University of West Bohemia in Pilsen
Department of Computer Science and Engineering
Univerzitní 8
30614 Pilsen
Czech Republic

Methods for Signal Classification and their Application to the Design of Brain-Computer Interfaces

The State of the Art and the Concept of Ph.D. Thesis

Lukáš Vařeka

Technical Report No. DCSE/TR-2013-4
April, 2013

Distribution: public

Technical Report No. DCSE/TR-2013-4
April, 2013

Methods for Signal Classification and their Application to the Design of Brain-Computer Interfaces

Lukáš Vařeka

Abstract

This thesis summarizes state-of-the-art signal processing and classification techniques for P300 brain-computer interfaces (BCIs). BCIs allow paralyzed subjects to communicate with the outside world without using their muscles. P300 BCIs are based on intermixing frequent and rare stimuli which elicit different responses of the brain. The main challenge we have to deal with is very low signal-to-noise ratio. Furthermore, the EEG response related to stimuli shows great subject-to-subject variability. The related state-of-the-art techniques differ both in feature extraction and classification. Currently, there is no approach to be state-of-the-art, instead, many approaches have been successfully applied to different data-sets. Unfortunately, BCI researchers also have to cope with weaknesses of the state-of-the-art P300 BCIs. They have low bit rates and typically require new training for each individual user. In this theses, a novel approach for the design of P300 BCIs is proposed. The approach is based on unsupervised neural networks.

Copies of this report are available on http://www.kiv.zcu.cz/publications/ or by surface mail on request sent to the following address:

University of West Bohemia
Department of Computer Science and Engineering
Univerzitni 8
30614 Pilsen
Czech Republic

Contents

1 Introduction

A growing interest has been devoted to understanding the human brain. Brain research investigates the brain from different perspectives. In medicine and in neurobiological research, many brain imaging and monitoring techniques provide use with knowledge that was previously unavailable, e.g. electroencephalography (EEG), magnetoencephalography (MEG), positron emission tomography (PET), functional magnetic resonance imaging (fMRI) and optical imaging. Although modern neuroimaging techniques have helped to discover new knowledge by measuring blood flow in the brain, traditional EEG still maintains its position because it is relatively cheap, non-invasive and has very high temporal resolution. Typically, EEG can measure changes in brain activity on a millisecond-level.

One of the most promising technical applications of EEG are brain-computer interfaces (BCIs). They allow a direct communication between the brain and the computer without traditional pathways using muscles. This is especially important for paralyzed people who do not have any other possibility to communicate with the outside world. However, brain-computer interface design is not straightforward since the intention of the user cannot be read from EEG directly. Instead, special training of the subject is often required. This thesis focuses on the BCIs that are based on event-related potentials (ERPs). ERP-based BCIs, commonly referred to as P300 BCIs, are very popular since they do not require special training of the subjects, instead, visual or auditory stimulation is required. Unfortunately, signal-to-noise ratio is typically low, so the correct feature extraction and classification techniques are necessary to achieve good results both in accuracy and speed.

The main objective of this thesis is to introduce the most common preprocessing, feature extraction and classification algorithms that have been studied regarding the P300-based BCIs. For classification, both linear and non-linear classifiers are described, since they have been frequently applied to the classification problem. In addition, clustering-based neural networks are also introduced. So far, they have rarely been used for BCI research, however, they represent an interesting field for further exploration.

Section 2 introduces electroencephalographic signal and event-related potentials. Brain-computer interfaces and different approaches for their design are explained in Section 3. State-of-the-art techniques for preprocessing and feature extraction of the EEG/ERP data are introduced in Section 4 and classification is explored in Section 5. Finally, in Section 6, the problems of the current P300 BCIs yet to be addressed are discussed, and a novel approach for BCI design is proposed.

2 Electroencephalography

2.1 Introduction

Electroencephalography (EEG) is a technique based on recording the electrical activity along the scalp. EEG measures voltage fluctuations which result from ionic current flows within the neurons of the brain [1]. The resulting EEG activity reflects the summation of the synchronous activity of many groups of neurons that have similar spatial orientation. The neurons of the cortex are thought to produce most of the EEG signal because they are well-aligned and fire together. In contrast, activity from deep sources in the brain is generally more difficult to detect. Unfortunately, the sources of the signal can be recovered from the EEG signal only approximatively. [2] EEG uses electrodes for measuring EEG signals from multiple areas on the skull, the signal at each electrode is a time variation of the electrical potential difference between the electrode and the reference electrode. The recorded signal is stored as electroencephalogram for evaluation.

EEG is commonly used in clinical practice, neurological and psychological research. The main advantage of EEG is its good resolution in time domain and its non-invasiveness. However, this technique has also drawbacks. One of the biggest disadvantages is the fact that the EEG signal represents many sources of neural activity. Most of this activity is undesired so signal-to-noise ratio is typically very low. [1]

2.2 Recording of the EEG signal

The conventional electrode setting for both research and clinical purposes is called 10–20 system. It typically includes 21 electrodes (excluding the earlobe electrodes). The earlobe electrodes called A1 and A2, connected to the left and right earlobes, are often used as the reference electrodes. It is also possible to attach the reference electrode at the root of the nose. The 10–20 system considers some constant distances by using specific anatomic landmarks from which the measurement would be made and then uses 10 or 20% of that specified distance as the electrode interval. The odd electrodes are on the left and the even ones on the right. The system is illustrated in Fig. 2.1. [3]

Since almost all EEG systems are computer-based, the conversion from analog to digital EEG is required. The conversion is performed by means of multichannel analog-to-digital converters. Fortunately, the effective bandwidth for EEG signals is limited to approximately 100 Hz. Therefore, to satisfy the Nyquist criterion, a minimum frequency of 200 Hz is often enough for sampling of the EEG signals. [3]

2.3 Normal EEG activity

Although EEG is stochastic, certain brain rhythms commonly manifest in the EEG signal. In healthy adults, the amplitudes and frequencies of such signals

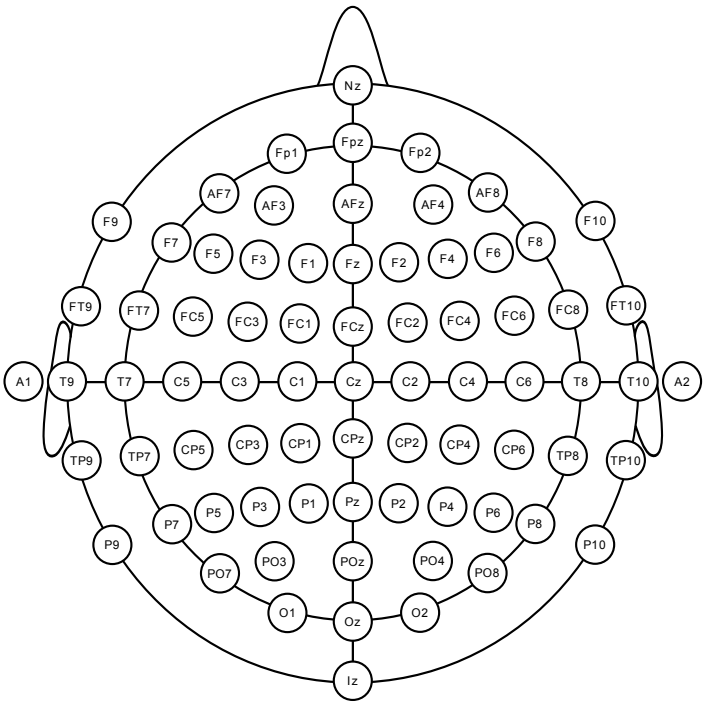

Figure 2.1: 10-20 system [4].

change from one state of consciousness to another, e.g. wakefulness or sleep. The characteristics of the waves may also change with age. There are five major brain waves distinguished by their different frequency ranges. These frequency bands from low to high frequencies are called delta, theta, alpha, beta and gamma (depicted in Fig. 2.2) [3]:

Delta waves Delta waves lie within the range of 0.5 – 4 Hz. These waves are primarily associated with deep sleep and may also be present in the waking state.

Theta waves Theta waves lie within the range of 4 – 7.5 Hz. They appear as consciousness slips towards drowsiness. Theta waves have been associated with creative inspiration and deep meditation.

Alpha waves Alpha waves appear in the posterior half of the head and are usually found over the occipital region of the brain. For alpha waves the frequency lies within the range of 8 – 13 Hz, and commonly appears as a round or sinusoidal shaped signal. Alpha waves have been thought to indicate a relaxed awareness without any attention or concentration. The alpha wave is the most prominent rhythm in brain activity. Most subjects produce alpha waves with their eyes closed. It is reduced or eliminated by opening the eyes, by hearing unfamiliar sounds, by anxiety, mental concentration or attention.

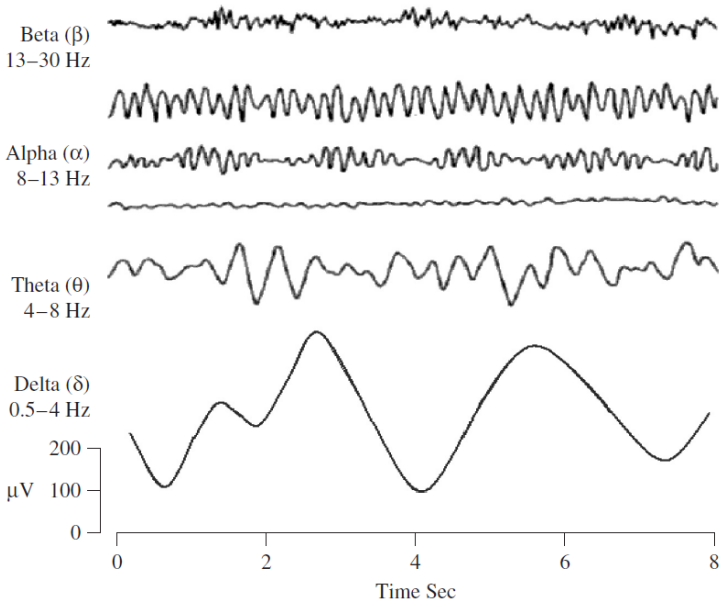

Figure 2.2: Brain rhythms in healthy adults [3].

Beta waves A beta wave is the electrical activity of the brain varying within the range of 14 – 26 Hz. It is the usual waking rhythm of the brain associated with active thinking, active attention, focus on the outside world, or problem solving, and is found in normal adults.

Gamma waves The gamma range is associated with the frequencies above 30 Hz (mainly up to 45 Hz). Although the amplitudes of these rhythms are very low and their occurrence is rare, detection of these rhythms can be used for confirmation of certain brain diseases.

2.4 Event-related potentials

Event-related potentials (ERPs) are the changes of the EEG signal associated with something that occurs either in the external world or within the brain itself. They are further classified as exogenous or endogenous. Exogenous ERPs are determined by the physical characteristics of the stimulus while endogenous ERPs are determined by its psychological effects. There are several ERPs differing by their latency, polarity and amplitude. The labeling reflects their polarity (P for the positive, N for the negative) and the latency (time after stimulus). For example, the N100 is a negative event-related potential occurring approximately 100 ms after the event in the signal. Since latency may vary among individuals and even among different recording situations within the same individual, the waveform is often identified by its typical latency. Therefore, the names of ERP components may also be based on sequential numbering of the peaks (e.g. P1,

N1, P2, N2, P3). [5] Some ERPs are associated with any type of visual or audible events (e.g. the N100 component), the others are triggered only by the events following some semantic pattern (e.g. the P300 or the N400 component). The ERP experiments usually aim to elicit certain ERP components using regular stimulation.

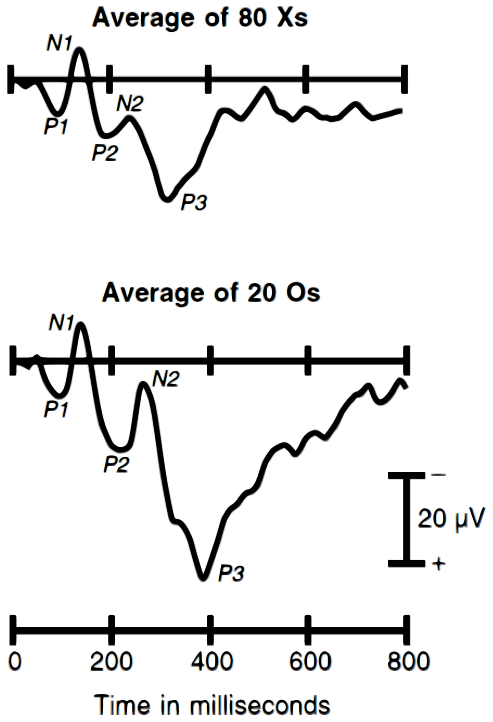

Figure 2.3: Comparison of averaged EEG responses to non-target stimuli (Xs) and target stimuli (Os). There is a clear P3b component following the Os stimuli. Negative is plotted upward. [6]

For example, oddball paradigm [6] is commonly used for the P300 elicitation. In this technique, low-probability target stimuli are mixed with high-probability non-target stimuli. Both stimuli trigger a reaction which can be measured and detected shortly after the event in the EEG signal and consists of multiple ERP components. However, the target stimuli tend to cause a different reaction, with the P300 waveform (sometimes referred to as the P3 component) being most significant. Fig. 2.3 shows an example of averaged event-related potentials for target and non-target stimuli. The P300 waveform (and especially its sub-component P3b [7]) is probably related to the process of decision making - it is elicited when the subject classifies the last stimulus as the target (for example by silent counting). The P300 is usually the strongest ERP component and it occurs 250 - 400 ms after the target stimulus as a positive peak. Its amplitude and latency may be influenced by different factors. For example, the P300 amplitude gets larger as target probability gets smaller. The amplitude is also larger when subjects

devote more effort to a task. [6]

2.5 Artifacts

Unfortunately, event-related potentials or different useful information in the signal are usually hidden in noise. In EEG, disturbing signals are commonly referred to as artifacts. The main artifacts can be divided into patient-related (physiological, biological) and system (technical) artifacts [3].

Physiological artifacts include any biological activity arising from other sources than the brain. There are several types of biological artifacts, e.g. blinks, eye movements, muscle activity, and skin potentials. These artifacts can be problematic in two ways. First, they are typically very large compared to the ERP signals and may greatly decrease signal-to-noise ratio of the averaged ERP waveform. Second, some types of artifacts may be systematic rather than random, occurring in some conditions more than others and being time-locked to the stimulus so that the averaging process does not eliminate them. For example, some stimuli may be more likely to elicit blinks than others, which could lead to differences in amplitude in the averaged ERP waveforms. [6]

Detection of eye-blinking artifacts is especially important since the artifacts may distort the data to an unacceptable extent [6]. Depending on the position of the reference electrode, a blink can be seen in the EEG signal as a positive or negative peak appearing at EOG electrode (if any) and a peak with opposite polarity appearing at the scalp electrodes. Deflection decreases with the increasing distance between the eyes and the electrode. A typical eye-blink response is represented by a peak with the amplitude of 50 - 100 μV with the duration of 200 - 400 ms [6]. Figure 2.4 shows an example of a blink in the signal.

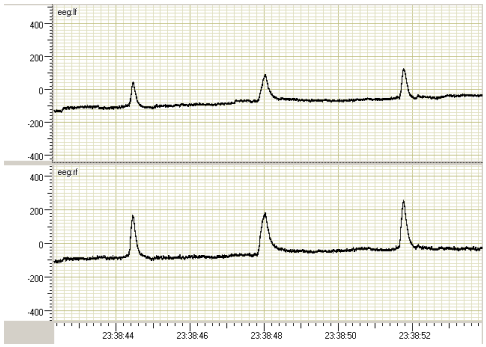

Figure 2.4: Blinks in EEG signal. The x-axis presents time and y-axis voltage in μV. [8]

The most important method to deal with physiological artifacts is to ask the subject to sit comfortably, not to move and to limit blinking. There is no substitute for clean data. [6] However, in many cases, it is also important to reject or correct the artifacts that necessarily more or less distort the data.

The system artifacts include 50 Hz power supply interference, impedance fluctuation, cable defects, electrical noise from the electronic components, and unbal-

anced impedances of the electrodes. [3] They are usually easier to eliminate than the biological artifacts, e.g. power supply interference can be eliminated using temporal filtering [6].

3 Brain-computer interfaces

Recent advances in cognitive neuroscience and brain imaging techniques have started to provide us with the ability to interface directly with the human brain. The scientific interest in brain-computer interfaces is primarily driven by the needs of people with neuromuscular diseases (e.g. amyotrophic lateral sclerosis, brainstem stroke, brain or spinal cord injury, cerebral palsy, muscular dystrophies, multiple sclerosis, and numerous other diseases). [9] Typically, these diseases damage the neural pathways that control muscles. Nearly two million people are affected in the United States alone, and far more around the world [10]. Those most severely affected may lose all voluntary muscle control and may be completely locked in to their bodies, unable to communicate with the outside world [11].

In the first international meeting on BCI technology, which took place in 1999, at the Rensselarville Institute of Albany (New York), Jonathan R. Wolpaw formalized the definition of the BCI system [12]:

> A brain-computer interface (BCI) is a communication or control system in which the user's messages or commands do not depend on the brain's normal output channels. That is, the message is not carried by nerves and muscles, and, furthermore, neuromuscular activity is not needed to produce the activity that does carry the message.

Instead of using brain's normal output pathways, users explicitly try to manipulate their brain activity to produce signals that can be used to control computers. The used recording techniques include, besides electroencephalography (EEG) and more invasive electrophysiological methods, magnetoencephalography (MEG), positron emission tomography (PET), functional magnetic resonance imaging (fMRI), and optical imaging. However, MEG, PET, fMRI, and optical imaging are still technically demanding and expensive. Furthermore, PET, fMRI, and optical imaging, which depend on blood flow, have long time constants and therefore, they are less appropriate for on-line communication. Currently, only EEG and related methods can function in most environments, and require relatively simple and inexpensive equipment. [11]

Any BCI has input (e.g. electrophysiological activity from the user), output (i.e. device commands), components that translate input into output, and a protocol that determines the operations. The EEG signal is acquired by electrodes on the scalp and processed to extract specific signal features (e.g. amplitudes of evoked potentials) that reflect the decision of the user. These features are translated into commands that operate a device (e.g. a simple word processing

program). The user must develop and maintain good correlation between his or her intent and the signal features employed by the BCI and the BCI must select and extract features that the user can control and must translate those features into device commands correctly and efficiently. [11]

3.1 Different paradigms for BCIs

Present-day BCIs generally fall into 5 groups based on the electrophysiological signals they use: visual evoked potentials, slow cortical potentials, mu and beta rhythms, cortical neurons and the P300 ERPs. The most important paradigms include [11]:

3.1.1 Visual evoked potentials (VEP)

Visual evoked potentials (VEP) are ERPs with short latency that represent the exogenous response of the brain to a rapid visual stimulus. They are characterized by a negative peak around 100ms (N100) followed by a positive peak around 200ms (P200). [13] The N100 component is significantly modulated by attention [6]. These potentials were used by the system introduced by Vidal in the 1970s [14] that used the VEP recorded from the scalp over visual cortex to determine the direction of eye gaze. Therefore, the VEP-based communication systems depend on the individual ability to control gaze direction. They perform the same function as systems that determine gaze direction from the eyes themselves, and can be categorized as dependent BCI systems. [11]

3.1.2 Slow cortical potentials

Low-frequency voltage changes generated in cortex occur over 0.5 – 10.0 s and are called slow cortical potentials (SCPs). Negative SCPs are typically associated with movement and other functions involving cortical activation, while positive SCPs are usually associated with reduced cortical activation. People can learn to control SCPs and thus, they can control movement of an object on a computer screen. [11]

3.1.3 μ and β rhythms

These electrical activities are observable inside a frequency range from 8 Hz to 12 Hz (μ) and 12 Hz to 30 Hz (β). These signals are associated with those cortical areas most directly connected to the motor output of the brain and can be willingly modulated with a movement, a preparation for movement or an imaginary mental movement. Movement is typically accompanied by a decrease in the μ activity. Its opposite, rhythm increase, occurs in the post-movement period and with relaxation. Since the changes are independent of activity in the normal output channels of peripheral nerves and muscles, the increases or

decreases of this rhythm have been used several times as a support for a BCI. [1, 13, 15]

3.1.4 P300 Event-related potentials

As previously mentioned, the P300 is an event-related potential elicited by oddball paradigm. Because of its amplitude and the fact that the P300 is a cognitive reaction to outside events, many brain-computer interfaces are based on the P300 detection [16]. However, the detection of the P300 is challenging because the P300 component is usually hidden in underlying EEG signal. [6]

This BCI paradigm was successfully used for attention-based typewriting introduced by [17]. The Matrix speller consists of a 6x6 symbol matrix. The symbols are arranged in rows and columns. Throughout the course of a trial, the rows and columns are flashed one after the other in a random sequence. Since the neural processing of a stimulus can be modulated by attention, the ERP elicited by target intensifications is different from the ERP elicited by nontarget intensifications. Fig. 3.5 shows examples of the P300 spellers based on two different scenarios.

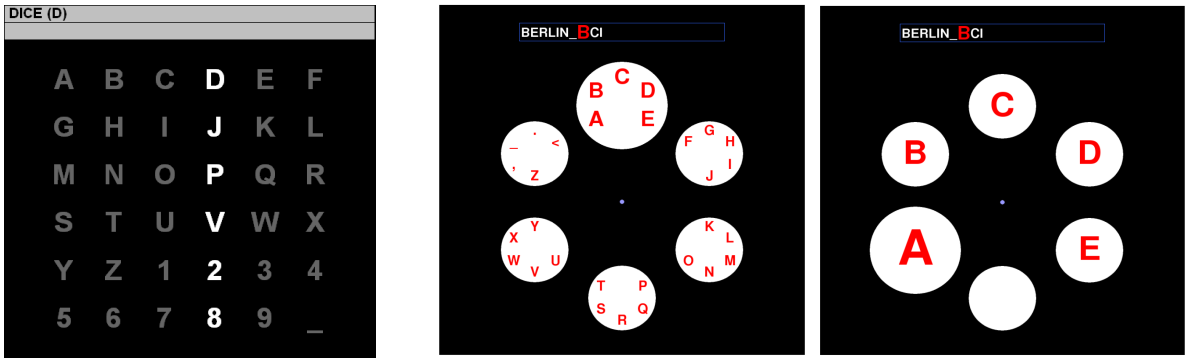

Figure 3.5: Comparison of two P300 speller experiments. The screenshot on the left shows the original P300 speller as it was suggested by [17]. The screenshots on the right show one of many improvements of the original scheme - Hex-o-Spell [18]. Symbols can be selected in the mental typewriter Hex-o-Spell in a two level procedure [18]: 1) At the first level, a disc containing a group of 5 symbols is selected. 2) For the second level, the symbols of the selected group are distributed to all discs (animated transition). The empty disc can be used to return to the group level without selection. Selection of the backspace symbol allows to erase the last written symbol.

Recently, it has been shown that the P200 component can also contribute to improving of the accuracy of P300 spellers. Therefore, it has been recently recommended to focus on the ERP signal as a whole rather than considering the P300 component only. [18]

3.1.5 Steady-State Visual Evoked Potentials (SSVEP)

These signals are natural responses to visual stimulations at specific frequencies. When the human eye is excited by a visual stimulus ranging from 3.5 Hz to 75 Hz, the brain generates an electrical activity at the same (or multiples of the) frequency of the visual stimulus. The SSVEP signals are strongly modulated by a selective spatial attention process: these signals are well defined within the extent, determined by the visual attention. Outside this area, flashing visual stimuli do not generate the same activity. They are used for understanding which stimulus the subject is looking at in case of stimuli with different flashing frequency. [13]

3.2 BCI illiteracy

The BCI systems do not work for all users. A universal BCI that works for everyone has never been developed. Instead, about 20% of subjects have troubles using a typical BCI system. Some groups have called this phenomenon "BCI illiteracy". Some possible solutions have been proposed, such as improved signal processing, training, and new tasks or instructions. However, these approaches have not resulted in a BCI that works for all users, probably because a small minority of users cannot produce detectable patterns of brain activity necessary to a particular BCI approach.

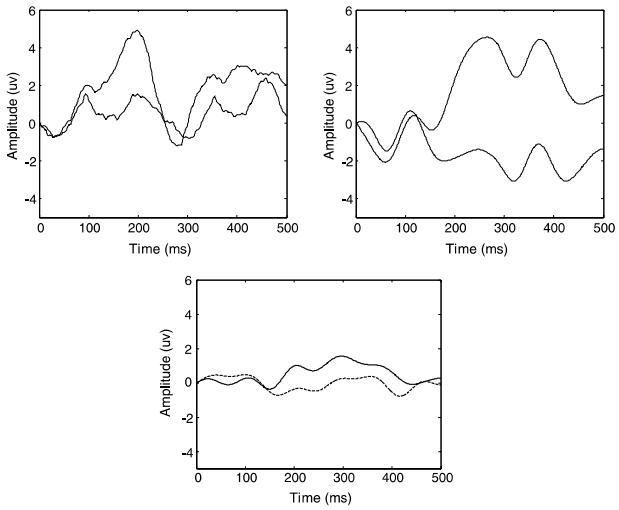

Figure 3.6: The P300 waveforms for different subjects. Only the subject whose response is in the top right figure has a strong P300. [9].

While all people have brains with the same cortical processing systems, in roughly the same locations, there are individual variations in brain structure. In some users, neuronal systems needed for control might not produce electrical activity detectable on the scalp. This is not because of any problem with the user. Their necessary neural populations may be healthy and active, but the activity they produce is not detectable by EEG. The key group of neurons neurons may

be located too deep for EEG electrodes or too close to another, louder group of neurons. For example, about 10% of subjects do not produce a robust P300. [9]

Consider the examples in Fig. 3.6 which depicts ERP activity from three users of a P300 BCI. Each figure represents an average of many trials. The top left panel shows a subject who did not have a strong P300. The solid and dashed lines look similar in the time window when the P300 is typically prominent, which is about 300–500 ms after the flash. However, these two lines differed during an earlier time window. The top right panel shows a subject who did have a strong P300. The bottom panel shows a subject whose ERPs look similar for target and nontarget flashes throughout the time window. This subject cannot use a P300 BCI.

3.3 Design of the BCI systems

In order to control a BCI, the user must produce different brain activity patterns that will be identified by the system and translated into commands. Typically, this identification relies on a classification algorithm. [11] The accuracy of classification depends on preprocessing, feature extraction and classification.

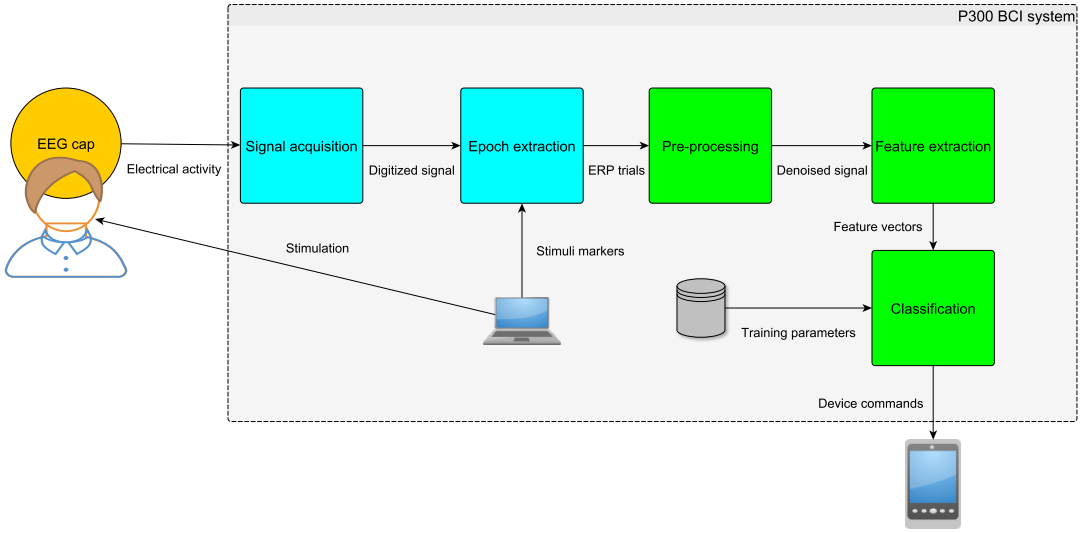

Figure 3.7: Diagram of the P300 BCI system. The EEG signal is captured, amplified and digitized using equidistant time intervals. Then, the parts of the signal time-locked to stimuli (i.e. epochs or ERP trials) must be extracted. Preprocessing and feature extraction methods are applied to the resulting ERP trials in order to extract relevant features. Classification uses learned parameters (e.g. distribution of different classes in the training set) to translate the feature vectors into commands for different device types.

The purpose of preprocessing is to improve SNR to allow feature extraction. Feature extraction selects the most relevant features for classifiers. For classification, different approaches are commonly used. Support vector machines (SVM),

multilayer perceptrons (MLP) and linear discriminant analysis (LDA) are among the most frequently used methods [19]. Recently, there has been growing tendency to treat feature extraction and classification as a complex algorithm rather than as separate operations [20].

Since this thesis focuses on P300 BCIs, Fig. 3.7 depicts the structures of P300 BCIs. From general BCIs, they only differ in one step: for further processing, it is necessary to extract the parts of the EEG signal that are time-locked to stimuli.

4 Preprocessing and feature extraction techniques for P300 BCIs

4.1 Introduction

ERP components are characterized by their temporal evolution and the corresponding spatial potential distributions. Therefore, as raw material, we have the spatio-temporal matrix $\boldsymbol{X}^{(k)} \in \mathbb{R}^{\mathbb{M}.\mathbb{T}}$ for each trial k with M being the number of channels and T being the number of sampled time points. Time is typically sampled in equidistant intervals, but it might be beneficial to select specific intervals of interest corresponding to ERP components. It can be helpful for classification to reduce the dimensionality of those features, e.g. by averaging across time in certain intervals, or by removing non-informative channels. The time intervals may result from sub-sampling, or they may be specifically chosen, preferably such that each interval contains one ERP component in which the spatial distribution is approximately constant. [18] The subset of channels can be chosen according to the used BCI paradigm, e.g. for the P300 speller it appears that 8-channel electrode set (Fz, Cz, P3, Pz, P4, PO7, PO8, Oz) is sufficient [20].

In case there is only one time interval, we call the features purely spatial. In this case, the dimensionality of the feature corresponds to the number of channels. Otherwise, when a single channel is selected, the feature is the time course of scalp potentials at a given channel, sampled at time intervals. [18] Fig. 4.8 illustrates both spatial and temporal features in ERP trials.

4.1.1 Feature vector properties

To design an EEG-based BCI system, the following properties must be considered [19]:

- *low SNR*: BCI features are noisy since the EEG signal generally has poor signal-to-noise ratio.
- *high dimensionality*: Usually, several features are generally extracted from several channels over several time segments before being concatenated into a single feature vector.

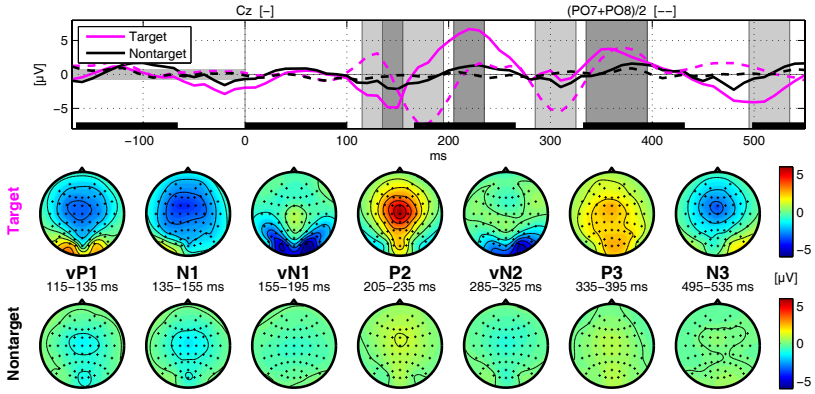

Figure 4.8: ERP components - spatial and temporal features. The upper plot shows averaged event-related potentials. Areas in gray correspond to the occurrence of ERP components. The lower part of the figure shows scalp maps representing spatial contribution of EEG channels related to the specific ERP components for both target and non-target responses. Note that for the P200 and the P300 components, the difference between target and non-target responses is the most significant. [18]

- *time information*: BCI features are usually based on time information since brain activity patterns are related to specific time variations of EEG (e.g. the P3 component)

4.2 Temporal features

4.2.1 Introduction

Temporal preprocessing and feature extraction generally treat the EEG signal as an one-dimensional signal. The signal is typically sampled in equal time intervals, e.g. 1 ms. The purely temporal signal may be taken from the most informative channel regarding the P300 classification (e.g. the Pz channel [6]).

The most used methods to improve signal-to-noise ratio of the EEG/ERP signal include averaging, temporal filtering, discrete wavelet transform and matching pursuit. [21, 22]

4.2.2 Averaging

As mentioned above, one of the main challenges of classification in P300 BCIs is low signal-to-noise ratio. In single trials, the P300 response is hard to distinguish from the background noise. This is due to artifacts in the signal caused for example by blinking which disrupt the signal a lot, or due to the latency of ERP which may slightly change from trial to trial even for the same subject. The problem with SNR may be overcome by averaging together many subsequent trials associated with the same stimulus. [22] This strategy is common in P300 BCI systems. The averaging generally suppresses random noise and makes the

ERP with a repeated pattern stand out. The problem with averaging is that more averaged trials mean slower data transfer. For example, the artifacts significantly increase the amount of trials needed to detect ERPs (usually the P300) with reasonable accuracy. Several solutions have been proposed to solve this problem, e.g. artifact removal or ANOVA averaging [23]. Fig. 4.9 shows how averaging gradually amplifies the differences between target and non-target trials and thus increases SNR.

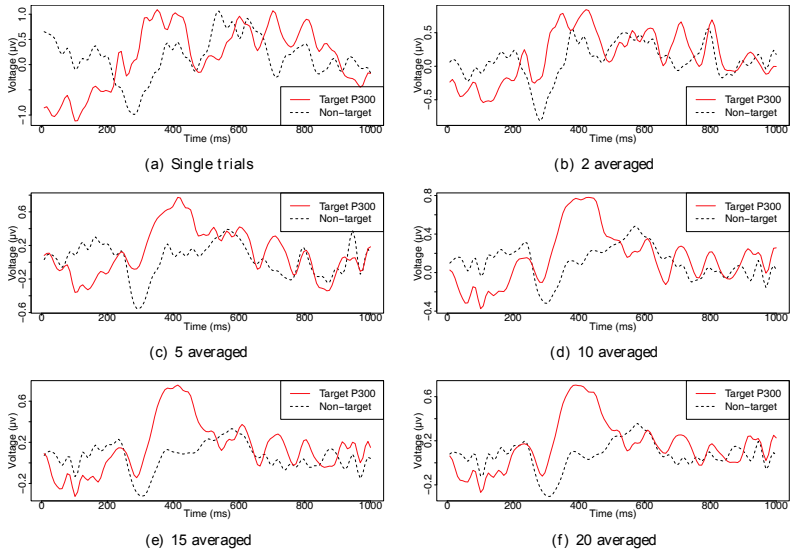

Figure 4.9: These figures show the effect of averaging trials together. For comparison, each figure depicts both target and non-target responses. [22]

4.2.3 Temporal filtering

Temporal filtering is absolutely necessary for EEG/ERP processing [6]. First, to fulfill the Nyquist Theorem, the rate of digitization must be at least twice as high as the highest frequency in the signal being digitized in order to prevent aliasing. Since the real filters do not have rectangular frequency response, the common practice is to set the digitization rate to be at least three times as high as the cut-off value of the filter [6].

The second main goal of filtering is the reduction of noise, and this is considerably more complicated. The basic idea is that the EEG consists of a signal plus some noise, and some of the noise is sufficiently different in frequency distribution from the signal that it can be suppressed simply by eliminating certain frequencies. For example, most of the relevant portion of the ERP waveform consists of frequencies between 0.01 Hz and 30 Hz, and contraction of the muscles leads to an EMG artifact that primarily consists of frequencies above 100 Hz. Therefore, the EMG activity can be eliminated by suppressing frequencies above 100 Hz and this will cause very little change to the ERP waveform. However, as the frequency distribution of the signal and the noise become more similar, it

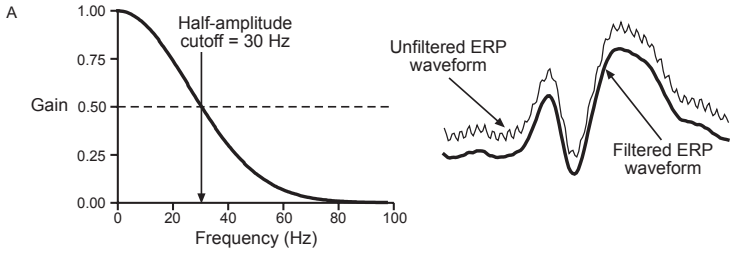

Figure 4.10: This figure illustrates the effects of low-pass filtering. The frequency response of the low-pass filter is on the left, the original P300 response and the low-pass-filtered response on the right. Although the filter deals with high-frequency distortions, it also decreases the amplitude of peaks and shifts their latencies. [6]

becomes more difficult to suppress the noise without significantly distorting the signal. For example, alpha waves can provide a significant source of noise, but because they are around 10 Hz, it is difficult to filter them without significantly distorting the ERP waveform. [6]

High-pass frequency filters may be used to remove very slow voltage changes of non-neural origin during the data acquisition process. Specifically, factors such as skin potentials caused by sweating and drifts in electrode impedance can lead to slow changes in the baseline voltage of the EEG signal. It is usually a good idea to remove these slow voltage shifts by filtering frequencies lower than approximately 0.01 Hz. This is especially important when obtaining recordings from patients or from children, because head and body movements are one common cause of these shifts in voltage. [6]

Although filters may increase SNR in temporal domain, they also more or less distort the ERPs. For example, as Fig. 4.10 shows, the low-pass filtering causes the filtered ERP waveform to start earlier and end later than the unfiltered waveform. The low pass filters also decrease the amplitude of peaks in the signal.

4.2.4 Discrete Wavelet Transform

Wavelets [24] were suggested by J. Morlet as a method for seismic data processing. The mathematical foundation was written by J. Morlet with A. Grossman. The theory of wavelets is based on signal decomposition using a set of functions that is generated by one or two base functions using dilatation or translation (modifying scale and position parameters).

The most commonly used is Discrete Wavelet Transform (DWT) which has linear computational complexity. It is based on restricting position and scales. Typically, they are based on powers of 2. [24]

The DWT of a signal is calculated by applying several filters to the signal. At every iteration, a high-pass and a low-pass filter is applied to the signal. The high-pass-filtered signal is called detail coefficients, the low-pass-filtered signal is called approximation coefficients. Both the signal are down-sampled by factor of

2 in each iteration to remove redundancy. The process may be repeated as needed for approximation coefficients. The algorithm is illustrated in Fig. 4.11.

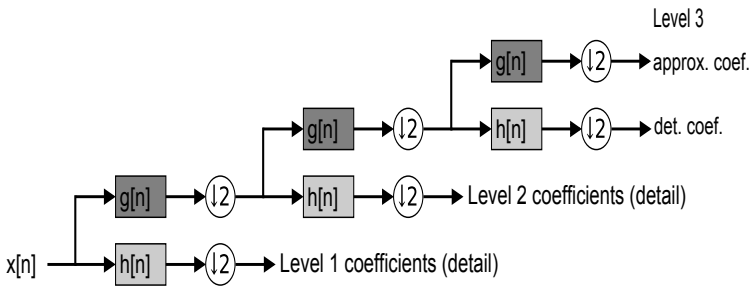

Figure 4.11: 3-Level Discrete Wavelet Transform

DWT for P300 BCIs Wavelet Transform is suitable for ERP analysis because of its optimal resolution in both the time and frequency domain [25]. In [26], the activity of the average ERP was analyzed using 5-level DWT. The wavelet coefficients related to the ERPs were identified and the remaining ones were set to zero. The inverse transform was applied to obtain a de-noised average ERP. The algorithm could also be applied to single trials. The authors identified the approximation coefficients of level 5 to be most correlated with the P300 component. However, the use of the processing for the detection of the P300 component was not explored.

Another solution is presented in [27]. It is based on blind source separation of 14 EEG channels using Independent Component Analysis. For feature extraction, 11-level DWT using Daubechies-4 wavelet was applied to the Independent Component 2 that was correlated with the P300 component. The accuracy was 60% when false positive events were taken in account.

Our research group has also published a few papers regarding the benefits of DWT for the P300 detection (see [28] and [29]). For the detection of the P300 component, the cross-correlation was calculated between a wavelet (scaled to correspond to the P300 component) and the ERP signal only in the corresponding part of the signal, where the P300 could be situated. If the maximum correlation coefficient exceeded threshold, the P300 was considered detected. The threshold was set for each wavelet separately. Daubechies6, Mexican hat, Gaussian, Haar and Symmlet8 wavelets were tested.

In [21], multi-level DWT was applied to purely temporal single trials. 7-level approximation coefficients were the feature vectors. The accuracy of approximately 75% was achieved. Furthermore, it could be increased up to more than 90% when averaging together six trials.

4.2.5 Matching Pursuit

Matching pursuit decomposes any signal into a linear expansion of waveforms that are selected from a redundant dictionary of functions. At each iteration, a waveform is chosen in order to best match the significant structures of the signal. Typically, this part is approximated by a Gabor atom, which has the highest scalar product with the original signal, and then it is subtracted from the signal [30]. This process is repeated until the whole signal is approximated by Gabor atoms with an acceptable error. Suppose we have a function g as follows:

$$g(t) = e^{-\pi t^2} \tag{4.1}$$

The Gabor atom has the following definition:

$$g_{s,u,v,w}(t) = g(\frac{t-u}{s})\cos(vt+w) \tag{4.2}$$

where s means scale, u latency, v frequency and w phase. These four parameters define each individual atom.

After a reasonable amount of iterations, the signal is decomposed into a set of Gabor atoms. Given this set of N atoms (g_n), the signal can be reconstructed:

$$f(t) \approx \sum_{n=1}^{N} a_n g_n(t) \tag{4.3}$$

When computed, the set of atoms is always finite so there cannot be a perfect match and a residuum remains.

Matching Pursuit for P300 BCIs Matching pursuit has not yet been extensively explored regarding the processing the EEG/ERP signal. However, it has been used for continuous EEG processing [31]. Since biological signal processing is one of the most important matching pursuit applications, it seems promising in the case of ERPs as well: the Gabor atoms are very flexible and can resemble any part of the signal including the localized ones such as the P300 component.

Some of the atoms found may be associated with ERPs, the others may correspond to some artifacts or noise in the signal. The interpretation of the Gabor atom can be based on its parameters. The position is especially important since each ERP component has its typical delay from the stimulus onset. However, significant differences in parameters may occur in different subjects and even the same subject may show different parameters for the same ERP component.

Filtering of the signal using a reconstruction of a few matching pursuit atoms appears to be the most promising. This approach was proposed in [23], described also in [29] and successfully tested on supervised classifiers [21, 32]. In [32], a multi-layer perceptron with eight input, five hidden and one output neuron was used to test the method on the data set of 752 epochs from five healthy participants. The accuracy of approximately 77% for single trials could be increased

up to over 90% when 6 trials were selected for averaging. Recall was significantly higher than precision, so false positives were a bigger problem than the error rate.

4.3 Spatio-temporal features and filtering

4.3.1 Introduction

Although purely temporal features may be sufficient if the most informative channel has high signal-to-noise ratio, in many cases, it is beneficial to combine the results of more channels to improve SNR. This approach relies on assumption that measured signals are mutually independent [33]. While the model assuming independent sources in the brain is considered partially inaccurate, they are also not completely dependent on each other and in many cases, assuming the independence can still lead to impressive results ([18], [34]).

4.3.2 Blind source separation

The basic macroscopic model of EEG generation [35] assumes the tissue to be a resistive medium and only considers effects of volume conduction, while neglecting the marginal capacitive effects. Therefore, each current source $s(t)$ contributes linearly to the scalp potential $\boldsymbol{x}(t)$, i.e.:

$$\boldsymbol{x}(t) = \boldsymbol{a}s(t) \tag{4.4}$$

with $\boldsymbol{a} \in \mathbb{R}^{\mathbb{M}}$ representing the individual propagation of the source s towards the M surface electrodes. Since there are multiple sources contributing ($\boldsymbol{s}(t) = (s_1(t), s_2(t), ...)^T$), the propagation vectors form a matrix $\boldsymbol{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, ...)$ and the overall surface potential results in:

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t) \tag{4.5}$$

In Eq. 4.5, $\boldsymbol{n}(t)$ is noise (i.e. it will not be a subject of investigation). The propagation matrix $\boldsymbol{A}$ is often called the forward model, as it relates the source activities to the signal acquired at different sensors. The propagation vector $\boldsymbol{a}$ of a source s can be visualized by means of a scalp map.

The reverse process of relating the sensor activities to originating sources is called backward modeling and aims at computing a linear estimate of the source activity from observed signals:

$$\hat{\boldsymbol{s}}(t) = \boldsymbol{W}^T\boldsymbol{x}(t) \tag{4.6}$$

A source $\hat{s}$ is therefore obtained as a linear combination of the spatially distributed information from the multiple sensors. A solution can be obtained only approximatively, e.g. by means of least mean squares estimator. [18] Therefore, the goal of backward modeling is to improve signal to noise ratio of signals of interest (e.g. ERP components). The rows $\boldsymbol{w}^T$ of the matrix $\boldsymbol{W}^T$ are commonly referred to as spatial filters.

However, this approach has its limitation for EEG data. Consider the following simplified noise free example of two sources s_1 and s_2 given with their corresponding propagation vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, respectively. The task is to recover the source s_1 from the observed mixture $\boldsymbol{x} = \boldsymbol{a}_1 s_1 + \boldsymbol{a}_2 s_2$. Any linear filter $\boldsymbol{w}^T$ yields $\boldsymbol{w}^T\boldsymbol{x} = \boldsymbol{w}^T\boldsymbol{a}_1 s_1 + \boldsymbol{w}^T\boldsymbol{a}_2 s_2$. If the two propagation vectors are orthogonal, i.e., ${\boldsymbol{a}_1}^T\boldsymbol{a}_2 = 0$, the best linear filter is directly $\boldsymbol{w}^T = \boldsymbol{a}_1^T$. In case of orthogonal sources the best filter corresponds to the propagation direction of the source. However, if assuming non-orthogonal propagation vectors, the signal along the direction $\boldsymbol{a}_1$ also consists of a portion of s_2. In order to obtain the optimal filter to recover s_1, the filter has to be orthogonal to the interfering source s_2 while having a positive scalar product $\boldsymbol{w}^T\boldsymbol{a}_1 > 0$. As a consequence, the spatial filters depend on the scalp distributions not just of the reconstructed source, but also on the distribution of the other sources. Furthermore, the signal that is recovered by a spatial filter $\boldsymbol{w}^T$ also captures the portion of the noise that is collinear with the source estimate: $\hat{s}(t) = \boldsymbol{s}(t) + \boldsymbol{w}^T\boldsymbol{n}(t)$. As a result, a spatial filter which optimizes the SNR of a signal of interest must be approximatively orthogonal against interfering sources and noise signals. [18, 36]

For on-line BCI systems, the exact recovery of the sources in the brain is not required. Instead, we use linear projection that combines information from multiple channels into the one-dimensional signal whose time course can be analyzed with conventional temporal methods. The vector w is chosen using one of the blind source separation methods to amplify the desired ERP component, e.g. the P300, and thus increase signal-to-noise ratio. [36] One of the benefits of spatial filtering is that some channels may contribute to a reduction of noise in the informative channels. Fig. 4.12 shows an example [18].

4.3.3 Independent Component Analysis

Independent component analysis (ICA) [33] 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. ICA is defined as an optimization problem to minimize the mutual information between the source components [33]. An efficient algorithm using higher order statistics was presented to measure the notion of non-Gaussianity that corresponds to 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 $\boldsymbol{S}(t)$ is assumed to be a set of truly independent sources, the observed mixed signal $\boldsymbol{X}(t)$ will be more Gaussian by the central limit theory. A single estimated source $\hat{s}_i(t)$ is a linear mixture of $\boldsymbol{X}(t)$ given by the weights in the spatial filter $\boldsymbol{w}_i$. The $\boldsymbol{w}_i$ 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.

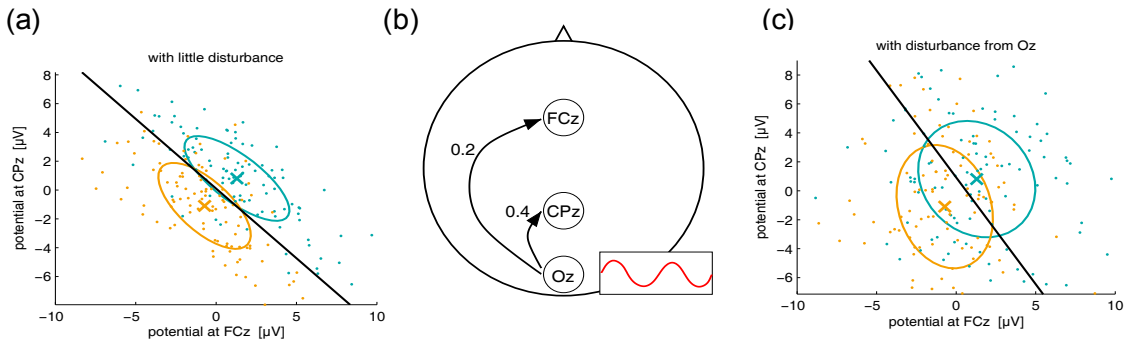

Figure 4.12: The benefits of spatial filtering. (a) Two dimensional Gaussian distributions were generated to simulate scalp potentials at two electrodes with relative high signal-to-noise ratio. (b) A disturbing signal is generated: simulated visual alpha at channel Oz, which is propagated to channels CPz and FCz. (c) This results in a substantially decreased separability in the original two dimensional space. However when classifying 3D data that includes data from sensor Oz, it becomes possible to subtract the noise from the informative channels CPz and FCz and classification becomes possible with the similar accuracy as for undisturbed data in (a). [18]

There are many different implementations of ICA that each use different metrics for statistical independence, e.g. FastICA which has good performance and uses kurtosis [22] as a measure of non-Gaussianity.

Furthermore, ICA can be implemented using neural networks, namely Infomax. [37]

ICA for P300 BCIs ICA has been proposed for ERP component analysis by [38]. This approach requires multiple EEG channels as inputs. This approach is explored and yields good results, however, it has high computational complexity. Therefore, it is unappropriate for on-line BCI systems. In [34], this problem has been addressed by applying ICA in training mode only. During the testing phase, a priori knowledge about spatial distribution (i.e. ICA demixing matrix) is used to decompose the signal efficiently. Although this technique has proven to boost classification accuracy of the P300 ERP component significantly (up to 100% classification accuracy with 5–8 averaged trials), it has also drawbacks. The ICA is trained on an individual subject and the information obtained cannot be easily applied to different subjects which may have different latencies and amplitudes of the ERP components.

5 Classification methods for P300 BCIs

5.1 Introduction

The purpose of classification is to divide the feature space into regions that correspond to different classification classes (decisions). For example, if a feature vector x, corresponding to an unknown pattern, falls in the region of "target" class, it is classified as "target", otherwise it is classified as "non-target". This does not necessarily mean that the decision is correct. If it is not correct, misclassification has occurred. The patterns (feature vectors) whose true class is known and which are used for the design of the classifier are known as training patterns (training feature vectors). [39]

While performing a pattern recognition task, classifiers may be facing several problems related to the features properties. In the field of BCI, two main problems need to be underlined: the curse-of-dimensionality and the Bias-Variance tradeoff [19].

The curse-of-dimensionality The amount of data needed to properly describe the different classes increases exponentially with the dimensionality of the feature vectors. Actually, if the number of training data is small compared to the size of the feature vectors, the classifier will most probably give poor results. It is recommended to use, at least, five to ten times as many training samples per class as the dimensionality.

The Bias-Variance tradeoff Formally, classification consists in finding the true label y^* of a feature vector x using a mapping f. This mapping is learnt from a training set T. The best mapping f^* that has generated the labels is, of course, unknown. If we consider the Mean Square Error (MSE), classification errors can be decomposed in three terms [19]:

$$\begin{aligned} MSE &= E[(y^* - f(x))^2] = E[(y^* - f^*(x) + f^*(x) - E[f(x)] + E[f(x)] - f(x))^2] \\ &= E[(y^* - f^*(x))^2] + E[(f^*(x) - E[f(x)]^2)] + E[(E[f(x)] - f(x))^2] \\ &= Noise^2 + Bias(f(x))^2 + Var(f(x)) \end{aligned} \tag{5.7}$$

These three terms describe three possible sources of classification error [19]:

- Noise: represents the noise within the system. This is an irreducible error.
- Bias: represents the divergence between the estimated mapping and the best mapping. Therefore, it depends on the method that has been chosen to obtain f (e.g. linear).
- Variance: reflects the sensitivity to the training set T used.

To attain the lowest classification error, both the Bias and the Variance must be low. Unfortunately, there is a natural Bias-Variance tradeoff. Actually, stable classifiers (e.g. LDA) tend to have a high Bias and a low Variance, whereas unstable classifiers (e.g. multi-layer perceptron) have a low Bias and a high Variance. Several techniques, known as stabilization techniques, can be used to reduce the Variance. Among them, combination of classifiers and regularization can be mentioned. EEG signals are known to be non-stationary. Training sets coming from different sessions are likely to be relatively different. Thus, a low Variance can be a solution to cope with the variability problem in BCI systems. [19]

5.2 Linear classifiers

Linear classifiers use linear functions to separate classes. Let us focus on the two-class case and consider linear discriminant functions. Suppose we have l-dimensional feature space, a weight vector $\boldsymbol{\omega} = [\omega_1, \omega_2, ..., \omega_l]$ and a *threshold* ω_0. Then the corresponding decision hypersurface is a hyperplane [39]:

$$g(\boldsymbol{x}) = \boldsymbol{\omega}^T \boldsymbol{x} + \omega_0 = 0 \tag{5.8}$$

For any $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ on the decision hyperplane, Equation 5.9 directly implies that the difference vector $\boldsymbol{x}_1 - \boldsymbol{x}_2$ (i.e. the decision hyperplane) is orthogonal to the vector $\boldsymbol{\omega}$ [39].

$$0 = \boldsymbol{\omega}^T \boldsymbol{x}_1 + \omega_0 = \boldsymbol{\omega}^T \boldsymbol{x}_2 + \omega_0 \Longrightarrow \boldsymbol{\omega}^T (\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0 \tag{5.9}$$

The most popular linear classifiers for BCIs include Linear Discriminant Analysis and Support Vector Machines. [19]

5.2.1 Linear Discriminant Analysis

The Linear Discriminant Analysis (LDA, also known as Fisher's LDA) is one of the most popular linear classifiers. The separating hyperplane is obtained by seeking the projection that maximize the distance between the two classes means and minimize the interclass variance [40]. To solve a N-class problem ($N > 2$), several hyperplanes are used. [19] This technique has a very low computational complexity which makes it suitable for on-line BCI system. Furthermore, this classier is simple to use and generally provides good results. Consequently, LDA has been used with success in a great number of BCI systems such as motor imagery based BCI, P300 speller, multiclass or asynchronous BCI. [19]

For known Gaussian distributions with the same covariance matrix for all classes, it can be shown that Linear Discriminant Analysis (LDA) is the optimal classifier in the sense that it minimizes the risk of misclassification for new samples drawn from the same distributions. LDA is equivalent to Least Squares Regression. [18]

LDA for P300 BCIs The optimality criterion generally relies on three assumptions. The following text discusses the criterion regarding the P300 BCIs [18]:

1. Features of each class are Gaussian distributed: According to experience in [18], features of ERP data satisfy this condition quite well and the method is quite robust to deviations from the normality assumption. For other type of features it is necessary to find a preprocessing to approximately achieve Gaussian distributions.
2. Gaussian distributions of all classes have the same covariance matrix: This assumption implies the linear separability of the data. It is approximately satisfied for many ERP data sets as the ongoing activity is typically independent of the different conditions under investigation. But this is not necessarily so. When comparing ERPs related to different visual stimuli, the visual alpha rhythm may be modulated by each type of stimuli in a different way. Fortunately, LDA is quite robust in cases of different covariance matrices. On the other hand, modeling a separate covariance matrix for each class leads to a nonlinear separation that is much more sensible to errors in the estimation of the covariance matrices and therefore often yields inferior results to LDA unless a large number of training samples is available. [18]
3. True class distributions are known: This assumption is obviously never fulfilled in any real application. Means and covariance matrices of the distributions have to be estimated from the data. Due to the inevitable errors in those estimates the optimality statement might not hold at all, even when the assumptions (1) and (2) are met quite well. This is typically the case, when the number of training samples is low compared to the dimensionality of the features. [18]

It has been shown that for high-dimensional data, the estimated covariance matrix may be imprecise. Therefore, LDA with shrinkage has been proposed to compensate for the problem. [18]

5.2.2 Support Vector Machines

A Support Vector Machine (SVM) [41] also uses a discriminant hyperplane to separate classes. Such a hyperplane is not unique. A classifier may converge to any of the possible solution. For SVM, the selected hyperplane is the one that maximizes the margins, i.e., the distance from the nearest training points. Maximizing the margins is known to increase the generalization capabilities [39]. In Figure 5.13, the margin for direction "1" is $2z_l$ and the margin for direction "2" is $2z_2$. Our goal is to search for the direction that gives the maximum possible margin. For any linear classifier defined by Equation 5.8, the distance between a point and a hyperplane can be calculated as [39]:

$$z = \frac{|g(\boldsymbol{x})|}{||\boldsymbol{\omega}||} \tag{5.10}$$

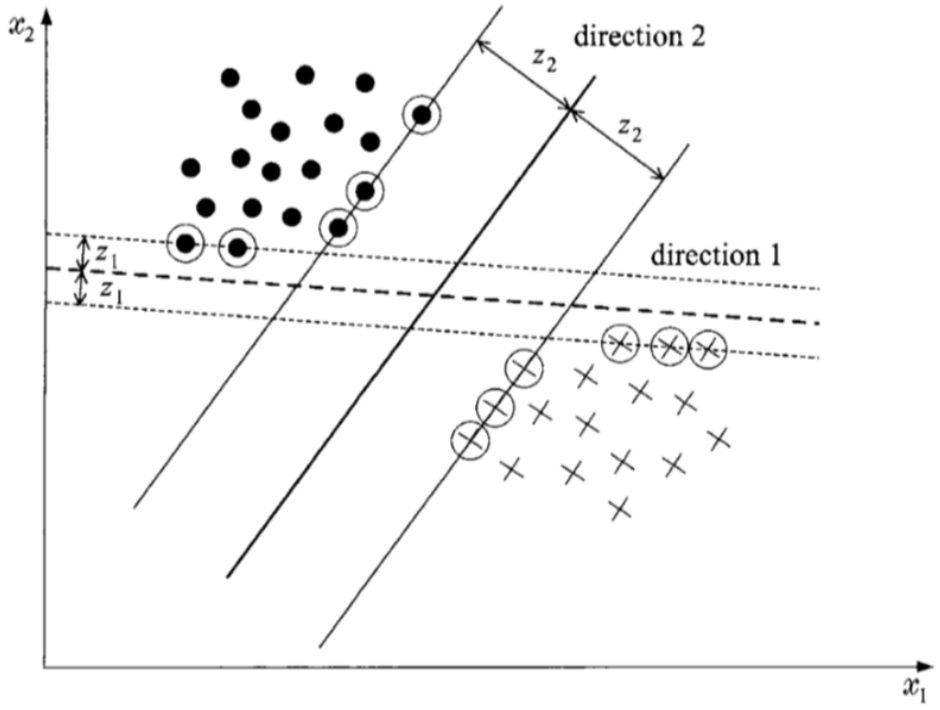

Figure 5.13: The figure depicts a linearly separable classification problem. However, there are multiple solutions for the decision hyperplane. The margin for direction 2 is larger than the margin for direction 1. Therefore, it is the preferable solution for the Support Vector Machine. [39].

We can scale $\boldsymbol{\omega}$, ω_0 so that the value of $g(\boldsymbol{x})$, at the nearest points in ω_1, ω_2 (circled in Figure 5.13), is equal to 1 for ω_1 and, thus, equal to -1 for ω_2. Assuming these conditions, the following can be stated:

- The margin equals: $\frac{1}{||\boldsymbol{\omega}||} + \frac{1}{||\boldsymbol{\omega}||} = \frac{2}{||\boldsymbol{\omega}||}$
- We require:

$$\boldsymbol{\omega}^T\boldsymbol{x} + \omega_0 >= 1, \forall \boldsymbol{x} \in \omega_1 \tag{5.11}$$

$$\boldsymbol{\omega}^T\boldsymbol{x} + \omega_0 <= -1, \forall \boldsymbol{x} \in \omega_2 \tag{5.12}$$

For each $\boldsymbol{x_i}$, we denote the corresponding class indicator by y_i (+1 for ω_1, -1 for ω_2.) Our task can now be summarized as: compute the parameters $\boldsymbol{\omega}$, ω_0 of the hyperplane so that to [39]:

- minimize $J(\boldsymbol{\omega}) = \frac{1}{2}||\boldsymbol{\omega}||^2$
- subject to $y_i(\boldsymbol{\omega}^T\boldsymbol{x_i} + \omega_0) >= 1$

Obviously, minimizing the norm makes the margin maximum. This is a quadratic optimization task subject to a set of linear inequality constraints. [39]

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 that must be chosen in advance. [22] SVMs are discussed in more detail in [39].

SVMs for P300 BCIs In [42], the authors report the results of a comparison of different classification algorithms, which show that stepwise linear discriminant analysis (SWLDA) and support vector machines (SVMs) perform better compared to the other classifiers. This conclusion is supported by [19].

5.2.3 Perceptron

The perceptron [43] is the simplest artificial neural network - it is essentially an artificial neuron; it simulates the functioning of a single biological neuron. The artificial neuron has the following definition:

$$y = f(\sum_{i=1}^{n} \omega_i x_i + \theta) \tag{5.13}$$

where y is the output of the neuron, ω_i are the weights of the neuron, x_i are the inputs of the neuron, θ is the threshold and f is the neural activation function.

For a single perceptron, the learning algorithm gradually adjusts its parameters to increase the probability of correct classification in the next step. At the beginning, the weights are set to initial values, typically chosen by random. The weights are updated according to the classification error, i.e. the Euclidean distance between the real and expected output. The problem with the perceptron is that it finds a separating hyperplane but not the optimal one. The algorithm is based on the following steps [43]:

1. Weights and a threshold are initialized. Weights $\omega_i(0)$ and the threshold θ are set to random low values.
2. The pattern and expected output are accepted. The input vector $X = x_1, x_2, .., x_n$ is applied to the perceptron and the expected output $d(t)$, being either +1 or -1, is stored.
3. The current output is calculated as:

   $$y(t) = f_h(\sum_{i=1}^{n} \omega_i(t) x_i(t) - \theta) \tag{5.14}$$

   with f_h being threshold function returning -1 for any $x < 0$ and +1 for any $x > 0$.
4. The weights are updated:

   $$\omega_i(t+1) = \omega_i(t) + \eta[d(t) - y(t)]x_i(t) \tag{5.15}$$

   with $d(t)$ being:

   - +1, if the pattern belongs to the first class

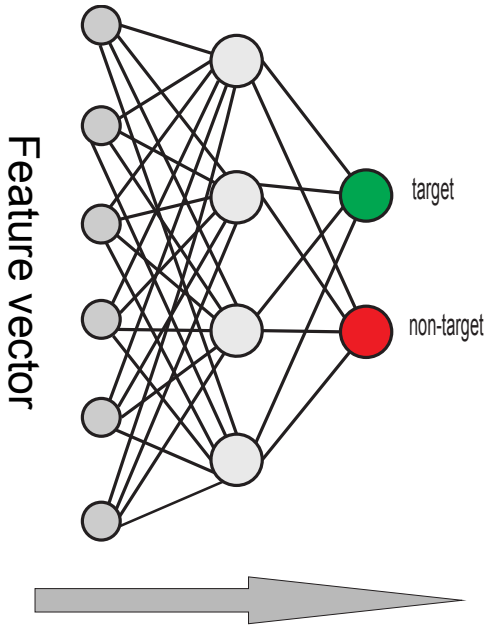

Figure 5.14: The figure depicts how the ERPs can be classified using multi-layer perceptron. Feature vectors are accepted with the input layer and propagated throughout the network. The decision about the class can be based on comparing the outputs of two output neurons, the higher output decides the class.

- - 1, otherwise

The constant η represents learning rate.

5. The process is iterated until stopping condition is fulfilled.

After training, classification is based on applying Step 3.

The perceptron is important because many neural networks are based on building more complex structures from perceptrons. The single perceptron has rarely been used for P300 BCIs. However, it has been successfully applied to finger movements in ECoG. [19]

5.3 Non-linear classifiers

Non-linear classifiers have non-linear decision boundaries. They may be superior to linear classifiers if the features are not linearly separable.

5.3.1 Multi-layer perceptron

Multi-layer perceptron (MLP) is a widely used neural network. It consists of two or more layers of perceptrons and follows a supervised learning model. From structural point of view, it is based on perceptrons connected in a form of more layers. Output of each neuron is connected to all neurons from the next layer. [43] An example of classification using MLP is shown in Fig. 5.14.

Since one perceptron can classify using one decision hyperplane, two perceptrons in the same layer represent two hyperplanes. Adding an additional layer enables the neural network to separate a more complex shape. [43]

Backpropagation In the 80s, the discovery of Backpropagation algorithm sparked a renewed interest in artificial neural networks. The algorithm is based on error minimization that leads to a gradual update of weights and thresholds. The parameters are updated starting from the last layer of MLP and finishing with the first layer. [43]

MLP for P300 BCIs Multi-layer perceptrons can approximate any continuous function. Furthermore, they can also classify any number of classes. This makes MLP very flexible classifiers that can adapt to a great variety of problems. Therefore, MLP, which are the most popular networks used in classification, have been applied to almost all BCI problems. However, the fact that MLP are universal classifiers makes them sensitive to overtraining, especially with such noisy and non-stationary data as EEG. Therefore, careful architecture selection and regularization is required. [19]

In [21], a voting classifier containing MLP was applied to the P300 classification problem. When using matching pursuit or wavelet transform for feature extraction, accuracy of more than 70% was achieved on single trials and over 90% for averaging.

5.3.2 Other non-linear classifiers

More non-linear classifiers have been explored. For example, Bayes quadratic, k Nearest Neighbors, or Hidden Markov Model. However, they are not as widespread as linear classifiers or neural networks in BCI applications. [19]

5.4 Clustering-based neural networks

Apart from traditional approaches based on supervised learning, clustering-based neural networks may also be considered for BCI design. They generally perform cluster analysis based on finding similarities within feature vectors. In most cases, these classifiers do not require any supervising. Therefore, it is necessary to interpret the results. So far, these methods have not been frequently used for P300 BCIs. However, they represent an interesting field for further exploration. [44, 45]

5.4.1 Self-organizing maps

Self-Organizing Maps (SOM) are neural networks in the unsupervised-learning category. In its original form the SOM was invented by the founder of the Neural Networks Research Centre, Professor Teuvo Kohonen in 1981-82, and numerous

versions, generalizations, accelerated learning schemes, and applications of the SOM have been developed since then.

The SOM converts complex, nonlinear statistical relationship between high-dimensional data items into simple geometric relationship on a low-dimensional display. [46] To allow this, a topological structure among the cluster units is assumed. There are m cluster units, arranged in a one- or two-dimensional array and the input signals are n-tuples.

The weight vector for a cluster unit serves as an input pattern associated with that cluster. During the self-organization process, the cluster unit whose weight vector matches the input pattern most closely (typically, by means of the square of the minimum Euclidean distance) is chosen as the winner. The winning unit and typically also its neighboring units (in terms of the topology of the cluster units) update their weights. The weight vectors of neighboring units do not have to be close to the input pattern. The architecture and corresponding algorithm can be used to cluster a set of p continuous-valued vectors $x = (x_1, x_2, .., x_n)$ into m clusters. [43] The architecture of the SOM network is shown in Fig. 5.15.

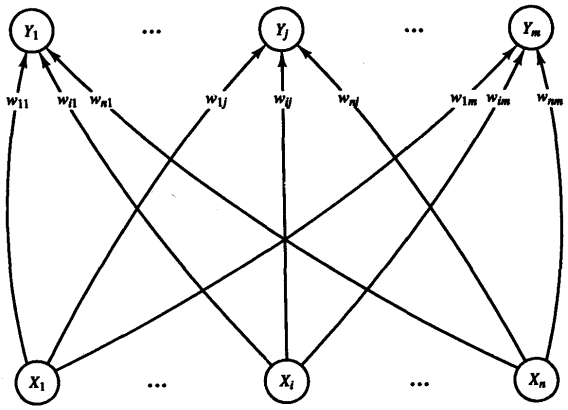

Figure 5.15: SOM network (X - input vector, w - weight vectors, Y - cluster units). [43]

Algorithm SOMs operate in the following steps [43]:

1. Initialize weights w_{ij}. Set topological neighborhood parameters. Set learning rate parameters.
2. While stopping condition is false, do:
   (a) For each input vector x, do i - iii.
      i. For each j, compute:

$$D(j) = \sum_i (w_{ij} - x_i)^2 \tag{5.16}$$

      ii. Find index J such that $D(j)$ is a minimum. (the closest cluster unit)

iii. For all units j within a specified neighborhood of J, and for all i:

$$w_{ij}(new) = w_{ij}(old) + \alpha[x_i - w_{ij}(old)] \tag{5.17}$$

(b) Update learning rate.

(c) If required, reduce radius of topological neighborhood.

(d) Test stopping condition.

The learning rate α is a slowly decreasing function of time (or training epochs). Kohonen [46] indicates that a linearly decreasing function is satisfactory for practical computations, a geometric decrease would produce similar results. The radius of the neighborhood around a cluster unit also decreases as the clustering process progresses. The formation of a map occurs in two phases: the initial formation of the correct order and the final convergence. The second phase takes much longer than the first and requires a small value for the learning rate. Many iterations through the training set may be necessary, at least in some applications. [43, 46]

Displaying the results When visualizing the trained self-organizing maps, the data clusters and the metric-topological relations of the data items are clearly visible. If the data items are vectors, the components of which are variables with a definite meaning such as the descriptors of statistical data, or measurements that describe a process, the SOM grid can be used as a groundwork on which each of the variables can be displayed separately using grey-level or pseudocolor coding. This kind of combined display has been found very useful for the understanding of the mutual dependencies between the variables, as well as of the structures of the data set. [46]

The U-Matrix is frequently used for visualization. It is a representation of a SOM where the Euclidean distance between the codebook vectors of neighboring neurons is depicted in an image. It is used to visualize the data in a high-dimensional space using a 2D image. [47] Fig. 5.16 shows an example.

SOMs for ERP decomposition SOMs have not been used for P300 BCIs. However, they have already been proposed for ERP decomposition and the results were promising [45]. In addition, SOMs were used for classification of features in continuous EEG [48].

5.4.2 Learning Vector Quantization

Learning Vector Quantization (LVQ) algorithms [46] are similar to Self-organizing maps. However, they are based on supervised approach. There are different variants of LVQ algorithms, e.g. LVQ1, OLVQ1, LVQ2.1, LVQ3, OLVQ3. In the following text, I will focus on the standard LVQ1 model.

As for SOM networks, a set of codebook vectors V^i with the same dimension as the input X store the cluster vectors. Each cluster is pre-assigned to a classification class. Several clusters can be assigned to the same class. The percentage of

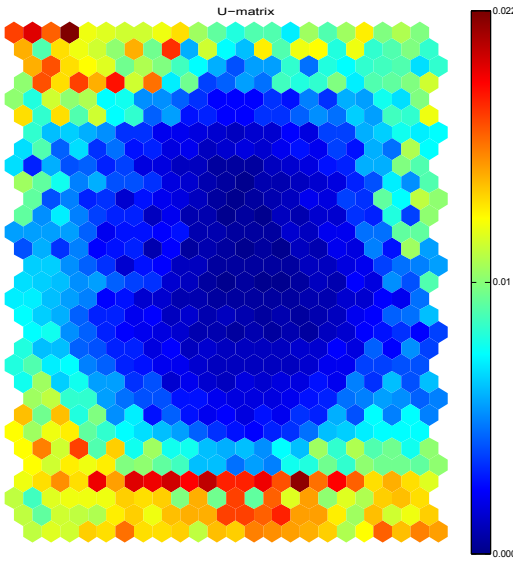

Figure 5.16: U-Matrix for the visualization of Self-organizing maps. The colors of the units represent Euclidean distances between the neighboring cluster units.

clusters assigned to one class is usually proportional to the percentage of training samples of this class.

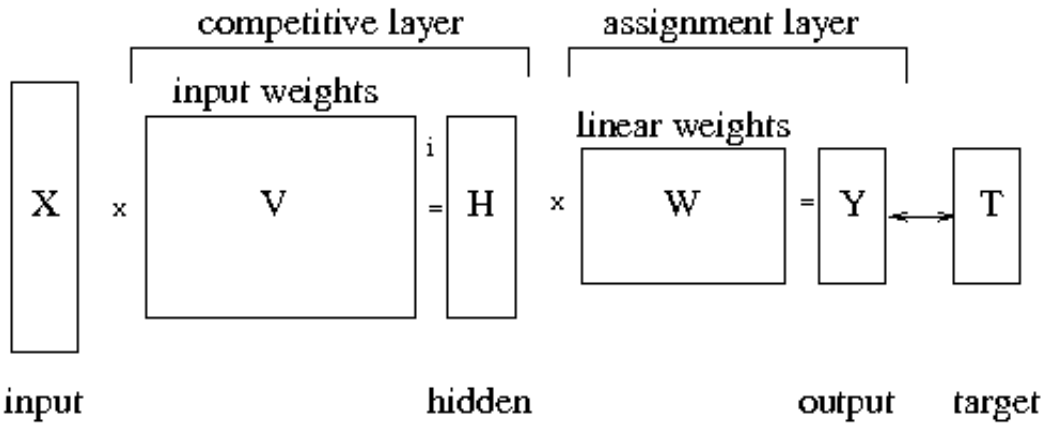

Figure 5.17: Learning Vector Quantization [49]

The architecture of LVQ, illustrated in Fig. 5.17, can be defined as a two-layer neural network containing a competitive layer and an assignment layer. The competitive layer is based on the similar principle as the SOM. Each neuron of the competitive layer (hidden layer) corresponds to a cluster and is characterized by a codebook vector V^i. A classic competitive step is used to determine which neuron is the closest one to the current input using the following formula: $c = argmin_i||X - V^i||$. The closest neuron is the winner neuron. The output vectors H of the competitive layer are computed as follows:

$h_i = winner\_takes\_all(||X - V^i||)$, where the *winner_takes_all* function produces 1 as the output for the winner neuron, and 0 for other neurons. The classes of the competitive layer are transformed into target classifications through the layer weights W.

Each neuron of the second layer corresponds to a specific class. Thus, the weights of the second layer relate clusters with classes. In the standard first version LVQ1, each cluster is exclusively assigned to one class. For example, suppose that neurons $i-1$, i, $i+1$ in the competitive layer are assigned to class k, then these competitive neurons will have W weights of 1 that link to the class k output neuron and 0 to other class output neurons. The output Y of LVQ1 can be written as: $Y = W * H = W * winner\_takes\_all(||X - V^i||)$. Y_k produces 1 if any of neurons $i-1$, i, $i+1$ wins the competition. Y is compared to the target vector T to compute the accuracy. In the learning procedure of LVQ1, W is fixed and only the codebook vector of the winner neuron c is updated according to the following rules: $V^c(t+1) = V^c(t) + \alpha(t)[X(t) - V^c(t)]$ if X and V^c belong to the same class or $V^c(t+1) = V^c(t) - \alpha(t)[X(t) - V^c(t)]$ if X and V^c belong to different classes where α is the learning rate.

The application of LVQ1 requires to define the learning rate, the number of competitive neurons and the number of training epochs. [49]

LVQ1 for ERP decomposition In [49], LVQ1 has been successfully applied to event-related potential data. Furthermore, the combination of LVQ and Extreme Learning machine was proposed to improve the performance of the classifier.

5.4.3 Adaptive Resonance Theory

The ART (Adaptive Resonance Theory) network developed by Carpenter and Grossberg [50] is also based on clustering. Its output is a direct information about an output class. There are several ARTs (ART 1, ART 2, ARTMAP) differing by architecture and input feature vector type (binary or real valued) they are able to process. The simplified architecture of this network is illustrated in Fig. 5.18.

The network consists of two layers of elements labeled $\mathbf{F_1}$ (input units) and $\mathbf{F_2}$ (cluster units), each fully interconnected with the others, and unit **G** and **R** (called *gain control unit* and *reset unit*), which are used to control the processing of the input data vector and the creating of the clusters.

The input and interface layer $\mathbf{F_1}$ has the same number of processing units as it is the size of the feature vector. The clustering layer $\mathbf{F_2}$ consists of as many units as the maximum number of clusters. $\mathbf{F_1}$ and $\mathbf{F_2}$ layers are connected using the set of weight vectors. Weight vectors are modified according to the properties of the input feature vector. For detailed description of ART network and the training algorithm see [43].

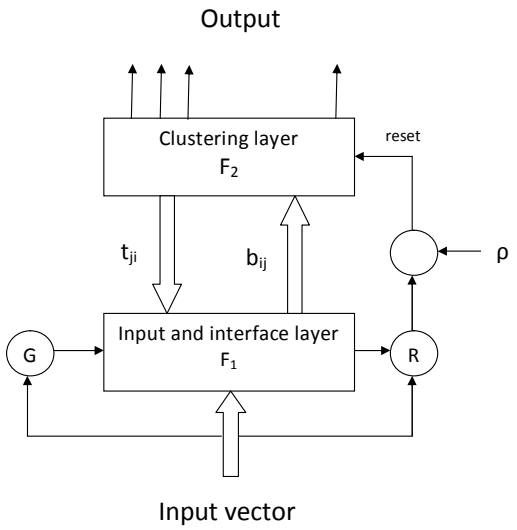

Figure 5.18: The simplified architecture of ART 2 network [44].

ART for ERP decomposition The ART 2 network has already been used for clustering of features obtained by processing the EEG/ERP data. The features were any waveforms that corresponded to significant EEG signal changes (e.g. ERP components or artifacts). Clustering and evaluation of its results can separate possible ERP components from artifacts and noise. The method was tested for the P300 component and the results were presented in [44].

Furthermore, Fuzzy ARTMAP (i.e. a modified ART network to allow supervised learning) has been proposed for BCI systems. [19]

6 Conclusion and Future Work

This thesis introduced the most common, but by no means all approaches that have been applied to P300 BCIs. Literature reviews of the field [19, 22] did not show any single P300 detection system to be the state-of-the-art. In [19], linear and non-linear classifiers were discussed regarding their benefits for EEG-based BCIs. Clearly, non-linear classifiers (e.g. multi-layer perceptrons) may be able to separate classes that are not linearly separable. However, linear classifiers may outperform non-linear classifiers if the features are very noisy since linear classifiers are simpler and thus less prone to overtrain. No clear conclusion can be made and there is ongoing discussion which approach is superior [18].

Many papers report different approaches for the P300 BCIs, however, it is difficult to directly compare between them because they use data recorded from different laboratories and different subjects. However, there is a benchmark P300 dataset provided from the BCI Competition 2003 [51] and some papers report their results on this dataset. Several approaches were able to achieve 100% accu-

racy using only 4-8 averaged trials on the BCI Competition 2003 data.

Although the P300 speller has been studied extensively and is one of the well established BCI systems, a recent review of the field [20] 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 paralyzed patients can display widely varying ERP responses between subjects. A reliable BCI system must be able to adapt to the unique ERP responses of each subject 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 paralyzed patients. [22]

Therefore, the universal BCI system should not only rely on priori information about expected event-related response, but should also be able to adapt and to provide reasonable accuracy for different subjects. This problem is commonly addressed for supervised classifiers which require training for each individual subject. The problem with this approach is that it usually consumes additional time for training on the data-sets from separate subjects. Furthermore, when traditional supervised methods are used, all attention is concentrated on separating the classes using class labels, and any other information is ignored by the classifier.

In the Ph.D. thesis, I will focus on unsupervised neural networks (e.g. self-organizing maps or Adaptive Resonance Theory) that has so far not been used for P300 BCIs. Instead of using class labels from a supervisor, unsupervised neural networks learn representation of the different kinds of data types that occur in the data sets. Furthermore, since no assumptions of the class structure of the data are made, the networks may discover new clusters that have not been apparent before. Therefore, the method may also contribute to understanding of the related feature vectors. Self-organizing maps were successfully applied to recognition of topographic patterns of EEG spectra in [48]. Six classes in total were used, for continuous alpha activity, flat EEG, theta activity, eye movements, muscle activity and bad electrodes contact. The authors concluded that SOMs were able to recognize similar topographic patterns in different EEGs, also in EEGs not used for the training of the map. According to [19], Learning Vector Quantization is the closest approach that has been investigated regarding P300 BCIs. In [49], supervised LVQ1 has successfully been applied to the P300 data. This further supports the hypothesis that similar models may be beneficial for P300 BCIs.

Unsupervised ANN, e.g. self-organizing maps can be trained on the data from a simple odd-ball experiment. At least two clusters and a "noise" cluster should appear after training. One cluster should be associated with target features, another one with nontarget features and in addition, the rest will probably be

undecidable. An expert can associate the clusters with classification classes, or features with target classes can be propagated through the network to create the associations. For each subject, the clusters will be distributed differently and the percentage of training features that will be associated with the undecidable cluster may indicate to which extent the subject is suitable for P300 BCIs. The trained neural network could be applied to a more complex BCI paradigm, e.g. the P300 speller. If the classification class of a single trial pattern cannot be decided, the trial can be averaged with the next corresponding trial to gradually increase SNR.

6.1 Aims of Ph.D. Thesis

To summarize, the idea of using unsupervised neural networks for ERPs is based on the following steps:

1. Preprocess the signal and extract the features to maximize signal-to-noise ratio.
2. Select a suitable unsupervised neural network and train it on the extracted features.
3. Based on expert knowledge, or on an automatic procedure, identify target and non-target responses in the trained ANN.
4. Verify the proposed approach by designing an on-line BCI system and testing the trained network on different subjects.
5. Compare the proposed classification approach with state-of-the-art classification techniques, e.g. LDA, or SVM.

References

[1] T. F, "Electroencephalography: Basic principles, clinical applications and related fields," *Archives of Neurology*, vol. 40, no. 3, 1983. [Online]. Available: http://dx.doi.org/10.1001/archneur.1983.04050030085025

[2] S. Klein and B. M. Thorne, *Biological psychology.* New York, USA: Worth., 2006.

[3] S. Sanei and J. A. Chambers, *EEG Signal Processing.* Wiley-Interscience, Sep 2007.

[4] M. t Hart. (2013) 10-20 system. [Online]. Available: http://www.mariusthart.net/

[5] T. W. Picton, O. G. Lins, and M. Scherg, "The recording and analysis of event-related potentials." in *Handbook of Neuropsychology*, F. Boller and J. Grafman, Eds. Amsterdam: Elsevier, 1995, vol. 10, pp. 3–73.

[6] S. Luck, *An introduction to the event-related potential technique*, ser. Cognitive neuroscience. MIT Press, 2005.

[7] J. Polich, "Updating P300: an integrative theory of P3a and P3b." *Clinical neurophysiology*, vol. 118, no. 10, pp. 2128–2148, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.clinph.2007.04.019

[8] A. Savelainen, "An introduction to eeg artifacts," *Mat-2.4108 Independent research projects in applied mathematics*, 2010.

[9] D. S. Tan and A. Nijholt, Eds., *Brain-Computer Interfaces - Applying our Minds to Human-Computer Interaction*, ser. Human-Computer Interaction Series. Springer, 2010.

[10] R. T. Abresch, J. J. Han, and G. T. Carter, "Rehabilitation management of neuromuscular disease: the role of exercise training." *J Clin Neuromuscul Dis*, vol. 11, no. 1, pp. 7–21, 2009. [Online]. Available: http://www.biomedsearch.com/nih/Rehabilitation-management-neuromuscular-disease-role/19730017.html

[11] D. J. McFarland and J. R. Wolpaw, "Brain-computer interfaces for communication and control," *Commun. ACM*, vol. 54, no. 5, pp. 60–66, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1941487.1941506

[12] J. Wolpaw, N. Birbaumer, W. Heetderks, D. McFarland, P. Peckham, G. Schalk, E. Donchin, L. Quatrano, C. Robinson, and T. Vaughan, "Brain-computer interface technology: a review of the first international meeting," *Rehabilitation Engineering, IEEE Transactions on*, vol. 8, no. 2, pp. 164–173, 2000.

[13] F. Beverina, G. Palmas, S. Silvoni, F. Piccione, and S. Giove, "User adaptive bcis: Ssvep and p300 based interfaces." *PsychNology Journal*, vol. 1, no. 4, pp. 331–354, 2003. [Online]. Available: http://dblp.uni-trier.de/db/journals/psychnology/psychnology1.html

[14] J. J. Vidal, "Real-time detection of brain events in EEG," *Proceedings of the IEEE*, vol. 65, no. 5, pp. 633–641, 1977. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1454811

[15] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "BCI2000: a general-purpose brain-computer interface (BCI) system." *IEEE transactions on bio-medical engineering*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004. [Online]. Available: http://dx.doi.org/10.1109/tbme.2004.827072

[16] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, "An Efficient P300-based Brain-Computer Interface for Disabled Subjects," *Journal of neuroscience methods*, vol. 167, no. 1, pp. 115–125, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1016/j.jneumeth.2007.03.005

[17] L. A. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, Dec. 1988. [Online]. Available: http://dx.doi.org/10.1016/0013-4694(88)90149-6

[18] B. Blankertz, S. Lemm, M. S. Treder, S. Haufe, and K.-R. Müller, "Single-trial analysis and classification of erp components - a tutorial," *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.

[19] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces," *Journal of neural engineering*, vol. 4, no. 2, Jun. 2007. [Online]. Available: http://dx.doi.org/10.1088/1741-2560/4/2/R01

[20] 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. 025 003+, Apr. 2011. [Online]. Available: http://dx.doi.org/10.1088/1741-2560/8/2/025003

[21] L. Vareka, "Neural networks in erp signal processing (in czech)," Master's thesis, 2011.

[22] Z. Cashero, *Comparison of Eeg Preprocessing Methods to Improve the Performance of the P300 Speller*. Proquest, Umi Dissertation Publishing, 2012.

[23] T. Rondik, "Methods of erp signals processing (in czech)," Diploma Thesis, University of West Bohemia, 2010.

[24] G. Kaiser, *A friendly guide to wavelets*. Cambridge, MA, USA: Birkhauser Boston Inc., 1994.

[25] E. Bartnik, K. Blinowska, and P. Durka, "Single evoked potential reconstruction by means of wavelet transform," *Biological Cybernetics*, vol. 67, pp. 175–181, 1992, 10.1007/BF00201024. [Online]. Available: http://dx.doi.org/10.1007/BF00201024

[26] R. Quiroga and H. Garcia, "Single-trial event-related potentials with wavelet denoising," *Clinical Neurophysiology*, vol. 114, no. 2, pp. 376–390, Feb. 2003. [Online]. Available: http://dx.doi.org/10.1016/S1388-2457(02)00365-6

[27] J. Ramirez-Cortes, V. Alarcon-Aquino, G. Rosas-Cholula, P. Gomez-Gil, and J. Escamilla-Ambrosio, "Anfis-based p300 rhythm detection using wavelet feature extraction on blind source separated eeg signals," in *Intelligent Automation and Systems Engineering, LNEE, Vol. 103*. Springer New York, 2011.

[28] T. Rondik and J. Ciniburk, "Comparison of various approaches for p3 component detection using basic methods for signal processing," in *BMEI*, 2011, pp. 698–702.

[29] R. M. Tomas Rondik, Jindrich Ciniburk and P. Mautner, "Erp components detection using wavelet transform and matching pursuit algorithm," in *Applied Electronics 2011*, 2011, pp. 1–4.

[30] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[31] P. Durka and K. Blinowska, "Analysis of EEG transients by means of matching pursuit," *Annals of Biomedical Engineering*, vol. 23, no. 5, pp. 608–611, Sep. 1995. [Online]. Available: http://dx.doi.org/10.1007/bf02584459

[32] L. Vareka, "Matching pursuit for p300-based brain-computer interfaces." in *TSP 2012*, 2012, pp. 513–516.

[33] P. Comon, "Independent component analysis, a new concept?" *Signal Process.*, vol. 36, no. 3, pp. 287–314, Apr. 1994. [Online]. Available: http://dx.doi.org/10.1016/0165-1684(94)90029-9

[34] 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 bio-medical engineering*, vol. 51, no. 6, pp. 1067–1072, Jun. 2004. [Online]. Available: http://dx.doi.org/10.1109/tbme.2004.826699

[35] P. L. Nunez and R. Srinivasan, *Electric Fields of the Brain: The Neurophysics of EEG, 2nd Edition*, 2nd ed. Oxford University Press, USA, Dec. 2005.

[36] L. P. Clay, L. C. Parra, C. D. Spence, Adam, and C. Paul Sajda, "Recipes for the linear analysis of EEG," *NeuroImage*, vol. 28, pp. 326–341, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.62.919

[37] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, Nov. 1995. [Online]. Available: http://dx.doi.org/10.1162/neco.1995.7.6.1129

[38] S. Makeig, T.-P. Jung, A. J. Bell, D. Ghahremani, and T. J. Sejnowski, "Blind separation of auditory event-related brain responses into independent components," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 94, no. 20, pp. 10 979–10 984, 1997.

[39] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Third Edition*, 3rd ed. Academic Press, Mar. 2006.

[40] K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition (Computer Science & Scientific Computing)*, 2nd ed. Academic Press, Oct. 1990.

[41] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[42] D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayoudh, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, "A comparison of classification techniques for the P300 Speller," *Journal of Neural Engineering*, vol. 3, no. 4, pp. 299–305, Dec. 2006. [Online]. Available: http://dx.doi.org/10.1088/1741-2560/3/4/007

[43] L. Fausett, Ed., *Fundamentals of neural networks: architectures, algorithms, and applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994.

[44] L. Vareka and P. Mautner, "The event-related potential data processing using art 2 network," in *The 5th International Conference on BioMedical Engineering and Informatics (BMEI 2012)*, 2012, pp. 467–471.

[45] T. Rondik and P. Mautner, "Clustering of gabor atoms describing event-related potentials," in *HEALTHINF 2013 International Conference on Health Informatics*, 2013, pp. 309–314.

[46] T. Kohonen, *Self-organization and associative memory: 3rd edition*. New York, NY, USA: Springer-Verlag New York, Inc., 1989.

[47] A. Ultsch and H. P. Siemon, "Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis," in *Proceedings of International Neural Networks Conference (INNC)*. Paris: Kluwer Academic Press, 1990, pp. 305–308. [Online]. Available: http://www.uni-marburg.de/fb12/datenbionik/pdf/pubs/1990/UltschSiemon90

[48] S. L. Joutsiniemi, S. Kaski, and A. T. Larsen, "Self-organizing map in recognition of topographic patterns of EEG spectra," *IEEE Transactions on Biomedical Engineering*, vol. 42, pp. 1062–1068, 1995.

[49] N. Liang and L. Bougrain, "Non-identity Learning Vector Quantization applied to evoked potential detection," in *Deuxième conférence française de Neurosciences Computationnelles, "Neurocomp08"*, Marseille, France, Oct. 2008, iSBN : 978-2-9532965-0-1. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00331590

[50] G. A. Carpenter and S. Grossberg, "The handbook of brain theory and neural networks," M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1998, ch. Adaptive resonance theory (ART), pp. 79–82. [Online]. Available: http://dl.acm.org/citation.cfm?id=303568.303586

[51] B. Blankertz, K. 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," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 6, pp. 1044–1051, 2004.