Increasing Information Transfer Rates for Brain-Computer Interfacing

Dissertation

zur Erlangung des akademischen Grades
doctor rerum naturalium
– Dr. rer. nat. –

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Potsdam

von
Guido Dornhege

Potsdam, im März 2006

Abstract

The goal of a Brain-Computer Interface (BCI) consists of the development of a unidirectional interface between a human and a computer to allow control of a device only via brain signals. While the BCI systems of almost all other groups require the user to be trained over several weeks or even months, the group of Prof. Dr. Klaus-Robert Müller in Berlin and Potsdam, which I belong to, was one of the first research groups in this field which used machine learning techniques on a large scale. The adaptivity of the processing system to the individual brain patterns of the subject confers huge advantages for the user. Thus BCI research is considered a hot topic in machine learning and computer science. It requires interdisciplinary cooperation between disparate fields such as neuroscience, since only by combining machine learning and signal processing techniques based on neurophysiological knowledge will the largest progress be made.
In this work I particularly deal with my part of this project, which lies mainly in the area of computer science. I have considered the following three main points:
Establishing a performance measure based on information theory: I have critically illuminated the assumptions of Shannon's information transfer rate for application in a BCI context. By establishing suitable coding strategies I was able to show that this theoretical measure approximates quite well to what is practically achieveable.
Transfer and development of suitable signal processing and machine learning techniques: One substantial component of my work was to develop several machine learning and signal processing algorithms to improve the efficiency of a BCI. Based on the neurophysiological knowledge that several independent EEG features can be observed for some mental states, I have developed a method for combining different and maybe independent features which improved performance. In some cases the performance of the combination algorithm outperforms the best single performance by more than 50 %. Furthermore, I have theoretically and practically addressed via the development of suitable algorithms the question of the optimal number of classes which should be used for a BCI. It transpired that with BCI performances reported so far, three or four different mental states are optimal. For another extension I have combined ideas from signal processing with those of machine learning since a high gain can be achieved if the temporal filtering, i.e., the choice of frequency bands, is automatically adapted to each subject individually.
Implementation of the Berlin brain computer interface and realization of suitable experiments: Finally a further substantial component of my work was to realize an online BCI system which includes the developed methods, but is also flexible enough to allow the simple realization of new algorithms and ideas. So far, bitrates of up to 40 bits per minute have been achieved with this system by absolutely *untrained* users which, compared to results of other groups, is highly successful.

Zusammenfassung

Ein Brain-Computer Interface (BCI) ist eine unidirektionale Schnittstelle zwischen Mensch und Computer, bei der ein Mensch in der Lage ist, ein Gerät einzig und allein Kraft seiner Gehirnsignale zu steuern. In den BCI Systemen fast aller Forschergruppen wird der Mensch in Experimenten über Wochen oder sogar Monaten trainiert, geeignete Signale zu produzieren, die vordefinierten allgemeinen Gehirnmustern entsprechen. Die BCI Gruppe in Berlin und Potsdam, der ich angehöre, war in diesem Feld eine der ersten, die erkannt hat, dass eine Anpassung des Verarbeitungssystems an den Menschen mit Hilfe der Techniken des Maschinellen Lernens große Vorteile mit sich bringt. In unserer Gruppe und mittlerweile auch in vielen anderen Gruppen wird BCI somit als aktuelles Forschungsthema im Maschinellen Lernen und folglich in der Informatik mit interdisziplinärer Natur in Neurowissenschaften und anderen Feldern verstanden, da durch die geeignete Kombination von Techniken des Maschinellen Lernens und der Signalverarbeitung basierend auf neurophysiologischem Wissen der größte Erfolg erzielt werden konnte.
In dieser Arbeit gehe ich auf meinem Anteil an diesem Projekt ein, der vor allem im Informatikbereich der BCI Forschung liegt. Im Detail beschäftige ich mich mit den folgenden drei Punkten:
Diskussion eines informationstheoretischen Maßes für die Güte eines BCI's: Ich habe kritisch die Annahmen von Shannon's Informationsübertragungsrate für die Anwendung im BCI Kontext beleuchtet. Durch Ermittlung von geeigneten Kodierungsstrategien konnte ich zeigen, dass dieses theoretische Maß den praktisch erreichbaren Wert ziemlich gut annähert.
Transfer und Entwicklung von geeigneten Techniken aus dem Bereich der Signalverarbeitung und des Maschinellen Lernens: Eine substantielle Komponente meiner Arbeit war die Entwicklung von Techniken des Machinellen Lernens und der Signalverarbeitung, um die Effizienz eines BCI's zu erhöhen. Basierend auf dem neurophysiologischem Wissen, dass verschiedene unabhängige Merkmale in Gehirnsignalen für verschiedene mentale Zustände beobachtbar sind, habe ich eine Methode zur Kombination von verschiedenen und unter Umständen unabhängigen Merkmalen entwickelt, die sehr erfolgreich die Fähigkeiten eines BCI's verbessert. Besonders in einigen Fällen übertraf die Leistung des entwickelten Kombinationsalgorithmus die beste Leistung auf den einzelnen Merkmalen mit mehr als 50 %. Weiterhin habe ich theoretisch und praktisch durch Einführung geeigneter Algorithmen die Frage untersucht, wie viele Klassen man für ein BCI nutzen kann und sollte. Auch hier wurde ein relevantes Resultat erzielt, nämlich dass für BCI Güten, die bis heute berichtet sind, die Benutzung von 3 oder 4 verschiedenen mentalen Zuständen in der Regel optimal im Sinne von erreichbarer Leistung sind. Für eine andere Erweiterung wurden Ideen aus der Signalverarbeitung mit denen des Maschinellen Lernens kombiniert, da ein hoher Erfolg erzielt werden kann, wenn der temporale Filter, d.h. die Wahl des benutzten Frequenzbandes, automatisch und individuell für jeden Menschen angepasst wird.
Implementation des Berlin Brain-Computer Interfaces und Realisierung von geeigneten Experimenten: Eine weitere wichtige Komponente meiner Arbeit war eine Real-

isierung eines online BCI Systems, welches die entwickelten Methoden umfasst, aber auch so flexibel ist, dass neue Algorithmen und Ideen einfach zu verwirklichen sind. Bis jetzt wurden mit diesem System Bitraten von bis zu 40 Bits pro Minute von absolut untrainierten Personen in ihren ersten BCI Experimenten erzielt. Dieses Resultat übertrifft die bisher berichteten Ergebnisse aller anderer BCI Gruppen deutlich.

Acknowledgements

My first acknowledgement is addressed to Prof. Dr. Klaus-Robert Müller who gave me the opportunity to be a member of the BCI project in his Intelligent Data Analysis (IDA) group at the Fraunhofer Institute for Computer Architecture and Software Technology (FhG-FIRST). Although he was often very busy he found the time to help both in organizational and scientific questions. I appreciated in particular his advice and his enthusiasm for the project. Thanks, Klaus, for the support I have gotten from you during the last years.
My most special thanks are dedicated to Dr. Benjamin Blankertz, who is another leading figure in the BCI project. Benjamin introduced the main ideas of the project to me during our first meeting. After the discussion that followed I decided to try to become a part of this project since it sounded very interesting to me and I had the feeling that my work could be very helpful. Furthermore, having Benjamin as a colleague and advisor was very fascinating. Thanks, Benjamin; my first positive feeling about the project and you were confirmed during the past years.
I would also like to thank Prof. Dr. Gabriel Curio, who is responsible for the neurological part of the project. The discussions with him usually opened many new perspectives and gave new insights to me. Furthermore I am very grateful to his team members Dr. Florian Losch and Dr. Volker Kunzmann in this project.
Additionally, I would like to thank Prof. Dr. Gabriel Curio, Prof. Dr. José del R. Millán and Prof. Dr. Stefan Jähnichen for agreeing to be reviewers of this thesis.
A very special thanks goes to my two colleagues Matthias Krauledat and Dr. Anton Schwaighofer. I share a room with them and we had many fruitful but also funny discussions during this time and I got much help from them. I would especially like to thank Matthias for all the work and time we spent together.
I am very grateful to Thorsten Zander who informed me about the Brain-Computer Interfacing project and organized my first contact with the group of Prof.Dr. Klaus-Robert Müller at Fraunhofer. Without him I would never written my thesis in this area.
Furthermore I am very grateful to Christin Schäfer, Pia Philippi, Dr. Andreas Ziehe, Dr. Florin Popescu, Frank Meinecke, Dr. Gilles Blanchard, Dr. Guido Nolte, Dr. Gunnar Rätsch, Dr. Jens Kohlmorgen, Dr. Julian Laub, Konrad Rieck, Dr. Masashi Sugyama, Dr. Michael Schröder, Dr. Mikio Braun, Dr. Motoaki Kawanabe, Dr. Olaf Weiss, Patrick Düssel, Dr. Pavel Laskov, Pradeep Shenoy, Rolf Schulz, Dr. Roman Krepki, Ryota Tomioka, Dr. Sebastian Mika, Siamac Fazli, Sören Sonnenburg, Dr. Stefan Harmeling, Steven Lemm, Timon Schröter, Yakob Badower and all our students during the whole time. It was great to be a member in this group and this not only from a scientific point of view. Thanks to you all! I am especially grateful to Dr. Stefan Harmeling who knew everything about getting a PhD at the university of Potsdam and could help me with all the important formal parts.
A very special thanks goes to Prof. Dr. Jürgen Elstrodt who was the supervisor of my diploma thesis. His fascinating way of giving lectures was one reason for my good mathematical education.

On a private note I am very grateful to my family who supported me during this time. I would also like to thank all my relatives and friends for their help during this time and for the fun I had with them. My special thanks are dedicated to my wife Monika who was there for me in good and bad times and has helped me whenever I was unhappy.
To all of you, thank you very much for your unlimited help and support.

Contents

1 Introduction

Ich bin ein Cursor ***(I am a cursor)*** was the title of a long article about the Berlin Brain-Computer Interface in the well known German newspaper, *Frankfurter Allgemeine Zeitung*, on 17th March 2004 (see [125]). A journalist visited us in Berlin and tried to use the interface, i.e., control a computer purely by thought alone. After less than one hour of training he was able to move a cursor horizontally on a computer screen reasonably well which made him want to publish his experience in the newspaper. Although a horizontal movement on a screen is not enough to be convenient for normal users as an additional communication path, it opens up new perspectives in this direction for the future, assuming the performance and measurement technique can be further improved.
Whereas in the beginning of Brain-Computer Interface (BCI) research the main contributions were made by neuroscientists and psychologists, this research field has become a hot topic in computer science with interdisciplinary links to neurophysics, psychology, electrical engineering and other research fields. To explain this I will start in this chapter by defining the goals of a Brain-Computer Interface (cf. section 1.1) and by describing the interdisciplinary nature of BCI research (cf. section 1.2). Afterwards I will introduce an overview of common approaches (cf. section 1.3) for readers not familiar with this topic as well as results achieved by other BCI groups (cf. section 1.4) in this relatively new research field without claiming to be exhaustive. A more complete overview can be found in Wolpaw et al. [141] or in the forthcoming book Dornhege et al. [49]. By defining the main ideas of the BCI approaches of the Berlin group which I belong to, the focus is turned to the computer science part in this research field (cf. section 1.5). Finally I will discuss my contribution to BCI research which directly leads to the outline of this work (cf. section 1.6).

1.1 Goal of a BCI system

The beginning of Brain-Computer Interface research goes back to the early 1970s. At that time Jacques Vidal defined a brain-computer interface by a computer-based system that produced detailed information on brain functions (cf. [133, 134]) and built a first BCI based on visual evoked potentials (cf. section 1.3.3). During the last decades the definition and the goal of a BCI has been refined and specialized. The most recent version is given by Wolpaw et al. [141]. Here a BCI is a system for controlling a device (e.g., a wheelchair, a neuroprosthesis or a computer) by human intentions without using activity of muscles or peripheral nerves (see Fig. 1.1).
Previous systems were mainly developed for patients suffering from several disabilities, especially for amyothropic lateral sclerosis (ALS) and spinal cord injuries if they have lost all other communication abilities to the outside world. If the brain is intact a BCI might be the last opportunity for them to communicate with other people. A BCI could also help patients like amputees to lead a more comfortable life. Recently, many groups have suggested using

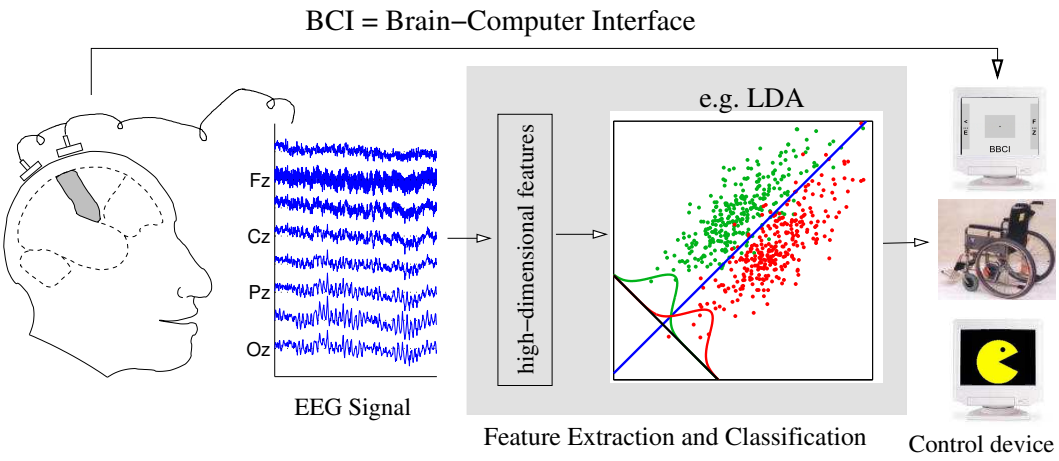


Figure 1.1: In a BCI the EEG signal is recorded, processed and classified such that a control device like a speller, wheelchair or game can be controlled. By the visual feedback provided for the user of the control device enhancements in performance due to the learning capabilities of humans can be achieved. The machine learning part of the interface is visualized in gray. In most BCI systems only a fixed setup without machine learning and thus without computer adaptation is used instead.

a BCI system for healthy people as a further communication path for gaming or in real life. However, the functionality of a BCI is so far very limited such that a BCI system is not convenient in workplace applications. Nevertheless, recent results have given reason to hope that the system can be improved to be useful for healthy users too (cf. [21, 22, 89]).

1.2 Contributing research areas

BCI is an interdisciplinary research area to which many researchers from different fields can contribute, e.g.

- Electrical Engineering: To make a BCI attractive to a user a suitable measurement technique has to be established, i.e., with small preparation times for the user, high quality and convenient ongoing use.

- Neurophysiology: One important point in BCI research is the specification of paradigms and the localization of EEG features. Furthermore BCI research can help to understand the main functions of the brain.

- Psychology: Since a user of a BCI is directly confronted with feedback on his own brain rhythms which he is able to modulate, this is also an interesting field for a psychologist, e.g., by discussing learning techniques, or by explaining human behavior.

- Computer science with subdisciplines:
  - ☛ Software Engineering: To allow feedback control one has to implement an interface which records the EEG data, applies some algorithms and techniques and finally controls a graphical feedback.
  - ☛ Signal Processing: Since EEG signals are time series, advanced signal processing techniques are necessary to reveal the relevant part of the signal.
  - ☛ Machine Learning: If one is interested in the machine adapting to the users requirements, machine learning techniques should be established and adjusted for use in this field.
  - ☛ Information Theory: Since a BCI opens a communication channel from the user to a machine one can try to discuss and evaluate the performance of the system by means of submitted information.

Consequently, BCI research contains research in physics (Electrical Engineering), medicine (neurophysiology, psychology), computer science and various other fields.

1.3 Several techniques to achieve a BCI

Many different approaches and realizations of a BCI system in the world exist. They can be grouped in several ways. For example, different measurement techniques exist, starting from invasive methods using electrodes within the brain, going on to methods where electrodes were placed subdurally, i.e., below the skull and above the brain and finally by non-invasive methods like the electroencephalogram. Of course the target group of a BCI system really depends on the measurement technique, since healthy people are usually not interested in implanting electrodes in their brain. A short overview about common measurement techniques are given in section 1.3.1. Further distinctions in BCI directions are given by the way to use the training capabilities of the human and the computer (see section 1.3.2) or by using evoked potentials or unstimulated brain signals (see section 1.3.3). Further possible distinctions are discussed in section 1.3.4.

1.3.1 Measurement techniques

All available acquisition techniques can be ordered by the extent of invasiveness of the method and thus by the size of the target group of such a system. Nevertheless, with higher invasiveness comes an increase in the quality of the signal such that these systems can not be ignored, especially for disabled patients. A further distinction is given by recording based on neuronal or vascular blood-flow activity. Hereby also a differentiation is made between the temporal resolution of the measurement techniques. Whereas neuronal activity can be measured and also controlled in a range of milliseconds, the temporal resolution of vascular activity is quite poor, i.e., it lies in the range of seconds. In this section I will first introduce two invasive methods based on neuronal activity, namely Multielectrode Arrays and Electrocorticograms. In a second step I will introduce three methods based on vascular activity, one of them with a low risk for humans, Positron Emission Topography, and two non-invasive methods, Functional Magnetic Response Imaging and Near Infrared Spectroscopy. Finally

I will introduce two methods which use the neuronal activity of the brain non-invasively, Magnetoencephalography and Electroencephalography.

Multielectrode Arrays. In this case microelectrodes are used to record action potentials of single neurons in the cerebral cortices. Since this method is highly invasive most studies were done in animals like monkeys. By presenting monkeys suitable feedback about the firing rate of single neurons they were able to learn to control the feedback. Consequently, the expectations arise that humans could develop a similar control (see [40, 104, 126]). As announced in a talk at the Neural Information Processing Systems Conference 2004 in Vancouver, Canada, a first group has recently implanted these electrodes successfully in a disabled human who was then able to control a device. But these results are not published so far. One big problem with these electrodes, besides the risk of the invasive approach, is given by the fact that the electrodes move relatively to the individual neurons and induce scar tissue, so that over time neurons come and go or the recording deteriorates. Consequently this leads to a decrease in performance or complete loss of control abilities.

Electrocorticogram (ECoG). For this measurement technique an electrode grid is placed subdurally below the skull. With this method one cannot measure the firing rate of single neurons any longer. Similar to the EEG (see below) one can measure the electrical field above the brain but with a better signal to noise ratio. Usually these electrodes are used for finding areas of epileptic seizures. But recent approaches also suggest the ability of ECoG electrodes for BCI control (see [56, 76, 82]).

Positron Emission Topography (PET). By using radioactively marked chemical substances like glucose the chemical functioning of an organ can be observed. One can use this techniques for the brain, too: brain areas which need glucose can be specified. Thus active brain regions can be determined. However, there is a small risk to the human due to the use of radioactively marked chemical substances. Furthermore this system has long time constants in the range of seconds since it measures vascular activity. Additionally PET is technically very demanding and expensive.

Functional Magnetic Response Imaging (fMRI). Hydrogen atoms are an essential component of the human body. These atoms are dipoles and thus produce a small magnetic field. Based on a strong magnetic field and radio waves these dipoles can be influenced and thus image slices about the hemoglobin flow can be computed. Usually the spatial resolution is very high, whereas the temporal resolution is quite poor in the range of seconds. Furthermore this method is technically very demanding and expensive and not applicable in common environments.

Near Infrared Spectroscopy (NIRS). The transmission of photons which impinge on biological materials depends on the combination of reflectance, scattering and absorption effects. The relatively good transparency of biological materials in the near-infrared (NIR) region of the spectrum permits sufficient photon transmission through organs in situ for monitoring cellular events. Furthermore, it has been known for many years that some intrinsic changes in the optical properties of brain cells are dependent on blood flow and electrical activity. Thus changes in brain activity can be determined by this technique. Unfortunately the spatial resolution of this technique is very poor, so far. Furthermore, another disadvantage of this technique lies in the delay and thus in the quite poor temporal resolution (in the

range of seconds) due to the neurovascular coupling.

Magnetoencephalography (MEG). In this case the magnetic field induced by electrical currents in the brain are measured directly at the head. It allows a high spatial and temporal resolution and provides similar signals as the EEG (see below). Unfortunately, compared to the EEG the technique is still very expensive and technically demanding since the MEG is highly distorted by movement artifacts and therefore a shielded room and huge equipment are required to have a sufficient quality of recording.

Electroencephalography (EEG). Based on the ongoing electrical activity of large populations of cortical neurons, voltage fluctuations as the sum of this activity can be measured by surface electrodes on the head. Consequently, changes in activity in regions close to the skull can be observed if the corresponding electrical dipole has the correct direction. The spatial resolution of an EEG system is quite good. Since EEG recordings are based on neuronal activity the temporal resolution lies in the range of milliseconds. Additionally, it can be technically easier and cheaper to realize than the methods discussed above. Therefore Wolpaw et al. [141] concludes that only the EEG is able to establish a practical BCI so far. Nevertheless, the preparation time of such a system is too high, so far, to be convenient for healthy users.

1.3.2 Subject or Machine training

In the beginning of BCI research the system usually worked on a-priori-defined neurophysiological features (cf. [133, 137, 11]). In this environment the subject is confronted with feedback based on a fixed setup such he is able to find how he can control the system, i.e., he has to learn to produce neurophysiology like the average human to control the system. In several studies (cf. [137, 11]) it was reported that subjects are able to do so within weeks or even months of individual training due to the adaptivity of the human brain. Recently, several groups came up with the idea to also adapt the system to the subject-specific brain functions such that control becomes easier (cf. [19, 90]). Here two different approaches exist. The first one starts with presenting general bio-feedback and adapting the system. The other one uses machine learning techniques based on one initial training session to have an individually optimal setup. For the latter adaptation during further feedback is also possible. Of course during feedback the human capability to learn will also enhance performance of the system. But usually the system starts on a higher level, i.e., with a higher performance, such that learning becomes more attractive and easier for the user.

1.3.3 Evoked Potentials or unstimulated brain signals

Several neurophysiological features exist which are relevant candidates for a BCI system. The resulting BCI systems can be grouped into two main fields: BCIs based on evoked potentials like SSVEP and P3 (see below) or BCI based on unstimulated brain signals like ERD/ERS of oscillatory features or ERPs of slow cortical potentials (see chapter 2).
The main difference between these two groups consists of the following: Evoked Potentials are based on stimuli given by the outside world. For example if a human focuses on a light, which blinks or changes color in a specific frequency, the same rhythm can be observed in the occipital area of the brain. This effect is called Steady State Visual Evoked Potentials

(SSVEP) (see [87, 31]). Another prominent evoked potentials is the so called P3 component. Here a user is confronted with a lot of standard and some rare deviant stimuli. After a short negativation a high positive peak arises at 300 ms after a deviant. Successful BCI applications based on the P3 phenomena can be found in Farwell and Donchin [52] and Donchin et al. [39]. Evoked potentials, especially SSVEP, require stable control of the eye muscles such that it is not applicable to all users. Furthermore, this control is reported by several subjects to be inconvenient and it also puts heavy restrictions on the range of possible applications.
Another way of BCI control is the use of unstimulated brain signals. Here the subject is called to change his mental states which can be detected by the system and used for control. For example, based on the imagination of left hand movement an attenuation of some rhythms in the right hemisphere can be observed and used for, say, controlling a cursor. Compared to the visual evoked potentials, stimulation of brain signals by sources from the outside world are not required any longer. It only depends on the active control and intentions of the user and thus could be more convenient for controlling devices.
The focus of my work and of the BCI group in Berlin lies on unstimulated brain signals. Thus a broader focus in this work is dedicated to this way to implement a BCI.

1.3.4 Other options for a BCI

Other distinctions for a BCI exist. For example, one can ask if the control should work independently of other output paths or not. So far the sole control of the system is the goal of many groups. However, combined use can be an interesting add-on in future applications. A further distinction is given by the fact whether control should only be possible at a predefined pace (synchronous control) or during the whole time (asynchronous control).

1.4 BCI - State of the art

During the last decades many new research groups were formed which work in the BCI field. Therefore I can only give a short overview of the most successful ones. I will briefly discuss the results of invasive methods in section 1.4.1. For non-invasive methods I have chosen the four prominent research groups led by Prof. Jonathan Wolpaw in Albany (see section 1.4.2), Prof. Niels Birbaumer in Tübingen (see section 1.4.3), Prof. Gert Pfurtscheller in Graz (see section 1.4.4) and Prof. José del R. Millán in Martigny (see section 1.4.5). Furthermore I will describe the Berlin approach in section 1.5.

1.4.1 Invasive methods

Most studies with invasive methods have been done with monkeys. In doing so a monkey is usually trained to move a prosthesis instead of its own arms which are fixated. In the beginning attempts to move the arm are usually clearly visible but after some time when the control of the prosthesis becomes more precise the monkey usually stops moving its own arm and only moves the prosthesis by its *thoughts*. It was reported that monkeys are able to learn to control the arm in 3D nearly perfectly if feedback is provided. Recently a monkey was additionally able to use the prosthesis for grasping in such a skilled way that it could

eat fruit. For more detailed information I refer to Donoghue and Sanes [40], Nicolelis et al. [104], Schwartz [126]. It is expected that one can transfer these results directly to humans. Recently it was reported at the Neural Information Processing Systems 2004 conference that for the first time a human was able to control a BCI by multi-electrode arrays, but a more detailed publication is not available so far.

1.4.2 The Wadsworth BCI

The Wadsworth BCI supervised by Prof. Jonathan Wolpaw uses the Event-Related Desynchronization of the $\mu-$rhythm during real or imagined movements. Based on a fixed setup the user of the system was able to move a cursor into one of two to four different targets on the right side of the screen relatively and vertically whereby the cursor moves with constant speed from left to right. Hereby the movement is controlled by suitable desynchronization of the $\mu-$ rhythm which the subject has to train over weeks using this feedback scenario (cf. [140]). After many feedback sessions subjects were able to achieve over 90 % hit rates for the binary decision problem with a selection rate of 4 to 5 seconds. Recently a first approach of controlling a cursor on a screen vertically and horizontally at the same time was presented in Wolpaw and McFarland [136]. Here the user was trained during many feedback sessions to control the device by suitable modulations of $\mu-$ and $\beta-$rhythm.

1.4.3 The Thought Translation Device (TTD)

The Tübingen Thought Translation Device (TTD) (cf. [11, 12, 74]) enables subjects to learn self-regulation of the slow cortical potentials (SCP) at central scalp positions during many feedback sessions. Here a cursor is controlled vertically by the negativation of EEG and patients are able to generate binary decisions with an accuracy of up to 85 % with a 4–6 second pace. Recently they (cf. [76]) have also reported that they have transferred their results to ECoG measurements successfully. It should be mentioned that the main focus of the Tübingen group lies in establishing a BCI for locked-in patients to enable for them a possibly sole communication channel to the outside world (see [11]).

1.4.4 The Graz BCI

The user of the Graz BCI system is able to control a device based on the modulations of the pericentral μ- and/or β-rhythms of sensorimotor cortices similarly to the Wadsworth BCI. While the Wadsworth BCI directly presents the power modulations to the user, the Graz BCI for the first time also uses machine adaptation for the control of the BCI. In Peters et al. [108] it was reported that they obtain accuracies of over 96 % for a ternary classification task with a trial duration of 8 seconds by evaluation of adaptive auto-regressive models (AAR). Recently they were also able to allow the grasping of the non functional arm of a disabled patient by Functional Electrical Stimulation (FES) of the arm controlled by EEG signals (cf. [95, 100, 111]).

1.4.5 The Martigny BCI

José del R. Millán (cf. [91]) with his machine learning background started some years ago, in parallel to the Berlin BCI, to introduce advanced machine learning methods into

the BCI thereby adapting the machine to the human and not vice versa. In Millán et al. [91] he suggests to use a local neural classifier based on quadratic discriminant analysis for the machine learning part. After a few days of training three subjects were able to achieve by imagination of left or right-hand movement or by relaxation with closed eyes in an asynchronous environment an average correct recognition rate of about 75 % whereas the wrong decision rate was below 5 %. Hereby it was possible to control a virtual keyboard and choose a letter approximately every 22 s for trained subjects and to control a motorized wheelchair (cf. [89]). In Millán et al. [92] they have added three further classes (cube rotation, subtraction and word association) and have exchanged relaxed with closed eyes to relaxed with opened eyes[1] and usually choose the best three class subset with successful effects. With these three classes, they control a robot which is moving in an artificial maze. The robot can be turned left and right, and a third option is to move the robot forward. In their control scenario, the user gives some control autonomy to the robot, i.e., if the robot is approaching the wall, it automatically turns into another direction and does not allow for turning towards the wall (see [92]).

1.5 The Berlin Brain Computer Interface

Since 2000 a small group based on a partnership between the Department of Neurology, Campus Benjamin Franklin, of the Charité Berlin represented by Prof. Gabriel Curio and the Intelligent Data Analysis Group at Fraunhofer FIRST (formerly GMD FIRST) represented by Prof. Klaus-Robert Müller and Dr. Benjamin Blankertz have started to implement a BCI driven by the idea of transferring the effort of training from the human to the machine. Based on both advanced machine learning techniques and neurophysiological knowledge they started by detecting and discriminating movements of different hands before the actual movement. They have reported (cf. [17, 19]) that classification rates of about 90 % between left and right hand keypress could be achieved more than 200 ms before keypress. The value of these results, for example in the area of safety technology, is obvious. Furthermore these results were achieved after less than one hour of recording data, so that the need of subject training is no longer essential.

Subsequently they transferred the idea of using machine learning techniques to reduce the amount of subject training to the BCI controls as described in the sections above. Based on the imagination of different motor tasks, using a priori neurophysiological knowledge about the accompanying ERD effects in the $\mu-$ and $\beta-$ rhythm and negativation effects in the SCP and introducing advanced machine learning techniques they were able to present promising offline results in Dornhege et al. [43, 44, 45]. Recently they were also able to achieve an accuracy of more than 95 % at a pace of about 2 s in a real online controlled binary decision task for an absolutely untrained subject during his first BCI feedback experiment (see [21, 22] and section 4.2). An extended review about their work is presented in Blankertz et al. [19, 23].

[1] Although closed eyes produce a strong ERS in the $\alpha-$rhythm which is usually easy to detect and thus a very good class to achieve good performances opened eyes are required for visual feedback environments.

1.6 My work in this project - Outline of this thesis

I have been working as a member of the Berlin group on the BCI project since 2002. My research interests mainly lie in the computer science part of the project. However, without understanding some important facts from electrical engineering and neurophysiology one can not contribute to this area. Thus important neurophysiological insights will be introduced in chapter 2. This chapter is mainly dedicated to readers who are novices in this field to understand the further parts in this work.
My main contributions to the project which I describe in the consecutive chapters consist of three points:

- **Establishing a performance measure based on information theory:** Many parameters are important evaluating the effectiveness of a BCI system, e.g., the preparation time and the comfort of the acquisition technology for the user, the mobility and price of the system, the degree of invasiveness, the time needed to be able to use the system, the duration a user can use the system and the amount of control, i.e., the amount of transmittable information. A suitable combination to compare all of these parameters has not been established so far, and is not obvious, either. In this work I only focus on the latter, the amount of transmittable information since the other parameters are fixed in our BCI system. Here I will use the information transfer rate provided by Shannon which I introduce and discuss in chapter 3. My work at this point does not consist of defining this measure. It consists of making this rate reasonable for BCI research. Shannon's bitrate is a theoretical measure which calculates the possible amount of data transfer via a noisy channel. By suitable codings Shannon finds a way to calculate the expected information transfer rate. Unfortunately these codings have to be performed by the source of the channel which is in the case of a BCI the user of the system. Consequently, the problem arises that codings that are too complex cannot be managed by the user. Therefore this theoretical measure cannot be achieved in a BCI. In chapter 3 I will discuss this measure and its problems in more detail. Furthermore, I will discuss solutions by finding optimal codings useable by humans and calculate their performance. Although they cannot achieve the Shannon information transfer rate they show that this measure is not too far away from realistic values. Thus Shannon's bitrate can be used as one reasonable measure.

- **Implementation of the BBCI and realization of suitable experiments:** One major part of BCI research are the experiments, which are described in chapter 4. Usually a BCI experiment in our group consists of two steps, the calibration measurement (also called training session for the computer) and the feedback experiment (also called online experiment). In the calibration measurement some labelled trials are provided, i.e., examples of some predefined mental states are recorded, without informing the subject about the characteristics of his brain rhythms. This calibration measurement serves as the training set for the machine learning techniques: Based on this data a subject and paradigm specific machine is built which should classify further mental states very well. The machine learning techniques used will be addressed in the next point of this enumeration. In the subsequent feedback session the user is confronted with information about the decision of this machine about his current mental state. Obviously, the success of a BCI has to be

shown in this feedback session, i.e., by the ability to control devices online. Thus one important part of my work was the implementation of the online interface. I will shortly introduce the ideas and main parts of this interface in chapter 4. Additionally I will report about the first successful online feedbacks which were performed with this interface.

➢ **Transfer and development of suitable signal processing and machine learning techniques:** In chapter 4 it was stated that the machine has to adapt to the subject based on the calibration measurement to provide suitable feedback. I haven't yet properly introduced how the machine learning part of a BCI works. Since it is a very important part of my work the next chapters of this thesis are dedicated to this topic. I will start in chapter 5 by making the reader more familiar with general ideas of machine learning. In the consecutive chapters 6, 7 and 8 I present three special BCI algorithms which I have developed:

① *Combining different features:* A review of the literature shows that many different features are used for control of a BCI which might contain different amounts of information for the same task. This field was examined in Dornhege et al. [43] for two features, namely for the slow cortical potentials and oscillatory effects during imagination of movements. It was found that the features are uncorrelated from each other. Nevertheless, nobody has used this fact to enhance the performance of a BCI. So far only BCIs based on one feature have been implemented. In chapter 6 I will discuss theoretically why the use of different uncorrelated features can enhance the performance of a BCI and to what degree. Furthermore I suggest methods and compare their results to the single feature results. Especially the theoretical solutions could be confirmed. Thus if the performance of the features is similar, high enhancements in performance can be achieved. With these algorithms I was able to win on one dataset for the BCI competition I (cf. [120]) and to be the second winner on one dataset from the BCI competition II (cf. [15, 20]). Furthermore several groups have used this idea for the BCI competition III (cf. [16]) and the winning approaches usually contain some feature combination techniques.

② *Using more than two classes/mental states:* One alternative step to increase the performance of a BCI system consists of the number of classes used. Obviously with more classes more information can be transferred, but unfortunately the quality of discrimination decreases. In chapter 7 I will illuminate the question of how many classes should theoretically be used for a BCI to achieve the highest information transfer rate. Furthermore I will discuss some interesting ideas to extend successful binary algorithms to multi-class ones and compare their results. Finally I will practically confirm the theoretical results about the number of classes used for a BCI system: With the performances reported so far for BCIs, 3 or 4 classes are usually the best choice (which is of course highly subject-dependent).

③ *Fitting spatio-temporal filters:* Another alternative to enhance performance is investigated in chapter 8. Usually the optimal choice of the frequency band for discriminating ERD effects varies strongly between different subjects. I will describe this problem in more detail and will suggest and compare several methods in that chap-

ter. Interestingly, the most successful one combines ideas of signal processing with machine learning techniques by using filter techniques and optimization approaches with sparsity constraint.

Note that the third point I will discuss in this work – the machine learning and signal processing part – is required for the second point, namely for the successful realization of the Berlin brain computer interface. However, without an understanding of the goal and the experiments carried out, main issues of this third part do not become totally clear. Thus I have decided to present the interface with the achieved results before I discuss the machine learning part.
In chapter 9 I will summarize this work and its results and try to give an outlook about possible future directions and opportunities in this field.
Although BCI research contains many different research fields as discussed in section 1.2 and is an interdisciplinary field, main parts of my work presented in this thesis are in the field of computer science: The development of algorithms for signal processing and machine learning, the theoretical analysis of the communication channel, the allocation of suitable BCI feedback environments and the implementation of the interface are located in main subareas of computer science and can also help to improve results in common computer science research.

2 Neurophysiological Background

To understand the options and challenges in BCI research, one should start with analyzing the main processes in the human brain. The goal of this chapter is the introduction of the most prominent features which are being used for a BCI so that the terminology can be used in further explanations, however it does not claim to be complete. An attempt at a complete taxonomy was made at the BCI meeting 2005 in Albany (see [1]). The approaches can be roughly grouped into two types, namely the event related potentials discussed in section 2.1 and the oscillatory features discussed in section 2.2. Since the Berlin Brain Computer Interface focuses on motor-related brain activity I will discuss the differences between real and imagined movements in section 2.3. Finally the important role of a closed loop feedback system for a BCI will be illuminated in section 2.4.
All features discussed below have a specific temporal structure but also a specific location in the brain, i.e., different brain areas are responsible for different tasks. Fig. 2.1 shows a coarse overview over some important brain areas.

2.1 Event Related Potentials (ERP)

According to the most widely accepted model (see [10]), ERPs are signals generated by neural populations which are activated time-locked regarding some event. This is reflected in a modification in the electrical activity of some brain areas in the time-domain. These EEG changes are also referred to as Slow Cortical Potentials (SCPs). Usually one denotes by a stimulus the event the subject reacts to, and by a response the action of the subject. ERPs are obtained by calculating the average about many independent recorded trials locked to the stimulus or response event. Hereby the assumption of independent trials reduces the signal to noise ratio by a factor of $\sqrt{\text{Number of trials}}$ which can be easily seen by the law of large numbers and the central limit theorem so that the underlying relevant signal is revealed if enough trials are given. Nevertheless, in addition to the detection of ERPs without having the exact time-points of the event one major challenge is the classification on a single-trial level. I will introduce in the following the P300, the error potential and the readiness potential based on fixed triggers. All these neurophysiological phenomena are largely studied in the literature. A first attempt at classification on readiness potentials without knowledge of the exact timepoint can be found in section 4.2.3.

2.1.1 P300

The P300 (or P3) is a very prominent component whose existence has been known for a long time (see [127]). Therefore many extensive reviews to this topic exist, e.g., [114]. Although it is not yet clarified which cognitive functions are reflected by this component, there are well established theories about it. One has observed that the P3 usually consists of two peaks,

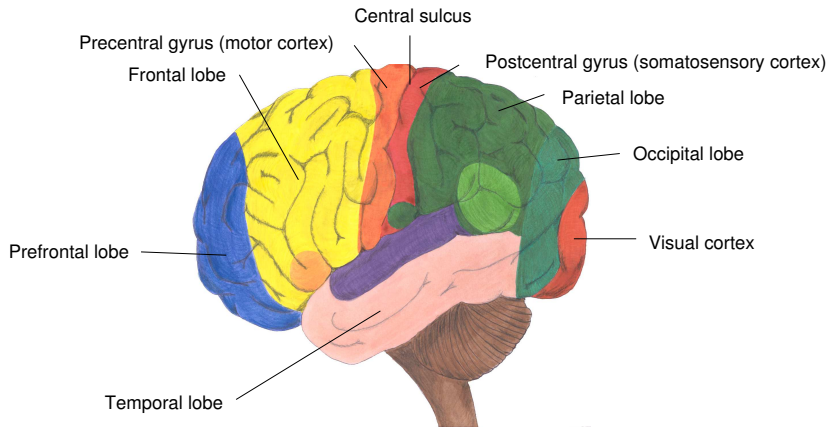


Figure 2.1: The figure shows a coarse overview taken from Krepki [73] about some important brain areas.

the P3a and P3b. The first component is more strongly pronounced for novel stimuli over the frontal and central cortex and is thought to reflect an alerting process originating in the frontal cortex (see [34]). To reveal the structure of the P3 component the oddball paradigm is usually used. In this paradigm standard stimuli are presented frequently to the subject, which are interrupted in random order by rare non-standard objects called deviants. Here the P3b component is strongly pronounced in response to deviant objects. If in this oddball paradigm one non-target, i.e., a novel target, is presented a very pronounced P3a component is observed (see [34, 32]). The P3 component consists of a positivity approximately 300 ms after the stimulus. This component can be observed for both auditory and visual stimuli. Knight [68] suggests that the P3a reflects the interruption of the usual brain processing by infrequently presented stimuli.
In Donchin et al. [39] the use for BCI of the P3 in a speller paradigm has been demonstrated: All 26 letters and 10 digits are visualized on a computer screen in 6 rows and 6 columns. The user is asked to focus his attention on the desired letter. The rows and columns are flashed in random order several times. A P3 component can be expected after flashing the focused row or column. Thus one could choose a letter after suitable detection of the P3 component.

2.1.2 Error Potentials

Another component called error potential with similar structure as the P3 component appears during the process of evaluation of the correctness of an event. The brain reaction varies strongly if the event contains an error compared to correct events. Mainly two deviating components can be observed: a negative wave called error negativity and a following broader positive peak called error positivity (cf. [51]). The latter looks similar to the P300. The error negativity can be observed in both correct and wrong trials with a delay and is less intense for correct trials, whereas the error positivity can be only observed in wrong trials and thus is more specific to errors. Falkenstein et al. [51] and Nieuwenhuis et al. [105] claim that the error negativity component reflects some kind of comparison process, while the error positivity can be connected to conscious error detection.

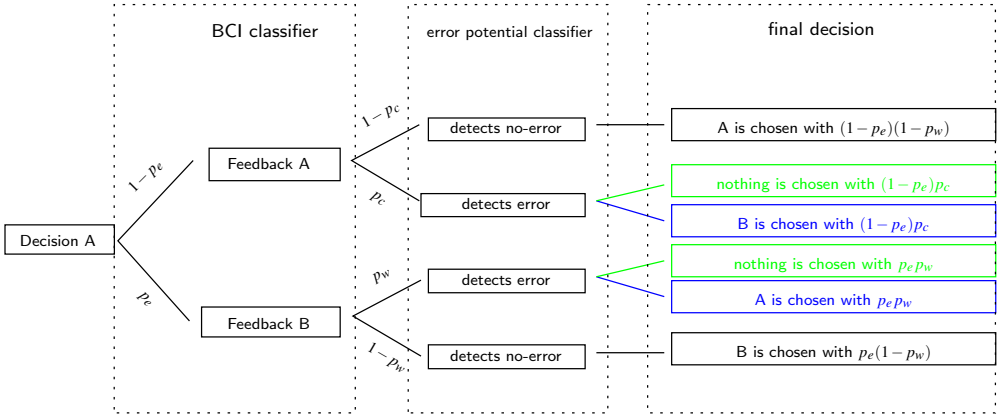


Figure 2.2: The figure visualizes the decisions in a BCI with error detection. Based on two possible decisions A and B in this case the user wants to have decision A. With some probability $1-p_e$ the BCI system recognizes this decision correctly and presents A as feedback, otherwise B. Since the user is confronted with this decision he is able to evaluate the correctness of the BCI system. If it is not correct an error potential appears, otherwise not. An error potential detector can detect this to some degree. If it does not detect an error potential the visualized decision is confirmed, otherwise two options exist: the decision is ignored and a new run starts (green) or the other decision is chosen (blue).

Schalk et al. [122] and Blankertz et al. [18] suggest using the error potential as an add-on to other active controlled BCI-systems. Usually control of a BCI-system with presented feedback is accompanied by errors. Let us assume that this error appears with probability p_e which is of course highly subject and feedback dependent. If an error occurs one usually expects the error potential which can be detected by a suitable P3-detector. In this case one can repeat the old decision or choose the other one in the two class case. Under the assumption that this system detects a correct trial as error with probability p_c, and a wrong trial as error with probability p_w, the probability to make an error in the binary case with flipped decision is given by $p_e(1-p_w)+(1-p_e)p_c$ or in the *repeat*-case by $p_e(1-p_w)/((1-p_e)(1-p_c)+p_e(1-p_w))$ regarding all accepted trials but with the need of more runs by a factor of $((1-p_e)(1-p_c)+p_e(1-p_w))^{-1}$ (see [53]). This setup is visualized in Fig. 2.2. Obviously there exist error rates where this error correction is worthwhile. In Fig. 2.3 the situation is shown for $p_c = 0.03$ (an error potential is detected after a correctly classified trial with a probability of 3 %) and $p_w = 0.8$ (an error potential is detected after a wrongly classified trial with a probability of 80 %). The resulting bitrate per decision is shown with varying accuracy of the BCI system between 50 % (random classification) and 100 % (perfect classification). One can observe that in this special case a decision flip after detection of an error potential should be preferred if the accuracy is below 87 %. With higher accuracies repetition of trials should be preferred, until with 99 % no error correction is advisable anymore.

As a preliminary study for Blankertz et al. [18], a so called d2-test was performed. Here the letters d and b with up to two horizontal lines above and below the letter were visualized. The user has to react as fast as possible by pressing a key with the left hand if a d with two

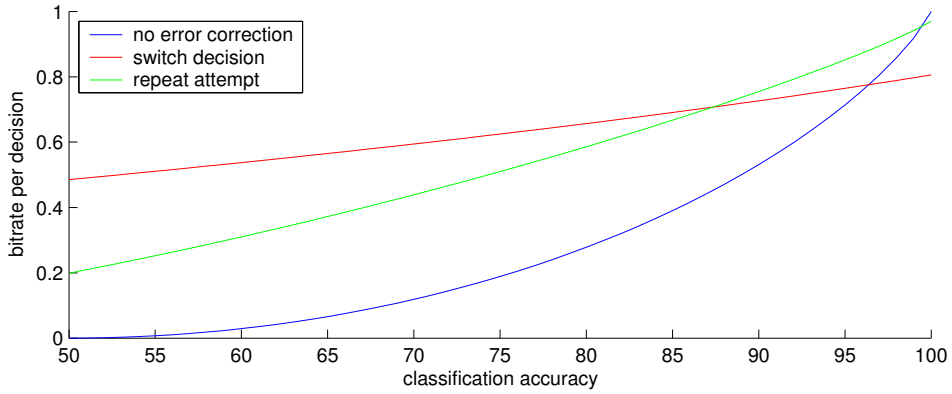


Figure 2.3: With $p_c = 0.03$ and $p_w = 0.8$ the resulting bitrate per decision is shown if no error correction is performed, a decision is switched and an attempt is repeated, if the error potential is detected. The classification accuracy is varied on the x-axis. For a given accuracy the algorithm with highest value should be chosen. Note that for the repetition of decisions the higher amount of trials is suitable recognized by multiplying $(1-p_e)(1-p_c)+p_e(1-p_w)$.

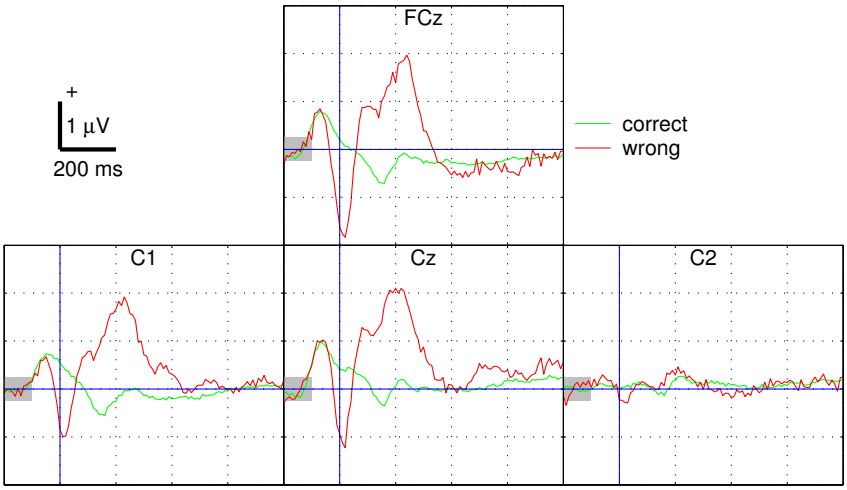


Figure 2.4: In the figure the ERP for one subject is plotted. At timepoint zero (blue line) the subject has pressed a button based on the command given by an $d2$-test. The correctness of the button press is immediately visualized by a green (correct) or red (wrong) flash. In the figure the ERP for correctly and wrongly answered trials is shown. It reveals the expected error negativity/positivity complex for the wrong decisions.

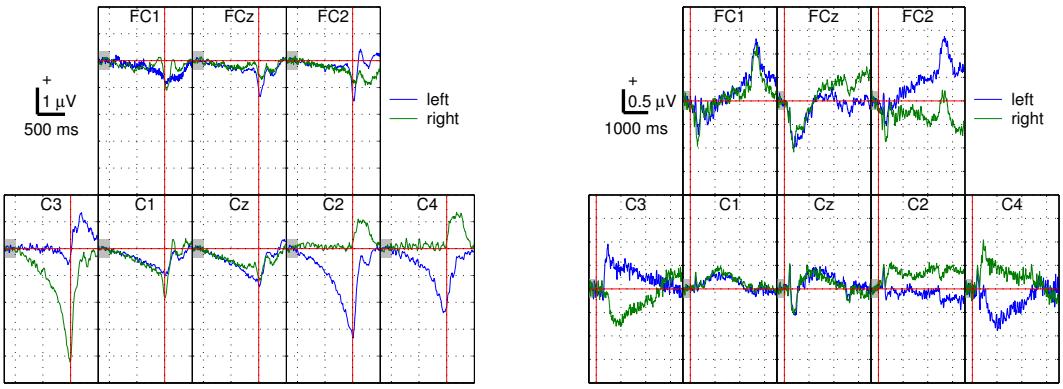


Figure 2.5: On the left an ERP for one subject for left hand and right hand movements is shown which reveals the contralateral negativity preceding the movements. The movement was performed at timepoint zero (red line). On the right the ERP for imagined left and right hand movements is visualized that shows the same contralateral characteristics. The task was ordered at timepoint zero (red line).

bars appears, otherwise he should press a button with the right hand. Afterwards the screen flashed green if the answer was correct, red if it was not correct. One could clearly observe after a wrong decision the error potential as visualized in Fig. 2.4. It should be mentioned that the subjects in that experiment usually recognized the error before they pressed the button but were not able to stop the decision any more, so that the error positivity/negativity complex is pronounced so early in the figure.

2.1.3 Readiness Potential (Bereitschaftspotential)

Self-initiated movements are preceded by the readiness potential (Bereitschaftspotential, RP) in the mesial fronto-central cortex including the supplementary motor area and in the primary motor cortex. The amplitude for the latter is contralaterally more strongly pronounced than ipsilaterally (cf. [38]). These results were backed by studies in nonhuman primates (cf. [128, 118]). The functionality of different brain areas preceding a self-initiated movement is controversial in the literature, see Deecke et al. [38], Lang et al. [79], Cui et al. [36] for a broad overview.
Cui et al. [36] has shown that the RP can start at about 1.5 s before movement onset over the medial-wall motor area (supplementary motor area). Here a reference far away from the supplementary motor area was chosen. 750 to 500 ms before movement onset the topography of the RP changes by an increasing lateralization more pronounced contralaterally. This effect is called Lateralized Readiness Potential (LRP).
This is visualized on the left of Fig. 2.5 for left or right index or little finger movements for one subject at some electrodes, which corresponds to locations where this neurophysiological effect can be observed. In this experiment the subject was asked to press buttons on a keyboard, self-initiated in an arbitrary order at an approximate pace of 2 s. See chapter 4 for a more detailed description of the experimental setup. In the figure the means over many trials separated between left and right hand finger movements are distinguished and triggered at the key-press (zero point).

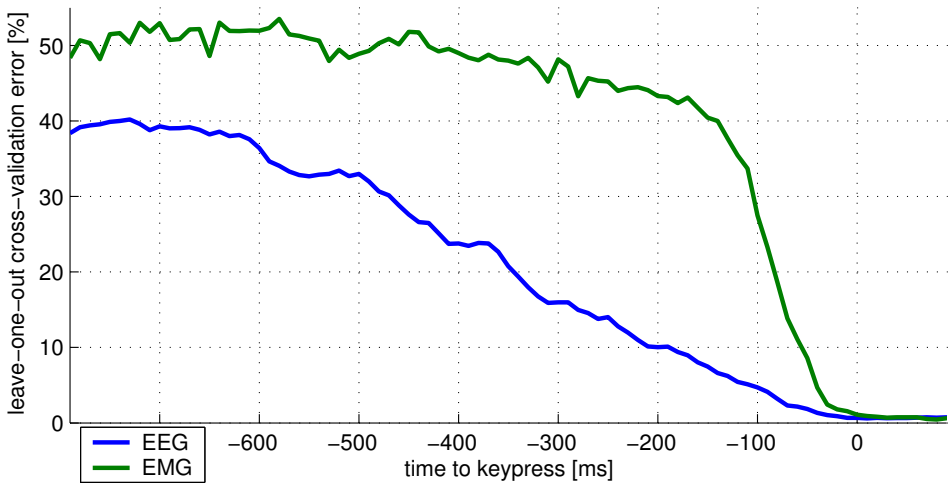


Figure 2.6: The figure shows leave-one-out cross-validation errors based on classification on EEG activity or EMG electrodes fixed at the arms up to a specific time regarding key-press between left and right finger movement during a selfpaced experiment.

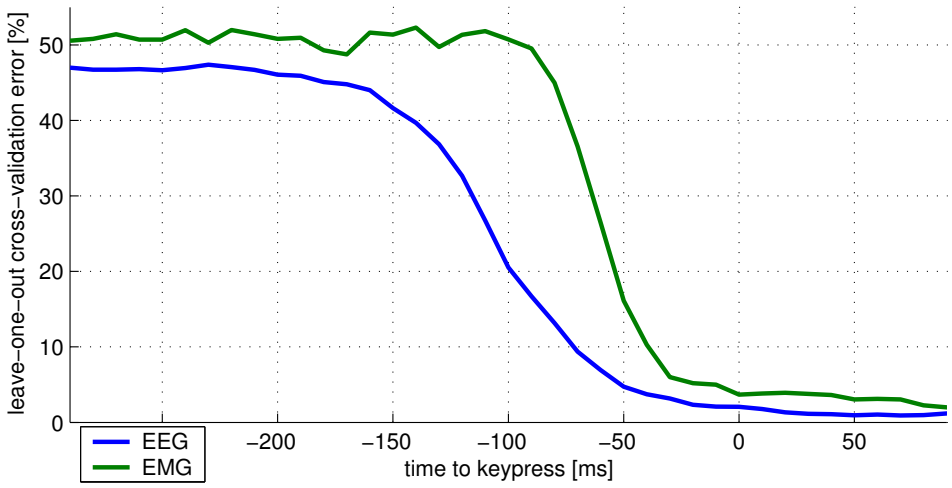


Figure 2.7: The figure shows leave-one-out cross-validation errors based on classification on EEG activity or EMG electrodes fixed at the arms up to a specific time regarding key-press between left and right finger movement during a d2 experiment.

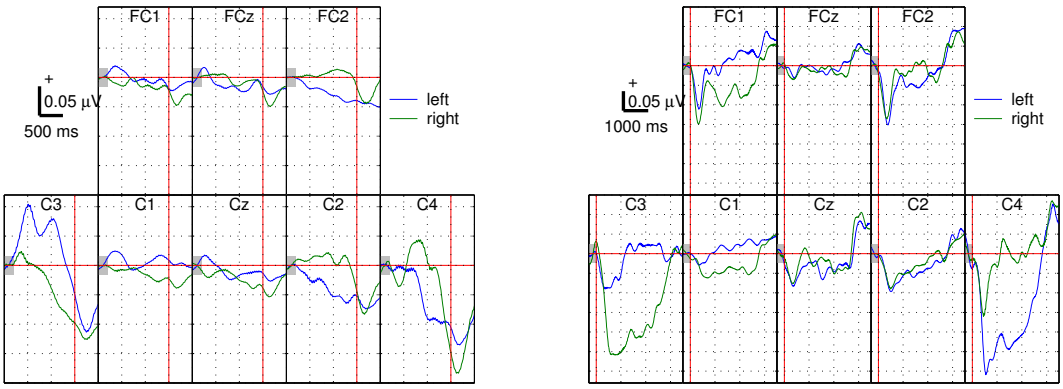


Figure 2.8: On the left the ERD of the μ-rhythm for left and right finger movements is shown which reveals the contralateral desynchronization preceding the movements. The movement was performed at timepoint zero (red line). On the right the ERD of the μ-rhythm for imagined left and right hand movements is visualized which shows the same contralateral characteristics. The task was ordered at timepoint zero (red line).

Beisteiner et al. [6] has shown the existence of similar modulations in the EEG during imagined movements, too. I will call this effect Movement Related Potential (MRP) in the following, since the effect in real movements would be similar. On the right of Fig. 2.5 this is visualized for one subject. In this experiment the trigger (zero point) is given by the visual command of the corresponding class. After this trigger the subject starts to imagine to move the corresponding hand for at least 3 s. After 500 ms the contralateral MRP is clearly visible.
During analysis of real movements, e.g., the described selfpaced experiment, one observes that the Bereitschaftspotential and the ERD (see section 2.2) precede the upcoming movement (see [17]). If one is able to detect and discriminate this it can be used to have a faster output channel or to confront a subject with his own upcoming actions. In Fig. 2.6 the classification performance in discriminating left vs right hand movements for one subject based on EEG and EMG is visualized (see [17]). In this experiment the time-point of classification is varied. One can conclude that EEG-based classification is possible 200 ms before key-press with less than 10 % error. But the possibility to classify with EMG electrodes fixed at the arm does not start until 100 ms before key-press.
This effect can also be observed in reactive movements, this means if a movement has to be done as fast as possible at a certain point in time. Of course the preparation time as shown in Fig. 2.6 does not start so early (see [71, 70]). Fig. 2.7 visualizes the classification performances similar to Fig. 2.6 for reactive left and right hand movements based on the d2-paradigm for one subject. This figure strengthens the hypothesis that an upcoming reactive movement can be detected earlier than the real movement takes place. Thus one can use this effect for example in the field of safety technology, e.g., by preparing a car if the driver wants to brake as fast as possible.

2.2 Oscillatory features

Some brain states can be described by different brain rhythms over specific brain areas. The most prominent rhythm α is around 7 to 13 Hz and is mainly focused on the parietal

and occipital area of the brain, but due to volume conduction it is radiated over the whole cortex. Since this rhythm is very strong compared to others, it is very visible in almost all electrodes. The α-rhythm varies depending on visual processing, fatigue and sleep (see [10, 66, 37]). In approximately the same frequency range the so called μ-rhythm can be observed over the motor area. This rhythm is attenuated during (real or imagined) movement in the corresponding brain region. This attenuation is called event-related desynchronization (ERD). The effect in the other direction is called event-related synchronization (ERS). Other rhythms like β, γ, δ exists with different functionality which I will not describe further (see [10, 66, 37, 109] for more details). Note that with higher frequencies the amplitude decreases. It should be mentioned that for movements a similar effect in the β-rhythm compared to the μ-rhythm can be observed.
The ERD during movement appears both for real and for imagined movements. Fig. 2.8 shows the ERD curves for the μ-rhythm of the experiments described for the same subjects as in section 2.1.3, which have the expected characteristics.

2.3 Real vs. imagined movements

The imagination of a movement compared to a real movement for a healthy subject is very unnatural. Nevertheless, a strong functional similarity between real and imagined movements was observed in several studies (cf. [6]). Furthermore it was shown that although small changes can be observed due to the missing tactile feedback, the neurophysiology in general is preserved for disabled or ALS patients who are able only to imagine the specific movement (cf. [131, 57, 119]). Furthermore existing BCI applications have shown that patients have similar neurophysiological features as healthy subjects (cf. [142, 75]).
Thus it should be concluded that it does not matter if real or imagined movements are chosen. Unfortunately, this is not fully correct. There are a few points to discuss:

- Any movement in the face (e.g., tongue movement, jaw activity) has its direct electrical correlate called EMG measured also in the EEG. Therefore EEG-controlled BCIs based e.g., on imagined tongue movements should be checked carefully, since two changes can be observed if the tongue is really moved: the neurophysiological (EEG) and the physiological muscle component (EMG). The latter is usually more pronounced so that pure EEG control on movements of the face is highly questionable if a real movement happens.

- Experience in our lab has shown that EEG-based BCIs are highly distorted by other brain activities. During imagination the concentration is usually much higher than during real movements so that one could assume that the quality of the imagined signal could be best. Furthermore patients which are only able to imagine the movement probably need the same amount of concentration.

- Imagined movements are absolutely unnatural for a healthy subject and one could ask why a subject should be interested in controlling something by his thoughts and not with his hands. However, some studies have reported that after some time the imagined movements become a skill during feedback (see [141, 121]) so that one can hope that

thought-based control can be an easy add-on to other human communication channels for the healthy subject too.

- Real movements can be easily controlled for their correctness and timing.
- Healthy subjects have to order and inhibit an imagined movement. The inhibition is different to the process in disabled subjects.

2.4 Closed-Loop Feedback

Closed-loop feedback is one of the main issues when considering BCI systems. Of course one could control applications without having the feedback but it seems to be more natural if one knows about the reaction of the system to be able to interact with it. Assuming that a subject is able to use arbitrary complex coding strategies to transmit his thoughts (e.g., transmitting checksums of some old decisions) and that the classification performance does not change during time, theoretically no gain can be achieved by providing feedback to the user, i.e., the information if the decision was classified correctly or not (see chapter 3). However, both assumption are highly questionable. As discussed in chapter 3 arbitrary coding strategies can not be used for a BCI; one is forced to restrict the system to ergonomic codings, i.e., codings which can be handled by a human. In my opinion, ergonomic coding strategies can only be applied successfully in the sense of achieved performance with the use of the feedback, i.e., with the information when mental states are classified wrongly by the system (see chapter 3). Thus a closed-loop feedback is required to achieve good performance. Furthermore, establishing an error potential based system to correct other BCI systems to enhance performance as discussed in section 2.1.2 requires feedback too. Finally an important advantage which contradicts the assumptions above is the human capability of adapting to the environment. Once a subject is within this environment he will interpret the feedback and change his natural behavior to get a more appropriate feedback. Thus many BCI systems were successfully built using human learning capability based on fixed EEG-processing which works independently of the subject (see [140, 136]).
In systems where the EEG-processing is adapted to the subject, human learning capability is still there and can enhance the performance. However, the learning effect is usually very slow, but it was reported that subjects of a BCI with subject-independent EEG-processing were able to learn within 100 hours to control feedback to a certain degree. For good learning effects a suitable psychological program has to be used ([101, 102, 103]), otherwise the risk arises that the user gets frustrated and does not learn to control the feedback or even looses all control abilities. Both effects, learning and frustration, were visible in recent experiments at our lab, too. Although in our case a user is only confronted with a feedback for a few hours, some have reported that they were able to increase their performance over time and get a more natural feeling of the control, whereas others have reported that they became angry and frustrated with errors which results in a decreasing performance. However, the advantage of a closed-loop feedback far outweighs the disadvantage of frustration effects so that a successful BCI system can only be established with feedback in my opinion.

3 Measuring Performance: The Bitrate

3.1 Motivation

In BCI research several goals exist to enhance the usability of the interface. Besides fast preparation of the system (i.e., attaching the electrodes and training) one big focus is to increase the performance of the system, i.e., the ability to control complex scenarios. To do this, several parameters can be considered, e.g., the following three

- the number of available choices/options in one decision (N),
- the accuracy, i.e., the ratio of correctly detected to total number of choices (p),
- the decision rate, i.e., number of decisions which can be made every minute (ϑ).

Obviously the value of the BCI increases with the accuracy of the computer in interpreting human thought. Faster decision rates allow more complex and faster control of the device. Finally with the number of available decisions the opportunities of the user can be enhanced. Unfortunately, these three values N, p and ϑ cannot be controlled independently. For example, with increasing number of classes the accuracy of the system decreases (see chapter 7). Therefore these three values should be combined in one performance value to allow a fair comparison of different BCI systems. E.g., the κ-value based on Carletta [30] or the mutual information suggested in Schlögl et al. [123] are prominent candidates for this combined value. Wolpaw et al. [139] claim a method based on ideas of information theory. I will shortly summarize the ideas of this measure, called information transfer rate (ITR) in section 3.2. Afterwards I will discuss the problems with this approach and introduce some solutions in section 3.3. Finally I will compare the results in section 3.4.

3.2 Shannon's Information Transfer rate

To understand the idea of the ITR I should first introduce the concept of entropy. Based on the definition and the meaning of the entropy given in section 3.2.1 the ITR can be derived in section 3.2.2. In the following a short description without the full mathematical background is given since the idea is to present the intuition of ITR rather than to prove it. For more details and full proofs I refer to MacKay [83] and Cover and Thomas [35].

3.2.1 Entropy

Consider the set of all possible bitstrings, i.e., strings with elements in $\{0,1\}$, of length n. Obviously each element of this set informs about n different Yes-No-decisions. One says that each string contains n bits information. To get a suitable measure for the *information content* of an arbitrary but finite alphabet $\mathscr{A}$ with underlying probability distribution P about

the elements of $\mathscr{A}$ one maps the alphabet uniquely to a set of bitstrings. Hereby the mapping should be optimal in the sense that the length of the bitstrings is as small as possible. In this case the entropy informs about the length n of these optimal bitstrings. Due to the finiteness of the alphabet and the discreetness of the length of bitstrings a more precise meaning of the entropy is given by the averaged length of the bitstrings achieved by the optimal and unique mapping elements of $\mathscr{A}^n$ to bitstrings divided by n if n goes to infinity. Here $\mathscr{A}^n$ denotes the concatenation of elements of $\mathscr{A}$ of length n.
Obviously a finite set of 2^s elements can be uniquely coded by bitstrings of length s and no less. Thus the raw bit content of a finite set is defined by $H_0(\mathscr{A}) = \log_2 \#\mathscr{A}$, namely by the average length of the most efficient bitstrings which uniquely correspond to all elements of $\mathscr{A}$. Here $\#\mathscr{A}$ denotes the number of elements in $\mathscr{A}$. Additionally I have to introduce the essential bit content of $\mathscr{A}^n$ which is defined by $H_\delta(\mathscr{A}^n) = \log_2 \#S_\delta$ where S_δ is one example of all smallest subsets of $\mathscr{A}^n$ with $P_n(x \in S_\delta) \geq 1-\delta$. Roughly speaking, the essential bit content of length n measures the bit-code content of almost all except a few unlikely elements of length n. Here P_n denotes the induced probability of P if one draws the repetitions independently. Note that one can choose S_δ by the most probable elements of $\mathscr{A}^n$ until the sum of the probabilities of these elements achieves the desired probability $1-\delta$. However, this choice is not unique, i.e., other sets with the same number of elements with at least combined probability of $1-\delta$ could exist. However, the specific choice is not important for the following theorem, only the amount of elements of a smallest subset is really important. With

$$H(\mathscr{A}) = -\sum_{x \in \mathscr{A}} P(x) \log_2(P(x)) \tag{3.1}$$

the following theorem holds true:

3.2.1 Theorem: ***Shannon's source coding theorem*** *Let $\mathscr{A}$ be an finite alphabet with underlying distribution P. Let $\varepsilon > 0$ and $0 < \delta < 1$. Then there exists a positive integer n_0 such that for all $n \geq n_0$*

$$\left| \frac{1}{n} H_\delta(\mathscr{A}^n) - H(\mathscr{A}) \right| < \varepsilon.$$

Proof: see [83, 35]. □

This theorem states that $\lim_{n\to\infty} \frac{1}{n} H_\delta(\mathscr{A}^n) = H(\mathscr{A})$ independent of the choice of δ. Consequently the value of $H(\mathscr{A})$ defines a suitable measure for the information content of the alphabet in the sense described above and thus is used for the entropy. A more detailed construction and explanation of this result can be found in MacKay [83] and Cover and Thomas [35].
For this theorem it is important that n has to be big to be able to explain almost all concatenations of length n based on the alphabet and probability distribution. Consequently, it does not mean that we can code the alphabet $\mathscr{A}$ directly into bitstrings of length of $H(\mathscr{A})$. This is only true for an *infinitely* long concatenation of elements of $\mathscr{A}$ on average. Nevertheless it defines a comparable measure for different alphabets and underlying probabilities.
If the alphabet $\mathscr{A}$ and the probability distribution are defined by a random vector X, $H(X)$ is defined by $H(\mathscr{A})$. For independent random vectors X and $\hat{X}$ it holds true that $H(X,\hat{X}) := H((X,\hat{X})) = H(X) + H(\hat{X})$. Furthermore for arbitrary random vectors X and $\hat{X}$ the value $H(X|\hat{X}) := H((X|\hat{X}))$ is equal to $H(X,\hat{X}) - H(\hat{X})$. Finally, the mutual information is defined by $I(X;\hat{X}) := H(\hat{X}) - H(\hat{X}|X) = H(X) - H(X|\hat{X})$ which is symmetric in X and $\hat{X}$.

3.2.2 From Entropy to Information Transfer Rate

Let us consider an one-way noisy communication channel, i.e., a transmitter sends signals to a receiver but not all signals are received correctly. Thus the receiver is not able to get all the transmitted information. Since the transmitter is interested in submitting all desired information over this noisy channel, he has to submit further control signals so that a better reconstruction by the receiver can take place. Ideally as many control signals should be as needed to ensure that a reconstruction is almost surely possible, if the channels is used infinitely long. The information transfer rate (also called bitrate in this work) should define a measure for this information loss, i.e., it states how many bits are received. Note that this information transfer rate is a relative measure: one could ask for the received bit amount if one bit is transmitted, or for the received bit amount if one decision out of a finite (or maybe infinite) alphabet is transmitted, or for the received bit amount per time if decisions can be submitted at a predefined rate.
To get the information transfer rate the model of a channel is used, where a transmitter X submits elements of a fixed finite alphabet $\mathscr{A}$ via a noisy channel to a receiver $\hat{X}$. I assume that the channel is memoryless, i.e., the output distribution of the channel only depends on the input of the channel and does not depend on older inputs or outputs. Furthermore I will first assume that the channel is one-way, i.e., the transmitter does not know anything about the received signal. Later this assumption will be discussed. To create a *perfect* channel one needs a coding and decoding algorithm which allow a *perfect* reconstruction (i.e., with arbitrarily small probability of transmission errors) of the transmitted signal. Thus redundancy has to be added during transmission. Driven by the idea to reduce the reconstruction error one maps the signals to longer codewords such that the space of codewords is *sparse*. Sparseness in this context means that two codewords coming from different submitted elements are very far away from each other. For reconstruction one calculates the probability of the received signal originating from all possible submission signals and choose the one with the highest probability. Thus the error probability for transferring information via the channels decreases with increasing sparseness of the submitted codeword space. This sparseness is described by the rate R which is defined by the ratio of the logarithm of the amount of transmitted codewords and the length of the codewords. Intuitively, the mutual information defined above is an important candidate to measure the performance of the optimal coding, which is stated in the following theorem:

3.2.2 Theorem:

- *For any $\varepsilon > 0$ and $R < C := \max_P I(X;\hat{X})$ there exist n_0 such that for all $n \geq n_0$ there exists a code of length n and rate $\geq R$ and a decoding algorithm, such that the maximal probability of decoding errors is $\leq \varepsilon$.*

- *If an error $0 < p_e < 1$ is allowed, rates up to $R(p_e) = \frac{C}{1+p_e \log_2 p_e + (1-p_e)\log_2(1-p_e)}$ can be achieved.*

- *Rates greater than $R(p_e)$ are not achievable.*

Proof: see [83, 35]. □

The value $C = \max_P I(X;\hat{X})$ is called the capacity of the channel. Here the maximum is calculated over all possible distributions on the alphabet of the transmitter.
In the BCI case the transmitter is the human with his decision, the channel is the EEG system and the classification algorithms, and the receiver is the output of the classification and therefore the device to be controlled. Thus the capacity of this channel given by the theorem above informs us about the maximal possible but also achievable transfer rate (i.e., the achievable reconstructable information) via the channel.
For simplification reasons let us assume that N classes and an accuracy p are given so that $P(\hat{X} = \hat{x}|X = x) = p$ for $x = \hat{x}$ and $P(\hat{X} = \hat{x}|X = x) = \frac{1-p}{N-1}$ for all $x \neq \hat{x}$. Here X describes the transmitted and $\hat{X}$ the received signal using both the same alphabet. Roughly speaking, I assume that the BCI user can choose between N different options. The classifier of the system, i.e., the receiver, detects these mental states as the desired choice with probability p and makes a mistake with probability $1-p$. If the system fails there is no further bias towards another class, i.e., the distribution of mistakes is laplace distributed on the remaining options, i.e., each other option except the desired one is taken with probability $\frac{1-p}{N-1}$.
Under this assumption one gets $H(\hat{X}|X) = -\sum_x p(X = x)\sum_{\hat{x}} p(\hat{X} = \hat{x}|X = x) \log_2 p(\hat{X} = \hat{x}|X = x) = -p\log_2 p - (1-p)\log_2 \frac{1-p}{N-1}$ independent of the distribution of the alphabet. Furthermore $H(\hat{X}) = -\sum_{\hat{x}} p(\hat{X} = \hat{x}) \log_2 p(\hat{X} = \hat{x})$ which is maximized if the distribution is uniform with maximal value $\log_2 N$. Therefore the capacity C is equal to $\log_2 N - p\log_2 p - (1-p)\log_2 \frac{1-p}{N-1}$. Note that the bitrate can also be calculated if the confusion matrix of the classification problem is not symmetric but I will only refer to this special form of the capacity in this work.
The so calculated capacity of the channel is called the information transfer rate per decision I_d of the system since it defines the maximal possible communication rate which can be achieved by this channel, i.e., the maximal achievable received information after exactly one decision by the user. If decisions can be performed at a specific rate ϑ one obtains the information transfer rate per time by $I_\vartheta = I_d\vartheta$.
The mutual information approach by Schlögl et al. [123] for comparing different BCI systems seems to be similar to the ITR but has one different aspect, namely that it has a stronger focus on the evidence of the system by interpreting human thoughts and not so strong a focus on the classification accuracy. However, the values will usually not show high differences in comparing different BCI systems.
Compared to the general Shannon model, there is a big difference to the usual BCI model. In Shannon's theory there is no feedback from the receiver to the transmitter, in other words the transmitter does not know when the errors appear. In a normal BCI system the received information is directly presented to the transmitter (the human) who can perfectly evaluate if the transmission was correct or not. Therefore one could think of more efficient codings which could use this feedback information and thus enhances the channel capacity. But the following theorem holds true:

3.2.3 Theorem: *The capacity of a memoryless channel with feedback is equal to the capacity of a memoryless channel without feedback.*

Proof: see [35]. □
Assuming fixed error rates and access to suitable codings this theorem states that feedback cannot enhance the performance of the BCI. However, both assumption are critical in the

BCI situation: Feedback allows adaptation for the subject, i.e., humans can learn and increase the classification accuracy due to the feedback, and it possibly allows more intuitive coding strategies. The former is illuminated briefly in the following; I will spend more time discussing the latter problem afterwards.
Theoretically the adaption ability of the user contradicts the assumption of the memoryless channel. Based on old transmissions the subject tries to improve his signals so that the noise model of the channel varies. Of course one could recalculate the capacity of the channel and in doing so one could take this performance increase into account. However, the statement of theorem 3.2.3 that feedback can not increase performance fails here: Feedback can and will increase performance since the human can learn based on old behavior. However, one could ask how much the information transfer rate is influenced if one calculates and then recalculates this value over and after short time periods. Since learning for a subject takes a long time in BCI experiments, in my opinion the use of the value over short time periods as an approximation for the real value without assuming a memoryless channel works reasonably well.
Another problem with this value is in my opinion more critical: Shannon's theorem only says that a coding exists to achieve the performance and that there is no better way of doing so, but not what the coding looks like. Moreover, no constructive way for the optimal coding is known so far. Therefore one has to find a way which is very close to the most optimal one. However, this coding has to be performed by the user of the BCI interface. Therefore there should be a limit to the complexity: Codings should be easy enough for a human to handle them, e.g., calculations of checksums of many old decisions could be too complex in this context. Consequently, the set of useful codings in the BCI context is restricted to simple ones, which of course decreases the achieved transfer rate compared to Shannon's result. But the question remains how big this loss could be. Some strategies in this direction will be discussed and their performance will be calculated in the next section and compared to the ITR in section 3.4.

3.3 Coding strategies for humans

The idea of this section is to define coding strategies which are easy to handle by a human and to estimate their expected performance. At this point a coding should satisfy two criteria: First of all it should be the optimal one out of the set of codings suitable for humans in that the maximum possible information can be transferred. The second criteria of a coding is the extension of the number of achievable decisions. For example if one is able to control a specific number of mental states, say left vs. right hand imagination, but has to choose out of many decisions for the feedback, say selecting a letter, one has to find a suitable command code that consists of left and right hand imaginations. In this section I will directly try to incorporate both criteria, optimality in transmitted information based on ergonomic codings and extensions to many decisions. Note that it is hard to find a limit for when a coding becomes too complex for a subject. Furthermore this limit mainly depends on the ability of the subjects which can vary significantly.
For all codings I assume that the human can handle N different classes/mental states with accuracy p. For simplicity I assume that $P(\hat{X} = \hat{x}|X = x) = p$ for $x = \hat{x}$ and $P(\hat{X} = \hat{x}|X = x) = \frac{1-p}{N-1}$ for all $x \neq \hat{x}$, i.e., the errors are equally distributed over all wrong

states. Here X describes the desired mental state of the subject and $\hat{X}$ the mental state detected by the system. Furthermore with these N classes a device should be controlled with $M \geq N$ different opportunities, e.g., the digits of a calculator, a speller or some fancy menu navigation.
In the following I will introduce concepts for suitable codings, try to analyze their behavior and finally find analytical solutions for the probability and expected number of steps to achieve a decision. Some of the results can be found in the appendix since the calculations can become rather technical.
However, for the analysis of the codings I have to introduce the Catalan Numbers (see [33]) and to prove two important formula (see lemma 3.3.1) about these numbers which relevance for this work becomes clear during analysis of the suggested codings. The Catalan numbers c_n are defined as the number of different $\{-1,1\}$-sequences $\{\varsigma_1,...,\varsigma_{2n+2}\}$ so that $\sum_{i=1}^{2n+2}\varsigma_i = 0$ and $\sum_{i=1}^{j}\varsigma_i > 0$ for $j < 2n+2$. The following lemma informs about important analytical properties of these numbers:

3.3.1 Lemma: *Let $q \geq 0$, $a,b,h_1,...,h_q \in [0,1]$ with $a+b+h_1+...+h_q = 1$, $h_1,...,h_q < 1$ and $t_a,t_b,t_{h_1},...,t_{h_q} \in \mathbb{R}_0^+$. Then:*

$$\sum_{n=1}^{\infty}\sum_{(2k,j_1,...,j_N)=n-1}\binom{n-1}{2k,j_1,...,j_q}c_k a^{k+1}b^k h_1^{j_1}\cdot...\cdot h_q^{j_q} = \begin{cases}1 & a \geq b\\ \frac{a}{b} & a < b\end{cases}$$

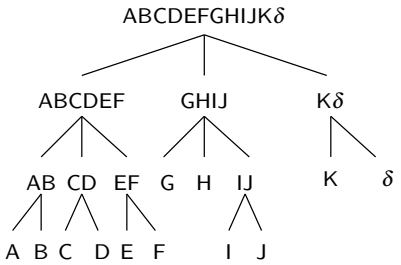
and for $a > b$

$$\begin{aligned}&\sum_{n=1}^{\infty}\sum_{(2k,j_1,...,j_N)=n-1}\binom{n-1}{2k,j_1,...,j_q}c_k a^{k+1}b^k h_1^{j_1}\cdot...\cdot h_q^{j_q}(t_a(k+1)+t_b k+t_{h_1}j_1+...+t_{h_q}j_q)\\ =\;& t_a\frac{a}{a-b}+t_b\frac{b}{a-b}+t_{h_1}\frac{h_1}{a-b}+...+t_{h_q}\frac{h_q}{a-b}.\end{aligned}$$

Proof: see A.1.
Here $\binom{n}{j_1,...,j_q} := \frac{n!}{(j_1)!\cdot...\cdot(j_q)!}$ for $n,j_1,...,j_q \geq 0, j_1+...+j_q = n$ denotes the multinomial coefficients and $(j_1,...,j_q) = n$ denotes all non-negative integers $(j_1,...,j_q) \in \mathbb{N}_0$ with $\sum_i^q j_i = n$.
Note that the performance of the following codings can sometimes be solved by recursions too. But then not all approaches can be solved and furthermore the existence of the expectations is not clear in every case. Therefore I choose the described distinct way. It should be mentioned that all these results can also be achieved by simulation of the approaches by repeated runs to achieve a desired decision. By the law of large number one knows that one gets the expected results. However, only results but not the formula can be achieved this way but the simulations can be used to verify that the achieved formulas are correct which was done for all described approaches successfully.

3.3.1 Standard tree with delete option (ST)

Concept. In this case the set of all M opportunities is split up into at most N subsets and the user is asked for a decision between these subsets. Afterwards the algorithm goes on with the chosen subset in the same manner until one single opportunity remains. For the



$\bar{\delta} : \delta$
$\bar{\delta} : GB\delta C\delta\delta K\delta\delta$
$\bar{C} : C$
$\bar{E} : GE\bar{\delta}\bar{\delta}E$
$\bar{K} : \delta * I\bar{\delta}K$

Figure 3.1: On the left a standard tree with eleven symbols and δ, based on $M = 3$, opportunities in each decision is visualized. On the right possible sequences are shown: Before the colon the desired letter is shown. After the colon possible sequences to achieve the goal are shown. Here the bar above the symbols denotes that this is a grouping for other sequences with final received symbol, e.g., $\bar{E}$ could be E, but also $F\delta E$ or even longer. The star for the last example denotes some arbitrary symbol. In this case a symbol was desired, but δ achieved. Consequently the last written symbol has to be written again. Afterwards the subject tries to get the K again.

choice of a suitable tree, i.e., the splittings into suitable subsets, efficient algorithms (see [83]) exist which depend on the prior distribution of the opportunities. If this is uniform an absolutely symmetric tree is chosen. Due to the assumption that errors will be made the ability to correct wrong decisions via a symbol is added. If this element is chosen the last decision is cancelled. I will use the symbol δ for this deletion action. An example tree is shown in Fig. 3.1 on the left with letters for each decision.

In the following I will call a single path of choices until a decision is achieved an attempt, whereas a run describes repeated attempts (maybe with different goals) until the right decision is achieved including deletions or repetition of wrongly deleted decisions. Here an infinitely long past of decisions is assumed.

Analysis. Suppose $p_{i,j}$ denotes the probability of achieving i during an attempt, starting with the full set of symbols, if decision j was desired (I will write $p_i := p_{i,i}$ for short), P_i the probability of a run to get the decision i if desired, E_i the corresponding expectation of used steps in the run to get i and d_i the depth of the decision i in the tree. Thus $p_i = p^{d_i}$ for all i and $p_\delta = p^{d_\delta}$. Furthermore $f_i > 0$, $\sum_{i \neq \delta} f_i = 1$ denote the prior distribution over all decisions ignoring δ.

Only situations with $P_i = 1$ for all i are relevant. Otherwise the convergence and thus the achievement of a decision cannot be guaranteed almost surely. Then $\bar{E} = \sum_{i \neq \delta} f_i E_i$ denotes the expected number of decisions for a successful choice of a decision recognizing correction of wrong attempts.

Usually the trick is to map the run to $\{-1, 0, 1\}$-sequences (which I will call extended Catalan sequences in the following) with the goal that the sum is equal to -1 and that -1 is achieved for the first time with the last element of the sequence. Here -1 denotes a step towards the goal, 1 a step backwards away from the goal, and 0 some in between steps without a direction (with possibly different meanings). If a goal is achieved in n steps, this usually consists in $k+1$-times -1-, k-times $+1$- and $n-2k-1$-times 0-steps, where all possible positions are somehow described by Catalan numbers (the described subsequence

with $\{-1,+1\}$ corresponds to a Catalan sequence with adding $+1$ in the beginning). Here the 0 sequences can be arbitrarily ordered in this sequence except for being at the last position. The number of suitable positions of 0-symbols can be perfectly described by the multinomial coefficients as used in lemma 3.3.1.

The meaning of the single elements of the $\{-1,0,+1\}$-sequences differ in the following based on the given situation and will be explained individually.

For (**ST**) I use them as follows: To calculate the probability for the run to delete a decision if desired, the subject tries to get a δ ($\cong -1$), if this fails ($\cong +1$) he has to delete the wrong decision ($\cong -1$) and has to try again. Therefore all successful runs for deleting a decision consists of finite sequences described above where a 0-part does not exist.

Now let us consider the attempt to achieve at some arbitrary decision $i \neq \delta$. Again positive runs can be described by the extended Catalan sequences. Here -1 describes a correct attempt at i, 0 an attempt at some different decision except δ with following successful deletion and $+1$ an attempt at δ. In the latter case (under assumption of an infinite past of decisions) an old decision $j \neq \delta$ has to be repeated. Therefore this is really a step backwards. I assume here that this old decision j is independent from i.

Possible Catalan-Sequences for both runs to δ and some other decisions are visualized in Fig. 3.1.

Results. P_δ can be calculated by summing over all probabilities of all possible decisions sequences described by the Catalan numbers above which is equal to the sum which has to be calculated in lemma 3.3.1. Note that one has to choose $a = p_\delta$ (namely the probability of a successful attempt at δ), $b = 1 - p_\delta$ (namely the probability of an unsuccessful attempt at δ) and $q = 0$. Consequently $P_\delta = 1$, if $p_\delta > \frac{1}{2}$. For the expectation E_δ I again use lemma 3.3.1 with $t_a = d_\delta$ (the number of steps to achieve at δ) and $t_b = \sum_{i\neq\delta} \frac{p_{i,\delta} d_i}{p_{i,\delta}}$ (the averaged depth of all wrong decisions). This results in $E_\delta = \frac{p_\delta d_\delta + (1-p_\delta)\sum_{i\neq\delta} \frac{p_{i,\delta} d_i}{p_{i,\delta}}}{2p_\delta - 1}$, if $p_\delta > \frac{1}{2}$.

P_i can be again calculated by summing over all probabilities of the sequences described above to achieve decision i. The corresponding probabilities to use lemma 3.3.1 are $a = p_i$ (namely the probability of a successful attempt at i), $b = p_{\delta,i}$ (namely the probability of achieving δ if i was desired), $q = 1$ and $h_1 = (1 - p_i - p_{\delta,i})P_\delta$ (namely the probability of an attempt to achieve some decision except i and δ if i was desired with consecutive successful deletion run). Here one directly sees that $P_\delta = 1$ is necessary to guarantee that one is almost surely able to achieve decision i (except if $p = 1$). Note that there is a small inaccuracy in the calculation since the retrieval of an old decision j can be different to i. For simplification I approximate p_i resp. $p_{\delta,i}$ by the – by the single decision frequencies f_i weighted – mean $\bar{p}$ of all p_i except δ resp. $\bar{p}_\delta$ of all $p_{\delta,i}$ except δ. This approximation makes sense if one remembers that $\sum_{(k_1,\ldots,k_q)=k} \binom{k}{k_1,\ldots,k_q} \prod_{i=1}^q (f_i p_i)_i^k = (\sum_{i=1}^q f_i p_i)^k = \bar{p}^k$ for all k. Note that $\bar{p}_\delta$ and p_δ are different, both describe probabilities if δ is achieved, but the first one if a different symbol was desired and the second one if δ was desired. Furthermore let us define the – by the single decision frequencies f_i averaged – depths d_i by $\bar{d}$.

With lemma 3.3.1 one gets $\bar{P} = 1$ if $p_\delta > \frac{1}{2}$ (since $P_\delta = 1$ is required) and $\bar{p} > \bar{p}_\delta$, i.e., convergence is guaranteed if one can achieve δ if desired with probability of at least 0.5 in one attempt and if one can achieve a symbol better than the δ if the symbol is desired (as mean over all symbols). For the decision depths one uses $t_a = \bar{d}$ (namely the mean depth of

the decisions), $t_b = d_\delta$ (the depth of δ) and $t_{h_1} = \bar{d} + E_\delta$ (the mean depth of the decisions plus the expected number of steps to delete the wrongly made decision). The exact formula for $\bar{E}$ is given in section A.5. The formula only leads to a concise solution if the depths for all decisions including δ are equal to d ($\Rightarrow p_\delta = \bar{p}$): $\bar{E} = \frac{d}{2p_\delta - 1}$.

3.3.2 Confirmation tree (**CF1-CF3**)

Concept. The idea of a confirmation tree is to have instances where decisions can be cancelled by asking for confirmation of the correctness. In the case of rejection the last steps will be cancelled and the user can repeatedly try to make the correct decision. In the easiest case a usual tree like in section 3.3.1 is used. After one attempt, i.e., after one decision is achieved, a further two class confirmation question is asked, in the case of an accept, the decision is chosen, otherwise the attempt is ignored and a new attempt starts directly. This approach is called (**CF1**). There is a second interesting option for the use of the confirmation, namely within the decision trees. Here the computer could ask after each s-th choices for a confirmation[1]. Here two options are possible: after a rejection a repetition of the last s steps starts automatically (then a δ is needed) (**CF2**) or the last confirmation is repeated so that groups of s choices can be cancelled iteratively (then a δ is not needed) (**CF3**). In the latter case calculation of the formula is similar to (**OB1**) (see section 3.3.3) by grouping together the choices. Since the calculations of (**CF2 - CF3**) are rather technical I will only describe (**CF1**) in more detail here. The formula for (**CF3**) are given in section A.5. I will skip completely the formula for (**CF2**) since an exact solution is very long and goes beyond the scope of this work.

Analysis. Let us denote by p_c the probability to correctly answer the confirmation question. To calculate P_δ one again uses the extended Catalan sequences: a successful run consists in an extended Catalan-sequence, where -1 corresponds to an attempt where δ was achieved successfully and confirmed, $+1$ corresponds to an attempt where a wrong decision was achieved and confirmed and 0 to an attempt where the chosen decision was finally rejected. For the latter two cases we need to discuss: a correct attempt which was wrongly rejected or a wrong attempt which was correctly rejected.

To calculate the probabilities $\bar{P}$ and $\bar{E}$, the extended Catalan-sequences can be used again: -1 corresponds to correct confirmed attempts, $+1$ to confirmed attempts at δ and 0 to rejected attempts (either rejection of a wrong δ, rejection of a wrong decision or rejection of a correct decision) or to confirmed wrong decisions except δ with following successful rejection.

Results. For calculating P_δ and E_δ one uses with lemma 3.3.1 $a = p_\delta p_c$ (namely achieving desired δ with confirmation), $b = (1-p_\delta)(1-p_c)$ (namely achieving a different decision with confirmation if δ is desired), $q = 2$, $h_1 = p_\delta(1-p_c)$ (namely the rejection of a correct attempt at δ), $h_2 = (1-p_\delta)p_c$ (namely rejection of a wrong attempt), $t_a = d_\delta + 1$, $t_b = \bar{d} + 1$, $t_{h_1} = (d_\delta + 1)$ and $t_{h_2} = (1-p_\delta)p_c(\bar{d}+1)$. One is able to achieve δ if desired almost surely, if $p_\delta p_c > (1-p_\delta)(1-p_c) \Leftrightarrow p_\delta + p_c > 1$, i.e., the sum of the probability to achieve a δ if desired and the probability to answer the confirmation question correctly is bigger than 1. For E_δ one gets $\frac{(d_\delta+1)p_\delta + (\bar{d}+1)(1-p_\delta)}{p_\delta + p_c - 1}$. In the case that all decisions have the

[1] Note that the structure of such a tree could become very complex if the depth of the branches is not a multiple of s.

same depth d (without confirmation), this simplifies to $E_\delta = \frac{d+1}{p_\delta + p_c - 1}$.
To calculate $\bar{P}$ and $\bar{E}$ one uses (with similar approximations as for (**ST**)) $a = \bar{p}p_c$ (namely achieving the desired symbol plus confirmation), $t_a = \bar{d}+1$, $b = \bar{p}_\delta(1-p_c)$ (namely achieving and confirming δ if another decision was desired), $t_b = d_\delta + 1$ and $q = 4$, $h_1 = \bar{p}(1-p_c)$ (namely rejection of a correct attempt) $h_2 = (1-\bar{p}-\bar{p}_\delta)p_c$ (namely correct rejection of a wrong attempt not at δ) $h_3 = \bar{p}_\delta p_c$ (namely correct rejection of an achieved δ), $h_4 = (1-\bar{p}-\bar{p}_\delta)(1-p_c)$ (confirmation of a wrong decision except δ with consecutive deletion of the symbol), $t_{h_1} = t_{h_2} = \bar{d}+1$, $t_{h_3} = \bar{p}_\delta$ and $t_{h_4} = \bar{d}+1+E_\delta$.
Obviously $p_\delta + p_c > 1$ is required to ensure convergence since deletion is a necessary tool for successful runs (except for extreme cases like $p_c = 1$). Furthermore one gets $\bar{p}p_c > \bar{p}_\delta(1-p_c)$ as necessary condition for convergence. For the formula for $\bar{E}$ I refer to section A.5. However, in the case that all decisions have the same depth d, this formula can be simplified to $\bar{E} = \frac{d+1}{p_\delta + p_c - 1}$. For small p this solution seems to be better than the solution for the standard tree, but the positions of the δ should be used optimally in every case which often does not lead to the symmetric tree. Therefore it is not clear if this algorithm enhances the standard tree.

3.3.3 Tree with one class to delete the last choice (OB1-OB2)

Concept. If more than two classes are available further interesting strategies exist, e.g., one could have one class in every choice to cancel the last choice. If this is done iteratively one can delete more than one old choice and consequently decisions, too. Consequently a δ is not needed anymore. This coding is called (**OB1**). For example, if four classes are given, one could build a binary tree similar to Fig. 3.1 on the left (without δ). At each position in the tree the user has four options, use the left, the middle or the right branch or go one step upwards in the tree to the parent node and go on with the choices there. Hereby the parent node of the root is the last choice of the last made decision, i.e., the last decision is cancelled.
An interesting alternative denoted by (**OB2**) to this approach is to use one class in a choice for a restart of the attempt. In this case a δ is needed again.
Analysis. The path for (**OB1**) can again be described by $\{-1,1\}$-sequences but here on the individual choices: -1 is a correct choice which could also be a deletion of the last choice. This in the case of an error is obviously a step towards the goal. $+1$ corresponds to a wrong choice one step away from the goal. But in this case it is not enough to arrive at -1 with the extended Catalan-sequences. One has to achieve $-d$ with the individual depth d of the desired decision. But these sequences resulting in $-d$ can be mapped one to one to d repeated extended Catalan-sequences[2]. Therefore the probability of achieving a decision can be calculated by P^d where P describes the probability of achieving the next step. Consequently the expected number of steps $\bar{E}$ to achieve a decision can be calculated by $\bar{d}E$ if E describes the expected number of steps to successfully go one choice further in the tree.
The path for (**OB2**) can also be described by the Catalan sequences but here again by con-

[2] that d such sequences result in the desired sequence is obvious, for the other direction one should note that a concatenation of extended Catalan-sequences is achieved if one starts from the beginning and cuts the whole sequence into sub-sequences at the points, the next depth is achieved first time.

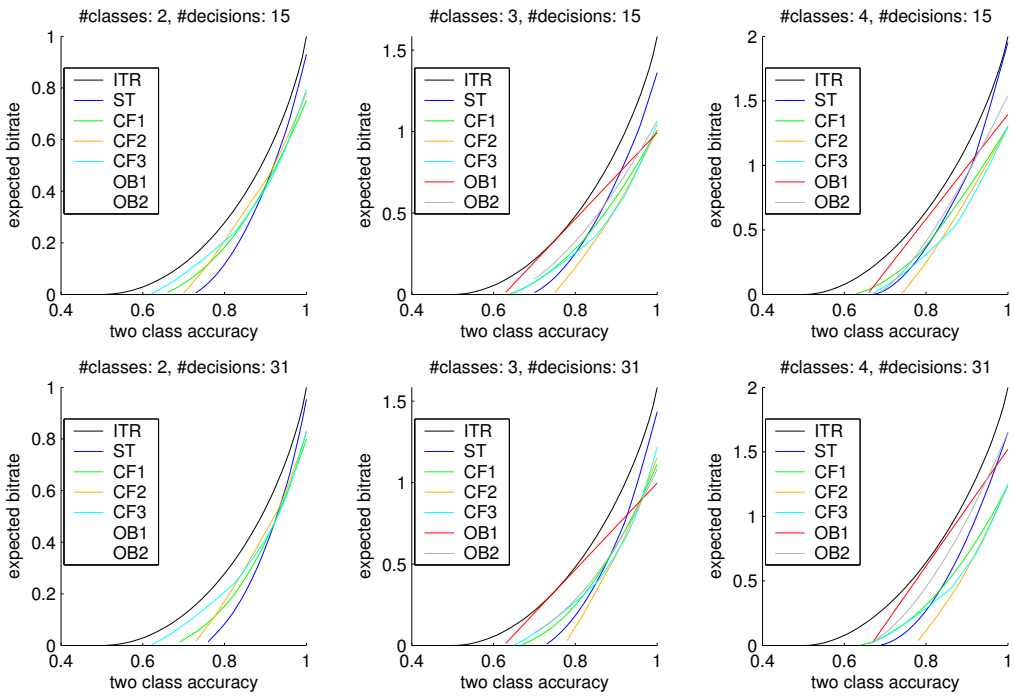


Figure 3.2: For each plot the number of classes and number of decisions was chosen as fixed. The classification accuracy for two classes is varied. For more than two classes the corresponding accuracy described by the κ-value is used. The plots show the calculated theoretical bitrate and the bitrate of the suggested methods. Note that (**OB1-OB2**) require at least three classes.

structing first P_δ and E_δ. The calculations are rather technical and are thus skipped completely.

Results. For (**OB1**) convergence is guaranteed if P as used above is equal to 1. With $a = p$ (here now the single choice probability), $b = 1 - p$ and $q = 0$ this results in $P = 1$ if $p > \frac{1}{2}$, i.e., convergence is guaranteed almost surely, if one is able to make the right choice with at least probability 0.5. With $t_a = 1$, $t_b = 1$ the expectation is given by $E = \frac{1}{2p-1}$ and thus $\bar{E} = \frac{\bar{d}}{2p-1}$.

3.4 Efficiency of coding strategies

In this section the ITR will be compared to the discussed strategies for human coding. I will do this on examples, where the number of choices, the number of decisions and the classification accuracy are compared. In each plot of Fig. 3.2 the number of classes and number of decisions was chosen as fixed, whereas the classification accuracy was varied in a suitable range. For all these parameters the calculated theoretical bitrate and the bitrate of the suggested methods are shown. Note that a uniform prior distribution over all decisions except δ is assumed. All possible positions of the δ in the tree are tested for all trees and the best performance is chosen. The same was done for s ($1 \leq s \leq$ maximum depth of the tree)

in (**CF2-CF3**). Note that the classification accuracy depends on the number of classes. To take this into account only variations on the accuracy of the two-class-decision are allowed. For the accuracies of more than two classes the κ-value is used. This value is defined by mapping the classification accuracy linearly to the interval $[0,1]$. The accuracy is a value between $\frac{1}{\text{Number of Classes}}$ (random classification) and 1 (perfect classification). A two class accuracy of 75 % is equal to a three class accuracy of 66.6 % and a four class accuracy of 62.5 % and so on. Thus with each 2-class accuracy corresponding accuracies for more than two classes are given. The 2-class accuracy is plotted on the x-axis in the figure. Since all suggested coding strategies above calculate the expected number $\bar{E}$ of choices to achieve one of M possible decisions the expected bitrate $\bar{B}$ is given by $\bar{B} = \frac{\log_2 M}{\bar{E}}$. Finally note that (**OB1-OB2**) require at least three classes, therefore they are only visualized if more than two classes are used.

First of all the figures show that the performances of all suggested coding algorithms are below the Shannon ITR, but the difference is not too big. One conclusion that one can draw from this investigation is that the ITR, although being primarily theoretical, is an admissible performance measure for a BCI system, since it can be almost achieved.

A second observation of the figures show that there is not one overall best method. This really depends on the specific situation. If the classification accuracy is not too high, usually the confirmation trees outperform the standard tree, whereas for very good accuracies, the standard tree performs best. This is not surprising, since for high accuracies the confirmation question is a waste of time compared to a δ with a high depth in the tree. If more than two classes are available one additionally observes that (**OB1**) has the best performance if the classification accuracy is not too high. Of course for almost perfect classification the use of one class as a backward step is a waste of capacity, thus (**OB1**) is not useful in this case.

Based on the results of the figures the use of the standard tree is advisable, if classification accuracy is very high. However, if the classification accuracy is not very high the use of the confirmation tree (**CF1**), if there are only two classes, or (**OB1**), if there are more than two classes, is more appropriate.

The results and suggested methods in this chapter can be used for BCI feedback experiments. Based on the performance on some training data and classes one could use these results to find the optimal coding strategy in each specific situation. Thus the BCI communication ability can be optimized individually. Recently some strategies were applied successfully during online feedback experiments in our lab (see section 4.2.2 for one example).

Finally, one should note that all approaches use feedback and the ability of the user to recognize the errors of the system. Obviously this is important to achieve this performance. Although Shannon's theorem says that this performance can also be achieved without feedback, suitable codings which can be handled by humans can presumably not be found achieving similar results. In other words the theoretical result that the use of feedback does not change the performance of the channel does not match the situation in a BCI interface where the coding class is limited.

4 Experiments

In this chapter I will discuss two types of experiments: the calibration measurement and the online feedback experiment. During the calibration measurement the user of the BCI system is asked to perform different tasks. The idea is that after this session the computer should adapt to the specific brain signals and be able to perform online feedback for human decisions. The recorded data are also used for evaluation of the performance of the algorithms described in later chapters. I will briefly present two types of calibration measurement used in our lab in section 4.1. Based on the calibration measurement for some recent datasets a classifier is immediately trained on the recorded data and applied online to present the user with his own feedback. The design of the online interface and first online feedbacks and their results will be presented in this chapter in section 4.2. I will describe how the machine training works later in this work, starting with chapter 5.

Note that the data recorded during feedback experiments is only used in section 4.2 in this work. For further comparison of different algorithms only the calibration measurement data is used since once feedback is presented, an adaption process of the subject happens. Thus the results are biased by the applied feedback algorithm which makes different algorithms hard to compare.

4.1 Calibration Measurement

The idea of a calibration measurement consists of getting a good amount of datasets to train a subject-specific classifier for subsequent online feedback. Since this recording serves to train a classifier for the feedback, I will call it the training session. Note that training here refers to machine training, not to subject training. Furthermore, these datasets serve as a testbed to compare different machine learning algorithms (see chapters 5, 6, 7 and 8).

In all experiments a multi-channel EEG amplifier with 32, 64 or 128 channels (see Fig. 4.1 for the location of the channels in a 128 electrode cap) band-pass filtered between 0.05 and 200 Hz and sampled to 1000 Hz is used. For off-line analysis all signals are down-sampled at 100 Hz. Depending on the involved limbs, surface electromyogram (EMG) at arms and legs as well as horizontal and vertical electrooculogram (EOG) signals are recorded. These signals are used neither for classification nor for feedback control. They only serve to check for correlated muscle activation and eye movements during imagined movements or to see the exact progress of real movements.

In this work I report on two different types of experiments, one called *selfpaced* and the other called *imag*, both designed to get a suitable training session for subsequent feedback applications, see section 4.2.

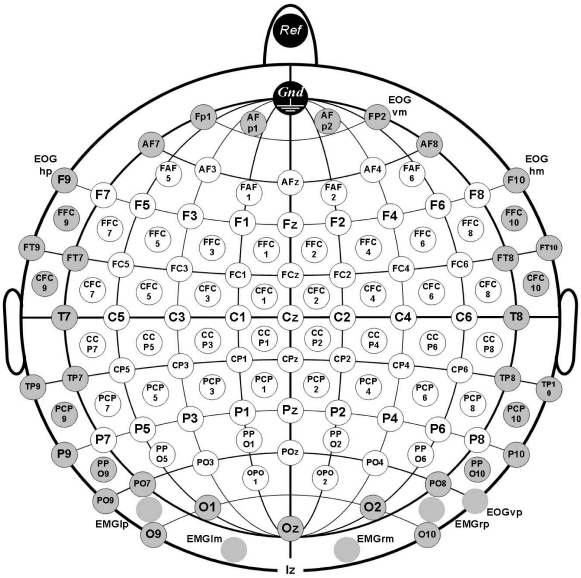


Figure 4.1: The map shows the channel locations for a 128-channel cap taken from Krepki [73]. Note that a few channels were used for measuring EMG and EOG.

4.1.1 Selfpaced Experiments

In this experiment type the subject sat in a comfortable chair looking at a cross on a computer monitor. His task was to press a button with the left or right index or little finger in a predefined pace of approximately 0.5, 1 or 2 s. This experiment was repeated with 8 different subjects, with some more than once. Since brain signals from healthy subjects who execute real movements are studied, no gain can be achieved by detecting this movement and using it to control a device, except this detection is possible before the movement really happens. In other words the goal in this type of experiment is the prediction of movement as early as possible and at least earlier than the movement can be detected by EMG activity. See Blankertz et al. [17, 19] for more details. The processing of such data works as follows: First of all a window of length 1280 ms is chosen (e.g., the interval $[-1400 - 120]$ msec regarding keypress). Then a cos-windowed FFT is applied to the data and the frequency band 0.8–5 Hz is chosen since this is the specific range a lateralized readiness potential lies (see Fig. 2.5). Afterwards an inverse FFT is applied to project the trials back into the time domain. Now the last 150 ms are chosen, which consist of 15 timepoints since the data are sampled at 100 Hz. By calculating jumping means of 5 consecutive timepoints this signal is reduced to 3 timepoints for each channel. Finally the feature vector is built by the concatenation over time and channels where a few non-relevant channels for the specific task are skipped.

For extracting ERD effects from selfpaced data for classification I first apply a broad band filter of 7–30 Hz (butterworth IIR filter of order 5) to the data. Note that this band was chosen since it comprises the μ- and β-rhythm. Other bands were tested but on average they do not perform better. However, it was seen that for some subjects a more appropriate fit to some frequency bands is advisable to enhance the performance. This point will be addressed in chapter 8. After filtering the time window -500– -100 ms regarding keypress is extracted and the CSP algorithm with 3 patterns per class is applied. The CSP algorithm will be explained in chapter 5 in more detail.

4.1.2 Imag Experiments

In this experiment type the subject sat in a comfortable chair looking at a computer monitor. During the experiment the subject was prompted by appropriate symbols or letters on the computer screen to imagine a specific task. For example an 'L' was used for the imagination of a left hand movement. Possible tasks were imagination of left and right hand, foot or tongue movement or auditory, visual or tactile sensations. The classes were prompted for approximately 3 seconds, i.e., the letter or symbol was shown during this time. The subjects were asked to perform the specific imagination during the whole time the letter was shown. After a short break of approximately 1.5 seconds the next run prompted by the next letter or symbol starts. In some training sessions these letters were exchanged for a gray rhombus which moved across the screen and was reflected by the edges. The stimulus consisted of a triangle that was colored red, pointing to the left, right, top or bottom which should correspond to left or right hand, tongue or foot. This was done to force uncorrelated eye movements to make classification robust against them.

Altogether approximately 60 experiments with around 20 different subjects were recorded. Every dataset consists of 100–200 trials per class. The aim on these datasets is to discriminate trials of different classes using the full period of imagination. Furthermore, the sensation classes (auditory, visual and tactile sensation) were chosen because their cortical activation patterns can be well discriminated at different locations of the brain so that discrimination for both slow potentials and oscillatory effects can theoretically be expected. However, since these classes do not appear natural enough for a BCI system they were only used for offline analysis but not for feedback experiments.

If not specified otherwise during this thesis, I only use 500–3500 ms of bandpass-filtered data between 7–30 Hz for offline analysis. Again the band was chosen since it comprises μ- and β-rhythm. Furthermore it was seen that even though other choices of frequency ranges can perform better in single datasets, on average over all datasets they do not perform better. Again an enhancement by fitting of frequency bands to the specific subject would be advisable. This task is addressed in chapter 8. After bandpass-filtering the CSP algorithm with 2 patterns per class is applied. The CSP algorithm will be explained in more detail in chapter 5.

To extract slow features I set a baseline at 0–300 ms regarding stimulus, extract the 500-2500 ms window regarding stimulus and calculate jumping means to retain 4 timepoints per channel. Finally, the feature vector consists of a concatenation over time and channels where a few unimportant channels for the task are skipped.

Note that performance could be enhanced if I adapt all these parameters to the specific datasets, but to compare different algorithms, a chosen fixed setup seems to be more appropriate.

4.2 Feedback experiments

Several fields of application for a Brain Computer Interface exist. First of all one could think about clinical solutions, i.e., an assistance system for the disabled by creating a new communication channel. It could especially help completely locked-in patients, enabling a possibly sole communication channel to the outside world [11]. Note that so far feedback experi-

ments in our lab were done with healthy people only. But it is safe to assume that one can transfer the feedback experiments described in section 4.2.2 directly to clinical applications for patients. Since their application seems to make more sense for disabled people I have chosen to call this section *applications for the disabled*. However, these methods could also be used as an additional communication channel for healthy people. Further applications for healthy people can be found in the area of movement prediction (e.g., for safety technology), since it was shown in off-line analysis that an upcoming movement can be detected before EMG-onset (see section 4.2.3). Other applications lie in the field of gaming, for example, which will be discussed in section 4.2.4. Finally one should note that by measuring the EEG many other states (e.g., fatigue, workload, attention) of a human can be evaluated and can be used to support the subject. For example, if the workload of a subject can be measured, a system can distribute the work the subject has to do, such that work during high workload is shifted into phases with low workload. In recent studies at our lab (see [48]) first interesting results were established in this direction. Since they do not touch the main issue of this work I will skip further details.
I should mention that the taxonomy into feedbacks for disabled and healthy people can not be very strict, of course. It is more a question for whom this feedback was originally developed.
In section 4.2.1 I will start by describing the design of the interface. It should be mentioned that there is an old interface version implemented by Roman Krepki (see [73]) which had some pioneering character. Unfortunately, some restrictions in the interface arose after some time, e.g.,

- The old interface was not flexible enough to allow other algorithms to be tested without a large amount of new programming.
- Since our toolbox is written in MATLAB, but the old interface was implemented in C++, many algorithms had to be reimplemented in C++.
- Interactions in the ongoing feedback (e.g., changing the speed of the application) were not possible.
- ...

After some experience with the interface I decided to implement a completely new version since it did not seem possible to fix all these problems. I will present the new interface in this chapter which replaced the old version of the interface. All results in this chapter except in section 4.2.3 were collected by this new interface.

4.2.1 Design of the interface

Many issues have to be considered for the implementation of the interface. First of all there is one of the central ideas of a BCI system which is the closed loop, namely that the subject is directly confronted with his own feedback. This requires fast algorithms with delays as short as possible. This immediate visualization of human thought is important, otherwise interaction with the feedback gets almost impossible or at least unattractive. Adaptation processes can additionally be used for further enhancements of the interface if the system is fast enough.

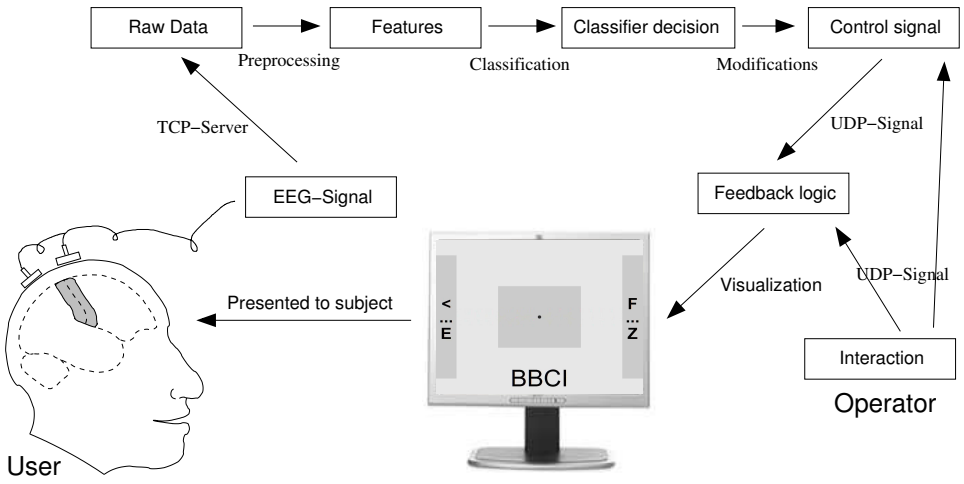


Figure 4.2: The figure shows the general BCI feedback design. Starting with the measured EEG the data are transported via TCP to a different machine that calculates a control signal which is finally used to visualize the decision of the machine about the estimation of the user's own decision. Modifications on some parameters to estimate the control signal (e.g., bias, scaling, see text) or to change the type of feedback can be done on the fly by an additional UDP control.

Furthermore, the implementation should be flexible enough so that new methods can be tested without having a high overhead in new programming. Thus a very general framework was a necessary characteristic of the system. Additionally one should note that the whole off-line BCI-toolbox at our lab is implemented in MATLAB. New algorithms are therefore developed in MATLAB. Consequently it was decided to implement the main part of the interface in MATLAB too, otherwise a parallel development of algorithms in two different languages would have been necessary. MATLAB often has very fast routines so that this decision works very well in many parts of the interface. But MATLAB also has some speed issues, e.g., the graphical visualization is very slow. For this reason it was decided to use several machines with a network communication between them so that one machine is responsible the graphical visualization only. Due to the ability of MATLAB to integrate *C*-code, communication interfaces with TCP or UDP protocols can be used. Thus the final setup for feedback (see Fig. 4.2) looks as follows: One machine to which the subject is connected records the data and provides the data on a TCP server. This part already exists in the recording software delivered by BrainProducts, the company who build the EEG-Hardware for our lab. The data are provided at 25 Hz. A second, very fast machine with a Linux system is used to acquire this data and to apply all machine learning algorithms until the control signal is received. Here a main loop is running with the following steps:

- acquisition of the data (the current packet to avoid delays), checking block numbers to be sure that no data gets lost,
- general preprocessing like channel selection, frequency filtering on the continuous data,
- choosing a window of some specific length backwards in time,

- applying the processing on this window (e.g., the feature extraction methods discussed in section 5.1),
- applying the classifier to the resulting feature vector,
- modifying of the classifier output in scaling, bias, integration etc. and combining possibly several classifiers,
- submission of the results via UDP to another machine,
- checking for changes in the used environment.

Most points are clear in the list except the third last and the last one. After the first experiments with this framework it was observed that a high enhancement can be achieved if some parameters like a bias, a scaling or a smoothness parameter are adapted manually during feedback on the subject's request. E.g., if the subject observes that the feedback has a tendency to one class, a readjustment of the classifier could take place with a bias to the other classes. So far this is done by adding a constant to the classifier output which is adjusted manually due to the user's request. Similar the scaling denotes a factor the classifier output is multiplied with. E.g., if one moves a cursor on the screen by the classifier output the speed of the cursor could be influenced by this factor. Finally a smoothness parameter describes how many points in time are averaged. With a higher smoothness parameter, more points are averaged and thus a more stable control signal, but with lower reaction time is achieved. Especially the speed of the control device can be influenced by the latter factors and was used on the subject's request, so far.
To update these parameters without stopping the interface the experimenter is able to submit changes of these values directly into the main loop by another network connection via a UDP protocol controlled by a graphical user interface (GUI). Note that one can also try to adapt these parameters automatically which is one further issue in our group. But so far the main idea was to let the user decide how reactive the system should be by varying scaling and integration. Additionally it seems that a few parameters can only be fitted on the users request such that a full automatic choice is arguable at this point.
A third machine is used to present the feedback. The data are received and checked for delays by using block numbers. Here the current package should be acquired. Afterwards the feedback is provided based on this control signal. Since different feedbacks need different control signals, individual control packages are used. It was found that some feedbacks like brainpong, cursor control feedback or a speller (like described below) are fast enough to run in MATLAB at 25 Hz on a usual machine if no other processes are interrupting them. For more complex feedbacks DirectX or OpenGL implementations are usually necessary.
Several parameters of the feedback like timing can be controlled by a GUI which submits changes into the main feedback loop via a UDP connection.
Note that all results are automatically logged for further analysis and for reconstructing the feedback.
For processing and classification purposes several routines are available. Before the main loop starts a setup file (defined beforehand) is loaded which defines the chosen routine and parameters. In general one can use every function out of the offline BCI-toolbox if the function call is of a specific type. Furthermore, several classifiers and processings can be determined to get more than one signal. Finally each EEG file consists of time markers

(e.g., Response and Stimulus) describing for example a specific timepoint the subject was prompted, say, to imagine a movement. This time information is also usable for the interface, e.g., for generating a classifier output at a certain time-point regarding the feedback which is provided for the subject.
Before applying the feedback the training of the classifier is necessary (see chapter 5–8). Here several setups in MATLAB can be implemented. For research purposes, every step of the training procedure can be executed manually or fully automatically. Finally a setup file of a specific format defining the variables for the main feedback loop results and is saved. The advantage of this saved file is that one could restart the feedback several times on different machines in different MATLAB environments.

4.2.2 Feedback applications for the disabled

The first goals of BCI systems were the development of a new communication channel for disabled subjects. Therefore first feedback implementations focus on simple choice options. In our group there are mainly two applications in this direction, the cursor control feedback (see below) and the basket feedback (see below). For these feedbacks I will also present the results of a recent study with 6 subjects from our lab with little or no BCI experience (cf. [21, 22]). Before feedback was presented to the participants of the study, a calibration measurement was performed as described in section 4.1 with the three classes: Imagination of left and right hand and foot movement of about 30 minutes. After a short analysis the most discriminable two class setup based on CSP features was chosen and the cursor control (see below) and the basket feedback (see below) presented to the user. After a few calibration runs to fit the parameters like bias and scaling on the user's request, several runs in the cursor control and basket feedback design were done.
Based on these feedbacks one can control more complicated environments or communication paths like a menu navigation or a speller which was recently successfully established in one experiment at our lab (see below). Finally I will present a feedback preparing use of a prosthesis, namely the virtual arm (see below). I should mention that many applications exist in this direction. However, I will only present applications here which are used in our lab within feedback systems. I should also mention that only studies with healthy subjects were done in our lab in this direction so far. However, it is safe to assume that these applications can be directly transferred to disabled persons.

Cursor control feedback

In this feedback the subject controls the horizontal position of a cursor and has to move it to the right or to the left for the next decision. This feedback is presented on a computer screen (see Fig. 4.3) with one small field on the left and on the right. The cursor is presented either as a black point (deactivated) or as a big red cross (activated). Only if the activated cursor is moved into one of the fields this decision is chosen. A cursor can be activated by moving it into the middle field after a specific time has elapsed since the last decision was made. The cursor is automatically deactivated after making a decision. To measure the performance of the feedback the decision is prompted by coloring the specific field the user should choose. Furthermore, on the subject's request a further small field was visualized which provides information about the upcoming decision. This increases the speed of the feedback.

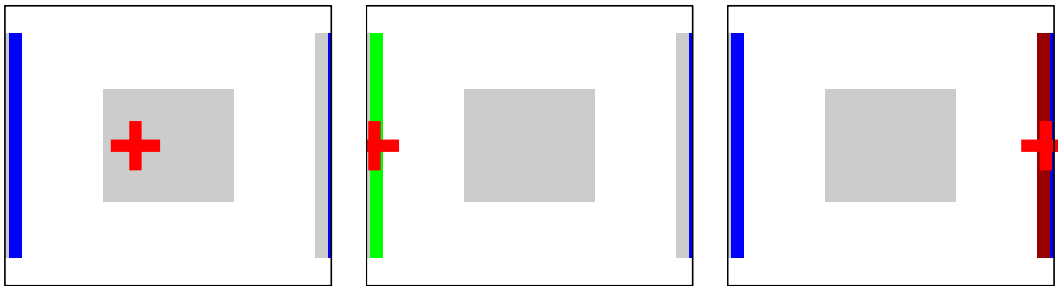


Figure 4.3: The figure shows a screenshot of the cursor control feedback. Here the cursor (red cross) can be moved by imagining different movements from left to right. The broad colored stripe on the edges prompts the desired class. If the cursor moves into the prompted class it flashes green, otherwise red. Afterwards the next decision was prompted which is announced by the small lines on the outmost edges of the target rectangles. The gray rectangle in the middle was used to activate the cursor (see text) in the position control feedback session. On the left the usual start of a trial is shown, in the middle a successful, on the right an unsuccessful trial.

Two different control options exist, position control and rate control feedback. In the position control feedback the position of the cursor is directly controlled by the control signal ($p(t) = \frac{s}{n}\sum_{i=t-n+1}^{t}(c(i)+b)$ with $p(t)$ position at time-point t, s as a scaling factor, n as an integration factor, b as a bias and $c(t)$ the classifier output at time t. Hereby $c(t)$ is the continuous output of the classifier (e.g. LDA) before taking the sign to decide for the class. One can think of it as a certainty value for each class, i.e., how certain the classifier is that the class is desired.). In the rate control feedback the position is changed by the control signal ($p(t) = p(t-1) + \frac{s}{n}\sum_{i=t-n+1}^{t}(c(i)+b)$). In the latter the cursor is moved back to the middle after a decision and fixed there for a specific time before being automatically activated. All specific times or general values like bias b, scaling s and integration n are adjustable on the user's request. Usually a run consists of 25 decisions. Finally the bitrate per minute for the

subject	**training acc [%]**	**position control [bits/min]**		**rate control [bits/min]**	
		overall	peak	overall	peak
1	95.4	7.1	15.1	5.9	11.0
2	64.6	–	–	–	–
3	98.0	12.7	20.3	24.4	35.4
4	78.2	8.9	15.5	17.4	37.1
5	78.1	7.9	13.1	9.0	24.5
6	97.6	13.4	21.1	22.6	31.5
mean	85.3	10.0	17.0	15.9	27.9

Table 4.1: The table shows the expected accuracies of the subjects in the study introduced in section 4.2.2 based on the training session and the mean and best achieved bitrate per minute during the cursor control feedback. Note that no feedback was presented to subject 2 due to his bad performance in the training session.

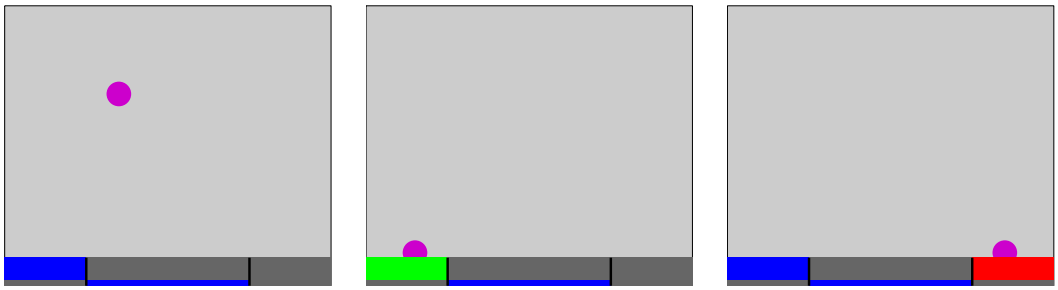


Figure 4.4: The figure shows a screenshot of the basket feedback. Here the ball can be horizontally and relatively controlled; the vertical movement was downwards at a constant speed. On the bottom the desired target is colored by a broad blue stripe, whereas the small blue stripe denote the next target. Correct decisions were colored green, wrong red. These situations are visualized in the middle and right screenshot.

subject is visualized, based on the formulas in chapter 3.
The results are presented in table 4.1 for the six subjects who took part in this feedback study. Note that for one subject it was decided after the training session to skip the feedback part due to bad performance. See Blankertz et al. [21, 22] for a more detailed overview of this study.
First of all one observes that the subjects usually perform better in the rate control feedback. Three of the 6 subjects were able to achieve peak performances of more than 30 bits/minute and mean performances around 20 bits/minute. Two other subjects were able to achieve performances between 5 and 10 bits/minute. In Wolpaw et al. [141] it was stated that a few subjects, which were trained over weeks or even months in BCI experiments, were able to achieve up to 25 bits/min in their best sessions. Consequently, the result of the presented study outperforms the results established by other groups considerably in two directions: First three out of six subjects were able to achieve peak performances of more than 30 and up to 37 bits/minute and a fourth achieved 25 bits/minute in his best session and second these results were performed without any subject training.
It should be mentioned that in this study the CSP algorithm based on ERD effects in $\mu-$ and β-rhythm was used. Further enhancements by integrating feature combination (see chapter 6) and CSSSP extensions (see chapter 8) can be expected.

Basket feedback

In the same study a further feedback was presented, namely the so called basket feedback. Here three fields are visualized on the bottom of the screen (see Fig. 4.4) each representing one decision. At the beginning of a trial the target field is colored blue, the next target is visualized by a small further field below, if the user requires this information. By this additional information the user is able to prepare the next movement earlier since he knows which the next desired target after the current one will be. Then a ball falls down vertically with a constant velocity. The horizontal position of the ball is controlled by the rate control feedback with the same classes as described above. A decision is made when the ball reaches

subject	training acc [%]	basket [bits/min] overall	peak
1	95.4	2.6	5.5
2	64.6	–	–
3	98.0	9.6	16.1
4	78.2	6.6	9.7
5	78.1	6.0	8.8
6	97.6	16.4	35.0
mean	85.3	8.2	15.0

Table 4.2: The table shows the expected accuracies of the subjects in the study based on the training sessions and the mean and best achieved bitrate per minute during the basket feedback. Again no feedback was presented to subject 2 due to his bad performance in the training session.

the bottom of the screen. Afterwards the ball starts again in the middle at the top of the screen. Finally the bitrate is again visualized after 25 decisions.
The results are visualized in table 4.2. Here the performance gets worse compared to the cursor control feedback. Only one subject was able to achieve peak performance higher than 30 bits/minute and mean performance higher than 15 bits/minute. Considering that the subjects had little or no experience with a BCI and especially with this type of feedback, it is likely that the performance can be further enhanced on familiarization process with the BCI and especially with this type of feedback.

Speller feedback

A simple but very effective communication channel can be built using the results of the cursor control and basket feedback: the speller (see Fig. 4.5). With this application the user is able to write text. In this environment the fields in the cursor control and basket feedback were exchanged for groups of letters. One starts with all 26 letters, a space and a deletion symbol and splits them into several groups, depending on the number of decisions which can be achieved (for cursor control feedback two, for basket it depends on the number of fields). The subject is able to choose one of these decisions as described above (no prompting takes place anymore). Afterwards the selected group is split and the process is repeated further until one letter or symbol remains. Then the process starts again with the full set of symbols to write the next letter. Since errors could happen, a deletion symbol is added which cancels the last letter. This is the implementation of the standard tree, presented in chapter 3. One could also use different trees as described in chapter 3. However the first subject who did this experiment was subject 3 of the study described in this section who had such a high performance that one should choose the standard tree (see chapter 3). Based on the results above the rate control cursor feedback was chosen due to it achieving the best performance. The splitting was done alphabetically based on the probabilities of the German alphabet with deletion and space right at the beginning. One could also apply some Huffman coding algorithms. However, in an alphabetically ordered tree the search for the next decision is much easier for the user so this approach was preferred. For alphabetical trees the probabilities are used to split the tree so that the cumulative probabilities for the groups are as similar as possible. Note that this approach is not optimal for alphabetic trees. There are algorithms (see [35]) which could be used to get an optimal alphabetic tree, but it turns out that the difference is usually very small.
In the experiment performed subject 3 was able in 30 minutes to write the following German

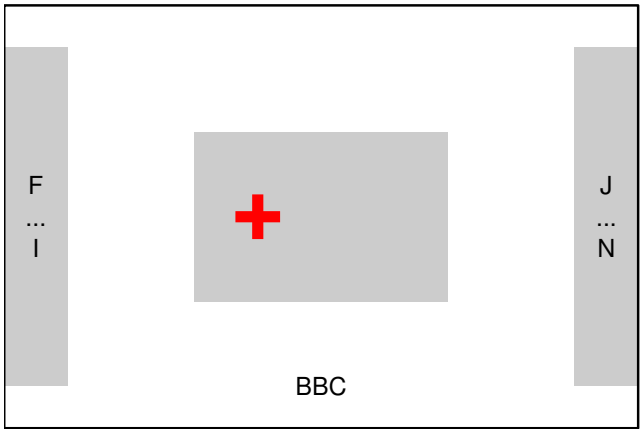


Figure 4.5: The figure shows a screenshot of the speller feedback. It looks similar to the cursor control feedback except that groups of letters appear in each field. Furthermore the written text is visualized on the bottom of the screen.

saying and two sentences (which were used for the first communication via a telephone):
AM BERG DA RAUSCHT DER WASSERFALL WENNS NICHT MEHR RAUSCHT ISTS WASSER ALL DAS PFERD FRISST KEINEN GURKENSALAT DIE SONNE IST VON KUPFER
During this process the subject performed more than 1000 decisions with around 60 mistakes which corresponds to around 20 bits/minute. According to formula (3.1) taking the probabilities of the letters in the German language into account which were used to build the alphabetical tree, the written text corresponds to around 544 bits, i.e., 18 bits/minute were really achieved with this interface. Thus the difference between the theoretical information transfer rate and the achieved rate is not very large which again strengthens the results of chapter 3.
It should be mentioned that some of the mistakes the subject encountered in this feedback setting were not due to misclassification on behalf of the classifier, but rather his own, since he sometimes recognized too late that the desired letter corresponded to the other decision. These mistakes will diminish automatically once the user becomes more familiar with the position of the letters over time. Furthermore compared to the cursor control feedback the time the cross was frozen in the middle was chosen to be longer due to the fact that the subject needs some time to be able to find on which side the desired letter is placed which is a harder task than finding the colored area.

Virtual arm

Using a totally different method, amputees could use a BCI to control a neuroprosthesis by thought. Based on the observation that different muscles on the arm are controlled by slightly different brain areas in the motor cortex one can try to use the movement of the shoulder or the finger to control a virtual arm on a computer screen (see Fig. 4.6). Here only full movements were used, gradual movements were not tried.
In this experiment one healthy subject performs a training session with real movements of shoulder and finger of the same arm in the selfpaced design. Since it was a first shot the first

Figure 4.6: The figure shows three pictures out of a movie of the virtual arm feedback. The subject were able to move the shoulder or the right finger by his own movements, but controlled only on the EEG signal. On the left the situation without movement, in the middle with finger and on the right with shoulder movement is shown.

goal was a good discrimination only, and not directly a detection of the ongoing movement in the feedback situation. Thus in the subsequent application of the feedback a discrimination was initiated by the key-press of the real movement. The classification was then performed on LRP features of EEG data only collected up to 50 ms before the key-press since one idea of this interface is the prediction of the movement before the movement happens. The decision of the classifier was then directly visualized by the movement of the virtual arm. Hereby correct classification rates of about 75 % could be achieved whereas classification by chance would be at 50 %. For real-life BCI which predicts the intended movement two enhancements are necessary: the accuracy should be increased and the detection of an upcoming movement is required to be able to really predict the movement (and not only the type of movement). The virtual arm feedback was initiated to allow control of a neuroprosthesis by an arm-amputee. Of course, to achieve a BCI usable in real life the same problems as above have to be solved, except that one is not forced to restrict oneself to the prediction of an upcoming movement: It is sufficient that a system detects the desired movement shortly after the subject initiates it.

4.2.3 Movement Prediction

Based on a selfpaced experiment with left and right hand finger movements two classifiers on LRP features are trained, one for the detection of a movement, one for the discrimination between left and right hand movements. Afterwards a cursor-feedback is presented during a further selfpaced session. Here the vertical position of the cursor was controlled by the continuous output of the detection classifier, the horizontal position was controlled by the discrimination. For example, the cursor should move upwards if an upcoming event is detected since the value of the detection classifier should increase and simultaneously the cursor should move to the left if a left hand movement is prepared. Two fields on the top left and top right are shown (see Fig. 4.7) defined as fields where the classifier has detected a movement. During the feedback the cursor is shown with a short tail showing its path, i.e., the position during the last 240 ms. If a key is pressed the cursor is frozen in that position for a short time.
In Fig. 4.7 the results for this experiment are visualized, but frozen at a time-point 80 ms

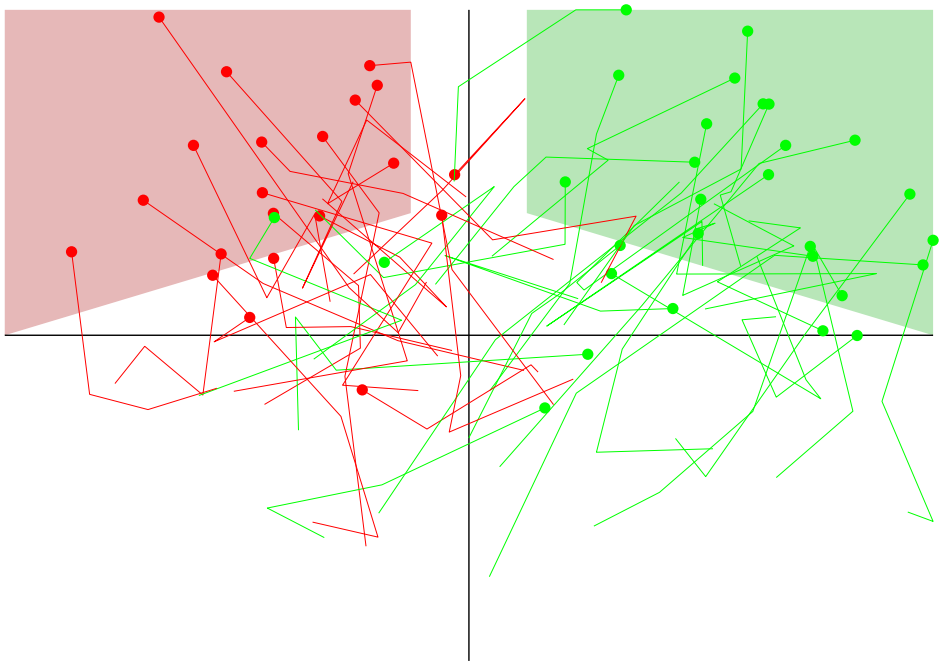


Figure 4.7: This figure visualizes the feedback for prediction of upcoming movements. There are two classifiers, one detecting the movement in the vertical direction (i.e., the vertical position is controlled by the continuous output of the detection classifier), and simultaneously one discriminating between left and right hand in the horizontal direction. The colored fields on the left and right correspond to fields where the classifier would inform about the detection of the movement. The points visualizes the position of the cursor 80 ms before the real key-press, the lines visualize the history 240 ms before this detection, i.e., the cursor positions during this time. Left key-presses are red, right key-presses are green.

before key-press. The line behind the point is 240 ms long. The shapes are colored by the pressed key: red (for left) or green (for right). Usually the red points lie in the upper left corner with a line coming from the lower middle, whereas green points lie in the upper right corner with a line also coming from the lower middle. Thus the classifier in this experiment was able to discriminate the movement 80 ms earlier in 96 % of all cases correctly, whereas 76 % of all movements are also detected correctly at this timepoint. Note that the movement prediction is usually a harder task than laterality prediction since in this experimental design it is hard to find good examples for a non-movement class. However, one can conclude that the ability of movement prediction was successfully proven by this experiment.

4.2.4 Gaming applications

Many gaming applications exist. However, I will briefly describe only one, called brainpong, which was successfully used in a recent online experiment in our group.
In brainpong a subject is able to move a racket on the bottom of the screen in a horizontal

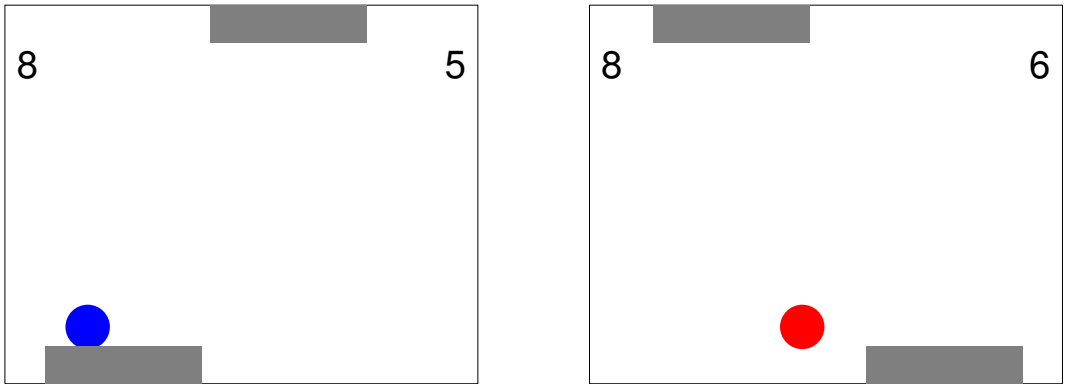


Figure 4.8: The figures show brainpong for two players. One player controls the racket on the bottom of the screen, the other the one at the top by imagination of movements. The ball moves around and is reflected by the rackets of the player or by the edges on the left and the right of the screen. The left figure shows a hit of the ball. Here the ball is now reflected and the other player has to hit the ball. On the right the ball is missed. In this case the ball is colored red, the other player gets a point, the game is paused for a short time usually less than one second and the game starts again.

direction by a one-dimensional classifier. Usually the racket has a width less than half of the screen but this depends on the ability of the user, since the games become easier if the racket is larger. A ball moves around on the screen reflected by the borders of the screen except at the bottom. If the ball reaches the bottom of the screen, the user has to reflect this ball by moving the racket between the ball and the border (see Fig. 4.8). In the case of a successful hit of the ball the subject gets a point, otherwise not. So far the model of the ball and racket is very strict (e.g., no drift of the ball so far). However, it would be easy to implement a more realistic physical model of the ball by e.g., incorporating spin, reflections etc.
A more interesting application in terms of game-play is to play this with a further person. Here the racket of the second player is placed on the other side of the screen. In this case a player gets a point if the other player misses the ball. This game was recently successfully played in our group. Here a game usually consists of at least 10 hits.

5 Signal Processing and Machine Learning

Following the leitmotif *let the machines learn*, a central role of the Berlin BCI is the use of suitable machine learning techniques for adapting the computer to the specific brain patterns of the subject. Several points have to be considered here. Beginning with the unprocessed EEG data one has to reduce the dimensionality of the data without loosing relevant information. This step is called feature extraction and is described in section 5.1. Since techniques from signal processing are usually important tools for feature extraction, a part of this section is dedicated to this topic. Based on the derived features classification has to be done, i.e., a function has to be learned which optimally separates the data in feature space. This will be described in section 5.2. This separation has to be done in such a way that it works optimally on new unseen data. The issue of generalization will be discussed in section 5.3. Finally, EEG data are usually distorted by artifacts which have to be reduced. This problem is usually called robustification in the machine learning world. I will briefly illuminate this point in section 5.4.
An overview of possible machine learning techniques is also given in Müller et al. [98].

5.1 Feature Extraction

Usually it is hard for classification algorithms to extract the relevant information if the dimensionality of the data compared to the number of existing examples is very high. This is called *Curse of Dimensionality* in the machine learning world. The dimensionality has to be reduced suitably in the sense that *undiscriminative* information is eliminated whereas *discriminative* information remains. There are several ways used in literature which can be separated more or less into two fields: On the one hand one incorporates neurophysiological a priori knowledge to find and extract neurophysiological features, e.g., calculating the band-power in prominent discriminative frequency ranges on well-known discriminative scalp locations. Here I will shortly introduce temporal filtering methods from signal processing like Finite Impulse Response Filter (FIR) (see section 5.1.2), Infinite Impulse Response Filter (IIR) (see section 5.1.1) and Fourier Based Filter (FFT) (see section 5.1.3) and spatial filtering methods like bipolar (see section 5.1.4), common average reference (CAR) (see section 5.1.5) and Laplace filtering (see section 5.1.6).
On the other hand there is the opportunity to use advanced machine learning techniques to blindly extract relevant features. For the latter one can use techniques like Principal Component Analysis (PCA) (see section 5.1.7), Independent Component Analysis (ICA) (see section 5.1.8) or Common Spatial Patterns (CSP) (see section 5.1.9). Alternatively one can use scoring functions like the Fisher Score (see section 5.1.10) or sparse classifiers (discussed for the Linear Programming Machine in section 5.2.7) for feature extraction. A broader overview about existing feature extraction methods can be found in Anderson [1].
In my opinion ignoring neurophysiological a priori knowledge completely and only using

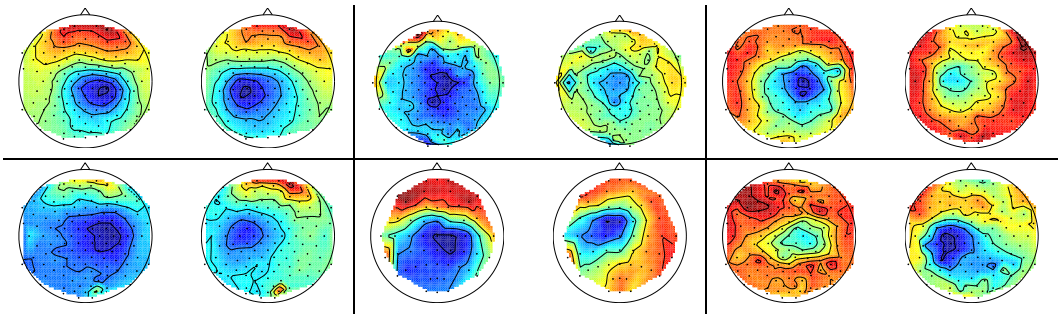


Figure 5.1: The plot shows activation for six different subjects, i.e., the negativation and positivation regarding some baseline, between 100–200 ms before a key-press. The left scalp corresponds to left hand key-press, right scalp to right hand key-press in each rectangle. In blue areas the EEG has a negative shift compared to a baseline interval beforehand, in red areas a positive shift.

advanced machine learning techniques is not advisable. But, the restriction to simple performance values described by neurophysiology suffers from the fact that there is a high trial and subject variability. The latter is visualized in Fig. 5.1. Here the activation before left or right hand finger movement is shown for six subjects. As can be seen, the variation is pronounced. The usual way of finding a feature in neurophysiology is to look at averages over many trials, and maybe for many subjects. It is obvious that this can help to find a stable and suitable feature, but that is usually not enough: To be able to interpret EEG data on a single-trial level with high performance for each subject this processing should adapt to the specific brain. Consequently, the best way is to use advanced feature extraction techniques on data which has been preprocessed using neurophysiological a priori knowledge. The CSP algorithm (see section 5.1.9) is one prominent example which combines ideas from machine learning with neurophysiological a priori knowledge to reduce the dimensionality of the problem.

5.1.1 Infinite Impulse Response Filter

If restrictions to some frequency bands are reasonable, several ways exist to do so. One common approach is the use of a digital frequency filter. Regarding the desired frequency range two sequences a and b with length n_a and n_b are required which can be calculated in several ways, e.g., butterworth or elliptic (cf. [106]). Afterwards the source signal x is filtered to y by

$$\begin{aligned} a(1)y(t) = \quad & b(1)x(t) + b(2)x(t-1) + \ldots + b(n_b)x(t-n_b-1) \\ - \quad & a(2)y(t-1) - \ldots - a(n_a)y(t-n_a-1) \end{aligned}$$

for all t. Usually one cannot construct a filter with a strict frequency range. In fact, there are small ranges at the border of the desired frequency range where only parts are filtered, i.e., some frequency component remains. However, for usual frequency ranges for BCI application suitable filters can be created.
One disadvantage of the IIR filter remains: The filtered signal is delayed, i.e., changes in power can only be recognized at a later point in time. To solve this problem the same filter

is often applied backwards to the data. In this case the time delay is moved in the other direction. Unfortunately, this procedure does not make sense in online environments. Since one has to decide in the present, one is only allowed to use data from the past and of course the actual data, but no future data. Considering that the goal of a BCI is online application, I will not use the backwards filtering during this work, neither for creating meaningful plots nor for measuring offline performance by validation.

5.1.2 Finite Impulse Response Filter

This filter is an IIR filter with a small modification: Here n_a and a are both fixed to 1. In other words only the sequence b remains and the signal x is filtered to y by

$$y(t) = b(1)x(t) + b(2)x(t-1) + ... + b(n_b)x(t-n_b-1)$$

for all t.

5.1.3 Fourier Based Filter

Another alternative for temporal filtering is Fourier based filtering. By calculating the Fast Fourier Transformation (FFT) (see [106]) of a signal one switches from the temporal to the spectral domain. The filtered signal is obtained by choosing a suitable weighting of the relevant frequency components and applying the Inverse Fast Fourier Transformation (IFFT). Since FFT calculation is based on complex numbers, one has to take the real part of the filtered signal.

5.1.4 Bipolar Filtering

During bipolar filtering the difference between two channels is calculated. Since the EEG consists of many signals which are highly overlapped, many parts measured at one electrode are also visible in neighboring electrodes. Thus most of the signal measured at one electrode does not belong to this single location. By calculating differences between suitable electrodes the common part in all electrodes is filtered out and the relevant part remains. However, this only works if one knows which electrodes should be subtracted.

5.1.5 Common Average Reference (CAR)

Given a set of EEG channels one can calculate the mean of these channels and subtract this signal from all other channels. Suppose $S \subset \{\text{all channels}\}$ is a subset of all available channels, one calculates the CAR filtered signal $\hat{s} = (\hat{s}_1, ..., \hat{s}_c)_{c=1,...,\text{number of channels}}$ of the source signal $s = (s_1, ..., s_c)_{c=1,...,\text{number of channels}}$ by

$$\hat{s}_j = s_j - \frac{1}{\#S} \sum_{i \in S} s_i$$

for all channels j. Usually one chooses the set S as the whole set of all channels if not specified otherwise.

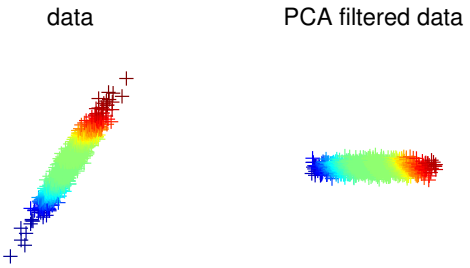


Figure 5.2: On the left Gaussian distributed data are visualized. After applying PCA the source signals on the right are retained. The points are colored so that for each point in each plot the same color is used.

5.1.6 Laplace filtering

One disadvantage of CAR is that channels at scalp positions far away from each other are combined which probably have no similar content. An alternative is the Laplace filter. Here for each channel j a neighborhood $N_j \subset \{\text{all channels}\}$ is defined. Then the Laplace filtered signal $\hat{s} = (\hat{s}_1, \ldots, \hat{s}_c)_{c=1,\ldots,\text{number of channels}}$ of the source signal $s = (s_1, \ldots, s_c)_{c=1,\ldots,\text{number of channels}}$ is defined by

$$\hat{s}_j = s_j - \frac{1}{\#N_j} \sum_{i \in N_j} s_i$$

for all channels j. The definition of the neighborhood of one channel remains an open issue. Several techniques like using all four/eight direct neighbors, or only using the horizontal or vertical neighbors exist and mainly depend on the nature of the EEG cap used. During this thesis I only apply the Laplace filter with four neighbors (two vertical and two horizontal one). Hereby electrodes that do not have all four neighbors are usually skipped.

5.1.7 Principal Component Analysis

Given some data $x_k \in \mathbb{R}^m$ for $k = 1, \ldots, n$ PCA tries to reduce the dimensionality of the problem by finding an optimal approximation of the data x_k by $x_k \approx b + Wa_k$ with $b \in \mathbb{R}^m$, $a_k \in \mathbb{R}^p, p \leq m$ and $W \in \mathbb{R}^{m,p}$. If this optimization is done by minimizing the squared error $\sum_{k=1,\ldots,n} \|x_k - (b + Wa_k)\|_2$ and simultaneously fixing the diagonal of $W^\top W$ to 1, one finds the solution by choosing $b = \frac{1}{n} \sum_{k=1,\ldots,n} x_k$, W by the eigenvectors of the highest p eigenvalues (suitably scaled) of the so called scatter matrix $\sum_{k=1,\ldots,n} (x_k - b)(x_k - b)^\top$ and $a_k = W^\top (x_k - b)$. Consequently, W consists of orthogonal vectors, describing the p-dimensional subspace of $\mathbb{R}^m$ which shows the best approximation to the data. For normal distributed data one finds the subspace of the covariance matrix where the most variation in the data is.
In Fig. 5.2 the principal components of a two-dimensional Gaussian distribution is visualized. In this case the data were only rotated.
In Müller et al. [96] this idea was extended to non-linear data by kernelization and is called kernel PCA (kPCA).

5.1.8 Independent Component Analysis

Suppose n recorded signals $x(t) = (x_1(t), \ldots, x_n(t))$ for $t = 1, \ldots, T$ are given. The basis assumption of ICA is that these n signals are modeled as linear combination of n un-

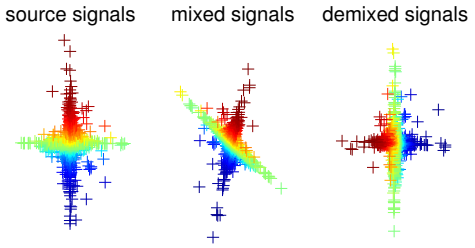


Figure 5.3: On the left two independent source signals (Gaussian to the power of 3) are shown. After multiplication of a mixing matrix the mixed signal in the middle is achieved. After applying JADE the signals on the right are revealed. The points are colored so that for each point in each plot the same color is used.

known source signals $s(t) = (s_1(t), ..., s_n(t))$ with $x_i(t) = \sum_{j=1}^{n} a_{i,j} s_j(t)$ for $i = 1, ..., n$ and $t = 1, ..., T$. This can be reformulated to $x(t) = As(t)$ with the so-called mixing matrix $A = (a_{i,j})_{i,j=1,...,n}$ which is assumed to be square and invertible. Obviously one needs further assumptions to be able to reconstruct A and s if both are unknown. A reasonable and thus the key assumption of ICA is the independence of the source signals, i.e., that the time course of $s_i(t)$ does not provide any information about the time course of $s_j(t)$ for $j \neq i$. Thus ICA tries to find a separating matrix B such that the resulting signals $y(t) = Bx(t)$ are spatially as independent as possible.

Driven by this goal one can find a solution (up to permutation and scaling) if at most one source has a Gaussian distribution, or the source signals have different spectra, or the source signals have different variances. Tools from information geometry and the maximum likelihood principle are used here to get an objective function for an optimization approach (see [65]).

Several algorithms exist depending on the initial situation. If one assumes non-Gaussianity one can use JADE (joint-approximate diagonalization of eigenmatrices) (cf. [28]), FastICA (cf. [64]) and infomax (cf. [7]). If one assumes time structure (like different spectra or variances) the prominent algorithms are TDSEP (cf. [144]) and SOBI (cf. [8]) which are both equivalent. If one assumes independent data (i.e., no time structure) but non-stationarity in the data SEPAGAUS (cf. [112]) is also an interesting tool. All these algorithms use the linear assumption $x(t) = As(t)$. For non-linear extensions of the TDSEP algorithm by kernelization I refer to Harmeling et al. [61, 62].

The typical ICA situation is visualized in Fig. 5.3. Here two independent source signals (Gaussian to the power of 3) were mixed by a random non-orthogonal matrix to get the mixed signals. Now the JADE algorithm was applied to the data so that the demixed signals remain which after suitable reordering and scaling is very similar to the source signal. PCA would fail here since the mixed signals are not orthogonal in general which is the key assumption for PCA.

5.1.9 Common Spatial Patterns

The CSP algorithm (see [55]) is very useful for determining spatial filters for ERD effects (see [69]) and thus for ERD-based BCIs (see [116]): Given two distributions in some arbitrarily high-dimensional space, the (supervised) CSP algorithm finds directions (i.e., spatial filters) with the biggest difference in variance between two classes, in other words, it tries to

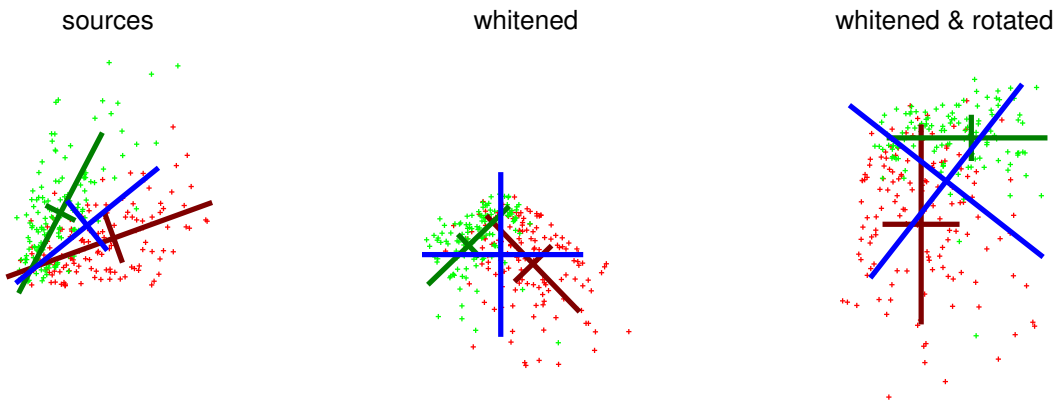


Figure 5.4: On the left two dimensional data for two classes are shown. They were whitened (middle figure) and then suitable rotated (right figure).

maximize variance for one class and at the same time minimize variance for the other class. Since ERD effects are only localized in specific brain rhythms (i.e., in specific frequency ranges) a band-pass filter focussing on the rhythms of interest is applied to the EEG signals beforehand. With the applied spatial filter a rhythm with large amplitude for one class and low amplitude for the other class is retained.
The CSP algorithm is trained on labeled data, i.e., a set of trials $s_i \in \mathbb{R}^{\#\text{channels},\#\text{samples}}$, $i = 1,2,\ldots$ is given. A spatial filter $w \in \mathbb{R}^{\#\text{channels}}$ projects these trials to the signal $w^\top s_i$ with only one channel. The idea of CSP is to find a spatial filter w such that the projected signal has high variance for one class and low variance for the other. In other words one maximizes the variance for one class whereas the sum of the variances of both classes remains constant, which is expressed by the following optimization problem:

$$\max_w \quad \sum_{i:\text{Trial in Class 1}} var(w^\top s_i), \quad \text{s.t.} \quad \sum_i var(w^\top s_i) = 1, \tag{5.1}$$

where $var(\cdot)$ is the variance of the vector. An analogous formulation can be found for the second class which has the same effect as calculating the minimum in the optimization problem (5.1).
The optimization problem (5.1) can be simplified to

$$\max_w \quad w^\top \Sigma_1 w, \quad s.t. \quad w^\top (\Sigma_1 + \Sigma_{-1}) w = 1, \tag{5.2}$$

where $\Sigma_y \in \mathbb{R}^{\#\text{channels},\#\text{channels}}$ is the covariance matrix of the trial-concatenated matrix of dimension [channels $\times$ concatenated time-points] belonging to the respective class $y \in \{\pm 1\}$. Formulating the dual problem one finds that the problem can be solved by calculating a matrix Q and diagonal matrix D with elements in $[0,1]$ such that

$$Q\Sigma_1 Q^\top = D \qquad \text{and} \qquad Q\Sigma_{-1} Q^\top = I - D \tag{5.3}$$

and by choosing the highest eigenvalue (and the lowest for the minimum). Several ways exist to solve this problem, for example by calculating generalized eigenvalues or by *whitening* the matrix $\Sigma_1 + \Sigma_{-1}$, i.e., determine a matrix P such that $P(\Sigma_1 + \Sigma_{-1})P^\top = I$ which

is possible due to positive definiteness of $\Sigma_1 + \Sigma_{-1}$ and by using spectral theory to get $P\Sigma_1 P^\top = RDR^\top$ ($\Rightarrow P\Sigma_{-1}P^\top = R(I-D)R^\top$). This process is demonstrated in Fig. 5.4 for two-dimensional data. The outgoing signal (left figure) is whitened (middle figure) and then suitably rotated (right figure). For CSP the highest and lowest value of the diagonal of D and the corresponding vectors of $Q = R^{-1}P$ have to be chosen. Typically one would retain some projections corresponding to the highest eigenvalues for each class to have several filters.

For feature extraction one uses this algorithm on suitable band-pass filtered signals. The used band can be given either by neurophysiological a priori knowledge or by techniques described in chapter 8. After applying the CSP algorithm band-power is calculated by variances, transferred to a logarithmic scale and the feature is retained. Thus this approach is driven by the idea of combining both neurophysiological a priori knowledge and advanced machine learning techniques: Based on the idea of using ERD effects, i.e., changes in prominent brain-rhythms, one calculates the band-power in the specific frequency band, but on suitably spatially filtered data, which is revealed by enhanced machine learning techniques.

Another motivation of the CSP algorithm arises from the following: Suppose that each timepoint of each trial $s_i(t)$ is derived by a Gaussian distribution $\mathcal{N}(0,\Sigma_y)$ where only the spatial covariance depends on the label and the trials are independent in time. One gets that $w^\top s_i(t)$ is Gauss-distributed $\mathcal{N}(0, w^\top \Sigma_y w)$. Thus with $v^2 := w^\top \Sigma_y w$ it holds true that $\log(w^\top s_i(t) s_i(t)^\top w) \sim \log((\mathcal{N}(0, w^\top \Sigma_y w))^2 = \log((\mathcal{N}(0,v^2))^2) = \log(v^2(\mathcal{N}(0,1)^2)) = \log v^2 + \log((\mathcal{N}(0,1))^2) = \log(w^\top \Sigma_y w) + \log \chi^2$ with χ^2 as the well-known χ^2-distribution (see [58]). Obviously, the optimal discrimination is given by maximizing the difference between $w^\top \Sigma_1 w$ and $w^\top \Sigma_{-1} w$. Thus the CSP approach is derived. Note that the assumption of independence, i.e., that the signal has no time structure, is contentious.

The CSP feature has been very successful in our lab when used on ERD phenomena of imagined or real movement datasets and was therefore also used for feedback applications. It was further applied successfully on slow potentials (see [42]).

5.1.10 Fisher Score

Another interesting opportunity to extract features is based on finding a scoring function for each dimension and choosing the highest ones. One example is the Fisher Score. Given the labels, the Fisher score for data $(x_k)_{k=1,\dots,N}$ with labels $(y_k)_{k=1,\dots,N}$ is defined for all dimensions i by:

$$s_i = \frac{|\mu_1^{(i)} - \mu_{-1}^{(i)}|}{\sigma_1^{(i)} + \sigma_{-1}^{(i)}}$$

with $\mu_y^{(i)} := \frac{1}{\#\{k:y_k=-y\}} \sum_{k:y_k=-y} x_{k,i}$ and $\sigma_y^{(i)} = \frac{1}{\#\{k:y_k=y\}} \sum_{k:y_k=y} (x_{k,i} - \mu_y^{(i)})^2$ for $y = \pm 1$. A few highest values, i.e., the most discriminative dimensions, are extracted. Alternatively one could also choose the r^2- or r-value (see [135]) for feature extraction. See Guyon et al. [59] for more scoring functions.

Since high scored features are often highly correlated (i.e., redundant) these scores are typically not used for feature selection but rather for visualization of discriminability or as a basis for heuristic methods or for semi-automatic feature selection by a human operator.

5.2 Classification

Given n labeled trials in the form (x_i, y_i) for $i = 1, ..., n$ with $x_i \in \mathbb{R}^m$ as data points in some Euclidean space and $y_i \in \{1, ..., N\}$ as class labels for $N > 2$ different classes or $y_i \in \{\pm 1\}$ as class labels for a binary problem. The goal of classification is to find a function $f : \mathbb{R}^m \to \mathbb{R}^N$ or $f : \mathbb{R}^m \to \mathbb{R}$ such that for an $x \in \mathbb{R}^m$ the function $\operatorname{argmax} f(x)$ or $\operatorname{sign} f(x)$ is a *very good guess* for the true label. For example, if the data can be described by a probability distribution X (for the data) and Y (for the label) one would try to minimize the misclassification risk $P(\operatorname{argmax} f(X) \neq Y)$ or $P(\operatorname{sign} f(X) \neq Y)$. Unfortunately the probability distributions are usually not given, only a finite number of samples coming from these distributions are presented. Thus in this case the probability distribution has to be estimated.

It should be mentioned that in the following I will use the one-dimensional classifier $f : \mathbb{R}^m \to \mathbb{R}$ instead of the two-dimensional classifier $f : \mathbb{R}^m \to \mathbb{R}^2$ for binary problems. Note that both formulations are equivalent since finding the maximum of two values can be decided by the sign of the difference.

Note that I will use the function argmax in two directions. First $\operatorname{argmax} z$ can define the index of the highest entry of vector z. The second meaning is given by $\operatorname{argmax}_{i \in Z} z_i$ with some set Z. In this case z_i has to be a real value and $\operatorname{argmax}_{i \in Z} z_i$ has to be calculated over the finite sequence $z_{i \in Z}$. One can distinguish these two cases by the use of the index of argmax. For example $\operatorname{argmax} f_i$ means the index of the highest entry of the vector f_i for a fixed i, whereas $\operatorname{argmax}_{i \in Z} f_i$ means the index of the highest element f_i for all $i \in Z$.

Several ways exist to fit a classifier to the data. Here one separates existing methods mainly in two directions, generative or discriminative classifiers. A generative classifier starts with some assumptions about the probability distribution of the data and estimates all important parameters. Based on the idea minimizing the misclassification risk, one tries to find a suitable classifier matching this requirement. A prominent classifier in this direction is Quadratic Discriminant Analysis (QDA) (see section 5.2.1) or its specialization Linear Discriminant Analysis (LDA) (see section 5.2.2). Furthermore, one interesting modification of these algorithms exists which takes care of overfitting effects by suitable regularization called Regularized (Linear) Discriminant Analysis (RDA or RLDA) (see section 5.2.3).

A discriminative classifier starts by defining a loss function and thus an optimization function on the data, e.g., minimizing the squared training error on the data. Several optimization functions exist. In this work I will discuss Least Square Regression (LSR) (see section 5.2.4), Fisher Discriminant Analysis (see section 5.2.5), Support Vector Machines (see section 5.2.6) and Linear Programming Machines (LPM) (see section 5.2.7). A totally different approach, the k-nearest neighbor algorithm, is briefly introduced in section 5.2.8. Further methods like Adaboost (cf. [86]) and Neural Networks (cf. [13]) are skipped. Since multiple kernel learning will be used in this work, this method in its linear version will be briefly introduced in section 5.2.10.

All these classifiers work very well if the optimal classification can be done linearly. If a more complex decision function is appropriate they usually fail since the function class is too restricted. The kernelization technique discussed in section 5.2.9 tries to solve this problem by mapping the data in a space where linear classification makes sense.

A broader overview about existing classification methods can be found in Anderson [1].

Finally, I will briefly illuminate the question of whether linear or non-linear classifiers (see

section 5.2.11) should be used. This question can not be answered in general, since it depends on the given situation.

5.2.1 Quadratic Discriminant Analysis

Let us consider the following situation, namely that the given data are normal distributed:

5.2.1 Theorem: *Let $X \in \mathbb{R}^m, Y \in \{1,...,N\}$ or $Y \in \{\pm 1\}$ random variables with $m,N \in \mathbb{N}, N \geq 2$ fixed and $X|Y = y \sim \mathcal{N}(\mu_y, \Sigma_y)$ normal distributed for $y = 1,...,N$ or $y = \pm 1$ with $\mu_y \in \mathbb{R}^m$ and $\Sigma_y \in \mathbb{R}^{m,m}$ positive definite. Furthermore define $\hat{f} : \mathbb{R}^m \to \mathbb{R}^N$,*

$$x \mapsto \left(-0.5x^\top \Sigma_y^{-1} x + \mu_y^\top \Sigma_y^{-1} x - 0.5\mu_y^\top \Sigma_y^{-1} \mu_y + \log(P(Y=y)) - 0.5\log(\det(\Sigma_y))\right)_{y=1,...,N}$$

resp. $\hat{f} : \mathbb{R}^m \to \mathbb{R}$

$$\begin{aligned} x \quad &\mapsto \quad \left(-0.5x^\top \Sigma_1^{-1} x + \mu_1^\top \Sigma_1^{-1} x - 0.5\mu_1^\top \Sigma_1^{-1} \mu_1 + \log(P(Y=1)) - 0.5\log(\det(\Sigma_1))\right) \\ &- \quad \left(-0.5x^\top \Sigma_{-1}^{-1} x + \mu_{-1}^\top \Sigma_{-1}^{-1} x - 0.5\mu_{-1}^\top \Sigma_{-1}^{-1} \mu_{-1} + \log(P(Y=-1)) - 0.5\log(\det(\Sigma_{-1}))\right). \end{aligned}$$

Then for all functions $f : \mathbb{R}^m \to \{1,...,N\}$ or $f : \mathbb{R}^m \to \{\pm 1\}$ with $\bar{f} := \text{argmax}(f)$ or $\bar{f} := \text{sign}(f)$ it holds true that

$$E(f(X) = Y) \leq E(\bar{f}(X) = Y).$$

In other words, $\bar{f}$ is the Bayes optimal classifier for this problem.

Proof: see A.2.
These results can be further simplified if equal class priors are assumed. However, this optimal classifier for normal distributed data is called Quadratic Discriminant Analysis (QDA). To use it one has to estimate the class covariance matrices and the class means. This is usually done by $\mu_y = \frac{1}{\#\{j:y_j=y\}} \sum_{j:y_j=y} x_j$ and $\Sigma_y = \frac{1}{\#\{j:y_j=y\}-1} \sum_{j:y_j=y} (x_j - \mu_y)(x_j - \mu_y)^\top$ if the data are given as column vectors. Note that the optimality of the classifier is only given, if the distribution is known. But if the distribution has to be estimated, which is usually the case, the required classifier is possibly not optimal anymore.

5.2.2 Linear Discriminant Analysis

Under specific assumptions theorem 5.2.1 can be simplified as follows:

5.2.2 Corollary: *In the situation of theorem 5.2.1 with $\Sigma = \Sigma_y$ for all $y \in \{1,...,N\}$ resp. $y \in \{\pm 1\}$ the optimal function $\hat{f}$ is given by*

$$\hat{f}(x) = \left(\mu_y^\top \Sigma^{-1} x - 0.5\mu_y^\top \Sigma^{-1} \mu_y + \log(P(Y=y))\right)_{y=1,...,N}$$

resp.

$$\hat{f}(x) = \left((\mu_1 - \mu_{-1})^\top \Sigma^{-1} x - 0.5(\mu_1 - \mu_{-1})^\top \Sigma^{-1} (\mu_1 + \mu_{-1}) + \log(P(Y=1)) - \log(P(Y=-1))\right).$$

□

This classifier is called Linear Discriminant Analysis (LDA). Again one can simplify this problem by assuming equal class priors. The parameters can be estimated as above where Σ is estimated by the – by the class priors weighted – mean of the Σ_i.
One could also estimate the expected classification accuracy of this linear classifier as stated in the following theorem:

5.2.3 Theorem: *Given the situation of corollary 5.2.2 the optimal classifier performance* $E(\bar{f}(X)=Y)$ *is given by*

$$E(\bar{f}(X)=Y) \geq 1 - \sum_{y_1=1,\dots,N} P(Y=y_1) \sum_{y_2 \neq y_1} erf\left(\frac{-0.5(\mu_{y_2}-\mu_{y_1})\Sigma^{-1}(\mu_{y_2}-\mu_{y_1}) + \log(P(Y=y_2)) - \log(P(Y=y_1))}{\sqrt{(\mu_{y_2}-\mu_{y_1})\Sigma^{-1}(\mu_{y_2}-\mu_{y_1})}} \right).$$

Especially for Laplacian distributed Y *this simplifies to*

$$E(\bar{f}(X)=Y) \geq 1 - \frac{1}{N} \sum_{y_1,y_2=1,\dots,N, y_1 \neq y_2} erf\left(-0.5\sqrt{(\mu_{y_2}-\mu_{y_1})\Sigma^{-1}(\mu_{y_2}-\mu_{y_1})} \right)$$

and for $N=2$ *and* $\mu_2 = -\mu_1$ *to*

$$E(\bar{f}(X)=Y) = erf\left(\sqrt{\mu_1 \Sigma^{-1} \mu_1} \right).$$

Here erf denotes the function $erf : \mathbb{R} \to [0,1],\ z \mapsto \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp(-0.5x^2)dx$.

Proof: see A.3.

5.2.3 Regularized (Linear) Discriminant Analysis

In LDA and QDA one has to estimate the mean and the covariance of the data. Especially for high-dimensional data with less trials this estimation is very imprecise. Thus overfitting and loss of generalization appears. To improve the performance Friedman [54] suggests introducing two parameters λ and γ into QDA. Both parameters modify the covariance matrices due to the fact that the risk of overfitting for the covariance matrix is higher than for the means.
The first parameter λ tries to robustify the estimation of the covariances for each class by taking the covariances for the other classes into account. If Σ_y denotes the estimated covariance for class $y = 1,\dots,N$ resp. $y = \pm 1$ the overall covariance Σ can be defined by $\Sigma = \frac{1}{N}\sum_{y=1}^{N} \Sigma_y$ resp. $\Sigma = 0.5(\Sigma_1 + \Sigma_{-1})$. Then λ moves Σ_y to Σ in the following way:

$$\hat{\Sigma}_y = (1-\lambda)\Sigma_y + \lambda\Sigma$$

with $\lambda \in [0,1]$. Obviously with $\lambda = 0$ normal QDA and with $\lambda = 1$ LDA is achieved.
The second parameter $\gamma \in [0,1]$ works on the single covariances $\hat{\Sigma}_y$. First of all one should note that it is more probable for Gaussian distributions to overestimate the directions coming from eigenvectors with high eigenvalues of Σ_y. Thus one introduces the parameter γ which decreases the higher eigenvalues and increases the lower eigenvalues of the estimated covariance matrix until with $\gamma = 1$ a sphere remains. One derives this *shrunken* covariance matrix by

$$\bar{\Sigma}_y = (1-\gamma)\hat{\Sigma}_y + \frac{\gamma}{m} \text{trace}\left(\hat{\Sigma}_y\right) I$$

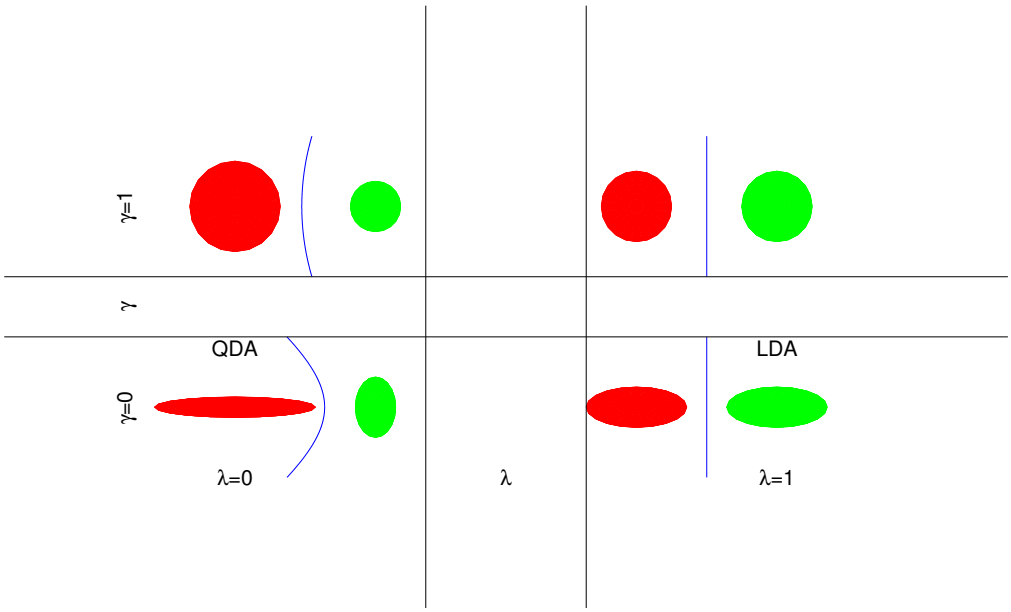


Figure 5.5: Starting with two estimated covariances and parameters $\lambda = \gamma = 0$ (the QDA situation) shown in the lower left plot one is able to modify this estimation by two parameters. With increasing λ the matrices are made more similar until with $\lambda = 1$ the same covariances are achieved (LDA) (lower right). The second parameter γ shrinks each individual covariance matrix until with $\gamma = 1$ a sphere remains (upper left). In the extreme case $\lambda = \gamma = 1$ two equal spheres are achieved (upper right). If $\lambda = 1$ (right column) this algorithm is called RLDA, since a linear classifier remains. In all cases the resulting classification hyperplane is visualized in blue.

with m as the dimensionality of the data. If $\hat{\Sigma}_y = V^\top DV$ is the spectral decomposition of $\hat{\Sigma}_y$ one gets

$$\bar{\Sigma}_y = (1-\gamma)V^\top DV + \frac{\gamma}{m}\text{trace}\left(\hat{\Sigma}_y\right)V^\top V = V^\top[(1-\gamma)D + \frac{\gamma}{m}\text{trace}\,(D)\,I]V.$$

Thus $\bar{\Sigma}_y$ has the same eigenvectors with modified eigenvalues in the required form. This approach of introducing parameters to avoid overfitting is called regularization.
QDA is applied with $\bar{\Sigma}_y$ instead of Σ_y. This modification is called Regularized Discriminant Analysis. In the special case where $\lambda = 1$ one calls this method Regularized Linear Discriminant Analysis. Fig. 5.5 shows the influence of the parameters λ and γ for a binary problem.

5.2.4 Least Square Regression

Although multiclass extensions exist for the following classifiers, I will only introduce the binary algorithms here.
Suppose an unknown function f projects elements of $\mathbb{R}^m$ to $\mathbb{R}$ (possibly with some noise). The idea of regression is to find a function g based on some given examples x_i and $f(x_i)$ that

optimally matches the unknown function f. Usually g is chosen based on some function class, e.g., all linear functions. One can use this approach for classification, too. Here the function f describes the mapping from the data to their class label. In Least Square Regression (cf. [50]) one tries to minimize the squared error made between the realization and the estimation by the function g. If a linear function class is assumed one consequently minimizes $g(w) = \sum_i (w^\top x_i + b - y_i)^2$ (or simplified $g(w) = \sum_i (w^\top x_i - y_i)^2$ by adding ones to x_i ($[x_i, 1]^\top$) and the b to w ($[w, b]^\top$)). If one defines $x = [x_1, ..., x_n]$ and $y = [y_1, ..., y_n]^\top$ this can be written as $\min g(w) = \min \|x^\top w - y\|_2^2$. Taking the derivative with respect to w and setting it equal to zero one gets $xx^\top w = xy$ and if $xx^\top$ is invertible $w = (xx^\top)^{-1}xy$. If it is not invertible one can introduce a small value ε and use $xx^\top + \varepsilon$ instead of $xx^\top$. Finally one can introduce regularization, too. To do so g is exchanged by $g(w) = w^\top w + C\|x^\top - y\|_2^2$ with some $C > 0$ where the unregularized solution is achieved if $C \to \infty$.

One can prove that the w calculated by this approach is equal to the w calculated by LDA, only the bias b can differ. Furthermore the regularization works similarly except that the range and the scale are different.

5.2.5 Fisher Discriminant Analysis

For some arbitrary w I define $\mu_y = \frac{1}{\#\{i|y_i=y\}} \sum_{i|y_i=y} x_i$, $\tilde{\mu}_y(w) = w^\top \mu_y$ and $\tilde{s}_y^2(w) = \sum_{i|y_i=y} (w^\top x_i - \tilde{\mu}_y)^2$. Note that one can easily add a bias term like in LSR, too. The idea of the Fisher Discriminant Analysis (cf. [50]) is to maximize the difference between the projected class means whereas the projected variance is minimized. In other words one looks for the maximum of

$$g(w) := \frac{(\tilde{\mu}_1(w) - \tilde{\mu}_{-1}(w))^2}{\tilde{s}_1^2(w) + \tilde{s}_{-1}^2(w)}.$$

Intuitively this makes sense for classification. One can calculate that $(\tilde{\mu}_1(w) - \tilde{\mu}_{-1}(w))^2 = w^\top S_B w$ with $S_B = (\mu_1 - \mu_{-1})(\mu_1 - \mu_{-1})^\top$ and $\tilde{s}_y^2(w) = w^\top S_y w$ with $S_y = \sum_{i|y_i=y} (x_i - \mu_y)(x_i - \mu_y)^\top$ and thus $\tilde{s}_1^2(w) + \tilde{s}_{-1}^2(w) = w^\top S_W w$ with $S_W = S_1 + S_{-1}$. S_W is called the within-class scatter matrix and S_B the between-class scatter matrix. Consequently $g(w) = \frac{w^\top S_B w}{w^\top S_W w}$. This quotient is the well-known Rayleigh quotient. One can determine the maximum of g by calculating the generalized eigenvalues λ_i and eigenvectors w_i between S_B and S_W (i.e., $S_B w_i = \lambda_i S_W w_i$) and choosing the highest one $\lambda_{\max}$ with corresponding eigenvector w (i.e., $S_B w = \lambda_{\max} S_W w$). An easier analytical solution can be obtained if S_W is invertible. Since $S_B w = c(\mu_1 - \mu_{-1})$ with some real-valued constant c (S_B has rank one) one gets $cS_W^{-1}(\mu_1 - \mu_{-1}) = \lambda_{\max} w$. Since the value of $g(w)$ does not depend on the scaling of w one can fix $w = S_W^{-1}(\mu_1 - \mu_{-1})$ as a solution. Finally one should note that the Fisher Discriminant can be regularized, too. Here one would exchange S_W by $S_W + CI$ with some constant $C \geq 0$. Unregularized Fisher Discriminant is then a special case of regularized Fisher Discriminant for $C = 0$.

One can prove that the w calculated by this approach is the same as calculated by LDA, only the bias b can differ. Furthermore the regularization works similarly except that the range and the scale are different.

5.2.6 Support Vector Machine

Suppose the given data can be separated by a hyperplane perfectly, i.e., a projection w and a bias b can be found such that $y_i(w^\top x_i + b) > 0$ for all i. Without loss of generality one can modify w and b such that $\min_{i|y_i=y} y(w^\top x_i + b) = 1$ for $y = \pm 1$. In this case one says the classifier is in canonical form. With these values the distance from the discriminating hyperplane to the closest point (which is called the margin) can be determined to $\frac{1}{\|w\|_2}$. For different hyperplanes in canonical form, those with smaller w and thus with higher margin should be preferred. Consequently, this can be formulated mathematically in the following optimization problem:

$$\min \frac{1}{2}\|w\|_2^2 \qquad \text{s.t. } y_i(w^\top x_i + b) \geq 1 \quad \text{for all } i.$$

Unfortunately, perfect separation is usually not possible. Thus one modifies this approach and allows errors by modifying the constraint to $y_i(w^\top x_i + b) \geq 1 - \xi_i$ for all i with $\xi_i \geq 0$ (*soft margin*) and additionally punishes the error made in the objective by adding $\frac{C}{n}\sum_i \xi_i$ with some constant $C > 0$. This machine is called $C-SVM$. By analyzing the dual problem one finds that w can be determined by $w = \sum_i \alpha_i y_i x_i$ with some real numbers α_i. For data points x_i with $y_i(w^\top x_i + b) > 1$ one additionally gets that $\alpha_i = 0$. Thus only a few trials (called support vectors) are required for calculating w. But note that usually all points are required to get this set of support vectors.
A slightly different formulation of the $C-SVM$ is given by the ν-SVM:

$$\min_{w,\rho,b,\xi} \frac{1}{2}\|w\|_2^2 - \nu\rho + \sum_i \xi_i \qquad \text{s.t. } y_i(w^\top x_i + b) \geq \rho - \xi_i,\ \xi_i \geq 0 \quad \text{for all } i, \rho > 0.$$

with some $0 \leq \nu < 1$. One can prove that the solution to the ν-SVM is equal to the solution of the $C-SVM$ with $C = \frac{1}{\rho}$.
The advantage of the $\nu-$SVM consists of the fact that the parameter ν informs about the number of support vectors, namely that the fraction of margin errors (data points with $\xi_i > 0$) is smaller than μ which again is smaller than the fraction of Support vectors.
A more detailed overview about Support Vector Machines can be found in Schölkopf et al. [124] and Müller et al. [96].

5.2.7 Linear Programming Machine

In SVM the trained hyperplane usually has only non-zero entries. If one is interested getting a sparse solution for w, i.e., with many entries equal to zeros, one can do so by a slight modification of the SVM approach in the following way:

$$\min \frac{1}{m}\|w\|_1 + \frac{C}{n}\sum_i \xi_i \qquad \text{s.t. } y_i(w^\top x_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0 \quad \text{for all } i.$$

Here the 1-Norm for w is used instead of the 2-Norm. One can prove that with higher C the number of zero entries in w increases. The sparsity of the hyperplane can be used, for example, for feature extraction, i.e., for excluding non-relevant features.
Note that analogously to the $\nu-$SVM, a ν-LPM can be formulated. More information about Linear Programming Machines can be found in Bennett and Mangasarian [9] and Campbell and Bennett [27].

5.2.8 k-nearest neighbor

Instead of calculating a hyperplane which discriminates the data optimally in some sense, the k-nearest neighbor algorithm determines for an unseen data point x the k-nearest neighbors in the training set based on some distance measure (usually the euclidean distance). The algorithm places this x into the class that most of its k neighbors belong to. In the case that both classes appear equally often one could add a measure based on the distances to the k neighbors.

5.2.9 The kernel trick

Obviously the space of linear functions is very limited and cannot solve all existing classification problems. Thus one interesting idea is to map all trials by a function ϕ from the data space to some (maybe infinite dimensional) feature space and apply a linear method there (see [96]). Although this sounds very complex one finds that for some classification algorithms like SVM (cf. [96]), LPM (cf. [27]), Fisher Discriminant (cf. [88]) etc. only the scalar product in feature space is required to get a classifier and to be able to apply it. This scalar product in feature space is called the kernel function $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$, $(x,y) \mapsto < \phi(x), \phi(y) >$. Lots of kernels exist like the RBF kernel ($K(x,y) = \exp(-\frac{\|x-y\|_2^2}{2\sigma^2})$) or the polynomial kernel ($K(x,y) = (< x,y > +c)^k$ with some further parameters. Furthermore, there are theorems about the existence for a feature mapping if a function $\mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is given (see [96]). However, with this kernel trick more complex (nonlinear) structures can be learned.
Note that the kernelization trick can also be applied successfully to feature extraction methods like PCA ((cf. [96]) called KPCA) and ICA (cf. [60]).

5.2.10 Multiple Kernel Learning

Recently an algorithm was suggested (see [77, 78, 3, 4, 5]), which combines F different features by weighted concatenation of kernels. In the linear case this can be formulated by the following optimization problem with regularization constant C

$$\begin{aligned} \min_{w,d} \quad & 0.5(\sum_{j=1}^{F} d_j \|w_j\|_2)^2 + C\sum_{i}^{n} \xi_i \\ s.t. \quad & w \in \mathbb{R}^{n_1} \times ...\mathbb{R}^{n_F}, \xi \in \mathbb{R}^n, \xi_i \geq 0, b \in \mathbb{R}, \sum_{j=1}^{F} d_j = 1, d \in \mathbb{R}^F, d_j \geq 0 \\ & y_i(\sum_j w_j^\top x_{j,i} + b) \geq 1 - \xi_i, \forall i \in \{1,...,n\}. \end{aligned}$$

Compared to the usual SVM approach, an additional weighting on the block structure of the features is used here. In the formulation the $\mathscr{L}_1$-norm of these weightings should be minimized, i.e., the solution should be sparse (depending on C). Consequently, the MKL approach finds a suitable weighting of important features, but tries to ignore features which are not useful for classification, since it tries to find sparse block weightings.

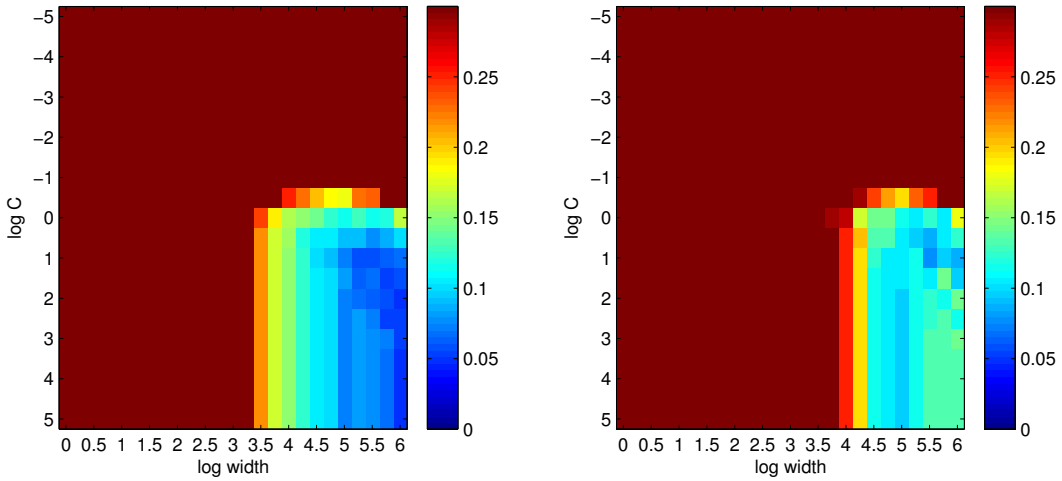


Figure 5.6: The leave-one-out cross validation error on the first 80 % of one selfpaced dataset is visualized in the left figure for different parameters of C and σ^2 of a SVM with RBF-kernel. For red values classification fails, for blue values the classification works well. The best values should be chosen in $C \in [10^2, 10^5]$ and $\sigma^2 = [10^5, 10^6]$. With each parameter combination the classifier on this 80 % percent of the data is determined and applied to the last 20 % to get the test error. This is visualized in the right figure with the same color coding. For the best chosen parameter combination on the left (which one would choose in model selection) a good test error is achieved, too.

By usual kernelization methods this approach can be kernelized. Here the weighting corresponds to the weighted sum of the kernels instead of using one kernel. Obviously the constraint on the norm of the weighting ($\sum d_j = 1$) remains.
By calculations (e.g., Lagrangian techniques) a semi-infinite learning problem remains which can be solved. For the simulations in this work an existing implementation of this algorithm by Sören Sonnenburg and Gunnar Rätsch is used to obtain the multiple kernel learning classifier.

5.2.11 Linear vs. non-linear classification

Since linear classifiers are usually a special case of non-linear machines, they cannot outperform non-linear methods. For example, if one uses an RBF-Kernel $K(x,y) = \exp(-\frac{\|x-y\|_2^2}{2\sigma^2})$ with some width σ^2, the resulting classifier becomes more linear as σ^2 tends to ∞. Thus if one performs suitable model selection for an SVM with RBF kernel over both parameters C and σ^2, then the linear case is included too. For the datasets described in this work I choose one selfpaced dataset and try several values for both parameters C and σ^2 and calculate the performance for these values by a leave-one-out cross-validation on the first 80 % of the dataset. The result is visualized in Fig. 5.6 on the left. The optimal classifier on this first 80 % was determined for each parameter combination and the test error on the last 20 % was calculated with this classifier to see that it generalizes, too. The result is shown in Fig. 5.6 on the right. One clearly observes in both figures that the best performances are achieved

if C and σ^2 are high and furthermore that the figures have the same structure such that one can conclude that generalization works too. Consequently, a linear SVM performs as well as a non linear one and the simplification of only using a linear classifier for this dataset is reasonable to save time by fitting parameters. Of course, one should test several other kernel functions to confirm this but the result with RBF kernels gives a first hint that linear classification is a sufficient choice. However, other kernels were tested too. Nevertheless, the performance of the linear machine was competitive in almost all datasets presented in this work so far.
There are many reason why one should prefer linear classification algorithms if they perform similar to the non-linear ones:

- A linear classifier allows an interpretation of the used weighting in terms of discriminability of importance for classification. For example one could estimate which channels or which timepoints are important. On the one hand one gets an insight into the specific brain of the subject and is able to learn from it from a neurophysiological perspective. On the other hand it is possible to check if the classifier makes sense neurophysiologically.

- Especially if sparse classifiers are used, they can be used to extract suitable features and to reduce the dimensionality. This can be useful both for machine learning and for the neurophysiological interpretation.

- If the assumption of Gaussian distributed data with known means and covariance matrices is reasonable QDA/LDA are optimal. Consequently, one can not do better.

- Linear classifiers are usually easier and faster to learn and less parameters have to be estimated.

- With the complexity of the classifier the possibility of overfitting the data increases, i.e., fitting a classifier to the training set with poor performance on new unseen data. Thus more careful model selection has to be done to get a classifier which generalizes very well.

- Under the assumption of Gaussianity one can easily adapt the classifier to situations where one wants to fix the error rate for only one class. For example, if one wants to detect a P3, one could be interested in restricting the probability of false detections. In Blankertz et al. [18, 19] this modification of LDA was reported. A direct formulation for other possibly non-linear machines is not obvious.

- One big problem in EEG recording is the non-stationarity of the data, i.e., that the data changes over time. Several reasons for the non-stationarity exist, e.g., variations in the quality of electrode and modulations of brain activity due to fatigue, concentration or other activity or non-activity. The influence of the non-stationarity can usually be easier interpreted for linear classifiers. Furthermore with the complexity of the classifier and the resulting overfitting the risk increases of falling for non-stationarity effects. Finally, online adaptation of the classifier, which is usually a good tool to work with non-stationarity data, can usually be handled more easily for linear classifiers.

Consequently, the advantages of linear classifiers compared to non-linear ones is high. Nevertheless, there is one big reason why one should not forget non-linear classifiers: They

achieve a higher performance than linear classifiers, if the data is non-linear. It is for this reason why non-linear classifier can be successfully applied in several environments like bioinformatics (cf. [77, 26, 147]), hand written recognition (cf. [24, 80]) and face recognition (cf. [107, 143]) where linear classifiers usually fail.
For EEG data of the type I use in this work, the performance of linear classifiers are competitive to the non-linear ones. Thus I have decided to choose only linear classifiers here. Since LDA, Fisher Discriminant and LSR are identical except in estimating the bias, LDA usually does not perform worse than SVM and LPM on these datasets and LDA has a direct multi-class formulation, I have decided to choose LDA only, unless stated otherwise. Here I use RLDA if the dimensionality of the data compared to the number of trials is high, otherwise not. Usually CSP filtered data are low-dimensional thus I choose only LDA for this feature.
A more detailed discussion about the choice of the classifier can be found in Müller et al. [97].

5.3 Validation and Model Selection

Generally, the performance of the classifier should be estimated on unseen data. However, it is only trained on given data for example by minimizing the training error, but this training error is in general meaningless. Furthermore, one generally cannot prove that the training error of the optimal function on the training set converges against the generalization error if the amount of training data increases. This depends mainly on the set of allowed classification functions (see [132]). Depending on the function class there exist bounds about the difference between the generalization error and the training error. Here the famous Vapnik-Chervonenkis-dimension (VC-dimension) plays an important role. For example, one can prove that the difference between the generalization and the training error is with probability $1-\delta$ smaller than $\sqrt{\frac{h\log(\frac{2en}{h})+\log(\frac{4}{\delta})}{8N}}$ with h as VC-Dimension and n number of training examples (see [132]). The VC-dimension describes the complexity of the allowed function class of the classifiers: With smaller complexity the VC-Dimension is smaller. For example, for linear classifiers the VC-dimensions is $m+1$, if m is the dimension of the feature space, for the SVM it depends on the margin ($\frac{2}{\|w\|_2}$) if the classifier is in canonical form (see 5.2.6). Unfortunately, to get meaningful bounds a small VC-dimension and a high number of trials is usually required. Regrettably this is not fulfilled in the BCI situation where high dimensional data and a small number of training examples are given. Thus these bounds are usually not meaningful in the BCI context. For example if one assumes $n = 500$, $\delta = 0.01$ and $m = 100$ one gets $h = 101$ and that the absolute difference between test and generalization error is smaller than 30 percentage points with probability 99 %.
To estimate the generalization performance one has to apply a different technique, called validation. The simplest way is the so called leave-one-out validation. Here a classifier is trained on all data except one data point and is evaluated for this excluded point. This is repeated for all data points. In this case the mean error is a good approximation for the generalization error. Alternatively one could perform an $n \times k$-fold cross validation. Here the data is split randomly into k disjoint subsets of nearly equal size. Now a classifier is trained on $k-1$ subsets and is applied to the excluded subset. This is repeated for all k subsets for n different splittings such that one gets $n \times k$ errors. The mean of them is also a

good approximation for the generalization performance. Due to the fact that slight variations depending on the random splitting in this performance exist if this approach is repeated a fix splitting for all algorithms and datasets is used in every case.
EEG data are usually highly non-stationary. Thus the performance of a classifier usually varies over time. If the performance of a classifier is validated on the first half of the data by one of the validation techniques above it could be that this result does not match the test error on the second half of the data with the classifier trained on the first half. This is a big problem of all validation strategies, namely that they assume that all future data are derived from the same distribution which is not the case for EEG data. Nevertheless, if one is interested in adapting the validation in such a way that classifiers are preferred which take this non-stationarity problem into account one could train the classifier on the first half and calculate the performance on the second half of the data. Consequently a classifier which focuses on features which are more stationary and thus do not change from the first half of the data to the second half are preferred. This approach I will call chronological validation. Nevertheless, with this one splitting the resulting performance is a very imprecise estimation of the generalization performance. However, I will use all introduced validation schemes in this work since a bias for one technique is not obvious.
Note that all preprocessings which depend on the class labels like CSP have to be performed on the training set only.
Usually one has to estimate further model parameters, like the λ in RLDA. This can be done either on each training set by another cross-validation or by introducing three disjoint sets: a training, validation and test set. In the latter case a classifier is trained for the parameters on the training set, and validated on the validation set. Then the classifier is trained for the best parameter on training and validation set and is applied to the test set. During the first approach (fitting on each training set within a cross-validation by another cross-validation) is very time consuming, the second approach with three disjoint sets often has the problem that only few examples for a test set remain. Consequently, the resulting test error mainly depends on the choice of the set and not of the general distribution of all datapoints (with more elements in a set the general distribution is more represented). Thus an estimation of the generalization error based on a small test set is highly arguable. Rätsch et al. [117] have suggested a third solution, namely by fitting the parameter with one global cross-validation with one – by the usual amount of used test data in the final validation – reduced dataset beforehand and then calculate with this parameter the test error by a usual cross-validation. In several studies in Rätsch et al. [117] it was found that this approach usually performs well in the sense of estimating the generalization error. However, the idea to fit the parameters globally (without excluding some data) suffers from the fact that overfitting to the optimal constant for the data but not for the problem could take place. In this thesis I will use the first method, fitting parameter on each training set, if this is possible in a conceivable time. Otherwise I will use the approach suggested in Rätsch et al. [117].

5.4 Robustification

Usually EEG data is contaminated by artifacts. In machine learning this corresponds to outliers in the data, i.e., points which have nothing to do with the underlying distribution. One should try to exclude these points in the data to get a better classifier (see [72]). In a

controlled training scenario, which is used for the data in this work, strong outliers usually do not exist. It is for this reason why tests with several outlier removal techniques like Harmeling et al. [63]) have not enhanced the performance in most datasets. Nevertheless it is important to exclude them if they exist. Thus performance in a few datasets can be greatly enhanced if strong outliers exist. However, if one leaves the controlled scenario and does experiments in a more natural environment, where the tendency to artifacts and thus outliers increases drastically, this becomes a very important issue. Therefore one should not forget this important machine learning issue in the context of real-word BCI. In this work I only focus on experiments in the laboratory where strong outliers usually do not exist. Thus robustification approaches are not used for this work.

6 Feature Combination

6.1 Motivation

If different features like ERD and LRP effects are available, suitable combination of them achieves at least the same performance as the best performing single feature, namely by ignoring all other features and only trusting the best one. The goal of combination is the use of additional information from several features and therefore to enhance performance. Roughly speaking, if in one trial one feature does not appear, another feature could contain the required information which can help to increase the performance. Furthermore if one feature of one trial is contaminated or overlaid by some artifact another feature could remain stable and can be used for the final decision. However, it is not clear what should happen if classifiers based on two different features decide on different classes. If more features are available one could do a voting or if something like a confidence for each feature exists one could compare these confidences. For example if one classifier decides on one class with a high confidence, while another classifier on a different feature decides on the other class but with low confidence, one should use the result of the first classifier. Unfortunately with the number of used features the dimensionality of the problem increases. Thus one should avoid the curse of dimensionality.
In section 6.2 I briefly present different EEG-features which are believed to be uncorrelated to each other during different mental states, based on neurophysiological knowledge. One finds in section 6.3 that theoretically the highest gain can be achieved if the underlying features are independent. Based on these ideas combination algorithms will be suggested in section 6.4 and their success will be shown and discussed in section 6.5.

6.2 Neurophysiological Background

In the EEG there are several well-known features for discriminating different tasks like imagination of movements, real movements and sensations as discussed in chapter 2. Toro et al. [130] demonstrate by invasive (subdural) EEG recordings during brisk self-paced finger and foot movements that ERDs in $\mu-$ and $\beta-$ rhythm and MRPs have different characteristics: MRPs start over widely distributed areas of the sensorimotor cortices (*Bereitschaftspotential*) and focuses at the contralateral primary (sensori-) motor cortex (M-1) hand cortex with a steep negative slope prior to finger movement onset, culminating in a negative peak approximately 100 ms after EMG onset. In contrast, a bilateral M-1 ERD preceding the movement appears to reflect a more widespread cortical 'alerting' function. Most importantly, the ERD response magnitude does not correlate with the amplitude of the negative MRPs slope. This study was backed by Babiloni et al. [2], namely that ERDs in $\mu-$ and $\beta-$ rhythm and MRPs have different spatio-temporal activation patterns across primary (sensori-) motor cortex, supplementary motor area (SMA) and the posterior parietal

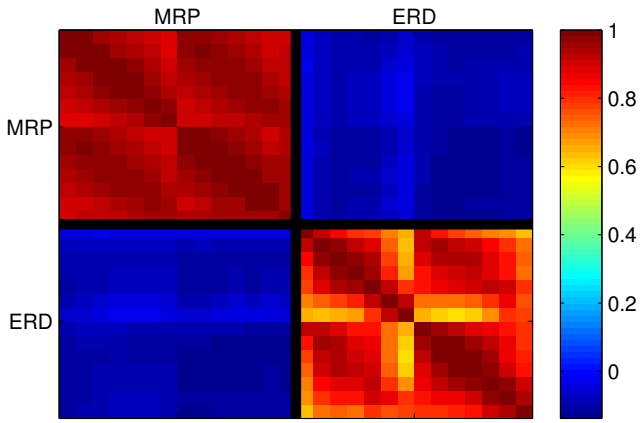


Figure 6.1: The figure shows the correlation between MRP and ERD features calculated for one subject in one experiment. Note that each feature consists of several dimensions. The black lines mark the border between both features.

cortex (PP). Note that both studies are based on experiments with real movements. However, one can assume a similar existence of MRP (cf. [6]) and ERD phenomena (cf. [140]) in imagined movements too. Encouraged by the results in section 6.5 one is able to presume the independence of these features. Note that the notation MRP is used for imagined movements in the same manner.
Of course the results in section 6.5 show that combination of both features increases performance compared to the best single feature performance if they both have performance in a similar range. This is only possible if these features are not correlated (compare section 6.3). Nevertheless, to illuminate the assumption of independence in the experiments the correlation matrix of the used features is calculated (visualized for one subject in Fig. 6.1). It shows the expected block structure which is evidence for independent data. Furthermore there is the opportunity to consider the set of *good* and *bad* trials for each feature for one subject, i.e., the trials which can be classified correctly or not. These sets can be determined for each feature by the results of a leave-one-out cross-validation, i.e., each trial is in the test set once and if it is classified correctly it is a good trial, otherwise not. If one feature A is independent of some other feature B, feature A can not classify the good trials for feature B better or worse than the bad trials for feature B. In other words the distributions of good trials for one feature in the good or bad trials of the other feature should be similar. If they are not independent this distribution could vary. Although it is not visualized here, tests in this direction were done and they back the assumption of independence considerably.
There is a second reason besides independence why combination of these features could be useful: MRP and ERD are distorted by artifacts outside the central nervous system (CNS), namely by eye (EOG) and muscular (EMG) movements in the skin. While MRP is contaminated by the EOG, EMG is detrimental for ERD phenomena (see [141]). Consequently a suitably constructed classifier based on a combination of both features can handle trials which are contaminated by exactly one of these artifact types. In this case the recognition of one of these artifacts should lead to the decision ignoring the corresponding feature and only concentrate on the other one.
Finally it should be remarked that this work focuses on these two features only. Extensions

to other features can be worthwhile if they show new, i.e., uncorrelated information. For example the phase information (see [85]) could be an interesting add-on to this approach. Furthermore, each of the neurophysiological characteristics (MRP and ERD) consists of several features (e.g., $\mu-$ and $\beta-$ rhythms for ERD) which could be used separately. Unfortunately in our studies no significant gain could be achieved by this approach since it seems that $\mu-$ and $\beta-$ rhythms are highly correlated.

6.3 Theoretical Background

With theorems (5.2.1), (5.2.2) and (5.2.3) one is able to estimate the gain of combining features under some specific assumptions. Of course some assumptions have to be made. To understand the necessity of assumptions consider the following example: Let X_1, X_2 be one-dimensional random variables with label $Y = \{\pm 1\}$ and $X_i|Y = y$ a Bernoulli-distribution with values in $\{\pm 1\}$ where 1 (or -1) is chosen with probability $p_i > 0.5$ if $Y = 1$ (or $Y = -1$) and -1 (resp. 1) otherwise. Furthermore both classes appear equally often (i.e., $P(Y = 1) = P(Y = -1) = 0.5$). Then the best classifier for each i is given by the value of X_i and the feature can be classified with accuracy p_i. With both features i this cannot be enhanced. In this case the best decision is given by the best performing feature, i.e., by the highest p_i. Consequently a performance gain cannot be achieved compared to the best single feature.
However, there are assumptions about the distributions where the performance can be enhanced, as stated in the following theorem:

6.3.1 Theorem: *Consider random variables $X_1, ..., X_F$ ($X_i \in \mathcal{F}_i$) with $F \geq 2$ and Laplace distributed label $Y \in \{\pm 1\}$. Furthermore suppose that the mappings $f_i : \mathcal{F}_i \to \mathbb{R}$ are normal distributed (i.e., $f_i(X_i)|Y = y \sim \mathcal{N}(\mu_{i,y}, \sigma_i^2)$) with equal variances σ_i^2 for each feature[1] for all i. Let $\mathcal{F} = (\mathcal{F}_1, ...\mathcal{F}_F)$ the concatenated feature space and $f : \mathcal{F} \to \mathbb{R}^F, f = (f_1, ..., f_F)$ the concatenation of the functions f_i. Suppose that $f(X)|Y = y \sim \mathcal{N}(\mu_y, \Sigma)$ which for example is the case if $(X_i)_{i=1,...,F}$ is pairwise independent. With the optimal classifier g on the trials $f(X)$ given by corollary 5.2.2 and the expected classification accuracies acc_g, acc_{f_i} on the combined/single problems based on the functions g and $(f_i)_{i=1,...,F}$ the following holds true:*

$$acc_g \geq \max_i acc_{f_i}.$$

Proof:

After suitable shifting one can assume that $\mu_{i,-1} = -\mu_{i,1}$ for all $i = 1, ..., F$ and $\mu_{i,1} > 0$. For the concatenated f it holds true that $f(X)|Y = y \sim \mathcal{N}(\mu_y, \Sigma)$ with $\mu_y := (\mu_{i,y})_{i=1,...,F}$ and the diagonal of Σ consists of σ_i^2. Consequently $\mu_{-1} = -\mu_1$. By theorem 5.2.3 this directly leads to

$$\text{acc}_{f_i} = \text{erf}\left(\frac{\mu_{i,1}}{\sigma_i}\right) \tag{6.1}$$

[1] sign f_i could for example define the optimal classifier for the i-th feature in the sense of minimizing the misclassification risk. In that case the assumption of Gaussianity is for example fulfilled if $X_i|Y = y \sim \mathcal{N}(\nu_{i,y}, \Sigma_i)$.

for all $i = 1, ..., F$. Theorem 5.2.3 also shows that

$$\text{acc}_g = \text{erf}\left(\sqrt{\mu_1^\top \Sigma^{-1} \mu_1}\right). \tag{6.2}$$

With suitable reordering and due to the fact that erf and $\sqrt{\cdot}$ are monotonely increasing functions the theorem is proved if one can show that

$$\mu_1 \Sigma^{-1} \mu_1 \geq \frac{\mu_{1,1}^2}{\sigma_1^2}.$$

Let $\Sigma = (\sigma_{i,j})_{i,j=1,...,N}$. Let us define the matrix $P = (p_{i,j})$ as follows:

$$p_{i,j} = \begin{cases} 1 & i = j \\ -\frac{\sigma_{1,j}}{\sigma_{1,1}} & i = 1, j \geq 2 \\ 0 & \text{otherwise} \end{cases}.$$

Thus

$$P^\top \Sigma P = \begin{pmatrix} \sigma_{1,1} & 0 \\ 0 & \tilde{\Sigma} \end{pmatrix}$$

with $\sigma_{1,1} = \sigma_1^2$. For an arbitrary $x \in \mathbb{R}^{F-1}$, $x \neq 0$ is holds true that

$$\begin{aligned} x^\top \tilde{\Sigma} x &= \begin{pmatrix} 0 \\ x \end{pmatrix}^\top \begin{pmatrix} \sigma_{1,1} & 0 \\ 0 & \tilde{\Sigma} \end{pmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} 0 \\ x \end{pmatrix}^\top P^\top \Sigma P \begin{pmatrix} 0 \\ x \end{pmatrix} \\ &= \left(P \begin{pmatrix} 0 \\ x \end{pmatrix}\right)^\top \Sigma \left(P \begin{pmatrix} 0 \\ x \end{pmatrix}\right) > 0 \end{aligned}$$

since Σ is positive definite. Consequently $\tilde{\Sigma}$ is positive definite. Furthermore P^{-1} is equal to P except that the elements on the non-diagonal are multiplied by -1. This results with $P^\top \mu_1 = \begin{pmatrix} \mu_{1,1} \\ w \end{pmatrix}$ with some $w \in \mathbb{R}^{F-1}$ in

$$\begin{aligned} \mu_1^\top \Sigma^{-1} \mu_1 &= \mu_1^\top \left((P^\top)^{-1} P^\top \Sigma P P^{-1}\right)^{-1} \mu_1 \\ &= (P^\top \mu_1)^\top (P^\top \Sigma P)^{-1} (P^\top \mu_1) \\ &= \frac{\mu_{1,1}^2}{\sigma_1^2} + w^\top \tilde{\Sigma}^{-1} w \geq \frac{\mu_{1,1}^2}{\sigma_1^2}, \end{aligned}$$

since $\tilde{\Sigma}^{-1}$ is positive definite and the inverse of a block matrix is the block matrix of the inverse blocks. Note that one can not conclude a strict $>$ in the theorem since w could be zero for some extreme cases. □

Two special cases should be considered. First the case that the features are independent results in the following theorem:

6.3.2 Theorem: *In the situation of theorem 6.3.1 with independent $X_1,...,X_F$ acc_g is given by*

$$acc_g = erf\left(\sqrt{\sum_{i=1,...,F}\left(erf^{-1}(acc_{f_i})\right)^2}\right).$$

Especially for $acc_{f_i} = acc_{f_1}$ for all $i = 1,...,F$ this simplifies to

$$acc_g = erf\left(\sqrt{F}erf^{-1}(acc_{f_1})\right).$$

Proof:

Using formulas (6.1) and (6.2) with Σ as the diagonal matrix with diagonal elements σ_i^2 directly gives the desired results. Σ is diagonal because of the assumed independence of the features. □

For the second case let us assume that two possibly dependent features are given. Then the following corollary holds true

6.3.3 Corollary: *In the situation of theorem 6.3.1 with $F = 2$ and correlation coefficient a between $f_1(X_1)$ and $f_2(X_2)$ it is*

$$acc_g = erf\left(\sqrt{\frac{\left(erf^{-1}(acc_{f_1})\right)^2 + \left(erf^{-1}(acc_{f_2})\right)^2 - 2aerf^{-1}(acc_{f_1})erf^{-1}(acc_{f_2})}{1-a^2}}\right).$$

Especially if $acc_{f_1} = acc_{f_2}$ this simplifies to

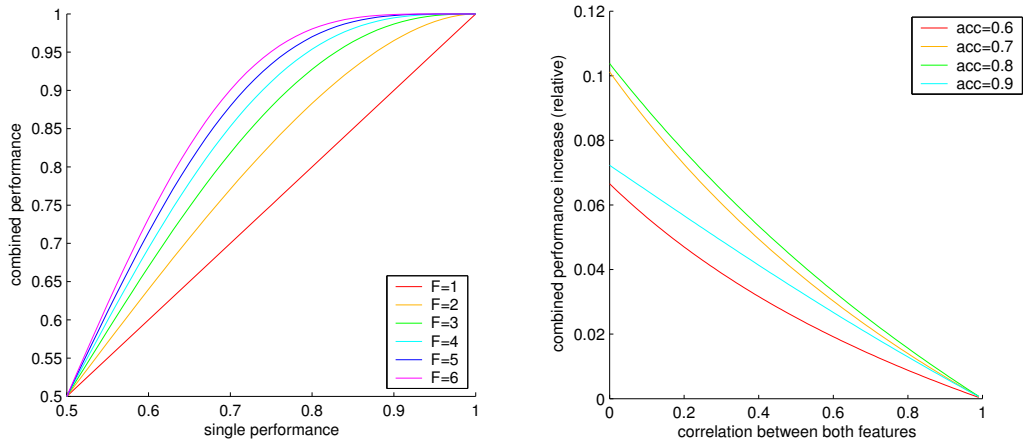
$$acc_g = erf\left(erf^{-1}(acc_{f_1})\sqrt{\frac{2}{1+a}}\right).$$

Proof:

With the notation of the proof of theorem 6.3.1 and $\Sigma = \begin{pmatrix} \sigma_1^2 & a\sigma_1\sigma_2 \\ a\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$ (a is the correlation coefficient) it holds true that

$$\begin{aligned}
\mathrm{acc}_g &= \mathrm{erf}\left(\sqrt{\mu_1^\top \Sigma^{-1}\mu_1}\right) \\
&= \mathrm{erf}\left(\sqrt{\frac{\mu_{1,1}^2\sigma_2^2 + \mu_{2,1}^2\sigma_1^2 - 2a\sigma_1\sigma_2\mu_{1,1}\mu_{2,1}}{\sigma_1^2\sigma_2^2 - a^2\sigma_1^2\sigma_2^2}}\right) \\
&= \mathrm{erf}\left(\sqrt{\frac{\left(\frac{\mu_{1,1}}{\sigma_1}\right)^2 + \left(\frac{\mu_{2,1}}{\sigma_2}\right)^2 - 2a\frac{\mu_{1,1}}{\sigma_1}\frac{\mu_{2,1}}{\sigma_2}}{1-a^2}}\right) \\
&= \mathrm{erf}\left(\sqrt{\frac{\left(\mathrm{erf}^{-1}(\mathrm{acc}_{f_1})\right)^2 + \left(\mathrm{erf}^{-1}(\mathrm{acc}_{f_2})\right)^2 - 2a\mathrm{erf}^{-1}(\mathrm{acc}_{f_1})\mathrm{erf}^{-1}(\mathrm{acc}_{f_2})}{1-a^2}}\right).
\end{aligned}$$

□



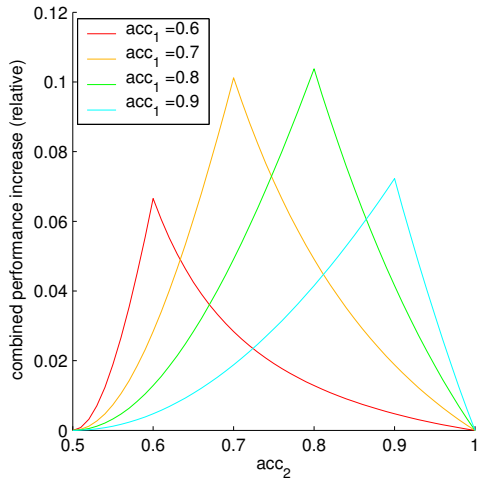


Figure 6.2: The figure on the top left shows the theoretical combined performance for F independent features with same classification performance varied on the x-axis. The figure on the top right shows the relative theoretical increase for different accuracies which are equal for both classes for two features with their correlation coefficient varied on the x-axis. The plot on the bottom left visualizes the influence of the similarity in performance between two independent features. With one performance fixed to 0.6, 0.7, 0.8 or 0.9 the other performance is varied on the x-axis and the relative accuracy increase compared to the best accuracy $max(\text{acc}_1, \text{acc}_2)$ (with $\text{acc}_i := \text{acc}_{f_i}$) is plotted on the y-axis.

To clarify the theoretical performance increase, three simulations are done: In figure 6.2 on the top left the accuracy for two classes and for different number of independent features with same classification performance acc the appropriate combined performance based on theorem 6.3.2 is shown. On the top right of figure 6.2 the dependence of the correlation coefficient between two features for two classes with variances equal to 1 as described by corollary 6.3.3 is visualized. Both show that there is an increase, but with a higher correlation coefficient this gain gets lost. Finally on the bottom left of figure 6.2 two independent features but with different accuracies are chosen. Here one accuracy is chosen fixed to 0.6,

0.7, 0.8 or 0.9 and one is varied. The relative performance increase to the better single feature is shown on the y-axis. One observes that the best performance increase can be achieved if the single performances are equal.

6.4 Algorithms

Combination of features are rather common in different fields, e.g., in speech recognition (e.g., [94]), vision (e.g., [25]) or robotics (e.g., [129]). Usually the approaches consist of concatenation of the single feature vectors (discussed as CONCAT below) or in a winner-takes-all strategy, which however cannot increase performance above the best single feature vector analysis. Nevertheless, the last strategy can be best in some context (see the example in the beginning of this chapter). Recently a new combination method was suggested (cf. [77, 78, 3, 4, 5]). This approach is based on the idea to concatenate the feature vectors with a weighting which is one further issue to learn. This method is called Multiple Kernel Learning (MKL, see section 5.2.10). In Dornhege et al. [43, 44] I have suggested two further methods. The first, which is called PROB, incorporates the independence assumption in the algorithm completely. The second algorithm which is called META allows individual fitting of a decision boundary to the single feature classifier results.
For all algorithms F given features are considered described by n training examples $x_{i,j}$ with labels $y_j \in \{1,...,N\}$ (N is the number of classes) for $i = 1,...,F$, $j = 1,...,n$. Furthermore let us assume that all classes appear equally often, i.e., the class priors are identical.
(CONCAT). This common approach consists of concatenation of the feature vectors and classification in the higher-dimensional space. In other words one defines $x_j := (x_{i,j})_{i=1,...,F}$ and classifies on the problem (x_j, y_j), $j = 1,...,n$. Note that in this case careful regularization has to be done (see [96, 97]).
(MKL). I will use MKL (see section 5.2.10) with linear kernels on the single features to find an optimal weighting and optimal classifiers for each feature simultaneously.
(PROB). Let us assume that the observed trials $x_{i,j}$, $j = 1,...,n$ derives from random vectors X_i based on some feature space $\mathscr{F}_i$ for all $i = 1,...,F$. Again $x_j := (x_{i,j})_{i=1,...,F}$ denotes the combined vector. Furthermore let us assume that functions $f_i : \mathscr{F}_i \to \mathbb{R}^N$ are given for all i such that the function $\operatorname{argmax} f_i$ is the Bayes optimal classifier[2] for each i, i.e., which minimizes the misclassification risk. By $X = (X_1,...,X_F)$ the combined random variable is denoted, by $g_{i,y}$ the densities of $f_i(X_i)|Y = y$, by f the optimal classifier on the combined feature vector space $\mathscr{F} = (\mathscr{F}_1,...,\mathscr{F}_F)$ and by g_y the density of $f(X)|Y = y$.
One gets for all $i = 1,...,F$ and all possible features $z = (z_1,...,z_F)$

$$\begin{aligned} \operatorname{argmax}(f_i(z_i)) &= \operatorname{argmax}_y g_{i,y}(z_i) \\ \operatorname{argmax}(f(z)) &= \operatorname{argmax}_y g_y(z). \end{aligned}$$

Let us assume that the features are independent. This assumption allows us to factorize the combined density, i.e., to compute $g_y(x) = \prod_{j=1}^{F} g_{j,y}(x_j)$ for the class labels $y = \{1,...,N\}$. This leads to the optimal decision function

$$f(z) = \operatorname{argmax} \sum_{i=1}^{F} f_i(z_i).$$

[2] At this point no assumptions about the distribution of the data are made.

If one additionally assumes that all feature vectors X_j's are Gaussian distributed with equal covariance matrices, i.e., $P(X_i|Y=y) = \mathcal{N}(\mu_{i,y}, \Sigma_i)$ the following classifier

$$\text{argmax} f(x) = \text{argmax}_y(\sum_{i=1}^{F}[w_i^\top x_i - \frac{1}{2}(\mu_{i,y})^\top w_i])$$

with $w_i := \Sigma_i^{-1}\mu_{i,y}$ is achieved.
In terms of LDA this corresponds to forcing the elements of the estimated covariance matrix that belong to different feature vectors to zero. Consequently since less parameters have to be estimated distortions by accidental correlations of independent variables are avoided. It should be noted that analogously to quadratic discriminant analysis (QDA) (see [54]) one can formulate a non-linear version of PROB with Gaussian assumption but different covariance matrices for each class.
To avoid overfitting PROB can be regularized, too. There are two possible ways: fitting one parameter to all features, or fitting one parameter for each feature. Extensive simulations on the datasets used in this chapter have shown that the gain by the second step is very low. Therefore only one parameter is fitted to all features simultaneously.
(META). This algorithm is the trade-off between CONCAT which allows absolute correlation between features and PROB which assumes independent features. Here a single classifier is trained on each feature and afterwards a META classifier is trained on the continuous output of all these classifiers. This has the further advantage that one can use different classifiers for each feature, e.g., linear and non-linear version. Here only RLDA is used with parameters fitted to each feature. For the meta classifier that combines the single classifier outputs regularization is not needed anymore in practice, since the meta classifier acts on very low dimensional feature vectors.
However, META extracts discriminative information from single feature vectors independently and may exploit inter-relations (also, for example, hidden dependencies) in the combining step based on the output of the individual decision functions. Therefore independence is assumed on a low level whereas possible high level relations are taken into account.
In the case where LDA is used as a classifier or more generally the logarithm of class densities, the difference between PROB and META consists of the following: PROB simply sums up all individual single classifiers, whereas META additionally learns a weighting between these outputs which is used for decision making. Furthermore, a bias on the classifier outputs can be learned. However, in practice the use of this bias is small and can therefore be neglected.

6.5 Results

To compare the algorithms they were applied to two different datasets. First of all I took the *imag* dataset and calculated the single feature performance with leave-one-out cross validation on all subsets of existing classes with at least two classes for two features: MRP and CSP (for the specific setup of the processing see section 4.1). For multi-class extensions I used the algorithms presented in chapter 7. The first big observation is, that the performance between subjects and features vary strongly. There are some subjects with good or average performance for both features, or for exact one, or for no features.

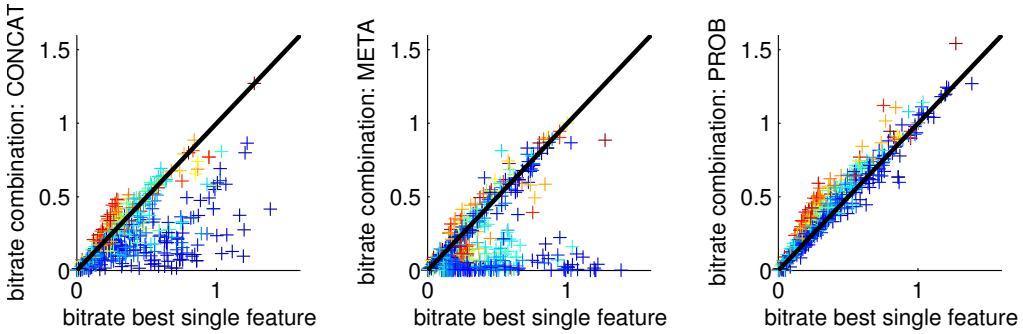


Figure 6.3: These figures show from left to right the best single feature performance on the x-axis against the combination performance on the y-axis from CONCAT, META and PROB on the discussed datasets with the bitrate as the performance measure. Points above the diagonal correspond to datasets where the combination algorithm outperforms the best single feature performance. The datasets were colored by the similarity of the single feature performances: red points correspond to very similar performances, blue to strongly varying.

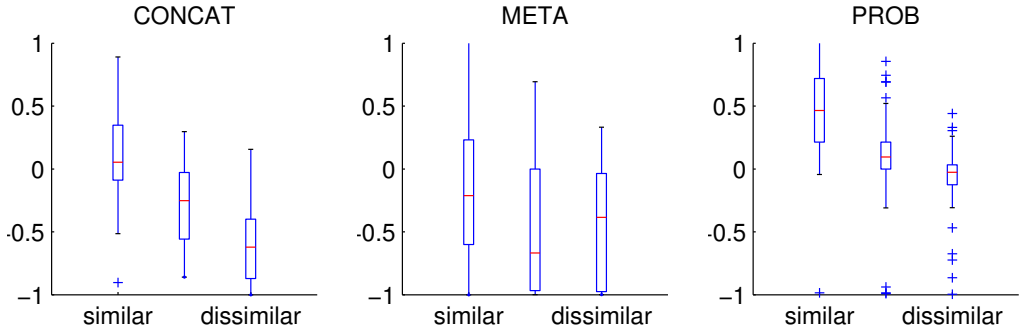


Figure 6.4: These figures show boxplots for the relative performance increase with the bitrate as the performance measure of the combination algorithms against the best single feature performance on the discussed datasets. From left to right the combination algorithms are varied from CONCAT, META to PROB. Within each plot a grouping based on the similarity used in Fig. 6.3 for the color coding is done: On the left the 25 % most similar (red points in Fig. 6.3), on the right the 25 % most dissimilar (blue points in Fig. 6.3) and in the middle the rest are shown. Each boxplot consists of the median, 25 %- and 75 %-percentiles, minimum and maximum value and some outliers.

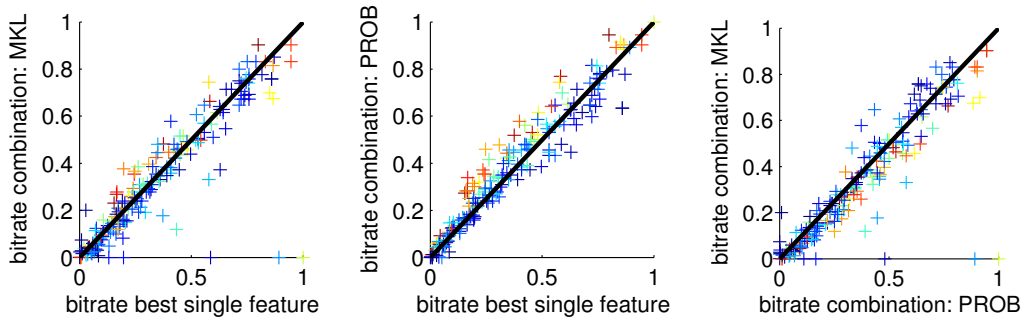


Figure 6.5: The two figures on the left show the best single feature performance on the x-axis against the combination performance on the y-axis from MKL and PROB on the discussed datasets (2-class problems only) with the bitrate as the performance measure. On the right the same is visualized for PROB (on the x-axis) against MKL (on the y-axis). Points above the diagonal correspond to datasets where the combination algorithm outperforms the best single feature performance or in the last plot where MKL outperforms PROB. The datasets were colored by the similarity of the single feature performances: red points correspond to very similar performances, blue to strongly varying.

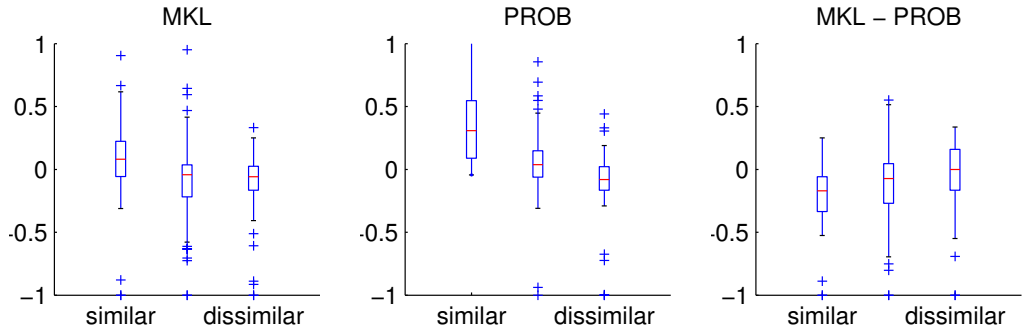


Figure 6.6: These figures show boxplots for the relative performance increase with the bitrate as performance measure of the combination algorithms MKL (left) and PROB (middle) against the best single feature performance or of MKL against PROB (right) on the discussed datasets (2 class problems only). Within each plot a grouping based on the similarity used in Fig. 6.5 for the color coding is done: On the left the 25 % most similar (red points in Fig. 6.5), on the right the 25 % most dissimilar (blue points in Fig. 6.5) and in the middle the rest are shown. Each boxplot consists of the median, 25 %- and 75 %-percentiles, minimum and maximum value and some outliers. On the right the performance increase of MKL against PROB is shown, i.e., for positive values MKL outperforms PROB.

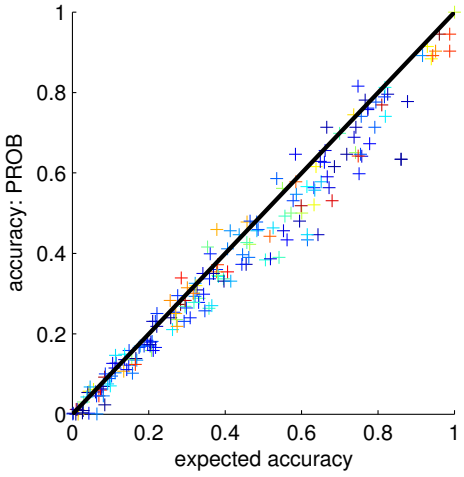


Figure 6.7: The figure shows the expected bitrate based on theorem 6.3.2 on the x-axis compared to the result of PROB for all datasets (2 class problems only). Points above the diagonal corresponds to datasets where PROB outperforms the expected accuracy. The datasets were colored by the similarity of the single feature performances: red points correspond to very similar performances, blue to strongly varying.

Due to the considerations in section 6.3 no gain should be achieved if the performances for at least one feature is bad. The gain is higher the more independent features and more similar feature performances are. In Fig. 6.3 the bitrates for the best performing single feature compared to the bitrate of the used combination algorithms CONCAT, META and PROB are visualized for these datasets by a leave-one-out cross-validation. Consequently, all points above the diagonal belong to datasets where the combination algorithms outperforms the best single feature result. Furthermore the points are colored by their relative difference on their single feature. Datasets with similar performance for both single feature correspond to red points in the plot, datasets with strongly varying performance to blue points. In Fig. 6.4 the same result is shown as boxplot separated based on the similarity of the single feature performances: The dataset was split into three groups, the 25 % with most similar performances, the 25 % with most dissimilar performances and the rest. For each group and combination algorithm the relative performance increase in bitrate of the combination algorithm against the best single feature is used for the boxplot. Each boxplot consists of the median, 25 %- and 75 %-percentiles, minimum and maximum value and some outliers.

Since MKL is very time-consuming and only comes up with a version classifying two classes the results here are presented for all binary subsets of the datasets with leave-one-out cross-validation. The performances are visualized in Fig. 6.5. On the left and in the middle the MKL and PROB solutions respectively are compared to the best single feature performance. On the right MKL is compared to PROB. The colors of the crosses are obtained analogously to Fig. 6.3. The results are also shown as boxplots similar to Fig. 6.4. Furthermore the performance increase of MKL against PROB is shown.

First of all it can be observed in Fig. 6.3 that PROB is better than CONCAT and META above a broad variety of EEG datasets. In Fig. 6.5 a slightly better performance for PROB against MKL is visible too. Especially for points where the single feature performances are similar, PROB outperforms MKL considerably, whereas MKL uses its capability to ignore one feature, if the discrimination of both features is highly different, such that a slightly better performance against PROB can be observed in this case. Fig. 6.4 and 6.6 confirm this observation. Especially the dependency on the similarity can be observed clearly.

Another interesting fact can be observed in the figures: With the similarity between the

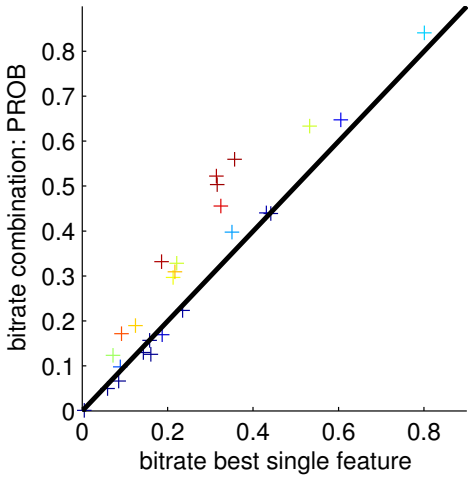


Figure 6.8: The figure compares the best single feature performance on the x-axis against the performance of PROB on the y-axis measured in bitrate on the dataset *self-paced* with leave-one-out cross-validation. Points above the diagonal correspond to datasets where PROB outperforms the best single feature performance. The datasets were colored by the similarity of the single feature performances, red points correspond to very similar performances, blue to strongly varying.

single feature performances the gain of the combination algorithm increases which confirms the ideas of section 6.3. Furthermore the results here back the idea of independent features.

Another interesting question appears based on theorem 6.3.2. Can PROB be enhanced if one assumes independence and normal distributed data by another approach? The comparison between the expected bitrate based on theorem 6.3.2 and PROB is visualized in Fig. 6.7. Since the theorem is only given for two class problems the visualization is restricted to such problems. The figure reveals that for almost all datasets the expected accuracy matches the achieved performance by PROB. Note that there are small variations in the results due to the finite sample size such that PROB can sometimes be slightly better than the expected bitrate. However, the figure confirms that one can not do better than PROB based on the these features if the assumption of Gaussianity is reasonable.

The performance gain of the algorithm PROB can also be observed in another dataset: Here PROB is applied to the *selfpaced* dataset. The results are visualized in Fig. 6.8 on the right as described above for PROB against the best single feature performance and again strengthen the case for combination.

Another interesting aspect of feature combination can be observed if one looks at the reaction time and persistence of the features. Hereby the classification traces over time and their medians and percentiles are calculated. This procedure is done in a leave-one-out validation scheme for generalization purposes. In Fig. 6.9 the class means of the classifier traces and 10−, 20−, 30− percentiles are shown for one subject with similar discrimination in both features during imagination of left vs. right hand movement. The first two figures on the left show these tubes for the MRP and CSP feature. The area when the 20 %- and 80 %-percentiles are separated is shown in black, the area where the 10 %- and 90 %-percentiles are separated is blue. On the right the same is shown for combination by PROB of both features.

The figures show that the MRP has a very fast reaction time whereas the CSP shows a long persistence for the whole time window. One further issue of combination is that both advantages appear, which is confirmed in the right figure. Thus combination of both features

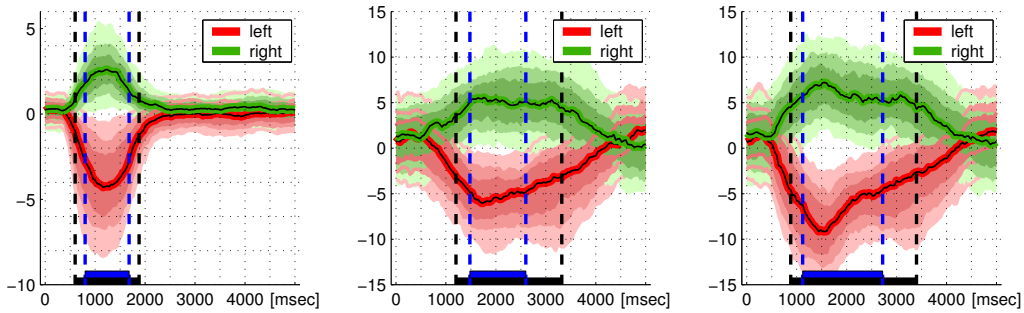


Figure 6.9: The figures show leave-one-out classifier traces (endpoint of classification) for different features. Here the mean and 10-, 20-, 30-percentiles are visualized. On the left MRP, in the middle CSP and on the right the combination PROB is shown. Areas where good discrimination starts are in black and the range where the tubes do not intersect anymore is shown in blue.

does not only increase performance, it also combines the fast reaction of one feature with the long persistence of the other feature. Of course, this is only true if both features can be used for the subject in the sense of good discrimination.
Finally, all these results motivate the following procedure for new datasets: First of all both features should be tested separately to get their individual performances. If their performances are in a similar range, the combination algorithm PROB should be applied and will outperform the best single feature performance considerably. If the performances are strongly different, the use of the best single feature only is advisable. Instead of calculating the individual performances one can also apply techniques like Multiple Kernel Learning to find a sparse representation of suitable features. Here MKL has to be preferred if a lot of features exist to save time and to get also a suitable subset.
To transfer the results of this chapter to online experiments one should first start to use both features individually to get a feeling about their online behavior. The CSP feature was successfully applied in chapter 4. The next goal will be to establish an online BCI based on the movement related potentials and finally to improve the BCI system by the suggested combination techniques considerably.

7 Multi-Class Extensions

7.1 Motivation

There are several ways to allow a user of a BCI system to choose between more than two decisions. For example, there are the codings described in chapter 3. Alternatively one could use two classes and time structure, e.g., the basket feedback described in section 4.2. Here how strongly a class is ordered and the duration a class is performed are important values which allow for more decisions. Unfortunately, there is a limit in the control of the timing of a BCI (see [138, 84]). Another option which is discussed in this chapter is the number of mental states which can be detected by the system. Mental states in this context could be imagination of different motor tasks, sensation of events or complex mental tasks. With the direct extension of mental states used one point arises which should be recognized in creating feedback for specific situations: On the one hand a user could become confused and overtaxed by too many classes. On the other hand in some feedback experiments more classes could be more intuitive for the user. Obviously, the solution to this problem is strongly feedback and subject dependent. In the following I will ignore this psychological problem and only illuminate the question of how to extend existing algorithms to multi-class versions and how many classes, i.e., different mental states, are adequate for BCI control in the sense of transmittable information.

7.1.1 Neurophysiological Background

Intuitively, a BCI system is more useful the more classes can be perfectly controlled since the bitrate of N perfectly controlled classes is $\log_2(N)$. Unfortunately, with the number of classes the achieved accuracy of the BCI system decreases as will be seen in section 7.1.2. Furthermore suitable classes in the sense of discriminability have to be found. In the literature there are several groups of suitable classes for controlling a BCI, namely (imagination of) movements, sensations and mental tasks. For imagined or real movements there is usually a corresponding region in the somatosensory and motor area of the neocortex for each part of the human body. Neighboring parts of the body are represented in neighboring parts of the cortex, which is shown in Fig. 7.1. Note that the corresponding parts of the body appears contralateralized on the vertex, i.e., the left hand is on the right hemisphere, the right hand on the left. Unfortunately the use of many different movements is restricted, since regions which are close together are hard to discriminate in the EEG. Consequently, movements of neighbored parts of the body (e.g., finger and hand) are hard to discriminate and thus are not suitable for combined use. Therefore one should choose only a few movements corresponding to brain areas which are far away from each other, e.g., one foot, left and right hand and tongue movements are very good candidates at this point. Besides movements one could use e.g., auditory, visual, tactile or haptic sensations. However, it is not clear in each case whether they are really different from imagined movement (a tactile sensation in the left

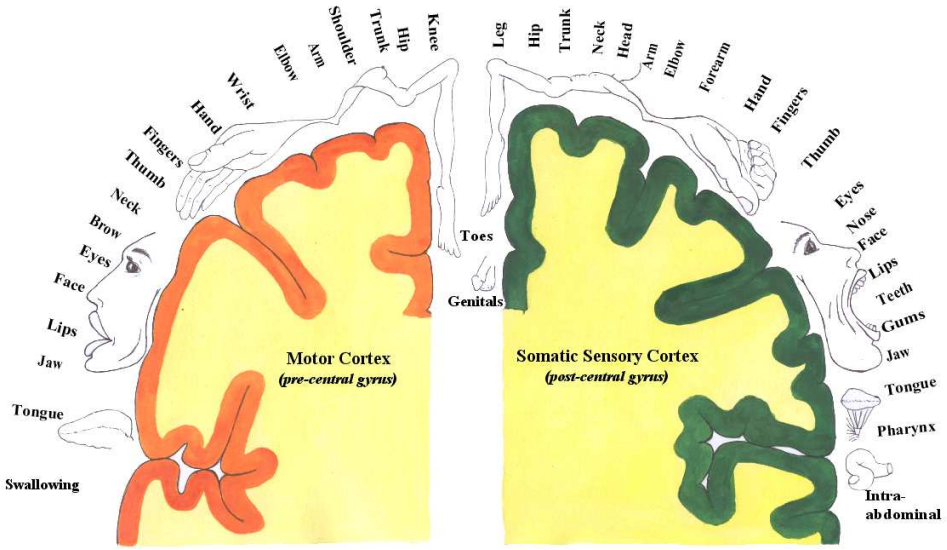


Figure 7.1: The figure taken from Krepki [73] shows the corresponding areas in the brain of parts of the human body. Note that both sides are symmetric except that the left side controls right body parts and the right side left body parts.

hand could be very similar to an imagined movement of the left hand in terms of the measured EEG). Furthermore, the use of these sensations in complex feedback environments is limited, e.g., visual sensation during a complex visual processing of the feedback seems inappropriate. Finally one could use mental tasks as classes, like mental arithmetics, mental rotation or mental spelling. First of all one should note that they could correspond strongly to other classes too, e.g., if a mental rotation is only done visually it could be very similar to visual sensations in terms of measured EEG. Finally, if a BCI system for cursor control uses mental tasks or sensations (e.g., moving a cursor to the left by a tactile sensation) the subject has to associate these tasks with movements on the screen which seems to be very unnatural in feedback control.
To summarize, at first glance it seemed that a lot of classes exist. But how many classes are really distinguishable (which is of course highly subject dependent) and how natural do they appear in feedback environments? Additionally, the result of the next section will be that it does not make sense to increase the number of classes for existing BCI performances arbitrarily. This will be clarified by experimental results in section 7.4.

7.1.2 Theoretical background

Let us start with some data presented by the random variable $X \in \mathbb{R}^m$ and label $Y \in \{1,...N\}$ with m as dimension of the feature space and N as number of classes. Furthermore equal class priors and normal distributed data with equal covariances for all classes, i.e., $X|Y = y \sim \mathcal{N}(\mu_y, \Sigma)$ for all $y = 1,...,N$ are assumed. By theorem 5.2.1 the optimal classifier in the sense of misclassification risk is defined by $\tilde{f} = \text{argmax}_{i=1,...,N} \tilde{f}_i$ with $\tilde{f}_i(x) = \mu_i^\top \Sigma^{-1} x - 0.5 \mu_i^\top \Sigma^{-1} \mu_i$ for all $x \in \mathbb{R}^n$ and $i = 1,...,N$. Additionally this theorem states that for each subset $S \subset \{1,...,N\}$ (with at least two elements) the optimal classifier

on this subset is given by $\tilde{f}_S = \text{argmax}_{i \in S} \tilde{f}_i$. If one chooses a subset $S \subset \{1,...,N\}$ with $\frac{1}{\#S} \sum_{s \in S} P(\tilde{f}(X) = s | Y = s) \geq P(\tilde{f}(X) = Y)$ one gets

$$\begin{aligned} P(\tilde{f}_S(X) = Y | Y \in S) &= \frac{1}{\#S} \sum_{s \in S} P(\tilde{f}_S(X) = s | Y = s, Y \in S) \\ &= \frac{1}{\#S} \sum_{s \in S} P(\tilde{f}_S(X) = s | Y = s) \\ &\geq \frac{1}{\#S} \sum_{s \in S} P(\tilde{f}(X) = s | Y = s) \\ &\geq P(\tilde{f}(X) = Y). \end{aligned}$$

The choice of such a subset S is possible since $P(\tilde{f}(X) = Y)$ is the mean about $P(\tilde{f}(X) = s | Y = s)$ for all $s \in \{1,...,N\}$. Excluding the lowest values here increases (or at least does not decrease) the calculated mean such that a suitable subset S exists.
Consequently, the classification accuracy on a well-chosen subset can not be worse than on the whole set of classes. However, the question arises how big the decrease of the accuracy is. In general, this question cannot be answered. One needs assumptions on the data like normal distributed data which I assume here. In this case an answer to the question can be found. However, the existence of a general analytical solution is not known to my knowledge, since calculations of the area of polyhedrons in the Gaussian space is analytically not possible in general. In Dornhege et al. [45] I have found that given an equal pairwise classification accuracy acc for a three-class classification problem the resulting classification is between $\text{acc} - \frac{\exp(-\frac{2(\text{erf}^{-1}(\text{acc}))^2}{3})}{6}$ and $\text{acc} - \frac{\exp(-\frac{(\text{erf}^{-1}(\text{acc}))^2}{2})}{6}$. To reveal these bounds the feature space was divided into several analytically calculable areas (see section A.4 for the complete proof). For more than three classes or more general assumptions of different pairwise classification accuracies two methods are available to estimate the expected error: First of all one could estimate the true error by numerically calculating the integral for the expected error based on approximation by small cuboids until a specified preciseness is achieved. The second idea based, on a Monte Carlo approach, uses the law of large numbers and the central limit theorem. In this case normal distributed data are drawn and applied to the classifier to measure the accuracy. By the law of large numbers this value converges almost surely against the true classification accuracy and the central limit theorem gives the speed of this convergence. Based on the latter one calculates how much data one has to draw such that with probability 99 % the resulting accuracy is in the range of 0.01 around the true value. In both cases (Monte Carlo or numerical approximation of the integral) the underlying geometry is calculated based on the pairwise classification accuracies by theorem 5.2.3. Note that the calculations can be simplified by letting $\Sigma = I$ and choosing a $N-1$-dimensional feature space (after suitable transformation on the manifold containing the means).
Both methods were simulated and the result is the same. In Fig. 7.2 the corresponding classification accuracies and bitrates for $N = 2,...,6$ classes are visualized based on the equal binary pairwise classification accuracies. One observes that (as noticed above) the accuracy decreases with the number of classes. However, a gain in bitrate can be observed by using more than two classes. Nevertheless, depending on the outgoing pairwise classification accuracy the gain is very small if one uses more and more classes. E.g., for 80 % the gain of using more than three classes is very small, for 90 % it is more than four classes and so on.

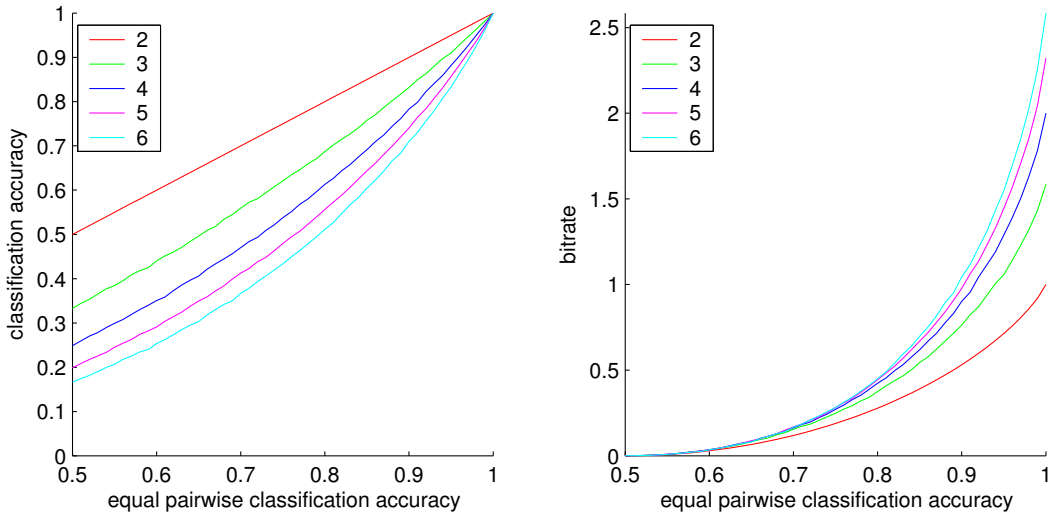


Figure 7.2: Based on a pairwise – for all classes equal – classification accuracy (varied on the x-axis) the accuracy (left figure) and the bitrate (right figure) of using $N = 2, ..., 6$ classes is numerically calculated and plotted on the y-axis.

Due to the fact that equal pairwise classification accuracies are a very strong and unrealistic assumption, especially in BCI, one should conclude that it is not useful to extend the number of classes arbitrarily. For the range of achieved accuracies in BCI literature one should use 3 or 4 classes.
In Fig. 7.3 the influence of varying pairwise classification performances for Gaussian distributed data with equal covariances $\Sigma = I$ are shown for $N = 3$ classes with the same simulation techniques as described above. Here one pairwise classification accuracy is fixed to $\text{acc}_1 = 0.95$ and the other two are varied between $\text{acc}_{2,3} = 0.5$ and $\text{acc}_{2,3} = 0.95$. It should be mentioned that the means of the classes are the edges of a triangle with side length fixed by theorem 5.2.3 based on the given classification accuracy. Obviously the triangle is unique up to rotation and mirroring. However, there is no such triangle for all sets of side length, in other words, not for all combination of classification accuracies used in the simulation does a real Gaussian example exist. In these cases no point is visualized in Fig. 7.3. For all other cases the resulting bitrate is shown. Furthermore the area is specified when ternary classification outperforms the best binary classification based on information transfer rate. The figure reveals that it does not make sense to increase the number of classes if pairwise accuracy varies too strongly, but if they are similar, a performance gain can be achieved.

7.2 CSP multi-class extensions

Often formulations of algorithms are only presented for binary classification tasks. Fortunately, the LDA (or more general the QDA or RDA) algorithm has a multi-class formulation such that one can directly use it. The Common Spatial Pattern (CSP) algorithm is one of the algorithms that is used in this work which has no direct multi-class formulation.
Based on the idea of CSP described in section 5.1 several opportunities exist to extend this

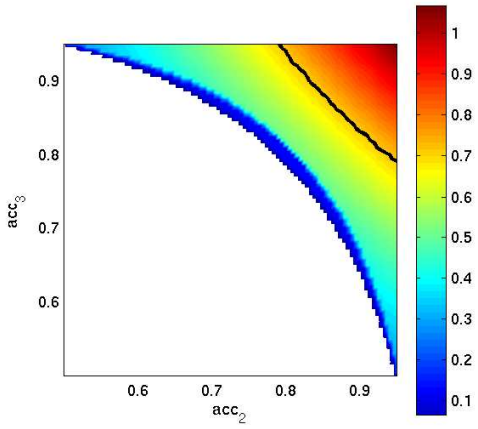


Figure 7.3: For $N = 3$ classes and pairwise classification accuracies $\text{acc}_1 = 0.95$, $\text{acc}_2 = 0.5, ..., 0.95$ varied on the x-axis and $\text{acc}_3 = 0.5, ..., 0.95$ varied on the y-axis the resulting ternary bitrate based on Gaussian distributed data with equal covariances is visualized. Only combinations of acc_1, acc_2 and acc_3 which are possible are shown in the colorplot. Additionally the boundary ternary classification which outperforms binary classification in the sense of better bitrate is shown as a black line. Above the black line one should prefer the ternary classification problem.

algorithm to multi-class. Some of these algorithms I have also presented in Dornhege et al. [44, 45]. The first two approaches are based on general multi-class extension ideas. In the third approach one uses an approach called simultaneous diagonalization. Finally two ideas based on the optimization approach in equation (5.2) are presented.
In the following $\Sigma_y = \frac{1}{\#\{i|y_i=y\}} \sum_{i|y_i=y} s_i s_i^\top$ denotes the covariance matrix as in section 5.1.9 for each class $y = 1, ..., N$.

(**OVR**). Binary combination strategies like one versus the rest or pairwise classification are

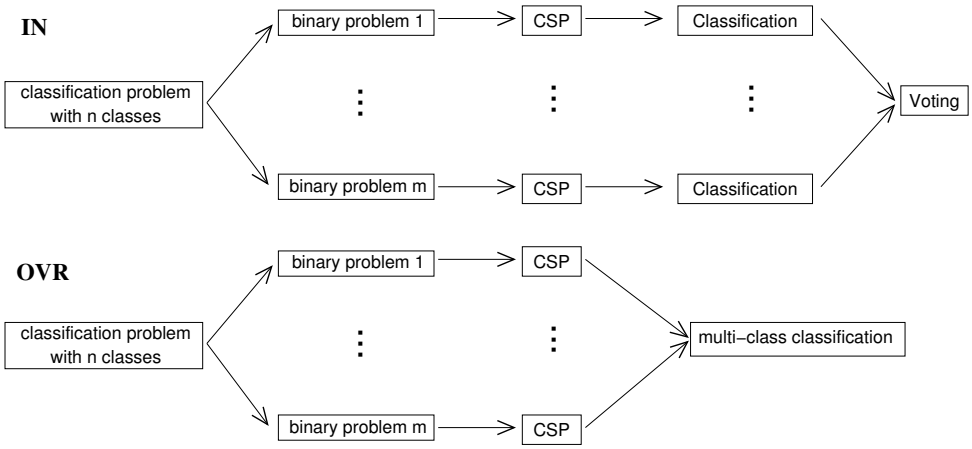


Figure 7.4: In the upper resp. lower plot the classification procedure for **IN** resp. **OVR** is shown. Here n classes were used and splitted in m binary classification subsets (e.g., $m = n$ for one versus rest classification or $m = \begin{pmatrix} n \\ 2 \end{pmatrix}$ for pairwise classification.

often used in such a situation. The idea is to use several binary classifications to get a multi-class decision. In the case of one versus the rest classification, binary classifiers are trained on each class against all other classes. Finally the class is chosen with the most votes (taking into account the confidence of each vote). Pairwise combination works similarly except that all binary classifiers on each two-class subset are used. I will use the one versus the rest approach to estimate all spatial patterns, and finally apply multi-class LDA. Note that pairwise classification was tested, too, but has not shown a significant difference. Thus these results are omitted here.

(**IN**). This approach, suggested by Ramoser et al. [116] for CSP, is based on the idea of doing the CSP approach within binary classification based on one versus rest or pairwise strategies. Pattern calculation and classification is done on binary tasks and a decision is made by voting. In **OVR** only pattern calculation is binary, whereas classification is done on all patterns. The difference between **IN** and **OVR** can be clearly observed in Fig. 7.4.

(**SIM**). In equation (5.3) the simultaneous diagonalization of two matrices is used to retain the result. For more than two classes one could try to do the same, namely finding a simultaneous diagonalization of the covariance matrices for each class which has to be calculated for CSP. Unfortunately, in general a simultaneous diagonalization of more than two matrices does not exist. But there are ways to approximate a simultaneous diagonalization (cf. [29, 113]). Obviously this approximation depends on the used error function. Here I choose the algorithm described in Ziehe et al. [145, 146] due to its speed and reliability, which minimizes the sum of the square values of the off-diagonals, i.e., which finds matrices V and D_y with $\Sigma_y = V^\top D_y V$ for all y with $\det V = 1$ and minimal square sum of the off-diagonals of all D_y. In contrast to the two class problem there is no canonical way for choosing the relevant multi-class CSP patterns. I investigated several opportunities (e.g., using the highest and lowest eigenvalues). However, the best strategy was based on the idea that two different eigenvalues for the same pattern have the same effect if their ratios to the mean of the eigenvalues of the other classes multiplies to 1, i.e., the ratios are multiplicatively inverse to each other. This results in choosing the highest score values of $\text{score}(\lambda) := \max(\lambda, 1/(1+(N-1)^2\lambda/(1-\lambda)))$ for each class, especially for two classes this results in $\max(\lambda, 1-\lambda)$. If a second class chooses the same pattern it is left out for this class and the next one, i.e., with the next highest score for this class, is chosen. Finally the signal is projected with all these patterns and after calculating the power by the logarithm of the variance in the signal usual multi-class LDA is applied.

(**OPT**). A closer look at (5.2) shows a further CSP-multi-class option, namely,

$$\max_w \quad w^\top \Sigma_j w, \quad s.t. \quad w^\top (\sum_{i=1,\ldots,N} \Sigma_i) w = 1 \tag{7.1}$$

for all classes $j = 1, \ldots, N$. In the two class case the maximization matches the minimization of the other classes, therefore only the maximum for each class is used. In the multi-class case this could be different, therefore one also calculates the minimum in equation (7.1) and uses both results. A solution to this problem can be found easily: it can be solved similar to the binary case by generalized eigenvalues or whitening and eigenvalue analysis.

(**OPTe**). In the formulation above solutions are found by comparing one class against the others. It could be that a solution to equation (7.1) has a good discrimination to some other

classes, but not to all since it is only compared to a mean performance of the other classes. For example, in a three class case let us assume a solution of equation (7.1) has value 0.45 for class 1. If one calculates with this pattern the values $w^\top \Sigma_i w$ for the other classes everything could happen as long as the results sum to $1-0.45=0.55$. E.g., one class could be 0.45 and the other 0.1, which would be bad for discrimination against all other classes since it would not work against one other class. On the other hand it could be 0.275 for both classes which would perhaps be better. The idea to have equal values for the other classes leads to the following optimization approach

$$\max_w \quad w^\top \Sigma_j w, \quad s.t. \quad w^\top (\sum_{i=1,\ldots,N} \Sigma_i) w = 1, \quad w^\top \Sigma_i w = w^\top \Sigma_k w \quad \forall (k,i) \neq j.$$

Note that the last constraints can be reduced to $N-2$ constraints as follows

$$\max_w \quad w^\top \Sigma_j w, \quad s.t. \quad w^\top (\sum_{i=1,\ldots,N} \Sigma_i) w = 1, \quad w^\top \Sigma_i w = w^\top \Sigma_k w \quad \text{with one } k \neq j \text{ and } \forall i \neq j,k. \tag{7.2}$$

More generally one tries to solve the following problem

$$\operatorname*{opt}_w \quad w^\top A w, \quad s.t. \quad w^\top B w = 1, \quad w^\top C_i w = c_i \quad \forall i = 1,\ldots,p \tag{7.3}$$

with some $p > 0$, $A, B, C_i \in \mathbb{R}^{n,n}$ symmetric and A, B positive semidefinite. Here opt generally denotes min or max. This problem can appear in several environments. For example for navigation of a 2d-system one needs four classes with high discrimination between two disjoint pairs. If a class is performed the corresponding classifier should work with high confidence whereas the other classifier should be unbiased. This could be achieved by using equation (7.3) with $A = \Sigma_1$, $B = \Sigma_1 + \Sigma_2$, $p = 2$, $C_1 = \Sigma_3$, $C_2 = \Sigma_4$, $c_1 = 0.5$, $c_2 = 0.5$ if classification for one direction should work on class 1 and 2 and in the other direction on class 3 and 4. A few simulations on data with four classes has shown that this gives the desired result. However, 2d-feedback is not the issue of this work, therefore I omit further details.

Calculating the dual of equation (7.3) one gets

$$\operatorname*{\hat{opt}}_\xi \quad \lambda_{\text{opt}} \left(A - \sum_{i=1,\ldots,p} \xi_i C_i, B \right) + \sum_{i=1,\ldots,p} \xi_i c_i. \tag{7.4}$$

Here $\hat{\text{opt}}$ denotes min for opt = max and max for opt = min. Furthermore $\lambda_{\min}(D,E)$ resp. $\lambda_{\max}(D,E)$ for two matrices D, E denotes the lowest resp. highest generalized eigenvalue of D, E (i.e., $\operatorname*{opt}_{w, w^\top w = 1} \frac{w^\top D w}{w^\top E w}$). Note that a solution for (7.4) in general does not exist. E.g., the positive definiteness of B guarantees the existence of a solution. However, in the situation of the optimization problem (7.2) the corresponding matrix is positive definite. Thus a solution exists. One can solve the problem by usual line-search optimization approaches, since the problem has usually only a few (namely $N-2$) dimensions.

By using more than one general eigenvalue of the final solution one could choose more patterns, too. Furthermore both, min and max are used.

Note that it is not clear that besides the absolute optimum further local optima exist. Therefore one should take care with this problem and should avoid falling into only local optima.

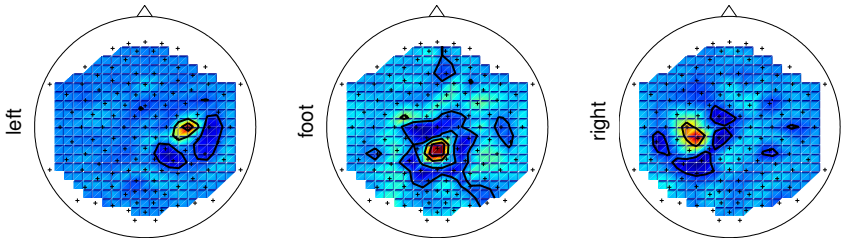


Figure 7.5: The figure show the CSP filter obtained by the CSP multi-class extension SIM for one dataset during left and right hand and foot movement. The filter matches the expected neurophysiological structure of motor areas.

7.3 Results

To demonstrate the algorithm SIM one dataset with imagined left and right hand and foot movements is used. The resulting first patterns for each class are shown in Fig. 7.5 which reveal the expected neurophysiology.
To compare the proposed algorithms I use all subsets of at least 3 classes of all *imag* datasets. I use leave-one-out cross-validation. To get comparable results I choose $4N$ patterns for each algorithm. Usually one could choose further patterns also by iteratively applying the algorithms in the orthogonal space of the chosen patterns. However, this was tested without significant differences.
Since the method **IN** was suggested as the standard approach in Ramoser et al. [116] this algorithm is compared to the other ones in Fig. 7.6. In the figures one cross corresponds to a result of one subset with the x-coordinate given by the bitrate of **IN** and the y-coordinate as the bitrate for the other algorithm. Thus crosses above the diagonal correspond to datasets where the proposed algorithm outperforms the algorithm **IN** which is the case for almost all algorithms and datasets. Therefore the choice of **IN** is not advisable. Comparing the other algorithms in similar graphs (which are skipped here) shows that there is not an overall best algorithm, the performance varies strongly between the algorithms for each dataset. There seems to be a small case in favor for **SIM** without being significant. One should individually choose the best algorithm.

7.4 How many classes should one use?

In section 7.1.2 it was stated that theoretically three or four classes could be the best choice in the sense of achieved ITR. To confirm this practically I use all our *imag* datasets with at least three classes and calculate the best binary classification bitrate (i.e., I take all subsets of the datasets which consist exactly of two classes, calculate their performance and take the subset with the highest performance for each dataset), the best 3-class subset bitrate and so on. The results are plotted in Fig. 7.7. Note that in each case I individually choose the best algorithm of the ones suggested in section 7.2. Furthermore the same results were plotted in Fig. 7.8 if the best algorithm is chosen within feature combination with the MRP feature (see chapter 6) with the algorithm PROB.
One can conclude that in most datasets a gain can be achieved by using a third class. For

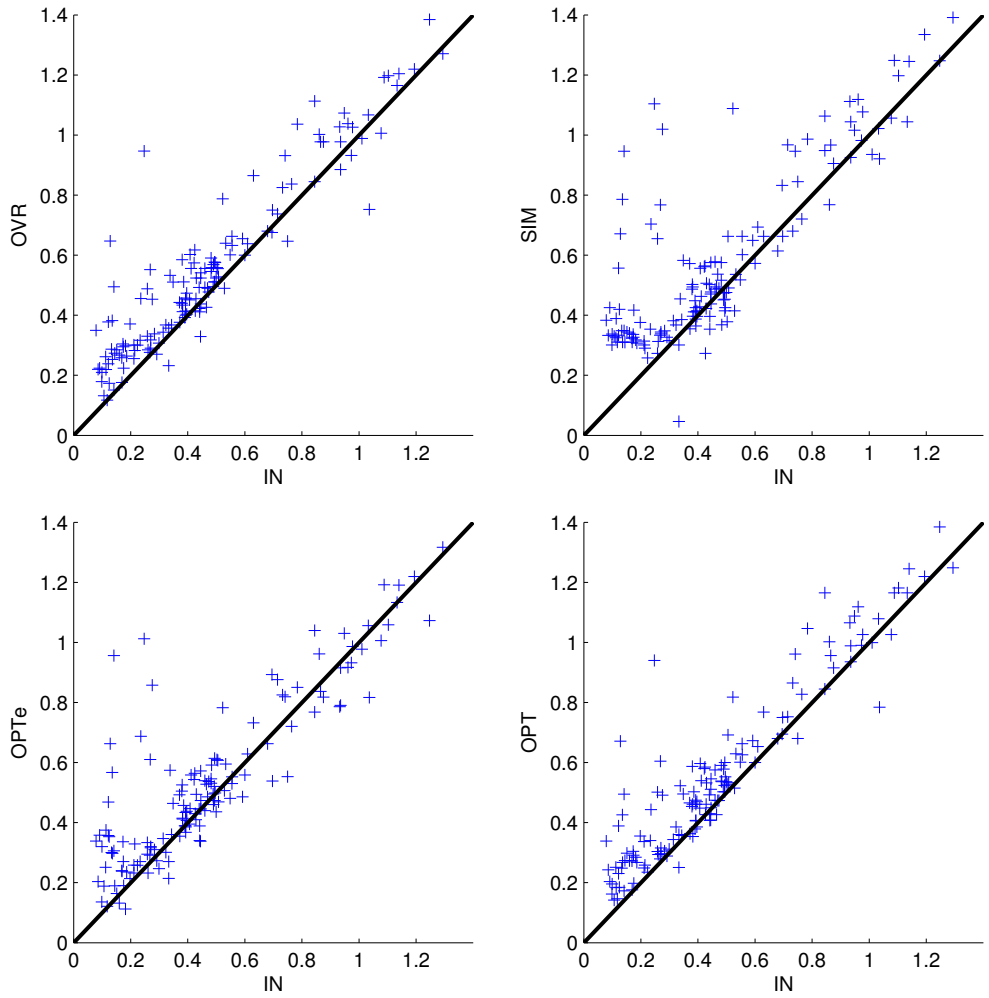


Figure 7.6: The figures shows for all at least 3 class subsets of all our *imag* datasets the performed leave-one-out bitrate for the algorithm **IN** against one other denoted at the *y*-axis. For values above the diagonal the algorithm on the *y*-axis outperforms the algorithm **IN**.

a few datasets this is not the case. A closer look at these datasets shows that in these cases the pairwise classification accuracies varies strongly such that a third class cannot really be distinguished from the others. If the number of classes is increased further the bitrate does not generally increase further. In some datasets four classes perform best, in some three. But there is not only one dataset where the use of more than four classes makes sense. The situation is similar for combination, except that the results are slightly better there, such that a slightly higher tendency to four classes exists. Nevertheless, as suspected in section 7.1.2, there is a limit of four classes which should be used for a BCI system. However, this is only true for the BCI performance that can be achieved so far. If the accuracy is enhanced drastically, the number of classes could be increased too. Finally it should be reiterated that the psychological meaning of the number of classes as discussed at the beginning of this

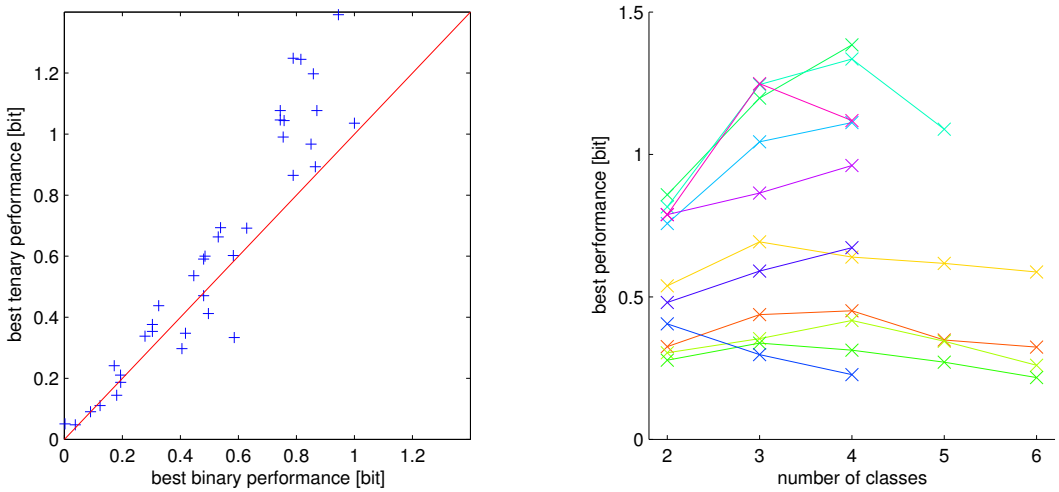


Figure 7.7: The plot on the left shows the best binary subset classification for one dataset against the best ternary subset classification in bitrate on leave-one-out validation on all our *imag* datasets with the suggested CSP algorithms. On the right for all datasets with at least four classes the best performances for all possible subsets of each number of classes are shown. This number of used classes is varied on the x-axis whereas on the y-axis the bitrate is visualized.

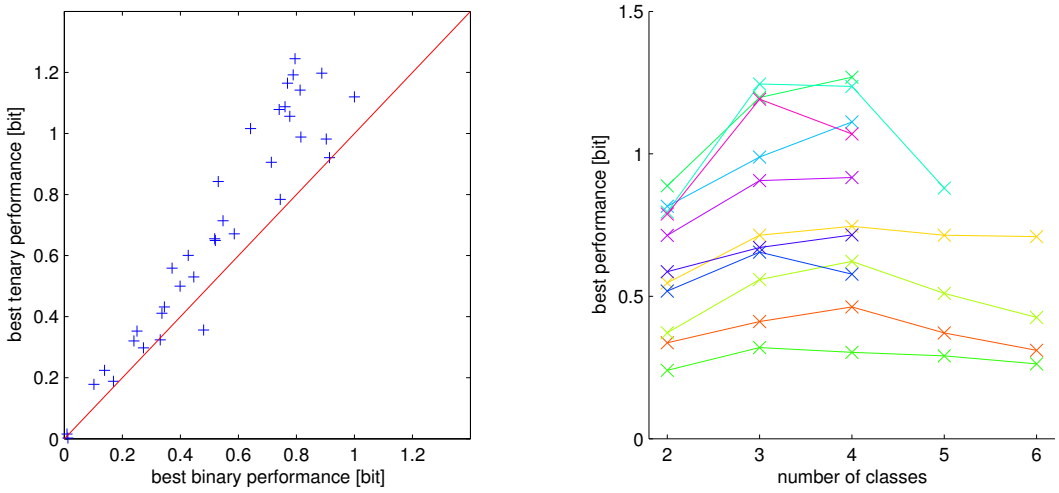


Figure 7.8: The plot on the left shows the best binary subset classification for one dataset against the best ternary subset classification in bitrate on leave-one-out validation on all our *imag* datasets with the suggested CSP algorithms in combination with the MRP feature with the algorithm PROB. On the right for all datasets with at least four classes the best performances for all possible subsets of each number of classes are shown. This number of used classes is varied on the x-axis whereas on the y-axis the bitrate is visualized.

chapter should not be ignored.
So far a systematic analysis of multi-class extensions in online experiments has not been done. However, first attempts with more than two classes give reason to believe that the main results of this chapter will be confirmed in BCI feedback applications too.

8 Spatio-temporal filters for CSP

As introduced in chapter 5, the Common Spatial Pattern algorithm (CSP) (see [116]) has proven to be very useful in extracting discriminative spatial filters based on ERD effects for each subject individually (see for example [45]). Unfortunately the used frequency band has to be chosen individually for each subject since a high subject variability in the presence of frequency rhythms in power and band exists. For off-line analysis a broad band filter (see [116, 110]) was chosen, but one observes that a more specific fit in several datasets is advisable. One could do that manually based on observation of meaningful plots like power spectra with r^2-values. In this chapter one algorithm is introduced into CSP which is able to fit to the specific suitable brain rhythms automatically. Hereby I start with a neurophysiological observation about brain rhythms for one subject in section 8.1 to clarify this problem. In a second section several suggestions to solve this problem are introduced (see section 8.2). Finally I will compare the results for these algorithms in section 8.3. Note that the presentation here is an extension of my publication (see [46, 47]).

8.1 Neurophysiological Background

Brain activity during wakefulness or rest can be described by several rhythms located at different brain areas (see [66, 67, 10, 37]). These rhythms are attenuated by real or imagined movements bilateral but more pronounced contralaterally in the corresponding area. Usually one differentiates several brain rhythms like α, β and μ. Hereby α and μ are usually in similar frequency ranges, but the former comes from the visual cortex, the latter from the sensorimotor cortex. Due to volume conduction these two rhythms interfere visibly over the motor cortex. Thus a μ-power based classifier can react or can depend on modulations of the posterior α-rhythm that varies with changes in visual processing and fatigue. If these two rhythms have different spectral peaks as in Fig. 8.1 a suitable frequency filter can help to eliminate the distorting effects of the variations in α-power.
In Fig. 8.1 the spectra for one dataset is plotted on sensorimotor and parietal areas where this effect can be observed. Here the subject was prompted to imagine a left hand or a foot movement. Below each channel r^2-values are plotted which describes the discriminability in this frequency range for this channel between left hand and foot. The chosen electrode Pz for the visual cortex shows only one peak at 8 Hz whereas the corresponding channels over the motor cortex show two peaks, one at 8 Hz and one at 12 Hz. Consequently, it could be that the first is mostly coming from the posterior α and thus does not have this high discrimination, whereas the second is the real motor rhythm with a good discrimination.
Unfortunately every brain shows its own behavior, so that the exclusion of the frequency range around 8 Hz which is advisable for this specific subject could eliminate relevant information for other subjects, decreasing the performance drastically. Thus an individual frequency fitting has to be done.

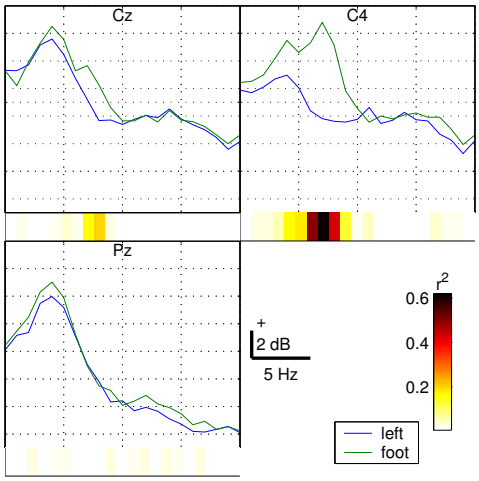


Figure 8.1: The plot shows the spectra for one subject during left hand (blue line) and foot (green line) motor imagery between 5 and 25 Hz at scalp positions Pz, Cz and C4. In both central channels two peaks, one at 8 Hz and one at 12 Hz are visible whereas at Pz only the peak at 8 Hz is pronounced. Below each channel the r^2-value which measures discriminability is added. It indicates that the second peak contains more discriminative information.

8.2 Algorithms

As discussed before suitable subject-specific spectral filters have to be constructed focussing on the frequency bands with the most discriminative information. This is a typical problem in machine learning, namely that based on some data an adaptation of the machine to this data has to be performed. In the following several approaches are suggested to solve this problem:

CSSP. In Lemm et al. [81] the following algorithm was introduced: Given s_i the signal s_i^τ is defined to be the signal s_i delayed by τ timepoints regarding the sampling rate. After concatenating s_i and s_i^τ in the channel dimension and treating the delayed signals as new channels, normal CSP is applied. Hereby the ability to emphasize or neglect specific frequency bands is given which strongly depends on the choice of τ. The estimation of a suitable τ can be achieved by validation on the training set. One could repeat this approach for several τ's to find more complex frequency filters. However, since the training set is usually small in BCI, Lemm et al. [81] discovered that increasing flexibility of the frequency filter by introducing more delayed taps results in extreme overfitting meaning that one delay tap is most effective.

Classification on several bands. For suitable temporal filtering one could also use CSP on data filtered to several frequency bands and combine the results into one new feature vector. In other words one could estimate a few CSP patterns individually for each used rhythm and apply them. Finally the different band-power values were concatenated and classification could take place (see [14]). I use it here together with **LDA**. Interestingly, there is also the option of applying Multi Kernel Learning (**MKL**) with a linear kernel (see chapter 5) on each band power feature group (note that one has several values for each rhythm). Following this idea one finds a sparse weighting of used frequency rhythms and gets an additional interpretation for the required frequency rhythms for each individual subject. Note that the MKL parameters are estimated on the training set only. The choice of the rhythms is crucial for these approaches. I have tried several things like many very small bands, or few overlapping bands. In this thesis I will present the results for the following rhythms θ =4–6 Hz, α_1 =7–10 Hz, α_2 =10–14 Hz, β_1 =15–20 Hz and β_2 =20–25 Hz since

I was not able to improve these results with another set of rhythms.

CSSSP. The CSSP algorithm allows fitting of a frequency filter individually to each channel. But usually the result of CSSP tends to an approximate global (i.e., identical for all channels) temporal filter. Thus a more stable estimation could be based on learning a global temporal filter directly, which is the idea of the CSSSP. Consequently with this restriction, the ability to fit to a more complex filter without the strong overfitting problems of CSSP results.

Usually signals are filtered by a digital frequency filter (see section 5.1.1) which consists of two sequences a and b with length n_a and n_b such that the signal x is filtered to y by

$$\begin{aligned} a(1)y(t) = \quad & b(1)x(t) + b(2)x(t-1) + \ldots + b(n_b)x(t-n_b-1) \\ - \quad & a(2)y(t-1) - \ldots - a(n_a)y(t-n_a-1). \end{aligned}$$

The next steps require the restriction to FIR (finite impulse response) filters by defining $n_a = 1$ and $a = 1$. Furthermore $b(1)$ is defined by 1 and the length of b is fixed to some T with $T > 1$. This restriction causes some flexibility of the frequency filter to get lost. But it allows us to find a suitable solution in the following way: The goal is to find a real-valued sequence $b_{1,\ldots,T}$ with $b(1) = 1$ such that the trials

$$s_{i,b} = s_i + \sum_{\tau=1,\ldots,T} b_\tau s_i^\tau \tag{8.1}$$

show a better behavior in the sense of discriminability.

Using equation (5.1) one has to solve the problem

$$\max_{w,b,b(1)=1} \quad \sum_{i:\text{Trial in Class 1}} var(w^\top s_{i,b}), \quad \text{s.t.} \quad \sum_i var(w^\top s_{i,b}) = 1. \tag{8.2}$$

Define $\Sigma_y^\tau := E(\langle s_i (s_i^\tau)^\top + s_i^\tau s_i^\top \,|\, i : \text{Trial in Class } y\rangle)$ for $\tau > 0$ and $\Sigma_y^0 := E(\langle s_i s_i^\top \,|\, i : \text{Trial in Class } y\rangle)$, namely the correlation between the signal and the by τ timepoints delayed signal. Since one can assume that $E(\langle s_i^\tau s_i^\top, | \, i : \text{Trial in Class } y\rangle) \approx E(\langle s_i^{\tau+j} (s_i^j)^\top, | \, i : \text{Trial in Class } y\rangle)$ for small $j > 0$, equation (8.2) can be approximately simplified to

$$\begin{aligned} &\max_{b,b(1)=1} \max_w \quad w^\top \left(\sum_{\tau=0,\ldots,T-1} \left(\sum_{j=1,\ldots,T-\tau} b(j)b(j+\tau) \right) \Sigma_1^\tau \right) w, \\ &s.t. \quad w^\top \left(\sum_{\tau=0,\ldots,T-1} \left(\sum_{j=1,\ldots,T-\tau} b(j)b(j+\tau) \right) (\Sigma_1^\tau + \Sigma_2^\tau) \right) w = 1. \end{aligned} \tag{8.3}$$

Since one can calculate for each b the optimal w by the usual CSP technique (see equation (5.2) and (5.3)) a $(T-1)$-dimensional (b(1)=1) problem remains which can be solved with usual optimization techniques like gradient or line-search methods if T is not too large.

Thus one gets for each class a frequency band filter and a pattern (or similar to CSP more than one pattern by choosing the next eigenvectors).

However, in order to avoid overfitting with increasing T the complexity of the frequency filter has to be controlled. One well-established way to do so is to enforce a sparse solution

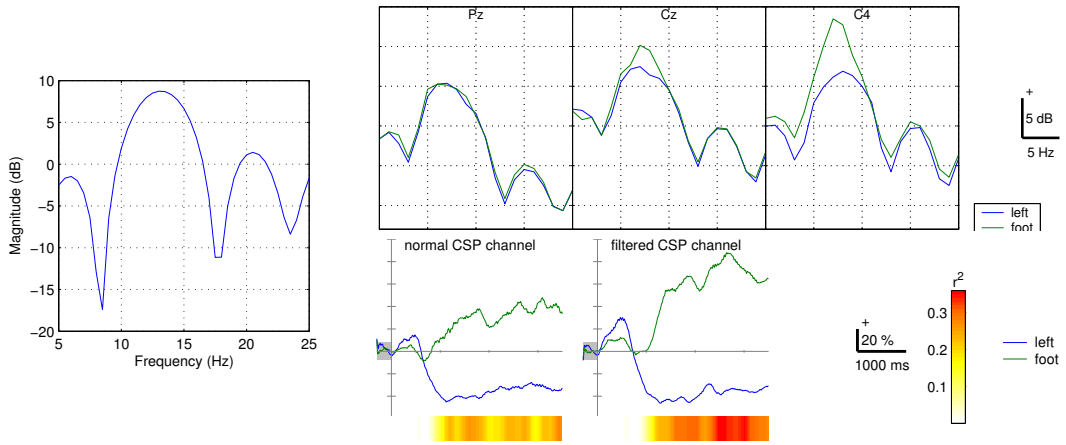


Figure 8.2: The plot on the left shows one – with CSSSP – trained frequency filter for the subject whose spectra was shown in Fig. 8.1. In the upper plot on the right the resulting spectra are visualized after applying the frequency filter on the left. In the lower plot on the right the ERD and the r^2-value for this ERD on C4 is shown for the normal filtered case (left) and the additionally by CSSSP filtered case for the same dataset. By this technique the classification error in chronological validation can be reduced from 17.4 % for CSP and CSSP to below 2 % with parameter C estimated by validation on the training set.

for b, i.e., a solution with only a few non-zero entries. This is done by introduction of a regularization term in the following way:

$$\begin{aligned} \max_{b,b(1)=1} \max_{w} \quad & w^\top \left(\sum_{\tau=0,\ldots,T-1} \left(\sum_{j=1,\ldots,T-\tau} b(j)b(j+\tau) \right) \Sigma_1^\tau \right) w - C/T\|b\|_1, \\ s.t. \quad & w^\top \left(\sum_{\tau=0,\ldots,T-1} \left(\sum_{j=1,\ldots,T-\tau} b(j)b(j+\tau) \right) (\Sigma_1^\tau + \Sigma_2^\tau) \right) w = 1. \end{aligned} \tag{8.4}$$

Here C is a non-negative regularization constant, which has to be chosen, e.g., by cross-validation. The 1-norm is used in this formulation to achieve a sparse solution for b: With higher C one gets sparser solutions for b until at one point the usual CSP approach remains, i.e., $b(1) = 1, b(m) = 0$ for $m > 1$. I will call this approach *Common Sparse Spectral Spatial Pattern* (CSSSP) algorithm.

8.3 Results

In Fig. 8.2 one chosen frequency filter of CSSSP is visualized for the subject whose spectra were shown in Fig. 8.1. Furthermore the remaining spectrum after using this filter is shown. As expected the filter detects that there is a high discrimination in frequencies around 12 Hz, but only a low discrimination in the frequency band around 8 Hz. Consequently a filter is trained which drastically decreases the amplitude in this very predominant band, whereas full power at 12 Hz is retained. In Fig. 8.2 the ERD (i.e., the relative changes in power regarding some baseline interval marked in gray) at electrode C4 for the normally filtered

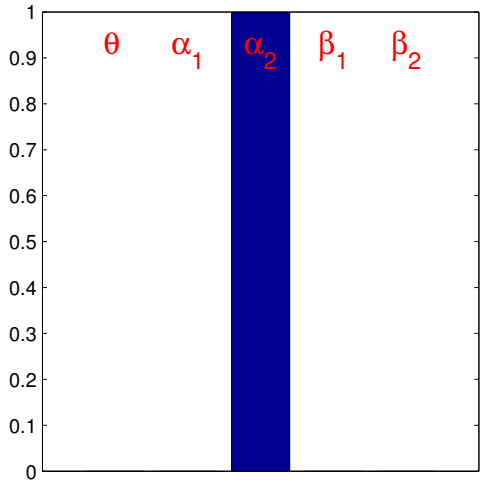


Figure 8.3: The plot shows the chosen weightings for classification with multiple kernel learning for the subject whose spectra was shown Fig. 8.1. By these techniques the classification error in chronological validation could be reduced from 17.4 % to below 2 % for both approaches with parameter C estimated for the second case by another validation on the training set only.

case and for the case is shown additionally filtered by CSSSP. The presented r^2-values reveal the improvements for the ERD effects due to the the additional filtering.

In Fig. 8.3 the chosen weightings of classification with multiple kernel learning is shown. Again a focus on the peak around 12 Hz is clearly visible.

Altogether the suggested algorithms enhance the classification performance compared to CSP and CSSP for this dataset considerably. Unfortunately, the situation looks different for the other datasets.

I have applied the algorithms in chronological validation to all subsets of the *imag* dataset which contains exactly two classes. Here only the classes with imagined movements are chosen. But similar effects for *selfpaced* data and other imagined classes can be expected. Note that all parameters were chosen on the training set by another validation only if a parameter has to be estimated. For each algorithm I choose 3 patterns per class, all other values remain fixed. For CSSP τ values between 0 and 20 were allowed, for CSSSP $T = 16$ was chosen. One should note that the choice of the parameter C varies strongly among the datasets. For example the parameter C of CSSSP usually is very small if datasets like the one visualized in Fig. 8.1 are given whereas the parameter is high, if discrimination in all rhythms are shown (in this case CSSSP finds the CSP solution).

The results of the validation are shown in Fig. 8.4. Here the results for CSP (on the y-axis) are compared to the results of the other algorithms (on the x-axis). Thus for crosses above the diagonal the chosen algorithm outperforms the CSP algorithm.

It can be observed that CSSSP has the best performance compared to all other algorithms. Nevertheless, this is not true for each dataset individually. High variations exist so that for some datasets CSP or one of the other algorithms outperforms CSSSP. CSSP shows very good behavior for some datasets, but not for others. Classification with MKL and LDA usually fails, since it focuses only on a very strict frequency range. But there are also datasets where they improve performance enormously. Again, a closer look reveals that this is usually the case if a fit to only one frequency range is advisable. However, the fixed setup with some rhythms seems to be a too strong restriction for good performance. Using more and smaller bands usually tends strongly to overfitting.

One should note that CSSSP usually does not have only one local maxima. There are

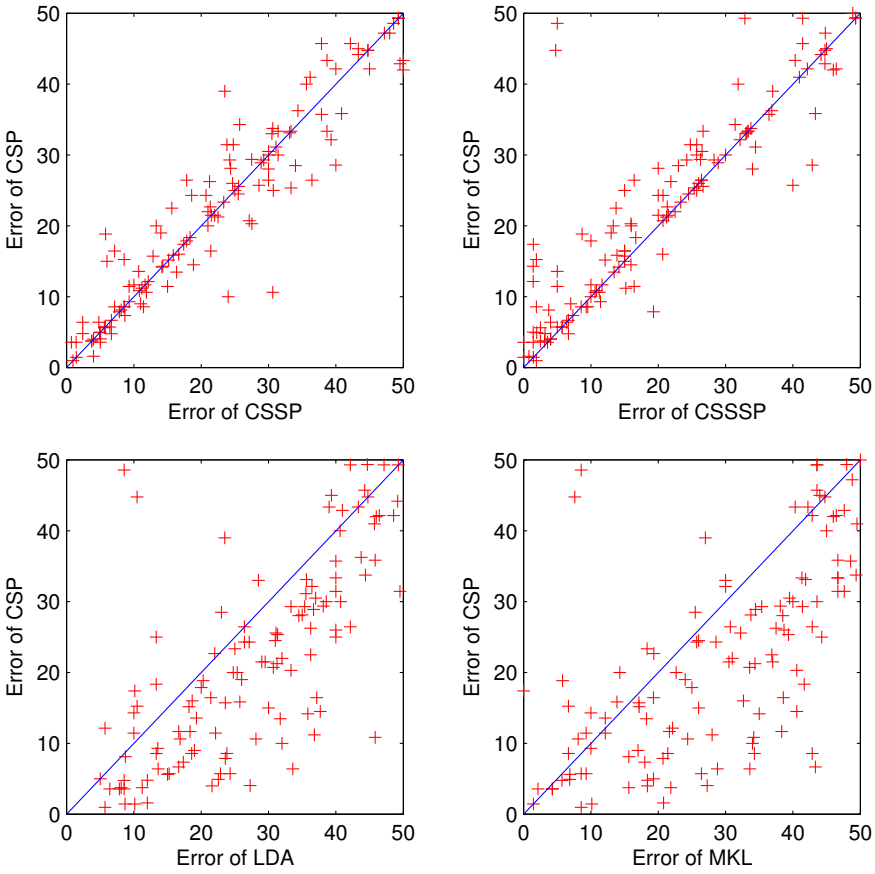


Figure 8.4: The figures compare the chronological validation error of the suggested algorithms on the x-axis against CSP on the x-axis. LDA resp. MKL refer to classification on several bands by LDA or MKL. Hereby points above the diagonal corresponds to dataset where the CSP algorithm could be enhanced.

datasets where at least two local maxima are found, one with a very good performance, the other not. However, validation on both maxima shows that the second best value sometimes should be preferred. Here a big problem can be observed: the CSP-value is not absolutely correlated to the real discriminability, since it ignores the inter-trial variance completely. But so far no extension of CSP is known which solves this problem. Consequently, I have not taken this problem into account, and have chosen the highest local (i.e., the global) maxima. A further enhancement can be expected if CSP can be modified to be more correlated to the original discriminability.

Furthermore there is a second point I should mention. Since I am using chronological validation here, I try to take care of the non-stationarity problem (see section 5.3), i.e., the test data could look different to the training set due to variation of brain activity over time. Thus it is possible that the algorithms find better or only more – over time – stable features. However, both cases would mean an enhancement for a BCI.

Altogether, one can conclude that there is no best method to extract the best spectral filter. One should decide individually on the best method. Nevertheless, a small advantage for

CSSSP is clearly visible, i.e., CSSSP outperforms the other methods significantly (by a Wilcoxon-Rank-Test with $p < 0.01$).
The choice of a suitable frequency band is so far done manually for our online system. Both in online systems and in this offline analysis the need of temporal filtering for the enhancement of the performance was observed. Consequently, the next step for our BBCI consists in automation of the training procedure – which is obviously necessary for further BCI applications – and thus of including the temporal filtering algorithm into this interface. Although it is not clear if it can outperform the manual choice, automation is one necessary requirement for further BCI systems and has to be done to enhance performance compared to usual CSP.

9 Summary

In this chapter I will summarize and discuss the results of this thesis. I mainly focus on my work but the relevance for our BBCI project and for BCI research in general will be briefly illuminated. Furthermore, I will give a brief outlook for further BCI directions.
As announced in the introduction, my contribution mainly consists of three parts:

- **Establishing a performance measure based on information theory:** The main criterion I use for the analysis in this work is the optimization of the transmittable information of the system. Theoretically this comprises the amount of decisions a user can choose, the rate at which decisions can be performed, and the accuracy by which a desired decision is really achieved. For the theoretically possible value for transmittable information, Shannon's information transfer rate, the human needs to be able to code the desired decisions in some arbitrary way. Although these assumptions lead to a theoretical solution, the meaning for the BCI situation is not clear because too complex codings might not be usable by a human. However, in chapter 3 I illuminate the question of ergonomic codings and find that one cannot achieve the theoretical bitrate, but that this theoretical value is not too bad an estimate for what is be possible with the interface. Thus the use of this theoretical value or, if the number of decisions and the decision rate is constant, the classification error is a suitable measure for further analysis and comparison of different algorithms.

- **Transfer and development of suitable signal processing and machine learning techniques:** Driven by the leitmotiv 'let the machines learn', I have developed or adapted several algorithms from the machine learning and signal processing community. Hereby it was possible to increase the performance of a BCI system considerably, which I have empirically shown for the following three points:

  ① *Combining different features:* Based on the neurophysiological knowledge of different features which accompany real or imagined movements in an uncorrelated form, I have developed and tested several algorithms. Besides state of the art methods, I also used some that I myself have developed. I have shown in a theoretical observation (see chapter 6) that performance can be enhanced if both features are used in a suitable way. Especially if they are uncorrelated and of similar single performance, the enhancement can be huge compared to the best single feature performance. In chapter 6 I have confirmed this theoretical insight by experimental results. Especially one method which I have developed outperforms all other existing methods which were used in similar situations in other application areas. Note that the presented approach was the first successful attempt for combining different features in the BCI literature. Based on these ideas many other groups were inspired to extract and combine different features too (see [16]).

② *Using more than two classes/mental states:* While the combination of different features aims for increasing classification performance, another option to enhance the bitrate of a BCI consists of using a suitable number of classes. In chapter 7 I have theoretically illuminated the question of how many classes one should use. I have found that with BCI performances reported so far the best choice is given by three or four classes, if a suitable discriminability for the specific subject exists. The gain for more than four classes is too small and only given under strong and unrealistic assumptions. I have successfully confirmed this theoretical insight in practice in chapter 7. Furthermore, I have successfully extended some algorithms working only on two class problems to multi-class versions which was important to achieve these results.

③ *Fitting spatio-temporal filters:* Based on another neurophysiological observation that the characteristics of different brain rhythms vary strongly in different subjects (see chapter 8), it is obvious that a suitable and subject-specific temporal filtering, i.e., the weighting of different frequency components in the signal, can enhance performance. This has to be done individually for each subject. Although one could do so manually by looking at suitable spectra plots, an automatic choice is more convenient for a useful BCI system. Several ways exist to do so and were introduced in chapter 8 based on one prominent and established BCI feature called Common Spatial Pattern (CSP). In this context, I have developed a new extension of the CSP algorithm which outperforms all other existing methods significantly and furthermore allows a neurophysiological interpretation.

➢ **Implementation of the BBCI and realization of suitable experiments:** All the introduced machine learning methods in the last point were evaluated on recorded data, which is often called offline analysis in the BCI context. However, the main goal of a BCI system is the realization of an online system, i.e., a computer directly interprets human brain signals to present feedback such that the subject can control a device. To do so I have implemented a suitable and flexible interface introduced in chapter 4. Here the easy exchange of different algorithms and approaches within the existing offline toolbox and the fast application were important considerations. Furthermore, small modifications should be easy to carry out and directly online. With the implemented interface several successful online applications were performed. The best subject was able to achieve up to 40 bits/minute within such a feedback and was able to write a sentence within minutes of his first experiments, i.e., without long periods of subject training. Furthermore, some first gaming applications were successfully established too. Nevertheless the BCI performance strongly varies between subjects. Besides some subjects who are able to control a BCI after a few minutes there are also a few subjects who develop no control abilities at all so far. Most subjects are in between these two extremes, i.e., they are able to control a BCI to some degree but mistakes happen regularly.

Many different approaches for implementing a BCI system exist. Many of them are very interesting and can be useful in some specific situations. In my opinion the most profitable way for a broad target group lies in combining neurophysiological knowledge and machine learning tools to have a good but possibly not perfect subject-specific system in the first step and in using further machine learning capabilities for adaptation and the ability of the subject to learn for the fine tuning of the system in the second step. This opinion is confirmed by

the results of this thesis, especially by chapter 4.
The described results in this thesis are important for the results of our group since it was possible to enhance existing methods and to establish a successful, working online system. Furthermore the results are of high interest to the BCI community because the BBCI system set new standards in terms of being ready to use after a calibration time of only 30 minutes, and of allowing high transmission rates (up to 40 bits/minute for untrained subjects). Other groups have started to use and adapt our ideas for their own systems (see [16]). Furthermore, the necessity of feedback to learn to control the device is helpful but not essential any longer.

9.1 Outlook

My future goals consist of transferring the algorithms which have proven to be successful in offline studies to online scenarios (e.g., feature combination, multi-class extensions (for 2d-feedback or for a rest-class)), finding other relevant features (e.g., the phase, see [85]) for BCI navigation (which could also help for finding solutions for subjects without any BCI control so far), online adaptation of the system during feedback, making BCI control independent of other mental states like fatigue, workload or lack of concentration and realization of movement predictions in online environments. Of course the ongoing research process will throw up many new interesting questions and research fields. Thus this list is not complete.
One big challenge for making BCI applications attractive for everyday life, which I cannot influence, is improvement of the acquisition technology. So far one needs one hour to prepare the cap (positioning and establishing of suitable conductance) before one can start with measuring EEGs. After some hours the electrode gel dries up making further recordings impossible. At this point other measurement techniques should be one important goal. Ideally one should be able to use the EEG cap like a baseball cap or a bicycle helmet without further preparation. Without solving this problem BCI can only be an interesting application for disabled patients, especially for those without any other communication capabilities or in special situations when the effort to prepare the cap is reasonable, e.g., studies on usability or psychology based on neurophysiological phenomena. But BCI would not become an application for healthy subjects in everyday life. However, if the problem of data acquisition is solved conveniently and if the system achieves a suitable accuracy the value of a BCI for human life could be huge and would open the field for many different applications. One big application field directly springs to mind: BCI would open an absolutely new way of playing games. Due to this novelty effect a big industry could be established and thus the price for the system could be become affordable for everyone.
But besides the gaming application there could be many other interesting real-word applications. The use of the EEG signal as an early decision instance could decrease reaction time of a user or let the system take emergency measures in extreme situations. For example, if a car can detect an emergency brake 100 ms earlier than it actually happens, the car could interrupt further acceleration shortly before or tighten the belt which could help to make driving safer.
A BCI could becomes attractive as an additional communication channel if a user can control the device on the side: giving full attention to a BCI without using other communication channels does not seem to be convenient for real-life applications. However, if one could

use BCI in addition to other channels the value would be huge.
Another interesting application of EEG-based online systems does not directly lie in an active interface, i.e., a system for controlling a device. One could also use the EEG to observe and classify mental states of a user. If a system could have information about the vigilance, concentration, mental workload or emotional state of a human user, it could render the working environment more ergonomic. For example, if a system could detect the workload of a subject other tasks could be suitably placed to improve human performance and satisfaction, i.e., during phases with less workload more tasks could be provided but during phases with high workload tasks could be delayed until less workload is detected (see [48]). For vigilance, concentration or emotions similar task management strategies could be introduced. With emotions one would also touch the field of usability. With the help of an EEG it could be possible to decide if the user of a machine understands or likes interacting with the system.
In my opinion, there are several problems for an everyday BCI which need to and can be solved. This goal will not be achieved during the next 2 or 3 years. But on a longer timescale BCI could become an important tool in everyday life.

A Appendix

In this part I will proof main statements of this work and collect the formula for the coding strategies in chapter 3.

A.1 Proof of theorem 3.3.1

Since the summand is positive it is

$$
\begin{aligned}
& \sum_{n=1}^{\infty} \sum_{(2k,j_1,\ldots,j_q)=n-1} \binom{n-1}{2k,j_1,\ldots,j_q} c_k a^{k+1} b^k h_1^{j_1} \cdot \ldots \cdot h_q^{j_q} \\
= & \sum_{n=1}^{\infty} \sum_{k=0}^{\lfloor \frac{n-1}{2} \rfloor} \sum_{(j_1,\ldots,j_q)=n-1-2k} \binom{n-1}{n-2k-1} \binom{n-2k-1}{j_1,\ldots,j_q} c_k a^{k+1} b^k h_1^{j_1} \cdot \ldots \cdot h_q^{j_q} \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k \sum_{n=2k+1}^{\infty} \binom{n-1}{n-2k-1} \sum_{(j_1,\ldots,j_q)=n-1-2k} \binom{n-1-2k}{j_1,\ldots,j_q} h_1^{j_1} \cdot \ldots \cdot h_q^{j_q} \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k \sum_{n=2k+1}^{\infty} \binom{n-1}{2k} (h_1 + \ldots + h_q)^{n-2k-1} \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k \sum_{n=2k}^{\infty} \binom{n}{2k} (h_1 + \ldots + h_q)^{n-2k} \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k \frac{1}{(1-h_1-\ldots-h_q)^{2k+1}} \\
= & \frac{a}{1-h_1-\ldots-h_q} \sum_{k=0}^{\infty} c_k \left(\frac{ab}{(1-h_1-\ldots-h_q)^2} \right)^k \\
\underset{[33]}{=} & \frac{1}{2b} (1-h_1-\ldots-h_q - \sqrt{(1-h_1-\ldots h_q)^2 - 4ab}) \\
\underset{a+b+h_1+\ldots+h_q=1}{=} & \frac{1}{2b} \left[a+b-\sqrt{(a-b)^2} \right] \\
= & \begin{cases} 1 & a \geq b \\ \frac{a}{b} & a < b \end{cases} .
\end{aligned}
$$

For the second equation for $a > b$ it holds true that

$$
\begin{aligned}
& \sum_{n=1}^{\infty} \sum_{(2k,j_1,\ldots,j_q)=n-1} \binom{n-1}{2k,j_1,\ldots,j_q} c_k a^{k+1} b^k h_1^{j_1} \cdot \ldots \cdot h_q^{j_q} (k+1) \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k (k+1) \sum_{n=2k+1}^{\infty} \binom{n-1}{2k} \sum_{(j_1,\ldots,j_q)=n-1-2k} \binom{n-2k-1}{j_1,\ldots,j_q} h_1^{j_1} \cdot \ldots \cdot h_q^{j_q} \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k (k+1) \sum_{n=2k+1}^{\infty} \binom{n-1}{2k} (h_1+\ldots+h_q)^{n-2k-1} \\
= & \frac{a}{1-h_1-\ldots-h_q} \sum_{k=0}^{\infty} c_k (k+1) \left(\frac{ab}{(1-h_1+\ldots+h_q)^2} \right)^k \\
= & \frac{a}{1-h_1-\ldots-h_q} \left[\frac{\partial}{\partial z} z \sum_{k=0}^{\infty} c_k z^k \right]_{z=\frac{ab}{(1-h_1-\ldots-h_q)^2}} \\
\underset{[33]}{=} & \frac{a}{1-h_1-\ldots-h_q} \left[\frac{1}{\sqrt{1-4z}} \right]_{z=\frac{ab}{(1-h_1-\ldots-h_q)^2}} \\
\underset{a+b+h_1+\ldots+h_q=1}{=} & \frac{a}{\sqrt{(a+b)^2-4ab}} \\
\underset{a>b}{=} & \frac{a}{a-b}
\end{aligned}
$$

and

$$
\begin{aligned}
& \sum_{n=1}^{\infty} \sum_{(2k,j_1,\ldots,j_q)=n-1} \binom{n-1}{2k,j_1,\ldots,j_q} c_k a^{k+1} b^k h_1^{j_1} \cdot \ldots \cdot h_q^{j_q} j_1 \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k \sum_{n=2k+1}^{\infty} \binom{n-1}{2k} \sum_{(j_1,\ldots,j_q)=n-1-2k} \binom{n-2k-1}{j_1,\ldots,j_q} j_1 h_1^{j_1} \cdot \ldots \cdot h_q^{j_q} \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k \sum_{n=2k+1}^{\infty} \binom{n-1}{2k} h_1 \left[\frac{\partial}{\partial z} \left(\sum_{(j_1,\ldots,j_q)=n-1-2k} \binom{n-2k-1}{j_1,\ldots,j_q} z^{j_1} h_2^{j_2} \cdot \ldots \cdot h_q^{j_q} \right) \right]_{z=h_1} \\
= & \sum_{k=0}^{\infty} c_k a^{k+1} b^k \sum_{n=2k+1}^{\infty} \binom{n-1}{2k} h_1 \left[\frac{\partial}{\partial z} \left((z+h_2+\ldots+h_q)^{n-2k-1} \right) \right]_{z=h_1} \\
= & h_1 \sum_{k=0}^{\infty} c_k a^{k+1} b^k \sum_{n=2k+1}^{\infty} \binom{n-1}{2k} (n-2k-1)(h_1+\ldots+h_q)^{n-2k-2} \\
= & h_1 \sum_{k=0}^{\infty} c_k a^{k+1} b^k \left[\frac{\partial}{\partial z} \left(\sum_{n=2k}^{\infty} \binom{n}{2k} z^{n-2k} \right) \right]_{z=h_1+\ldots+h_q} \\
= & h_1 \sum_{k=0}^{\infty} c_k a^{k+1} b^k \left[\frac{\partial}{\partial z} \left(\left(\frac{1}{1-z} \right)^{2k+1} \right) \right]_{z=h_1+\ldots+h_q} \\
= & h_1 \sum_{k=0}^{\infty} c_k a^{k+1} b^k \frac{2k+1}{(1-h_1-\ldots-h_q)^{2k+2}} \\
= & \frac{2ah_1}{(1-h_1-\ldots-h_q)^2} \sum_{k=0}^{\infty} c_k \left(\frac{ab}{(1-h_1-\ldots-h_q)^2} \right)^k (k+1) - \frac{ah_1}{(1-h_1-\ldots-h_q)^2} \sum_{k=0}^{\infty} c_k \left(\frac{ab}{(1-h_1-\ldots-h_q)^2} \right)^k \\
\underset{[33]}{=} & \frac{2ah_1}{(1-h_1-\ldots-h_q)^2} \left[\frac{\partial}{\partial z} z \frac{1-\sqrt{1-4z}}{2z} \right]_{z=\frac{ab}{(1-h_1-\ldots-h_q)^2}} - \frac{ah_1}{(1-h_1-\ldots-h_q)^2} \frac{1-\sqrt{1-4\frac{ab}{(1-h_1-\ldots-h_q)^2}}}{2\frac{ab}{(1-h_1-\ldots-h_q)^2}} \\
\underset{a+b+h_1+\ldots+h_q=1}{=} & \frac{h_1}{a-b}.
\end{aligned}
$$

The second formula in the lemma can now be easily calculated by these results. □

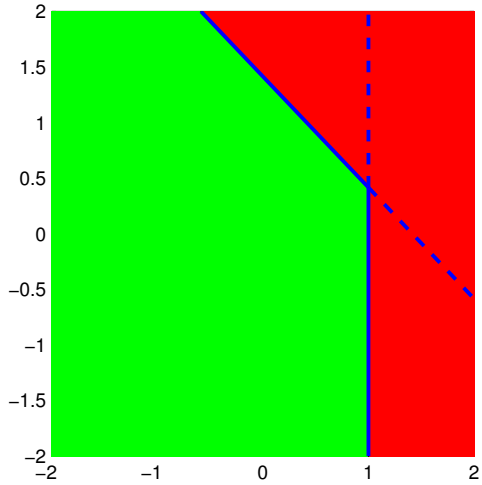


Figure A.1: The figure shows the area $\mu_k^\top x + c_k \leq 0$ for $k = 1,2$ with $n = 2$, $K = 2$, $\mu_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\mu_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $c_1 = c_2 = -1$ in green as required in lemma A.3.1. The other area is colored red. Furthermore the hyperplanes are marked.

A.2 Proof of theorem 5.2.1

Fubini's theorem shows directly that the optimal $\bar{f}$ has to satisfy $\bar{f}(x) = \text{argmax}_y P^{Y=y|X=x}$ (resp. $\bar{f}(x) = \text{sign}\left(P^{Y=1|X=x} - P^{Y=-1|X=x}\right)$) almost surely which is equivalent to $\bar{f}(x) = \text{argmax}_y P^{X=x|Y=y} P(Y = y)$ (resp. $\bar{f}(x) = \text{sign}\left(P^{X=x|Y=1} P(Y=1) - P^{X=x|Y=-1} P(Y=-1)\right)$) almost surely by Bayes rule. A short calculation completes the proof. The binary formulations can be directly derived by this result. □

A.3 Proof of theorem 5.2.3

Before one is able to prove this theorem one needs the following lemma:

A.3.1 Lemma: *Consider an m-dimensional random vector $X \sim \mathcal{N}(0, I)$ with I as identity matrix. Furthermore $\mu_1, \ldots, \mu_N \in \mathbb{R}^m$ and $c_1, \ldots, c_N \in \mathbb{R}$ with $N \geq 1$ are given. Then it holds true that*

$$P^X(x \in \mathbb{R}^m | \forall_{k=1,\ldots,N}\ \mu_k^\top x + c_k \leq 0) \geq 1 - \sum_{k=1}^{N} \text{erf}\left(\frac{c_k}{\sqrt{\mu_k^\top \mu_k}}\right).$$

Especially for $N = 1$:

$$P^X(x \in \mathbb{R}^m | \mu_1^\top x + c_1 \leq 0) = 1 - \text{erf}\left(\frac{c_1}{\sqrt{\mu_1^\top \mu_1}}\right).$$

In Fig. A.1 the situation of this lemma is exemplarily shown for $m = 2$, $N = 2$, $\mu_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, $\mu_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $c_1 = c_2 = -1$.

Proof:

Let us define $Z_k := \mu_k^\top X + c_k$ for $k = 1,...,N$. Z_k is a random variable with $Z_k \sim \mathcal{N}(c_k, \mu_k^\top \mu_k)$ (see [93]). Then the following holds true for all $k = 1,...,N$:

$$\begin{aligned} P^X(\mu_k^\top x + c_k > 0) &= P^{Z_k}(z > 0) \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi}\sqrt{\mu_k^\top \mu_k}} \exp\left(\frac{-(z-c_k)^2}{2\mu_k^\top \mu_k}\right) \mathrm{dz} \\ &= \int_{-\frac{c_k}{\sqrt{\mu_k^\top \mu_k}}}^\infty \frac{1}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}) \mathrm{dz} \\ &= \mathrm{erf}\left(\frac{c_k}{\sqrt{\mu_k^\top \mu_k}}\right). \end{aligned}$$

Thus

$$\begin{aligned} P^X(x \in \mathbb{R}^m | \forall_{k=1,...,N}\ \mu_k^\top x + c_k \leq 0) &= 1 - P^X(x \in \mathbb{R}^m | \exists_{k=1,...,K}\ \mu_k^\top x + c_k > 0) \\ &\geq 1 - \sum_{k=1}^N P^X(x \in \mathbb{R}^m | \mu_k^\top x + c_k > 0) \\ &= 1 - \sum_{k=1}^N \mathrm{erf}\left(\frac{c_k}{\sqrt{\mu_k^\top \mu_k}}\right) \end{aligned}$$

and for $N = 1$ the $\geq$ is obviously an $=$. □

Now one is able to prove the theorem 5.2.3 as follows:
The first step consists of calculating for $y_1 = 1,...,N$ the value of $P^{X|Y=y_1}(\bar{f}(x) = y_1)$. Remember that $X|Y = y_1 \sim \mathcal{N}(\mu_{y_1}, \Sigma)$. After mapping $X \mapsto \Sigma^{-1}(X - \mu_{y,1})$ it can be assumed that $X|Y = y_2 \sim \mathcal{N}(\Sigma^{-0.5}(\mu_{y_2} - \mu_{y_1}), I)$ for all $y_2 = 1,...,N$. Let $\hat{\mu}_{y_2} := \Sigma^{-0.5}(\mu_{y_2} - \mu_{y_1})$ and note that $\hat{\mu}_{y_1} = 0$. With lemma A.3.1 and corollary 5.2.2 this leads to

$$\begin{aligned} & P^{X|Y=y_1}(\bar{f}(x) = y_1) \\ \underset{5.2.2}{=} & P^{X|Y=y_1}\left(y_1 = \mathrm{argmax}_{y_2}(\hat{\mu}_{y_2}^\top x - 0.5\hat{\mu}_{y_2}^\top \hat{\mu}_{y_2} + \log(P(Y = y_2)))\right) \\ \underset{\hat{\mu}_{y_1}=0}{=} & P^{X|Y=y_1}\left(\forall_{y_2 \neq y_1} \hat{\mu}_{y_2}^\top x - 0.5\hat{\mu}_{y_2}^\top \hat{\mu}_{y_2} + \log(P(Y = y_2)) - \log(P(Y = y_1)) \leq 0\right) \\ \underset{A.3.1}{\geq} & 1 - \sum_{y_2 \neq y_1} \mathrm{erf}\left(\frac{-0.5\hat{\mu}_{y_2}^\top \hat{\mu}_{y_2} + \log(P(Y = y_2)) - \log(P(Y = y_1))}{\sqrt{\hat{\mu}_{y_2}^\top \hat{\mu}_{y_2}}}\right) \\ = & 1 - \sum_{y_2 \neq y_1} \mathrm{erf}\left(\frac{-0.5(\mu_{y_2} - \mu_{y_1})\Sigma^{-1}(\mu_{y_2} - \mu_{y_1}) + \log(P(Y = y_2)) - \log(P(Y = y_1))}{\sqrt{(\mu_{y_2} - \mu_{y_1})\Sigma^{-1}(\mu_{y_2} - \mu_{y_1})}}\right). \end{aligned}$$

Thus

$$\begin{aligned} & E(\bar{f}(X)=Y) \\ = & \sum_{y_1=1,\ldots,N} P(Y=y_1) P^{X|Y=y_1}(\bar{f}(X)=y_1) \\ \geq & 1-\sum_{y_1=1,\ldots,N} P(Y=y_1) \sum_{y_2\neq y_1} \operatorname{erf}\left(\frac{-0.5(\mu_{y_2}-\mu_{y_1})\Sigma^{-1}(\mu_{y_2}-\mu_{y_1}) + \log(P(Y=y_2)) - \log(P(Y=y_1))}{\sqrt{(\mu_{y_2}-\mu_{y_1})\Sigma^{-1}(\mu_{y_2}-\mu_{y_1})}} \right). \end{aligned}$$

Assuming $P(Y=y)=\frac{1}{N}$ this results in

$$\begin{aligned} E(\bar{f}(X)=Y) & \geq 1-\frac{1}{N}\sum_{y_1=1,\ldots,N}\sum_{y_2\neq y_1} \operatorname{erf}\left(\frac{-0.5(\mu_{y_2}-\mu_{y_1})\Sigma^{-1}(\mu_{y_2}-\mu_{y_1})}{\sqrt{(\mu_{y_2}-\mu_{y_1})\Sigma^{-1}(\mu_{y_2}-\mu_{y_1})}} \right) \\ & = 1-\frac{1}{N}\sum_{y_1=1,\ldots,N}\sum_{y_2\neq y_1} \operatorname{erf}\left(-0.5\sqrt{(\mu_{y_2}-\mu_{y_1})\Sigma^{-1}(\mu_{y_2}-\mu_{y_1})} \right). \end{aligned}$$

Since lemma A.3.1 equality holds here for $N=2$ which leads under the additional assumption $\mu_2=-\mu_1$ to

$$E(\bar{f}(X)=Y) = \operatorname{erf}\left(\sqrt{\mu_1\Sigma^{-1}\mu_1}\right).$$

□

A.4 Proof of the statement in chapter 7

A.4.1 Theorem: *Consider a classification problem (X,Y) with $X\in\mathbb{R}^m$ and $Y\in\{1,2,3\}$. Suppose $X|Y=y\sim\mathcal{N}(\mu_y,\Sigma)$ and $P(Y=y)=\frac{1}{3}$ for $y=1,2,3$. Let $\tilde{f}_y(x)=\mu_y^\top\Sigma^{-1}x-\frac{1}{2}\mu_y^\top\Sigma^{-1}\mu_y$ for $y=1,2,3$ the optimal classifier based on corollary 5.2.2. Furthermore $g_{y_1,y_2}:=\tilde{f}_{y_1}-\tilde{f}_{y_2}$ is the optimal classifier based on the same corollary on the two-class problem $(y_1,y_2)\in\{1,2,3\}$, $y_1\neq y_2$. Assume that all pairwise classification accuracies are equal, i.e., $acc_{g_{y_1,y_2}}=acc$ for all $y_1,y_2=1,2,3$, $y_1\neq y_2$. Then*

$$acc-\frac{\exp(-\frac{2(erf^{-1}(acc))^2}{3})}{6}\leq acc_{\tilde{f}}\leq acc-\frac{\exp(-\frac{(erf^{-1}(acc))^2}{2})}{6}.$$

Scaling, rotating and shifting appropriately, one can directly assume that $\Sigma=I$, $m=2$ and $\mu_1=0$. Since $\mathrm{acc}=\mathrm{acc}_{g_{1,y}}=\operatorname{erf}\left(0.5\sqrt{\mu_y^\top\mu_y}\right)$ (see theorem 5.2.3) for $y=2,3$ one gets $\mu_y^\top\mu_y=\left(2\mathrm{erf}^{-1}(\mathrm{acc})\right)^2=:\rho^2$ with $\rho>0$. Rotating appropriately one gets $\mu_2=(\rho,0)^\top$ and $\mu_3=(\frac{\rho}{2},\frac{\rho}{2}\sqrt{3})$ (since the distance between μ_2 and μ_3 has to be ρ, too.). This situation together with the optimal binary classifiers (blue lines) for class 1 against 2 (vertical line) resp. 3 (sloping line) based on corollary 5.2.2 are shown in Fig. A.2. Furthermore the connections lines between the means are shown in green. If a trial comes from class 1 it is classified wrongly if it is right of the vertical or above the sloping blue line. Since evaluation of probabilities for polyhedrons in the Gaussian space is difficult, I only estimate lower and

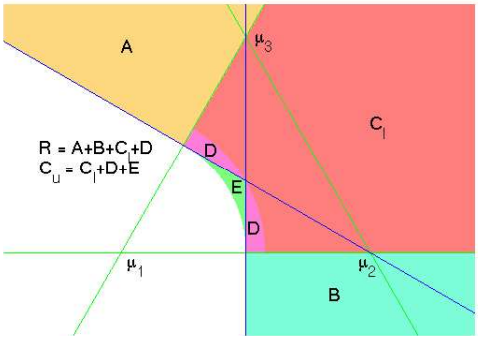


Figure A.2: The figure visualizes a method to estimate bounds for the ITR depending on the expected pairwise misclassification risk for three classes.

upper bounds. To do so I define the following sets (also visualized in Fig. A.2):

$$\begin{aligned}
A &:= \{x \in \mathbb{R}^2 \mid \mu_3^\top x > \rho^2/2 \wedge arg(x) > \pi/3\} \\
B &:= \{x \in \mathbb{R}^2 \mid \mu_2^\top x > \rho^2/2 \wedge arg(x) < 0\} \\
C_l &:= \{x \in \mathbb{R}^2 \mid \|x\|_2 > \rho/\sqrt{3} \wedge arg(x) \in [0, \pi/3]\} \\
C_u &:= C_l + D + E = \{x \in \mathbb{R}^2 \mid \|x\|_2 > \rho/2 \wedge arg(x) \in [0, \pi/3]\} \\
R &:= A + B + C_l + D = \{x \in \mathbb{R}^2 \text{ is classified as class 2 or class 3}\}
\end{aligned}$$

If $x = re^{i\phi}$ in the unique polar coordinates representation, in this situation $arg : \mathbb{R}^2 \to [-\pi, \pi)$ is defined by $arg(x) = \phi$.
To calculate the probability that a trial of class 1 is classified wrongly as class 2 or class 3 one has to calculate $P(R)$. Since $A \cup B \cup C_l \subset R \subset A \cup B \cup C_u$ this can be done by using symmetry and polar coordinates transformation to get

$$\begin{aligned}
P(A) &= 0.5 - 0.5\text{erf}\left(\frac{\rho}{2}\right) = 0.5 - 0.5\text{acc} \\
P(B) &= 0.5 - 0.5\text{erf}\left(\frac{\rho}{2}\right) = 0.5 - 0.5\text{acc} \\
P(C_u) &= \frac{1}{6}\exp(-\frac{\rho}{8}) \\
P(C_l) &= \frac{1}{6}\exp(-\frac{\rho}{6})
\end{aligned}$$

Consequently one gets for $\text{acc}_{\tilde{f}} = 1 - P(R)$ (one can get the same results for the other classes due to symmetry and equal class priors)

$$\frac{\exp(-\frac{\rho^2}{6})}{6} \leq \text{acc} - \text{acc}_{\tilde{f}} \leq \frac{\exp(-\frac{\rho^2}{8})}{6}$$

which directly leads to the statememt of the theorem. □

A.5 Overview of all formulas for the coding strategies of chapter 3

Coding	Convergence almost surely guaranteed if	Expected number of steps to achieve a decision
(**ST**)	$p_\delta > \frac{1}{2}$ and $\bar{p} > \bar{p}_\delta$	$\frac{p_\delta d + p_\delta d_\delta - \bar{p} p_\delta d_\delta - \bar{p} d + \bar{p} p_\delta d - p_\delta \bar{p}_\delta d + p_\delta \bar{p}_\delta d_\delta - \bar{p}_\delta d_\delta}{(\bar{p} - \bar{p}_\delta)(2p_\delta - 1)}$
(**CF1**)	$p_\delta + p_c > 1$ and $\bar{p} p_c > \bar{p}_\delta (1 - p_c)$	$\frac{(\bar{d}+1)(1-\bar{p}_\delta) + (d_\delta + 1)\bar{p}_\delta}{\bar{p} p_c - \bar{p}_\delta (1-p_c)} + \frac{\big((d_\delta+1)p_\delta + (\bar{d}+1)(1-p_\delta)\big)(1-\bar{p}-\bar{p}_\delta)(1-p_c)}{(p_\delta + p_c - 1)(\bar{p} p_c - \bar{p}_\delta (1-p_c))}$
(**CF3**)	$p_c p^s \geq 1 - p_c$ and $p_c \geq 0.5$	$\frac{\bar{d}}{s}\left(\frac{s+1}{2p_c - 1} + \frac{(s+2)p_c(1-p^s)}{(2p_c-1)(p_c + p_c p^s - 1)}\right)$
(**OB1**)	$p > 0.5$	$\frac{\bar{d}}{2p-1}$

Table A.1: The table shows the requirements for the variables for the codings introduced in chapter 3 so that convergence is guaranteed and the expected number of steps to achieve a decision. The solutions for (**CF2**) and (**OB2**) are skipped here because they are very long and go beyond the scope of this work.

Notations

C	(context dependent:) regularisation constant
C	(context dependent:) capacity of a channel
E_δ	expected number of steps to delete a made decision if desired (recognizing deletion of further wrong made decisions during this process)
E_i	expected number of steps to achieve decision i if desired (recognizing deletion of further wrong made decisions during this process)
$\bar{E}$	averaged expected number of steps to achieve a decision except δ (recognizing deletion of further wrong made decisions during this process)
F	number of features
$H_0(\mathscr{A})$	raw bit content of the finite set $\mathscr{A}$
$H_\delta(\mathscr{A})$	essential bit content of the finite set $\mathscr{A}$ for $\delta \geq 0$
$H(\mathscr{A})$	the entropy of the finite set $\mathscr{A}$
$H(X)$	entropy of the alphabet and the probability distribution described by a random vector X
I	Identity matrix
$I(X;Y)$	mutual information between X and Y ($I(X;Y) = H(X) - H(X\|Y)$)
N	number of classes/ discriminable mental states
M	number of decisions
P	probability distribution
P_δ	probability for a successful run to δ (recognizing deletion of further wrong made decisions during this process)
P_i	probability for a successful run to i (recognizing deletion of further wrong made decisions during this process)
$\bar{P}$	averaged probability for a successful run to an arbitrary decision except δ (recognizing deletion of further wrong made decisions during this process)
R	rate of a transmission
$S_\delta(\mathscr{A})$	on arbitrary example of all smallest subsets of $\mathscr{A}$ with $P(x \in S_\delta) = P(x \in S_\delta(\mathscr{A})) \geq 1 - \delta$
X	random variable describing data
Y	random variable describing the class label
a	(context dependent:) constant for lemma 3.3.1
a	(context dependent:) parameter vector of an IIR filter
$\mathbf{acc}_f$	accuracy of some classifier f
$\arg(x)$	for a two dimensional vector $x = r\exp(i\phi)$ the argument ϕ
b	(context dependent:) constant for lemma 3.3.1
b	(context dependent:) bias in classification

b	(context dependent:) parameter vector of an IIR filter, also used for CSSSP
c_n	Catalan numbers
d_δ	depth of decision δ in a tree
d_i	depth of decision i in a tree
$\bar{d}$	averaged depth of all decisions except δ in a tree
erf	$\text{erf} : \mathbb{R} \to [0,1], z \mapsto \int_{-inf}^{z} \frac{1}{\sqrt{2\pi}} \exp(-0.5x^2)dx$
f_i	prior distribution over all possible decisions
h	VC-dimension
$h_1, .., h_q$	constants for lemma 3.3.1
m	dimension of the feature space
n	number of trials
$p_.$	probability
p_c	probability to correctly answer a question
p_δ	probability to achieve δ if desired
$p_{i,j}$	probability to achieve decision i if decision j is desired
p_i	(in the context of coding trees:) $= p(i,i)$
$\bar{p}$	averaged probability to achieve a decision except δ if desired
$\bar{p}_\delta$	averaged probability to achieve δ if another decision is desired
q	constant for lemma 3.3.1
s_i	i-th EEG trial, usually a matrix with number of channels as rows, and number of time-points as columns
s_i^τ	the by τ timepoints delayed trial s_i
s	(for (**CF3**) and (**OB1**)) number of steps to the next confirmation question
t	timepoint
$t_.$	depth of a leaf in a tree
$var(\cdot)$	variance of the finite sequence in brackets
w	(context dependent:) spatial filter
w	(context dependent:) linear classifier
x_i	i-th realization of a random variable X
y_i	i-th realization of a random variable Y
$\mathscr{A}$	finite alphabet
$\mathscr{F}$	Feature space
$\mathscr{N}(\mu, \Sigma)$	normal distribution with mean μ and covariance Σ
$\mathbb{N}_0$	the space non-negative integers $\{0,1,2,3,...\}$
$\mathbb{R}$	the space of real numbers
$\Sigma_.$	covariance matrix
α	occipital brain rhythm at around 7–13 Hz
β	central brain rhythm at around 15–25 Hz
δ	(context dependent:) central brain rhythm at around 0.5–3 Hz
δ	(context dependent:) deletion symbol in a tree
θ	central brain rhythm at around 3–7 Hz
ϑ_i	i-th element of a catalan sequence
λ	regularisation constant
μ	central brain rhythm at around 7–13 Hz

ξ	slack variables
$\sigma_{\cdot}^2$	one-dimensional value denoting variance of a gaussian distribution
ϕ	mapping in some feature space
χ^2	χ^2-distribution
$\#A$	the number of elements of A, if A is a finite set
$\|x\|_1$	absolute norm of $x = (x_1, ..., x_n)$, $\|x\|_1 = \sum_i^n \|x_i\|$
$\|x\|_2$	euclidean norm of $x = (x_1, ..., x_n)$, $\|x\|_2 = \sum_i^n (x_i)^2$

Index

List of Figures

List of Tables

Bibliography

[1] C. Anderson. Taxonomy of feature extraction and translation methods for bci (web page), 2005. URL `http://www.cs.colostate.edu/eeg/taxonomy.html`.

[2] C. Babiloni, F. Carducci, F. Cincotti, P. M. Rossini, C. Neuper, G. Pfurtscheller, and F. Babiloni. Human movement-related potentials vs desynchronization of EEG alpha rhythm: A high-resolution EEG study. *NeuroImage*, 10:658–665, 1999.

[3] F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[4] F. Bach, G. Lanckriet, and M. Jordan. Fast kernel learning using sequential minimal optimization. Technical report, Division of Computer Science, University of California, Berkeley, 2004.

[5] F. Bach, R. Thibaux, and M. Jordan. Computing regularization paths for learning multiple kernels. In *Advances in Neural Information Processing Systems (NIPS)*, volume 17, 2004.

[6] R. Beisteiner, P. Hollinger, G. Lindinger, W. Lang, and A. Berthoz. Mental representations of movements. Brain potentials associated with imagination of hand movements. *Electroencephalography and Clinical Neurophysiology*, 96(2):183–193, 1995.

[7] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.

[8] A. Belouchrani, K. A. Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique based on second order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, 1997.

[9] K. Bennett and O. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[10] H. Berger. Über das Elektroenkephalogramm des Menschen. *Arch. Psychiat. Nervenkr.*, 99(6):555–574, 1933.

[11] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor. A spelling device for the paralysed. *Nature*, 398:297–298, 1999.

[12] N. Birbaumer, A. Kübler, N. Ghanayim, T. Hinterberger, J. Perelmouter, J. Kaiser, I. Iversen, B. Kotchoubey, N. Neumann, and H. Flor. The though translation device (TTD) for completly paralyzed patients. *IEEE Transactions on Rehabilitation Engineering*, 8(2):190–193, June 2000.

[13] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[14] G. Blanchard and B. Blankertz. BCI competition 2003 – data set IIa: Spatial patterns of self-controlled brain rhythm modulations. *IEEE Transactions on Biomedical Engineering*, 51(6):1062–1066, 2004.

[15] B. Blankertz. BCI Competition 2003 (web page), 2003. URL `http://ida.first.fhg.de/projects/bci/competition/`.

[16] B. Blankertz. BCI Competition III (web page), 2004. URL `http://ida.first.fhg.de/projects/bci/competition_iii/`.

[17] B. Blankertz, G. Curio, and K.-R. Müller. Classifying single trial EEG: Towards brain computer interfacing. In T. G. Diettrich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Inf. Proc. Systems (NIPS 01)*, volume 14, pages 157–164, 2002.

[18] B. Blankertz, C. Schäfer, G. Dornhege, and G. Curio. Single trial detection of EEG error potentials: A tool for increasing BCI transmission rates. In *Artificial Neural Networks – ICANN 2002*, pages 1137–1143, 2002.

[19] B. Blankertz, G. Dornhege, C. Schäfer, R. Krepki, J. Kohlmorgen, K.-R. Müller, V. Kunzmann, F. Losch, and G. Curio. Boosting bit rates and error detection for the classification of fast-paced motor commands based on single-trial EEG analysis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):127–131, 2003.

[20] B. Blankertz, K.-R. Müller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, and N. Birbaumer. The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering*, 51 (6):1044–1051, 2004.

[21] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio. The Berlin Brain-Computer Interface: Report from the feedback sessions. Technical Report 1, Fraunhofer FIRST, 2005.

[22] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, and G. Curio. The non-invasive berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects. *NeuroImage*, 2006. submitted.

[23] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Müller, V. Kunzmann, F. Losch, and G. Curio. The berlin brain-computer interface: EEG-based communication without subject training. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2006. submitted.

[24] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

[25] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition, 1996.

[26] M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. *Proceedings of the National Academy of Sciences*, 97(1): 262–267, 2000.

[27] C. Campbell and K. Bennett. A linear programming approach to novelty detection. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13, pages 395–401. MIT Press, 2001.

[28] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non gaussian signals. *IEE Proceedings-F.*, 140(6):362–370, 1993.

[29] J.-F. Cardoso and A. Souloumiac. Jacobi angles for simultaneous diagonalization. *SIAM J.Mat.Anal.Appl.*, 17(1):161 ff., 1996.

[30] J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 1996.

[31] M. Cheng, X. Gao, S. Gao, and D. Xu. Design and implementation of a brain-computer interface with high transfer rates. *IEEE Transactions on Biomedical Engineering*, 49(10):1181–1186, 2002.

[32] M. D. Comerchero and J. Polich. P3a and P3b from typical auditory and visual stimuli. *Clinical Neurophysiology*, 110(1):24–30, 1999.

[33] J. H. Conway and R. K. Guy. *The Book of Numbers*. Springer-Verlag, 1996.

[34] E. Courchesne, S. A. Hillyard, and R. Galambos. Stimulus novelty, task relevance and the visual evoked potential in man. *Electroencephalography and Clinical Neurophysiology*, 39(2):131–143, 1975.

[35] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley, 1991.

[36] R. Q. Cui, D. Huter, W. Lang, and L. Deecke. Neuroimage of voluntary movement: topography of the Bereitschaftspotential, a 64-channel DC current source density study. *Neuroimage*, 9(1):124–134, 1999.

[37] F. H. da Silva, T. H. van Lierop, C. F. Schrijer, and W. S. van Leeuwen. Organization of thalamic and cortical alpha rhythm: Spectra and coherences. *Electroencephalography and Clinical Neurophysiology*, 35:627–640, 1973.

[38] L. Deecke, B. Grozinger, and H. H. Kronhuber. Voluntary finger movement in man: cerebral potentials and theory. *Biological Cybernetics*, 23(2):99–119, 1976.

[39] E. Donchin, K. M. Spencer, and R. Wijesinghe. The mental prosthesis: Assessing the speed of a P300-based brain-computer interface. *IEEE Transactions on Rehabilitation Engineering*, 8(2):174–179, June 2000.

[40] J. P. Donoghue and J. N. Sanes. Motor areas of the cerebral cortex. *Journal of Clinical Neurophysiology*, 11:382–396, 1994.

[41] G. Dornhege. Charakterisierung von Räumen Maaßscher Spitzenformen. Master's thesis, Westfälische Wilhelms Universität Münster, 2001.

[42] G. Dornhege, B. Blankertz, and G. Curio. Speeding up classification of multi-channel brain-computer interfaces: Common spatial patterns for slow cortical potentials. In *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering. Capri 2003*, pages 591–594, 2003.

[43] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller. Combining features for BCI. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Inf. Proc. Systems (NIPS 02)*, volume 15, pages 1115–1122, 2003.

[44] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller. Boosting bit rates in non-invasive EEG single-trial classifications by feature combination and multi-class paradigms. *IEEE Transactions on Biomedical Engineering*, 51(6):993–1002, June 2004.

[45] G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller. Increase information transfer rates in BCI by CSP extension to multi-class. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 733–740. MIT Press, Cambridge, MA, 2004.

[46] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, and K.-R. Müller. Optimizing spatio-temporal filters for improving brain-computer interfacing. In *Advances in Neural Inf. Proc. Systems (NIPS 05)*, volume 18, 2006. accepted.

[47] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, and K.-R. Müller. Combined optimization of spatial and temporal filters for improving brain-computer interfacing. *IEEE Transactions on Biomedical Engineering*, 2006. accepted.

[48] G. Dornhege, M. Braun, J. Kohlmorgen, B. Blankertz, K.-R. Müller, G. Curio, K. Hagemann, A. Bruns, M. Schrauf, and W. Kincses. Improving human performance in a real operating environment through real-time mental workload detection. *Neural Computation*, 2006. submitted.

[49] G. Dornhege, J. del R. Millán, T. Hinterberger, D. McFarland, and K.-R. Müller, editors. *Towards Brain-Computer Interfacing*. MIT Press, 2006. in preparation.

[50] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, second edition, 2001.

[51] M. Falkenstein, J. Hoormann, S. Christ, and J. Hohnsbein. ERP components on reaction errors and their functional significance: a tutorial. *Biological Psychology*, 51(2-3):87–107, 2000.

[52] L. Farwell and E. Donchin. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and Clinical Neurophysiology*, 70:510–523, 1988.

[53] P. Ferrez and J. Millán. You are wrong!—automatic detection of interaction errors from brain waves. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, UK, August 2005.

[54] J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[55] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, 2nd edition, 1990.

[56] B. Graimann, J. E. Huggins, S. P. Levine, and G. Purtscheller. Visualization of significant ERD/ERS patterns in multichannel EEG and EcoG data. *Clinical Neurophysiology*, 113:43–47, 2002.

[57] J. B. Green, P. A. Arnold, L. Rozhkov, D. Strother, and N. Garrott. Bereitschaft (readiness potential) and supplemental motor area interaction in movement generation: Spinal cord injury and normal subjects. *Journal of Rehabilitation Research & Development*, 40(3):225–234, 2003.

[58] G. Grimmett and D. Stirzacker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.

[59] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors. *Feature extraction, foundations and applications*, chapter Filter Methods. Springer, 2004.

[60] S. Harmeling. *Independent component analysis and beyond*. PhD thesis, University of Potsdam, Potsdam, October 2005.

[61] S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller. Kernel feature spaces and nonlinear blind source separation. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

[62] S. Harmeling, A. Ziehe, M. Kawanabe, and K.-R. Müller. Kernel-based nonlinear blind source separation. *Neural Computation*, 15:1089–1124, 2003.

[63] S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, and K.-R. Müller. From outliers to prototypes: ordering data. *Neurocomputing*, 2005. accepted.

[64] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[65] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.

[66] H. Jasper and H. Andrews. Normal differentiation of occipital and precentral regions in man. *Arch. Neurol. Psychiat. (Chicago)*, 39:96–115, 1938.

[67] H. Jasper and W. Penfield. Electrocorticograms in man: Effects of voluntary movement upon the electrical activity of the precentral gyrus. *Arch. Psychiat. Nervenkr.*, 183:163–174, 1949.

[68] R. Knight. Contribution of human hippocampal region to novelty detection. *Nature*, 383(6597):256–259, 1996.

[69] Z. J. Koles and A. C. K. Soong. EEG source localization: implementing the spatio-temporal decomposition approach. *Electroencephalography and Clinical Neurophysiology*, 107:343–352, 1998.

[70] M. Krauledat, G. Dornhege, B. Blankertz, G. Curio, and K.-R. Müller. The Berlin brain-computer interface for rapid response. *Biomedizinische Technik*, 49(1):61–62, 2004.

[71] M. Krauledat, G. Dornhege, B. Blankertz, F. Losch, G. Curio, and K.-R. Müller. Improving speed and accuracy of brain-computer interfaces using readiness potential features. In *Proceedings of the 26th Annual International Conference IEEE EMBS on Biomedicine, San Francisco*, 2004.

[72] M. Krauledat, G. Dornhege, B. Blankertz, and K.-R. Müller. Robustifying EEG data analysis by removing outliers. *Chaos and Complexity*, 2005. accepted.

[73] R. Krepki. *Brain-Computer Interfaces: Design and Implementation of an Online BCI System of the Control in Gaming Applications and Virtual Limbs*. PhD thesis, Technische Universität Berlin, Fakultät IV – Elektrotechnik und Informatik, 2004.

[74] A. Kübler, B. Kotchoubey, T. Hinterberger, N. Ghanayim, J. Perelmouter, M. Schauer, C. Fritsch, E. Taub, and N. Birbaumer. The thought translation device: a neurophysiological approach to communication in total motor paralysis. *Experimental Brain Research*, 124:223–232, 1999.

[75] A. Kuebler, F. Nijboer, J. Mellinger, T. M. Vaughan, H. Pawelzik, G. Schalk, D. McFarland, N. Birbaumer, and J. R. Wolpaw. Patients with als can use sensorimotor rhythms to operate a brain-computer interface. *Neurology*, 64(10):1775–1777, 2005.

[76] T. N. Lal, T. Hinterberger, G. Widman, M. Schröder, N. J. Hill, W. Rosenstiel, C. E. Elger, B. Schölkopf, and N. Birbaumer. Methods towards invasive human brain computer interfaces. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 737–744. MIT Press, Cambridge, MA, 2005.

[77] G. Lanckriet, T. D. Bie, N. Cristianini, M. Jordan, and W. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626–2635, 2004.

[78] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[79] W. Lang, H. Obrig, G. Lindinger, D. Cheyne, and L. Deecke. Supplementary motor area activation while tapping bimanually different rhythms in musicians. *Experimental Brain Research*, 79(3):504–514, 1990.

[80] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Müller, E. Säckinger, P. Simard, and V. Vapnik. Learning algorithms for classification: A comparism on handwritten digit recognition. *Neural Networks*, pages 261–276, 1995.

[81] S. Lemm, B. Blankertz, G. Curio, and K.-R. Müller. Spatio-spectral filters for improved classification of single trial EEG. *IEEE Transactions on Biomedical Engineering*, 52(9):1541–1548, 2005.

[82] S. P. Levine, J. E. Huggins, S. L. BeMent, R. K. Kushwaha, L. A. Schuh, M. M. Rohde, E. A. Passaro, D. A. Ross, K. V. Elsievich, and B. J. Smith. A direct brain interface based on event-related potentials. *IEEE Transactions on Rehabilitation Engineering*, 8(2):180–185, 2000.

[83] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[84] D. J. McFarland and J. R. Wolpaw. EEG-based communication and control: Speed-accuracy relationships. *Applied Psychophysiology and Biofeedback*, 28(3):217–231, 2003.

[85] F. Meinecke, A. Ziehe, J. Kurths, and K.-R. Müller. Measuring Phase Synchronization of Superimposed Signals. *Physical Review Letters*, 94(8), 2005.

[86] R. Meir and G. Rätsch. An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, LNAI, pages 119–184. Springer, 2003.

[87] M. Middendorf, G. McMillan, G. Calhoun, and K. S. Jones. Brain-computer interface based on the steady-state visual-evoked response. *IEEE Transactions on Rehabilitation Engineering*, 8(2):211–214, June 2000.

[88] S. Mika. *Kernel Fisher Discriminants*. PhD thesis, University of Technology, Berlin, October 2002.

[89] J. D. R. Millán and J. Mouriño. Asynchronous bci and local neural classifiers: An overview of the adaptive brain interface project. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):159–161, 2003.

[90] J. D. R. Millán, J. Mouriño, F. Babiloni, F. Cinotti, and M. Varst. Local neural classifier for EEG-based recognition of mental tasks. *IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 2000.

[91] J. D. R. Millán, J. Mouriño, M. Franzé, F. Cinotti, M. Varsta, J. Heikkonen, and F. Babiloni. A local neural classifier for the recognition of EEG patterns associated to mental tasks. *IEEE Transactions on Neural Networks*, 13(3):678–686, 2002.

[92] J. D. R. Millán, F. Renkens, J. Mouriño, and W. Gerstner. Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, 2004.

[93] A. Mood, F. Graybill, and D. Boes. *Introduction to the theory of statistics*. McGraw-Hill Book Company, 1974.

[94] N. Morgan and H. Bourlard. Continuous speech recognition: An introduction to the hybrid hmm/connectionist approach. *Signal Processing Magazine*, pages 25–42, 1995.

[95] G. R. Müller, C. Neuper, R. Rupp, C. Keinrath, H. Gerner, and G. Pfurtscheller. Event - related beta EEG changes during wrist movements induced by functional electrical stimulation of forearm muscles in man. *Neuroscience Letters*, 340(2):143–147, 2003.

[96] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Neural Networks*, 12(2):181–201, May 2001.

[97] K.-R. Müller, C. W. Anderson, and G. E. Birch. Linear and non-linear methods for brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):165–169, 2003.

[98] K.-R. Müller, M. Krauledat, G. Dornhege, G. Curio, and B. Blankertz. Machine learning techniques for brain-computer interfaces. *Biomedizinische Technik*, 49(1): 11–22, 2004.

[99] K.-R. Müller, M. Krauledat, G. Dornhege, S. Jähnichen, G. Curio, and B. Blankertz. A note on the Berlin Brain-Computer Interface. In G. Hommel and S. Huanye, editors, *Human Interaction with Machines: Proceedings of the 6th International Workshop held at the Shanghai Jiao Tong University*, pages 51–60, 2006.

[100] G. R. Müller-Putz, R. Scherer, G. Pfurtscheller, and R. Rupp. EEG-based neuroprosthesis control: A step towards clinical practice. *Neuroscience Letters*, 2005. in press.

[101] N. Neumann and A. Kuebler. Training locked-in patients: a challenge for the use of brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):169–172, 2003.

[102] N. Neumann, A. Kuebler, J. Kaiser, T. Hinterberger, and N. Birbaumer. Conscious perception of brain states: mental strategies for brain-computer communication. *Neuropsychologia*, 41(8):1028–1036, 2003.

[103] N. Neumann, T. Hinterberger, J. Kaiser, U. Leins, N. Birbaumer, and A. Kuebler. Automatic processing of self-regulation of slow cortical potentials: evidence from brain-computer communication in paralysed patients. *Clinical Neurophysiology*, 115 (3):628–635, 2004.

[104] M. A. Nicolelis, A. A. Ghazanfar, C. R. Stambaugh, L. M. Oliveira, M. Lambach, J. Chapin, R. J. Nelson, and J. H. Kaas. Simultaneous encoding of tactile information by three primate cortical areas. *Nature Neuroscience*, 7:621–630, 1998.

[105] S. Nieuwenhuis, K. Ridderinkhof, J. Blom, G. Band, and A. Kok. Error-related brain potentials are differentially related to awareness of response errors: evidence from an antisaccade task. *Psychophysiology*, 38(5):752–760, September 2001.

[106] A. V. Oppenheim and R. W. Schafer. *Discrete-time signal processing*. Prentice Hall Signal Processing Series. Prentice Hall, 1989.

[107] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proceedings CVPR'97*, 1997.

[108] B. O. Peters, G. Pfurtscheller, and H. Flyvbjerg. Automatic differentiation of multichannel EEG signals. *IEEE Transactions on Biomedical Engineering*, 48(1):111–116, 2001.

[109] G. Pfurtscheller. Graphical display and statistical evaluation of event-related desynchronization (ERD). *Electroencephalography and Clinical Neurophysiology*, 43: 757–760, 1977.

[110] G. Pfurtscheller, C. Neuper, C. Guger, W. Harkam, R. Ramoser, A. Schlögl, B. Obermaier, and M. Pregenzer. Current trends in Graz brain-computer interface (BCI). *IEEE Transactions on Rehabilitation Engineering*, 8(2):216–219, June 2000.

[111] G. Pfurtscheller, G. R. Müller, J. Pfurtscheller, H. J. Gerner, and R. Rupp. 'thought' - control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia. *Neuroscience Letters*, 351:33–36, 2003.

[112] D.-T. Pham. Blind separation of instantaneous mixture of sources via the gaussian mutual information criterion. *Signal Processing*, 81:855–870, 2001.

[113] D.-T. Pham. Joint approximate diagonalization of positive definite matrices. *SIAM J. on Matrix Anal. and Appl.*, 22(4):1136–1152, 2001.

[114] J. Polich and A. Kok. Cognitive and biological determinants of p300: an integrative review. *Biological Psychology*, 41(2):103–146, 1995.

[115] H. Purwins, B. Blankertz, G. Dornhege, and K. Obermayer. Scale degree profiles from audio investigated with machine learning techniques. In *Audio Engineering Society 116th Convention*, Berlin, 2004.

[116] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE Transactions on Rehabilitation Engineering*, 8(4):441–446, 2000.

[117] G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001.

[118] G. Rizzolatti, G. Luppino, and M. Matelli. The classic supplementary motor area is formed by two independent areas. *Advances in Neurology*, 70:45–56, 1996.

[119] P. Sabbah, S. de Schonen, C. Leveque, S. Gay, F. Pfefer, C. Nioche, J.-L. Sarrazin, H. Barouti, M. Tadie, and Y.-S. Cordoliani. Sensorimotor cortical activity in patients with complete spinal cord injury: A functional magnetic resonance imaging study. *Journal of Neurotrauma*, 19(1):53–60, 2002.

[120] P. Sajda, A. Gerson, K.-R. Müller, B. Blankertz, and L. Parra. A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):184–185, 2003.

[121] A. W. Salmoni, R. A. Schmidt, and C. B. Walter. Knowledge of results and motor learning: a review and critical reappraisal. *Psychological Bulletin*, 95(3):366–386, 1984.

[122] G. Schalk, J. R. Wolpaw, D. J. McFarland, and G. Pfurtscheller. EEG-based communication: presence of an error potential. *Clinical Neurophysiology*, 111:2138–2144, 2000.

[123] A. Schlögl, C. Neuper, and G. Pfurtscheller. Estimating the mutal information of an EEG-based brain-computer interface. *Biomedizinische Technik*, 47(1-2):3–8, 2002.

[124] B. Schölkopf, C. Burges, and A. Smola, editors. *Advances in Kernel Methods – Support Vector Learning*. MIT Press, 1999.

[125] C. Schwägerl. Ich bin ein Cursor. Frankfurter Allgemeine Zeitung, March 2004.

[126] A. B. Schwartz. Motor cortical activity during drawing movement: population representation during sinusoid tracing. *Journal of Neurophysiology*, 70:28–36, 1993.

[127] S. Sutton, M. Braren, J. Zubin, and E. R. John. Evoked-potential correlates of stimulus uncertainty. *Science*, 150(700):1187–1188, 1965.

[128] J. Tanji. The supplementary motor area in the cerebral cortex. *Neuroscience Research*, 19(3):251–268, 1994.

[129] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Henning, T. Hofmann, M. Krell, and T. Schmidt. Map learning and high-speed navigation in RHINO. In D. Kortenkamp, R. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots*. MIT Press, 1998.

[130] C. Toro, G. Deuschl, R. Thather, S. Sato, C. Kufta, and M. Hallett. Event-related desynchronization and movement-related cortical potentials on the ECoG and EEG. *Electroencephalography and Clinical Neurophysiology*, 93:380–389, 1994.

[131] Y. Tran, P. Boord, J. Middleton, and A. Craig. Levels of brain wave activity (8-13 hz) in persons with spinal cord injury. *Spinal Cord*, 42(2):73–79, 2004.

[132] V. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.

[133] J. J. Vidal. Toward direct brain-computer communication. *Annu. Rev. Biophys.*, 2: 157–180, 1973.

[134] J. J. Vidal. Real-time detection of brain events in EEG. *IEEE Proc*, 65:633–664, 1977.

[135] B. Winer, editor. *Statistical Principles in Experimental Design*. McGraw-Hill, New York, 2nd edition, 1962.

[136] J. R. Wolpaw and D. J. McFarland. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proceedings of the National Academy of Sciences of the United States of America*, 101(51):17849–17854, 2004.

[137] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris. An EEG-based brain-computer interface for cursor control. *Electroencephalography and Clinical Neurophysiology*, 78:252–259, 1991.

[138] J. R. Wolpaw, D. Flotzinger, G. Purtscheller, and D. J. McFarland. Timing of EEG-based cursor control. *Journal of Clinical Neurophysiology*, 14(6):529–538, 1997.

[139] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan. Brain-computer interface technology: A review of the first international meeting. *IEEE Transactions on Rehabilitation Engineering*, 8(2):164–173, 2000.

[140] J. R. Wolpaw, D. J. McFarland, and T. M. Vaughan. Brain-computer interface research at the Wadsworth Center. *IEEE Transactions on Rehabilitation Engineering*, 8(2):222–226, 2000.

[141] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113:767–791, 2002.

[142] J. R. Wolpaw, D. J. McFarland, T. M. Vaughan, and G. Schalk. The Wadsworth Center brain-computer interface (BCI) research and development program. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):207–207, 2003.

[143] M.-H. Yang. Face recognition using kernel methods. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

[144] A. Ziehe and K.-R. Müller. TDSEP – an efficient algorithm for blind separation using time structure. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks, ICANN'98*, Perspectives in Neural Computing, pages 675 – 680, Berlin, 1998. Springer Verlag.

[145] A. Ziehe, P. Laskov, K.-R. Müller, and G. Nolte. A linear least-squares algorithm for joint diagonalization. In *Proc. 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 469–474, Nara, Japan, Apr 2003.

[146] A. Ziehe, P. Laskov, G. Nolte, and K.-R. Müller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5:777–800, Jul 2004.

[147] A. Zien, G. Rätsch, S. Mika, B. Schölkopf, T. Lengauer, and K.-R. Müller. Engineering Support Vector Machine Kernels That Recognize Translation Initiation Sites. *BioInformatics*, 16(9):799–807, Sept. 2000.