BAYESIAN MACHINE LEARNING APPLIED IN A BRAIN-COMPUTER INTERFACE FOR DISABLED USERS

THÈSE N^{O} 3924 (2007)

PRÉSENTÉE LE 30 NOVEMBRE 2007

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE DE TRAITEMENT DES SIGNAUX 1

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Ulrich HOFFMANN

Diplom-Informatiker, Eberhard Karls Universität, Tübingen, Allemagne
et de nationalité allemande

acceptée sur proposition du jury:

Prof. H. Bourlard, président du jury
Prof. T. Ebrahimi, Dr J.-M. Vesin, directeurs de thèse
Dr D. Barber, rapporteur
Dr D. Debatisse, rapporteur
Prof. P. Vandergheynst, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2007

Contents

Abstract

A brain-computer interface (BCI) is a system that enables control of devices or communication with other persons, only through cerebral activity, without using muscles. The main application for BCIs is assistive technology for disabled persons. Examples for devices that can be controlled by BCIs are artificial limbs, spelling devices, or environment control systems.

BCI research has seen renewed interest in recent years, and it has been convincingly shown that communication via a BCI is in principle feasible. However, present day systems still have shortcomings that prevent their widespread application. In part, these shortcomings are caused by limitations in the functionality of the pattern recognition algorithms used for discriminating brain signals in BCIs. Moreover, BCIs are often tested exclusively with able-bodied persons instead of conducting tests with the target user group, namely disabled persons.

The goal of this thesis is to extend the functionality of pattern recognition algorithms for BCI systems and to move towards systems that are helpful for disabled users. We discuss extensions of linear discriminant analysis (LDA), which is a simple but efficient method for pattern recognition. In particular, a framework from Bayesian machine learning, the so-called evidence framework, is applied to LDA. An algorithm is obtained that learns classifiers quickly, robustly, and fully automatically. An extension of this algorithm allows to automatically reduce the number of sensors needed for acquisition of brain signals. More specifically, the algorithm allows to perform electrode selection. The algorithm for electrode selection is based on a concept known as automatic relevance determination (ARD) in Bayesian machine learning. The last part of the algorithmic development in this thesis concerns methods for computing accurate estimates of class probabilities in LDA-like classifiers. These probabilities are used to build a BCI that dynamically adapts the amount of acquired data, so that a preset, approximate bound on the probability of misclassifications is not exceeded.

To test the algorithms described in this thesis, a BCI specifically tailored for disabled persons is introduced. The system uses electroencephalogram (EEG) signals and is based on the P300 evoked potential. Datasets recorded from five disabled and four able-bodied subjects are used to show that the Bayesian version of LDA outperforms plain LDA in terms of classification accuracy. Also, the impact of different static electrode configurations on classification accuracy is tested. In addition, experiments with the same datasets demonstrate that the algorithm for electrode selection is computationally efficient, yields physiologically plausible results, and improves classification accuracy over static electrode configurations. The classification accuracy is further improved by dynamically

adapting the amount of acquired data. Besides the datasets recorded from disabled and able-bodied subjects, benchmark datasets from BCI competitions are used to show that the algorithms discussed in this thesis are competitive with state-of-the-art electroencephalogram (EEG) classification algorithms.

While the experiments in this thesis are uniquely performed with P300 datasets, the presented algorithms might also be useful for other types of BCI systems based on the EEG. This is the case because functionalities such as robust and automatic computation of classifiers, electrode selection, and estimation of class probabilities are useful in many BCI systems. Seen from a more general point of view, many applications that rely on the classification of cerebral activity could possibly benefit from the methods developed in this thesis. Among the potential applications are interrogative polygraphy ("lie detection") and clinical applications, for example coma outcome prognosis and depth of anesthesia monitoring.

Keywords
Brain-Computer Interface, Disabled Users, Assistive Technology, Electroencephalogram, Evoked Potentials, P300, Bayesian Machine Learning, Linear Discriminant Analysis, Evidence Framework, Automatic Relevance Determination

Version abrégée

Un interface cerveau-ordinateur (ICO) est un système qui permet la commande de dispositifs ou la communication avec autres personnes, par l'activité cérébrale seule, sans employer des muscles. L'application principale des ICO est la technologie assistive pour personnes handicapées. Des exemples de dispositifs pouvant être commandés par un ICO sont les membres artificiels, les dispositifs pour épeler, ou les systèmes de contrôle d'environnement.

Pendant les dernières années la recherche sur les ICO a éveillé l'intérêt de beaucoup des chercheurs, et il a été montré de façon convaincante que la communication par le biais d'un ICO est en principe faisable. Cependant, les systèmes actuels ont toujours des imperfections qui empêchent une application répandue. Ces imperfections sont provoquées en partie par des limitations dans la fonctionnalité des algorithmes de reconnaissance de formes utilisés dans les ICO pour distinguer differentes types des signaux cérébraux. En outre, les ICO sont souvent testés exclusivement avec des sujets sains alors qu'il conviendrait de le faire avec le groupe d'utilisateurs ciblé, à savoir des personnes handicapées.

Le but de cette thèse est d'améliorer la fonctionnalité des algorithmes de reconnaissance de formes pour les ICO et de rendre à des systèmes utiles pour des utilisateurs handicapés. Nous discutons des extensions de "linear discriminant analysis" (LDA), qui est une méthode simple mais efficace pour la reconnaissance de formes. En particulier, une methode bayésienne pour la reconnaissance de formes, le "evidence framework", est appliqué à LDA. L'algorithme obtenu par l'application de cette methode permet un apprentissage rapide, robuste, et entièrement automatique. Une extension de cet algorithme permet de réduire automatiquement le nombre d'éelectrodes requises pour l'acquisition des signaux cérébraux. Plus spécifiquement, l'algorithme permet de sélectionner les electrodes importantes pour la classification. L'algorithme pour cette selection des electrodes est basé sur un concept connu en tant que "automatic relevance determination" (ARD) dans la reconnaissance de formes bayésienne. La dernière partie du développement algorithmique dans cette thèse porte sur des méthodes pour calculer précisement les probabilités de classe dans des classificateurs comme LDA. Ces probabilités sont employées dans un ICO qui adapte dynamiquement la quantité de données acquises, de sorte qu'une limite préréglée, approximative sur la probabilité de fausse classifications ne soit pas franchie.

Pour tester les algorithmes décrits dans cette thèse, un ICO spécifiquement adapté pour les personnes handicapées est présenté. Le système utilise des signaux de l'électroencéphalogramme (EEG) et est basé sur le potentiel évoqué P300. Des données enregistrées de cinq sujets handicapés

et de quatre sujets sains sont utilisées pour montrer que la version bayésienne de LDA surpasse le LDA simple en termes de qualité de classification. En outre, l'impact de différentes configurations statiques d'électrodes sur cette qualité de classification est examinée. Des expériences ultérieures avec les mêmes données démontrent que l'algorithme pour le choix d'électrodes est efficace, que les résultats sont physiologiquement plausibles, et que la qualité de classification est améliorée par rapport à des configurations statiques d'électrodes. La qualité de classification est encore améliorée par l'adaptation dynamique de la quantité de données acquises. Outre les données acquises des sujets handicapés et sains, des données de concours ICO sont utilisées pour prouver que les algorithmes discutés dans cette thèse sont concurrentiels avec des algorithmes "état de l'art".

Quoique les expériences dans cette thèse soient uniquement exécutées avec des données P300, les algorithmes présentés pourraient également être utiles pour d'autres types de ICO basés sur l'EEG. C'est le cas parce que fonctionnalités telles que l'apprentissage robuste et automatique de classificateurs, le choix d'électrodes, et le calcul des probabilités de classe sont utiles dans beaucoup des systèmes ICO. D'un point de vue général, beaucoup d'applications qui se fondent sur la classification de l'activité cérébrale peuvent probablement tirer bénéfice des méthodes développées dans cette thèse. Parmi les applications potentielles citons la polygraphie interrogative ("détection des mensonges") et des applications cliniques telles que par exemple le pronostic de coma et la surveillance de profondeur d'anesthésie.

Mots-Clés

Interface Cerveau-Ordinateur, Utilisateurs Handicapées, Technologie Assistive, Électroencéphalogramme, Potentiel Évoqué, P300, Reconnaissance De Formes Bayésienne, Linear Discriminant Analysis, Evidence Framework, Automatic Relevance Determination

Acknowledgments

Working towards this thesis during four years and finally writing up and presenting it was an interesting and exciting but at times also stressful experience. Finishing the thesis would have been impossible without the help, support, and encouragement of many people whom I would like to thank in the following.

First, I would like to express my gratitude to Prof. Touradj Ebrahimi for giving me the possibility to do a PhD at the ITS and for supporting me when it was necessary. I think I learned a lot about my research topic and even more about research in general while working in his group. Many thanks go to my thesis co-supervisor Dr. Jean-Marc Vesin from whom I received a lot of valuable feedback concerning drafts of publications and thesis chapters. Thanks for always being patient and for many interesting discussions about scientific and non-scientific topics. Also I would like to thank Prof. Pierre Vandergheynst, Med. Sc. Damien Debatisse, Dr. David Barber, and Prof. Hervé Bourlard for accepting to be part of the thesis committee, for reading the thesis, and for valuable comments that helped to improve my work.

For reading and correcting parts of my thesis I would like to thank Jonas Richiardi, Julien Meynet, David Marimón Sanjuán, Michael Ansorge, and my brother Johannes Hoffmann. Special thanks go to my office mate Michael for moral support during the week before the presentation and to David for listening twice to my thesis presentation. For accepting to read through a preliminary version of the material presented in chapter 6 of this thesis and for giving many valuable comments many thanks go to Michael Bensch. For introducing me to the basics of brain-computer interfacing and machine learning I would like to thank Dr. Gary Garcia. Gary finished his PhD when I started and without his knowledge and the work he has done it would have been much more difficult to get my PhD done.

Besides the work at EPFL a significant amount of work for this thesis was done at the “Fondation Plein Soleil” in Lausanne - a home for disabled persons. In this context I would first and foremost like to express my gratitude to the persons living at Plein Soleil who agreed to participate in my experiments. Communicating with you and doing the experiments was a good and very instructive experience that I will not forget. Many thanks also go to Nicolas Gremaud who initiated the cooperation with Plein Soleil and who spent a lot of time helping with experiments and tests. A further important person for the cooperation with Plein Soleil was Karin Diserens to whom I express my gratitude for her interest and support during the early phase of the project. Special thanks go to Viviane Chevalley who was extremely supportive and efficient in scheduling the experiments.

She always found the time to help and have a chat even if her own schedule was more than full. A lot of help also came from Eric Girardet and Serge Chiarardia to whom I express my gratitude for programming parts of the software needed for the experiments, for performing experiments, and for going back and forth between EPFL and Plein Soleil with all the BCI equipment more than once. Also I would like to thank Rosario Leroy for participating in many discussions, having stimulating ideas, and supporting the project.

For the exceptionally friendly and relaxed atmosphere at ITS I would like to thank all the people working there and having worked there. Working at ITS has been fun due to the possibility to interact with people from many different parts of the world and due to the many extracurricular activities organized by members of the ITS. Thanks to Markus, Sila, Gary, Lam, Julien, David, Mathieu, Yann, Karin, Zenichi, Patricia, Lisa, Phillipe, Grace, Yannick, Lorenzo, Lorenzo, Matteo, Gianluca, Gianluca, Nawal, and Anna, be it for cool babyfoot games, for coffee break and lunch time discussions, or for fun nights at SAT, in Lausanne, or elsewhere. For many non-EPFL related activities I would like to thank the “chilenian gang”: Gracias, muchas gracias Claribel, Paola, Alex, Cristobal y Andrea por enseñarme el español, comida rica, y mucho mas!

Last but not least I would like to thank my family for visiting me several times at Lausanne and for supporting and encouraging me during the last four years.

1 Introduction

1.1 Motivation

The ability to communicate with other persons, be it through speech, gesturing, or writing, is one of the main factors making the life of any human being enjoyable. Communication is at the basis of human development, makes it possible to express ideas, desires, and feelings, and on a more ordinary level simply allows to cope with daily life. Individuals suffering from the so-called locked-in syndrome do not have the above mentioned communication possibilities. The locked-in syndrome is a condition in which patients are fully conscious and aware of what is happening in their environment but are not able to communicate or move. In fact, the locked-in syndrome is caused by a nearly total loss of control over the voluntary muscles. A disease that is known to lead to the locked-in syndrome is amyotrophic lateral sclerosis (ALS), also known as Lou Gehrig's disease. ALS is a progressive, neurodegenerative disease and is characterized by the death of motor neurons which in turn leads to the loss of control over voluntary muscles. Besides ALS also multiple sclerosis, stroke or other cerebrovascular incidents leading to the infarction or degeneration of parts of the brain can cause the locked-in syndrome. Clearly, the quality of live of persons affected by the locked-in syndrome is strongly diminished by the lack of possibilities to communicate with other persons and by the complete loss of autonomy.

A promising means to give back basic communication abilities and a small degree of autonomy to locked-in persons are brain-computer interfaces (BCIs). The idea underlying BCIs is to measure electric, magnetic, or other physical manifestations of brain activity and to translate these into commands for a computer or other devices. More specifically, the idea underlying BCIs is to detect patterns of brain activity and to link these patterns to commands executed by a computer or other devices. Prototype systems allow for example to choose symbols from an alphabet by concentrating on specific mental tasks or to move artificial limbs, solely by imagining movements.

Basic research on BCI systems commenced in the early 1970's and has seen renewed interest in recent years. While increases in computing power and advances in measurement technology have led to a large variety of proof-of-concept systems, none of the systems described in the scientific literature is suited for daily use by disabled persons. This is due to the fact that the technology underlying BCIs is not yet mature enough for usage out of the laboratory. The main motivation for the research described in this thesis is to provide advances in technology which will lead to BCI systems featuring a performance beyond mere proof-of-concept systems.

1.2 Focus of the Thesis

Taking a closer look at the research area of brain-computer interfaces, one observes that a whole panoply of different systems exists. The general term brain-computer interface (or brain-machine interface) includes not only systems in which signals from the brain are recorded and analyzed but also systems in which signals are fed into the brain. A good example for the latter type of systems is given by cochlear implants, i.e. devices that transform sounds from the environment into electrical impulses which are in turn used to directly stimulate auditory nerves. Another example is given by retinal implants which can transform light into electrical impulses which in turn are used to stimulate nerve cells in the retina of blind persons. While cochlear implants are a relatively mature technology that is already used by thousands of patients, retinal implants are still under development.

Systems that directly stimulate nerve cells are by definition invasive which means that a surgical procedure is necessary to implant the device under question. The type of systems that is of interest in this thesis, namely systems which record and analyze signals from the brain, can be invasive or noninvasive. Invasive systems, such as for example systems using microelectrode arrays implanted in the motor cortex, have the advantage that the recorded signals have a high signal-to-noise ratio and that useful information can be extracted relatively easily from the signals. As a consequence, demanding applications such as for example the three-dimensional control of artificial limbs can be realized with invasive systems. Noninvasive systems on the contrary have the big advantage that potentially risky surgical procedures are unnecessary. This advantage, however, comes at the cost of decreased signal-to-noise ratio. Hence, the signals recorded with noninvasive approaches are often more difficult to analyze than those recorded with invasive approaches.

A particularly popular noninvasive method that allows to measure electric potentials of the brain at a temporal resolution on the order of milliseconds is the electroencephalogram (EEG). This measurement method is popular among neuroscientists and physicians because modern acquisition devices are relatively inexpensive and easily transportable and because the setup of recording sessions takes only little time. For the same reasons, the EEG is also used in many BCI systems. Among such BCI systems two subtypes can be discerned:

- Systems based on spontaneous activity use EEG signals that do not depend on external stimuli and that can be influenced by concentrating on a specific mental task. An example is given by so-called mu-rhythm BCIs. In these systems feedback training is used to let subjects acquire voluntary control over the amplitude of the mu-rhythm, i.e. EEG activity in the frequency

range of 8-12 Hz, located over the motor cortex. Changes in mu-rhythm amplitude are then linked to movements of a cursor or to other commands.

- Systems based on evoked activity use EEG signals that do depend on external stimuli. Particularly interesting are systems based on the P300 evoked potential. Roughly speaking, such systems work by detecting on which stimulus out of a random series of stimuli the user is concentrating. Since different commands or actions are linked to the stimuli, users can select a command simply by concentrating on the associated stimulus. The neurophysiological phenomenon underlying this approach is the P300 evoked potential. The P300 is a positive peak in the EEG that appears approximately 300 ms after the presentation of a task-significant stimulus in a random series of stimuli. Hence, detecting which stimulus the user is concentrating on is equivalent to detecting which stimulus has evoked a P300.

Advantages of the P300 are that it is relatively well understood from a neurophysiologic point of view and that it can be evoked robustly across different subjects. Moreover, feedback training is not necessary in P300 based BCI systems, as the P300 appears "automatically" whenever subjects concentrate onto one out of several stimuli presented in random order. The latter advantage is important when the goal is to build BCI systems for disabled subjects who might have difficulties in acquiring voluntary control over their brain activity.

Motivated by the aforementioned advantages, this thesis focuses on BCI systems based on the P300. An important component of any such system, but also of other EEG based systems, are pattern recognition methods that allow to discriminate EEG segments representing different types of brain activity. Hence, in this thesis special emphasis is given to algorithms that learn from a set of training data how to discriminate EEG segments containing a P300 from other EEG segments. In particular, the algorithms are built by making use of tools from Bayesian machine learning. Besides the theoretical and algorithmic aspects of BCI systems, emphasis is also put on the thorough testing of the presented algorithms with a relatively large database containing EEG records from disabled and able-bodied subjects.

1.3 Main Contributions

The main contributions of this thesis are the application of modern pattern recognition algorithms to BCI systems and the thorough testing of these algorithms with P300 data recorded from a BCI specifically adapted to disabled subjects. The applied algorithms are described and discussed in-depth and the communication rates achieved with the BCI for disabled users are significantly beyond those of previously described, comparable systems. The detailed contributions are listed in the following.

- The so-called evidence framework (MacKay, 1992), a Bayesian framework for estimating hyperparameters in neural networks or regression, is applied in the context of linear discriminant analysis. The resulting Bayesian linear discriminant analysis (BDA) algorithm is well suited for applications in BCI systems and can learn classifiers quickly, robustly, and fully automatically. Experiments show that BDA outperforms plain LDA in terms of classification accuracy.

- Application of the concept of automatic relevance determination (ARD) (MacKay, 1995; Tipping, 2001) to the problem of electrode selection in a BCI. The developed algorithm can automatically determine the size of an optimal electrode subset or find electrode subsets with a predetermined size. Experiments show that selecting electrodes with ARD is computationally efficient, improves classification accuracy, and yields physiologically plausible results.

- Development of an algorithm that sequentially computes probabilities for a set of hypotheses where the hypotheses concern the generation of a stream of data. The algorithm is used to dynamically adapt the amount of data recorded in P300 BCIs. More precisely, the algorithm is used to build a system in which data is acquired until the probability of misclassification is below a preset threshold. Experiments show that dynamically adapting the amount of recorded data improves the speed of communication compared to systems in which a fixed amount of data is used.

- Development of a P300 BCI system which is specifically adapted to the needs of disabled users. The impact of different fixed electrode configurations on the communication speed achievable with the system is explored. The system allowed several disabled users to achieve communication rates that are significantly beyond the rates previously reported in the literature. Possible reasons for the improved communication rates are discussed.

- P300 datasets recorded from four disabled and four able-bodied subjects are made publicly available on the internet. In addition, MATLAB implementations of some of the algorithms described in this thesis are made available on the internet. Datasets and algorithms allow to reproduce results presented in this thesis and can be downloaded from the address `http://bci.epfl.ch/efficientp300bci.html`.

1.4 Outline of the Thesis

The rest of this thesis is organized into seven chapters. Chapters 2 to 4 contain background material, Chapters 5 to 7 mainly describe research specific to this thesis, and Chapter 8 contains a summary and an outlook on future work. The detailed contents are listed in the following.

- In Chapter 2, a general introduction to the field of BCI research is given. Topics reviewed include different methods for measuring brain activity, the types of neurophysiologic signals that can be used in BCI systems, methods for extracting useful features from neurophysiological signals, and BCI applications.

- In Chapter 3, basic concepts of supervised machine learning are reviewed. In the first part of the chapter a general exposition of the supervised machine learning problem is given and important concepts such as overfitting, cross-validation, and model selection are discussed. In the second part of the chapter, probabilistic methods for supervised machine learning, i.e. maximum-likelihood estimation, maximum a posteriori estimation, and Bayesian learning are introduced. In the third part, concrete examples for the theory described in the first two

parts are given. In particular, several supervised learning algorithms that have been applied in the context of BCI are discussed.

- In Chapter 4, BCI systems using the P300 are reviewed. First, the P300 is described from a neuroscientific point of view. Conditions under which a P300 can be evoked and factors that influence the characteristics of the P300 are discussed. Then, the basic idea underlying P300-based BCIs is introduced and several systems implementing this idea are described. Finally, algorithmic aspects of P300-based BCIs are discussed and criteria for evaluating the different systems and algorithms are described.

- In Chapter 5 the supervised learning algorithms used in this thesis are introduced. First, the connection between least-squares regression and Fisher's discriminant analysis (FDA) is reviewed and used to motivate a Bayesian approach to discriminant analysis. Then, BDA is reviewed. In the second part of the chapter the BDA algorithm is extended to perform electrode selection in a BCI. The resulting algorithm is referred to as sparse Bayesian linear discriminant analysis (SBDA) in the rest of the thesis and uses a framework for sparse Bayesian learning, namely the ARD framework. In the last part of the chapter two methods are presented that allow for the computation of accurate class probabilities with BDA and SBDA. Moreover, an algorithm that sequentially computes probabilities for a set of hypotheses where the hypotheses concern the generation of a stream of data is introduced. This algorithm is applied in the context of P300 BCIs to adaptively stop data acquisition as soon as the probability of misclassification is smaller than a preset threshold. In other words, this algorithm allows to profit from fluctuations in the level of uncertainty of the recorded signals. If the level of uncertainty is small, classification decision are taken quickly. If the level of uncertainty is high, more data is recorded to avoid wrong decisions.

- In Chapter 6 a BCI system for disabled users is introduced. In the first part of the chapter the system itself is described, the patients from whom data is recorded are presented, and the experimental setup is presented. In the second part of the chapter results from offline experiments conducted with FDA and BDA on the recorded data are presented. Classification accuracy and bitrate achievable by using FDA or BDA in conjunction with different electrode configurations are discussed. Finally, differences to other P300 BCI systems for disabled subjects are discussed.

- In Chapter 7 the setup and results of experiments conducted with SBDA are described. Additionally, the algorithm for adaptively stopping data acquisition is explored. The chapter starts with a report about the classification accuracy that can be obtained with SBDA and with a comparison of SBDA and BDA. Then, the electrode subsets selected with SBDA are analyzed and compared to predefined, physiologically plausible electrode subsets. Furthermore, BDA and SBDA are applied to P300 datasets from past BCI competitions and it is shown that both algorithms lead to classification accuracies which are competitive with the state-of-the-art. In the second part of the chapter experiments with the adaptive stopping algorithm are described. It is shown that the adaptive stopping algorithm allows to obtain higher communication speed than decision schemes in which a fixed amount of data is used.

- In Chapter 8 the contributions of this thesis are summarized and an outlook on possible extensions of the presented work is provided.

Introduction to Brain-Computer Interfaces

2

2.1 Introduction

A BCI is a communication system that translates brain activity into commands for a computer or other devices. In other words, a BCI allows users to act on their environment by using only brain activity, without using peripheral nerves and muscles. The major goal of BCI research is to develop systems that allow disabled users to communicate with other persons, to control artificial limbs, or to control their environment. To achieve this goal, many aspects of BCI systems are currently being investigated. Research areas include evaluation of invasive and noninvasive technologies to measure brain activity, evaluation of control signals (i.e. patterns of brain activity that can be used for communication), development of algorithms for translation of brain signals into computer commands, and the development of new BCI applications.

In this chapter we review the aspects of BCI research mentioned above and highlight recent developments and open problems. The review is ordered by the steps that are needed for brain-computer communication (see Fig. 2.1). We start with methods for measuring brain activity (Section 2.2) and then give a description of the neurophysiologic signals that can be used in BCI systems (Section 2.3). The translation of signals into commands with the help of signal processing and classification methods is described in Section 2.4. Finally, applications that can be controlled with a BCI are described in Section 2.5, and a conclusion is given in Section 2.6.

The number of publications concerning BCI has strongly increased during the last few years. Hence, it is virtually impossible to give a balanced, exhaustive review of the field. The review provided here is biased towards electroencephalogram (EEG) based BCI systems. Other reviews can be found in the articles of Wolpaw *et al.* (2002), Lebedev and Nicolelis (2006), Birbaumer and Cohen (2007), and Mason *et al.* (2007). Detailed reports about the work in many BCI laboratories around the world can be found in the 2006 BCI special issue of IEEE Transactions on Neural Systems and Rehabilitation Engineering (Vaughan and Wolpaw, 2006).

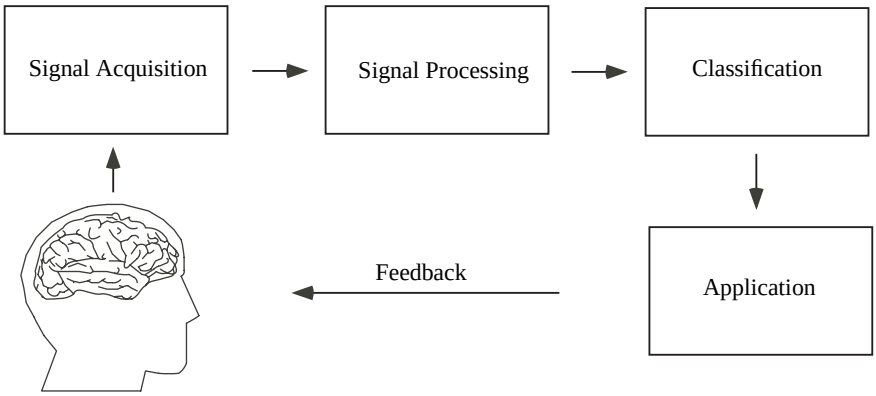

Figure 2.1 — Building blocks of a BCI. A subject performs a specific cognitive task or concentrates on a specific stimulus. Brain signals are acquired and then processed with signal processing and classification algorithms. The outcome of the classification is fed into an application, for example a spelling device. The application generates feedback to inform the subject about the outcome of classification.

2.2 Signal Acquisition

To enable communication with the help of a BCI, first brain signals have to be measured. Different methods to achieve this goal, ranging from the invasive measurement of electric potentials at single neurons to the noninvasive measurement of large-scale hemodynamic brain activity, have been reported in the literature. We review some of these methods below, starting from the EEG which allows for measurements of electric potentials at large spatial scales. We continue with the electrocorticogram (ECoG) and microelectrode arrays, which allow for measurement of potentials at smaller spatial scales. Next, methods for measuring magnetic brain activity and hemodynamic brain activity are described. The different methods are compared in terms of temporal and spatial resolution, invasiveness vs. noninvasiveness, and in terms of complexity of the apparatus needed for performing measurements.

2.2.1 Electroencephalogram

The EEG is one of the most widely used noninvasive techniques for recording electrical brain activity. Since its discovery by Hans Berger (Berger, 1929) the EEG has been employed to answer many different questions about the functioning of the human brain and has served as a diagnostic tool in clinical practice. The EEG is a popular signal acquisition technique because the required devices are simple and cheap and because the preparation of measurements takes only a small amount of time. EEG signals are recorded with small silver/silver chloride electrodes with a radius of about 5 mm, placed on the scalp at standardized positions (see Fig. 2.2). Conductive gel or saltwater is used to improve the conductivity between scalp and electrodes. To affix the electrodes to the scalp, often an electrode cap is used. EEG signals are always recorded with respect to reference electrodes, i.e. EEG signals are small potential differences (0 - 100 μV) between electrodes placed at different positions on the scalp.

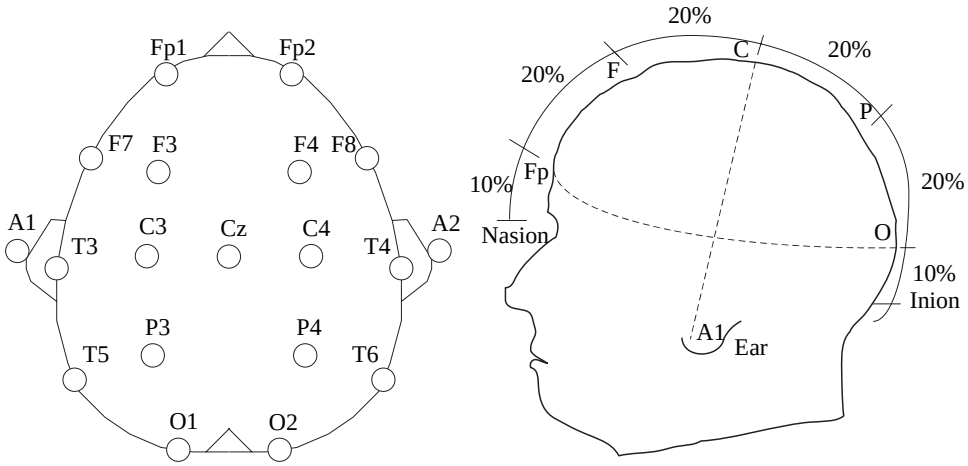

Figure 2.2 — Electrode placement according to the 10-20 international system. Odd numbers indicate electrodes located on the left side of the head. Even numbers indicate electrodes located on the right side of the head. Capital letters are used to reference each cortical zone, namely frontal (F), central (C), parietal (P), temporal (T), and occipital (O). Fp and A stand for frontal pole and auricular. The designation 10-20 comes from the percentage ratio of the inter-electrode distances with respect to the nasion-inion distance.

To understand how EEG signals are related to information processing in the brain, it is necessary to first review the structure and functioning of neurons. Neurons consist of a cell body (soma), an axon, and a dendritic tree (cf. Fig. 2.3). The axon serves as "output channel" of neurons and connects via synapses to the dendrites (the "input channel") of other neurons. The means of communication between neurons are action potentials, i.e. electrical discharges produced mainly at the soma of cells. Action potentials travel along the axon of cells and lead to a release of neurotransmitters when arriving at a synapse. The neurotransmitters in turn trigger a flow of ions across the cell membrane of the neuron receiving the action potential. The flow of ions across the cell membrane leads to a change in membrane potential, i.e. to a change in the potential difference between intracellular and extracellular space. If the membrane potential reaches a critical value of around -50 μV a new action potential is triggered, and information is transmitted via the axon to other neurons.

The signals measured with the EEG are thought to be mainly an effect of information processing at pyramidal neurons located in the cerebral cortex (Martin, 1991). Pyramidal neurons have a pyramid-like soma and large apical dendrites, oriented perpendicular to the surface of the cortex (see Fig. 2.3). Activation of an excitatory synapse at a pyramidal cell leads to an excitatory postsynaptic potential, i.e. a net inflow of positively charged ions. Consequently, increased extracellular negativity can be observed in the region of the synapse. The extracellular negativity leads to extracellular positivity at sites distant from the synapse and causes extracellular currents flowing towards the region of the synapse. The temporal and spatial summation of such extracellular currents, at hundreds of thousands of neurons with parallel oriented dendrites, leads to the changes in potential that are visible in the EEG. The polarity of the EEG signals depends on the type of synapses being activated and on the position of the synapses. As shown in Fig. 2.3, activation of excitatory synapses in superficial cortical layers corresponds to negative surface-potentials. Activation of ex-

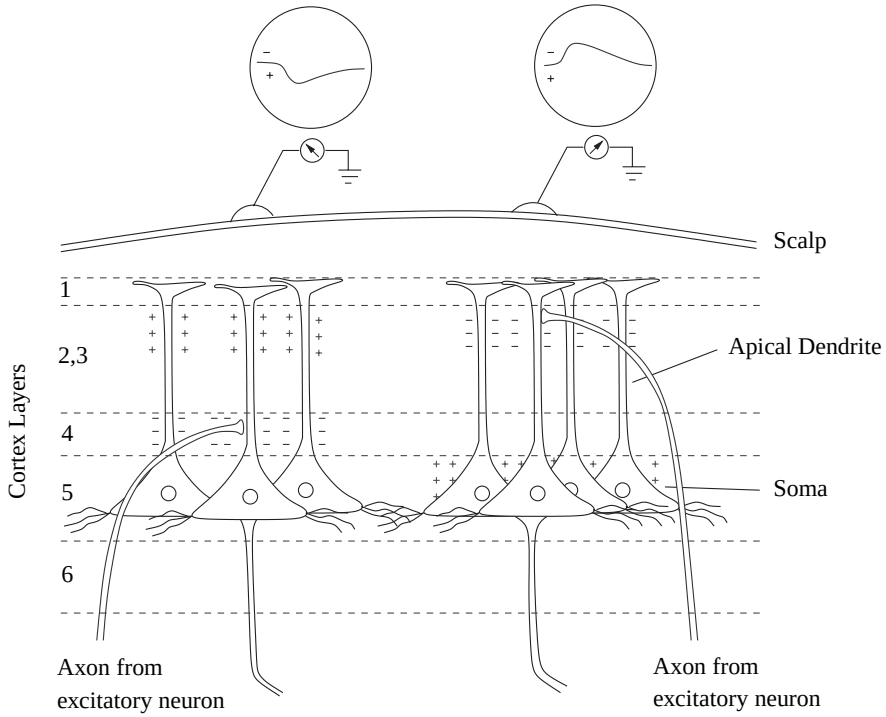

Figure 2.3 — Generators of the EEG. Pyramidal neurons in cerebral cortex receive excitatory input from synapses close to their soma (left) or from synapses connecting to their apical dendrites in superficial layers of the cortex (right). Excitation leads to a net inflow of positively charged ions and thus to an increased extracellular negativity in the region of the synapses. Extracellular currents flow towards the region of the synapse and cause an increased positivity at the dendrite (left) or at the soma (right). The extracellular currents lead to changes in potential on the scalp surface (adapted from Kaper (2006); Martin (1991)).

citatory synapses connecting close to the soma of a cell corresponds to positive surface-potentials. For inhibitory synapses the inverse is true: activation of synapses in superficial cortical layers corresponds to positive surface-potentials, and activation of synapses connecting close to the soma of a cell corresponds to negative surface-potentials. Without knowledge about the spatial distribution of synapses the type of synaptic action can thus not be inferred from the polarity of surface potentials (Martin, 1991).

The potential changes associated to extracellular currents at pyramidal neurons are mostly visible at electrodes placed over the active brain area. However, due to volume conduction in the cerebrospinal fluid, skull, and scalp, signals from a local ensemble of neurons also spread to distant electrodes. The potentials caused by the activity of a typical cortical macrocolumn (of diameter 3-4 mm) can spread to scalp electrodes that are up to 10 cm away from the macrocolumn (Srinivasan, 1999). A further effect of the tissue barrier between electrodes and neurons is that low-amplitude activity at frequencies of more than 40 Hz is practically invisible in the EEG. The EEG thus is a global measurement of brain activity. Consequently, it is difficult to use the EEG for drawing conclusions about the activity of small brain regions, let alone the activity of single neurons.

In addition to the effects of volume conduction, the analysis of the EEG is further complicated

by the presence of artifacts. Artifacts can be due to physiological or nonphysiological sources. Physiological sources for artifacts include eye movements and eye blinks, muscle activity, heart activity, and slow potential drifts due to transpiration. Nonphysiological sources for artifacts include power supply line noise (at 50 Hz or 60 Hz), noise generated by the EEG amplifier, and noise generated by sudden changes in the properties of the electrode-scalp interface. Artifacts often have much larger amplitude than the signals of interest. Therefore, artifact removal and filtering procedures have to be applied before an analysis of EEG signals can be attempted.

Despite the above mentioned shortcomings the EEG remains one of the most interesting methods for measuring electrical brain signals. It has been used in a variety of BCI systems and is also the measurement technique employed in this thesis. Besides BCI there are many other clinical and research applications of the EEG. These include research on different sleep stages, epilepsy monitoring, coma outcome prognosis, and many other, more theoretical, research questions.

2.2.2 Electrocorticogram

The ECoG is an invasive technique for recording electrical potentials in the brain. In a surgical procedure an array of electrodes, typically an 8×8 grid, is placed on the cortex surface. After the implantation, signals which are generated by the same mechanisms as the EEG can be measured. However, effects of volume conduction are less visible in the ECoG, i.e. the signals are less spatially blurred than EEG signals. Further advantages are that ECoG signals are barely contaminated with muscle or eye artifacts and that activity in frequencies up to about 100 Hz can be easily observed.

Due to the above mentioned positive properties, ECoG signals have generated a considerable deal of interest in the BCI community. Different experiments have been performed, mainly with epilepsy patients having ECoG arrays implanted over a period of one or two weeks for localization of epileptic foci or for presurgical monitoring purposes. The experiments have shown that patients can quickly learn to accurately control their ECoG signals through motor imagery (Graimann *et al.*, 2004; Hill *et al.*, 2006), motor and speech imagery (Leuthardt *et al.*, 2006), mental calculation (Ramsey *et al.*, 2006), or auditory imagery (Wilson *et al.*, 2006). This makes ECoG an interesting alternative to the EEG, however tests with severely handicapped subjects and research on the long term tissue compatibility of ECoG should be performed to validate the results.

2.2.3 Microelectrode Arrays

Microelectrode arrays are a technique for recording activity from single neurons or from small groups of neurons. As for ECoG, brain-surgery is necessary before signals can be recorded. The difference to ECoG is that electrodes are inserted *in* the cortex, i.e. the cortical tissue is penetrated by needle-like electrodes. A typical microelectrode array has a size of about 5×5 mm and contains around 100 electrodes, which can penetrate the cortex to a depth of several millimeters (Nicolelis *et al.*, 2003). Due to the invasive procedure that is needed to record signals, microelectrode arrays have been mainly tested in animal models (for example rhesus monkeys). An exception is the BCI system described by Hochberg *et al.* (2006), which is based on signals from microelectrode arrays implanted in human tetraplegic subjects.

Compared with other technologies for measuring brain activity, the advantages of microelectrode arrays are that signals are acquired at high spatial resolution and that the activity of single neurons can be detected. Recording the activity of neurons in the motor areas of the brain allows for complex applications such as realtime 3D control of a robot arm (Taylor *et al.*, 2002) which are difficult to realize with other measurement technologies. The disadvantage of microelectrode arrays is that brain-surgery is needed before signals can be recorded. During surgery an infection risk exists and moreover the reaction of brain tissue to the implanted electrode array is not well understood (Polikov *et al.*, 2005). Due to the death of neurons in the vicinity of the microelectrodes signal quality decays over time and data can only be recorded for a period of several months.

Despite the problems arising from the invasive nature of the measurements, microelectrode arrays are – together with the EEG – one of the most often used tools in BCI research. Current research issues and recent developments are nicely summarized in the review of Lebedev and Nicolelis (2006).

2.2.4 Other Methods for Measuring Brain Activity

- *Magnetoencephalogram*
  The magnetoencephalogram (MEG) is a noninvasive measurement of small changes ($\approx 10^{-15}$ Tesla) in magnetic field strength, which are caused by intracellular currents at pyramidal neurons. A small number of experimental studies have used the MEG in BCI systems (Kauhanen *et al.*, 2006; Lal *et al.*, 2005). These studies showed that MEG signals can be used for brain-computer interfacing and that a communication speed comparable to that of EEG based systems can be obtained. However, the equipment needed for MEG measurements is technically complex, expensive, and cannot be easily transported from one place to another. This rules out the use of current MEG devices in practical BCIs.

- *Functional Magnetic Resonance Imaging*
  Functional magnetic resonance imaging (fMRI) allows to noninvasively measure the so-called blood oxygen level dependent (BOLD) signal. The BOLD signal does not directly represent neuronal activation but rather depends on the level of oxygenated and deoxygenated hemoglobin and on the hemodynamic response to neuronal activation. The peak of the BOLD signal is typically very broad and observed four to five seconds after the neuronal activation, i.e. the temporal resolution of fMRI is relatively low when compared to methods that directly measure electrical brain activity. The spatial resolution is very good, structures of the size of a few millimeters can be localized with the fMRI. In addition, signals can be acquired from the whole brain and not only from the cortex, as for example with the EEG. In BCI research fMRI has been used in basic proof of concept systems (Weiskopf *et al.*, 2004; Yoo *et al.*, 2004) and to elucidate the brain mechanisms underlying successful self regulation of brain activity (Hinterberger *et al.*, 2003). To date, the use in practical BCI systems is impossible because fMRI devices are technically demanding, expensive and cannot be easily moved from one place to another.

- *Near-Infrared Spectroscopy*
  Near infrared spectroscopy (NIRS) is a noninvasive method to measure the hemodynamic activity of the cortex (similar to the BOLD signal). To measure NIRS signals, optodes emitting light in the near-infrared range are placed on the scalp of subjects. The emitted light is reflected by the cortical surface and measured by detector optodes. The light intensity measured at the detector optodes varies as a function of the amount of oxygenated and deoxygenated hemoglobin in the blood and thus allows to measure brain activity. NIRS provides a relatively low spatial resolution, and because hemodynamic brain activity is measured the temporal resolution is also low. Several publications describe the use of NIRS signals to classify different types of motor imagery in BCI systems (Coyle *et al.*, 2004; Sitaram *et al.*, 2007). These studies are proof of concept studies and further development is needed to make NIRS a real alternative for everyday use in a BCI.

Summary

Different methods to measure brain activity can be used in a BCI. The characteristics of the methods we reviewed are summarized in Table 2.1. As can be seen, each method has its own advantages and disadvantages and hence so far no method of choice exists. Consequently, BCI research will probably continue to explore the possibilities of all methods and real-world BCI applications based on different methods might emerge. Depending on their needs and on their willingness to undergo brain surgery, users will choose one of the methods. Clearly, the development of new methods for measuring brain activity has the potential to yield advanced BCI systems.

2.3 Neurophysiologic Signals

An ideal BCI system would directly detect every wish or intention of its user and perform the corresponding action. However, it is very difficult to clearly define how wishes or intentions are related to neurophysiologic signals. Consequently, it is virtually impossible to detect the intentions and wishes of a user from his brain activity. This is why present day BCI systems achieve only a much

Method	Measured Quantity	Invasive?	Spatial Resolution	Temporal Resolution	Equipment portable?
EEG	Electric Potentials	No	-	++	Yes
ECoG	Electric Potentials	Yes	+	++	Yes
Microel. Arr.	Electric Potentials	Yes	++	++	Yes
MEG	Magnetic Fields	No	+	++	No
fMRI	Hemodynamic Act.	No	++	–	No
NIRS	Hemodynamic Act.	No	- -	–	Yes

Table 2.1 — Methods for measuring brain activity. Characteristics important for practical BCI systems are indicated. The relative spatial and temporal resolution of the different methods is indicated with symbols ranging from - - (very low) to ++ (very high).

less ambitious goal, namely the discrimination of two to five different categories of neurophysiologic signals, or the mapping of neurophysiologic signals to continuous 1D, 2D, or 3D movements.

To allow for discrimination of different neurophysiologic signals or for mapping of such signals to movements, users have to acquire conscious control over their brain activity. Two fundamentally different approaches exist to achieve this. In the first approach subjects perceive a set of stimuli displayed by the BCI system and can control their brain activity by focusing onto one specific stimulus. The changes in neurophysiologic signals resulting from perception and processing of stimuli are termed event-related potentials (ERPs) and are discussed together with the corresponding BCI paradigms in Section 2.3.1. In the second approach users control their brain activity by concentrating on a specific mental task, for example imagination of hand movement can be used to modify activity in the motor cortex. In this approach feedback signals are often used to let subjects learn the production of easily detectable patterns of neurophysiologic signals. The types of signals resulting from concentration on mental tasks together with the corresponding BCI paradigms are described in Sections 2.3.2, 2.3.3, and 2.3.4.

2.3.1 Event-Related Potentials

ERPs are stereotyped, spatio-temporal patterns of brain activity, occurring time-locked to an event, for example after presentation of a stimulus, before execution of a movement, or after the detection of a novel stimulus. Traditionally, ERPs are recorded with the EEG and have been used in neuroscience for studying the different stages of perception, cognition, and action. Note that event-related changes can also be measured with other signal acquisition techniques like the MEG or fMRI, this is however not described here because in a BCI usually the EEG is used for measuring such changes.

- *P300*
  The P300 is a positive deflection in the EEG, appearing approximately 300 ms after the presentation of rare or surprising, task-relevant stimuli (Sutton *et al.*, 1965). To evoke the P300, subjects are asked to observe a random sequence of two types of stimuli. One stimulus type (the oddball or target stimulus) appears only rarely in the sequence, while the other stimulus type (the normal or nontarget stimulus) appears more often. Whenever the target stimulus appears, a P300 can be observed in the EEG. This principle was exploited by Farwell and Donchin (1988) in a BCI system. They described the P300 speller in which a matrix containing symbols from the alphabet is displayed on a screen. Rows and columns of the matrix are flashed in random order, and flashes of the row or column containing the desired symbol constitute the oddball stimulus, while all other flashes constitute nontarget stimuli. Since the seminal paper of Farwell and Donchin many studies about P300-based BCI systems have appeared. A review of these studies and a more detailed description of the P300 are provided in Chapter 4.

- *Steady-State Visual Evoked Potentials (SSVEPs)*
  SSVEPs are oscillations observable at occipital electrodes, induced by repetitive visual stimulation. Stimulation at a certain frequency leads to oscillations at the same frequency and at harmonics and subharmonics of the stimulation frequency (Herrmann, 2001). In a BCI,

SSVEPs are used by simultaneously displaying several stimuli flickering at different frequencies. Users can select one stimulus by focusing on it, which leads to increased amplitude in the frequency bands corresponding to the flickering frequency of the stimulus. Systems employing this principle are described in (Gao *et al.*, 2003; Lalor *et al.*, 2005; Middendorf *et al.*, 2000)

- *Motor-Related Potentials (MRPs)*
  Other than the previously described signals, motor-related potentials (MRPs) are independent of the perception or processing of stimuli. The events to which MRPs are related are the preparation or imagination of movements. MRPs are slow negative potentials, observable over the sensorimotor cortex before movement onset or during movement imagination. Since the sensorimotor cortex has a somatotopic organization the body part that will be moved, or for which a movement is imagined, can be inferred from the location of greatest amplitude of the MRP. This phenomenon has been used in combination with sensorimotor rhythms (see Section 2.3.2) in a BCI based on motor imagery (Dornhege *et al.*, 2004).

2.3.2 Oscillatory Brain Activity

Sinusoid like oscillatory brain activity occurs in many regions of the brain and changes according to the state of subjects, for example between wake and sleep or between concentrated work and idling. Oscillatory activity in the EEG is classified into different frequency bands or rhythms. Typically observable are the delta (1 - 4 Hz), theta (4 -8 Hz), alpha and mu (8 - 13 Hz)[1], beta (13 - 25 Hz), and gamma (25 - 40 Hz) rhythms.

- *Sensorimotor Rhythms*
  Mu-rhythm oscillations can be observed over the sensorimotor cortex when a subject does not perform movements. These oscillations are decreased in amplitude when movements of body parts are imagined or performed. In addition, imagined or performed movements of body parts lead to changes in beta-rhythm amplitude. The changes in the mu- and beta-rhythm are localized over the part of the sensorimotor cortex corresponding to the body part, and so imagined movements of different body parts can be discriminated. For example imagination of movement of the left hand corresponds to a decrease in mu-rhythm amplitude over the right sensorimotor cortex, whereas imagination of movement of the right hand corresponds to a decrease in amplitude over the left sensorimotor cortex. The changes in sensorimotor rhythms occurring in untrained users are usually not strong enough to be detected by a classification algorithm, and thus feedback training has to be used to let users acquire control over sensorimotor rhythms.

  BCI systems employing imagined movements of hands, feet, or tongue have been mainly introduced by the research group of Pfurtscheller in Austria (Pfurtscheller and Neuper, 2001)). The group of Wolpaw in the United States has also worked on such systems, and an impressive sensorimotor rhythm BCI allowing for fast control of a 2D cursor has been described by

[1] The term mu-rhythm is used for oscillatory activity with a frequency of about 10 Hz, localized over the sensorimotor cortex. The term alpha-rhythm is more general and can be used for any activity in the frequency range 8 - 13 Hz.

Wolpaw and McFarland (2004). Many other groups have performed research on such systems. For example research has been targeted on improving machine learning algorithms for classification of sensorimotor rhythms (Blankertz *et al.*, 2006b), on measuring and classifying neurophysiologic signal related to motor imagery with the help of NIRS (Coyle *et al.*, 2004; Sitaram *et al.*, 2007), ECoG (Graimann *et al.*, 2004; Leuthardt *et al.*, 2006), MEG (Kauhanen *et al.*, 2006), and on testing sensorimotor rhythm interfaces with severely handicapped subjects (Kübler *et al.*, 2005).

- *Other Oscillatory Activity*
  Cognitive tasks other than motor imagery can also be used to trigger changes in oscillatory brain activity. Examples for such tasks are mental calculation, auditory imagery, imagery of spatial navigation, or imagination of rotating geometric objects (Curran *et al.*, 2004; Garcia, 2004; Keirn and Aunon, 1990). The classification accuracy for such cognitive tasks seems to be comparable to that achievable with motor imagery. In addition, depending on the preferences of the users the alternative cognitive tasks might be easier to perform than motor imagery (Curran *et al.*, 2004). However, before such tasks can be routinely used in BCI systems, further research about the underlying neurophysiological mechanisms and tests with larger populations of subjects are necessary.

2.3.3 Slow Cortical Potentials

Slow cortical potentials (SCPs) are slow voltage shifts in the EEG occurring in the frequency range 1-2 Hz. Negative SCPs correspond to a general decrease in cortical excitability. Positive SCPs correspond to a general increase in cortical excitability. Through feedback training subjects can learn to voluntarily control their SCPs. The voluntary production of negative and positive SCPs has been exploited in one of the earliest BCI systems for disabled subjects (Birbaumer *et al.*, 1999). In their pioneering work, Birbaumer *et al.* showed that patients suffering from amyotrophic lateral sclerosis (ALS) can use a BCI to control a spelling device and to communicate with their environment. In other publications from the same group many different aspect related to SCPs were investigated, for example the use of self regulation of SCPs as a treatment for children with attention-deficit/hyperactivity disorder (Strehl *et al.*, 2006).

2.3.4 Neuronal Ensemble Activity

Action potentials are thought to be the basic unit of information in the brain and enable communication between different neurons. The number of action potentials per time (the firing rate) can be used in a BCI to predict the behavior of a subject. For example the firing rate of neurons in the motor and premotor-cortices can be used to predict hand positions or hand velocities. To make these predictions more reliable, usually the firing rates from ensembles of neurons, i.e. from populations of hundreds of neurons, are used to predict subject behavior. Furthermore, through feedback training subjects can learn to modulate the firing rates of neurons in the motor cortex. Neuronal ensemble activity can thus be employed as neurophysiological signal in BCIs, in particular in BCIs using microelectrode arrays (Hochberg *et al.*, 2006; Serruya *et al.*, 2002; Taylor *et al.*, 2002).

Summary

Different neurophysiologic signals can be used to drive a BCI. The advantage of ERPs is that no user training is necessary because ERPs occur as a natural response of the brain to stimulation. This might be of particular importance for subjects with concentration problems or for subjects not willing to go through a long training phase. A disadvantage is that communication depends on the presentation and perception of stimuli. Subjects are thus required to have remaining cognitive abilities. Moreover, BCI systems based on ERPs have only limited application scenarios because a device to present stimuli is needed and because users need to pay attention to stimuli, even in the presence of other unrelated, distracting stimuli.

If oscillatory activity or SCPs are used, more flexible BCI systems can be imagined because no computer screen or other device is needed to present stimuli. However, to gain voluntary control over brain activity, subjects have to perform feedback training, and it can take several weeks before subjects are able to reliably control a BCI. Therefore, BCI systems based on oscillatory activity or SCPs might be less suited for subjects with concentration problems or for subjects who are not willing to go through a long training phase.

The amount of training in systems using ECoG to measure brain activity or in systems using neuronal ensemble activity tends to be smaller than in other systems because the recorded signals have a better signal to noise ratio. A further advantage is that such systems, especially systems based on neuronal ensemble signals, allow for control of more complex applications than systems using the EEG. As already mentioned in Section 2.2.3, the biggest disadvantage of such systems is the brain surgery that is necessary before signals can be recorded.

Taken together, no paradigm for controlling and measuring neurophysiologic signals clearly rules out all other paradigms. Different paradigms will thus probably coexist. In specific cases paradigms have to be chosen depending on the abilities of the user, the application scenario, and the willingness of the user to undergo brain surgery.

2.4 Extracting Features from Neurophysiologic Signals

In the previous section we have discussed paradigms that let users control their brain activity and the neurophysiologic signals corresponding to the respective paradigms. To allow actual control of a BCI, the neurophysiologic signals have to be mapped to values that allow to discriminate different classes of signals, i.e. the neurophysiologic signals have to be classified.

The first step underlying most methods for classification of neurophysiologic signals is to acquire labeled training data. Acquiring labeled training data means that the subject has to perform prescribed actions, while neurophysiologic signals are recorded. Then, a computer is used to learn the desired mapping from signals to classes.

After the data acquisition phase, machine learning algorithms are applied to infer functions that can be used to classify neurophysiologic signals. For reasons of practicality and simplicity, machine learning algorithms are usually divided into two modules: feature extraction and classification. The feature extraction module serves to transform raw neurophysiologic signals into a representation that makes classification easy. In other words, the goal of feature extraction is to remove noise and

other unnecessary information from the input signals, while at the same time retaining information that is important to discriminate different classes of signals. Another, related, goal of feature extraction is to reduce the dimensionality of the data that has to be classified. After feature extraction, machine learning algorithms are used to solve two tasks. During training, the task is to infer a mapping between signals and classes. For this, the labeled feature vectors produced by the feature extraction module are used. During application of a BCI, the task is to discriminate different types of neurophysiologic signals and hence to allow for control of a BCI.

In this section we only review methods for feature extraction in BCIs. Machine learning algorithms are one of the main themes of this thesis and are described in a separate chapter (cf. Chapter 3).

To achieve the goals of feature extraction, neurophysiological a priori knowledge about the characteristics of the signals in the temporal, the frequency, and the spatial domain is used. Depending on the type of signals to be classified this knowledge can take many different forms. Consequently many different feature extraction methods have been described. Some basic and often used methods are described below. A more exhaustive review of feature extraction methods for BCIs can be found in (Bashashati *et al.*, 2007).

2.4.1 Time Domain Features

Time domain features are related to changes in the amplitude of neurophysiologic signals, occurring time-locked to the presentation of stimuli or time-locked to actions of the user of a BCI system. Good examples for signals that can be characterized with the help of time domain features are the P300, SCPs, and MRPs. A strategy that is often used to separate these signals from background activity and noise is lowpass or bandpass filtering, optionally followed by downsampling. This strategy is reasonable because most of the energy of the P300, SCPs, and MRPs is concentrated at low frequencies. Lowpass filtering, together with downsampling thus allows to remove unimportant information from high frequency bands. In addition, the dimensionality of the signals is reduced. Examples for systems in which filtering and downsampling have been employed are the P300 BCI described by Sellers and Donchin (2006), the SCP based system described by Birbaumer *et al.* (1999), and the system for classification of MRPs described by Blankertz *et al.* (2002).

An alternative to filtering is to use the wavelet transform of the signals. Systems based on the discrete wavelet transform (DWT), as well as systems based on the continuous wavelet transform (CWT) have been described in the literature. A crucial step in systems using wavelets is to select a subset of wavelet coefficients that is relevant for classification. This is equivalent to selecting regions in the time-frequency plane at which signals can be classified with high accuracy and can be achieved with the help of feature selection algorithms. An example for the use of the DWT is the P300-based BCI system described by Donchin *et al.* (2000). In this system Daubechies wavelets were used for feature extraction, and relevant wavelet coefficients were selected with stepwise discriminant analysis (SWDA). An example for the use of the CWT is the algorithm described by Bostanov (2004). In this algorithm the Mexican hat wavelet was used for feature extraction from P300 and SCP signals, and a t-test was used to select relevant wavelet coefficients.

Besides the use for the EEG signals P300, SCP, and MRP, time domain features are also used in BCI systems based on microelectrode arrays. A feature that is often used in such systems is

the number of spikes occurring in a certain time interval. Many different techniques for counting spikes and for sorting spikes recorded with the same electrode from several neurons exist. These techniques will however not be further discussed here.

2.4.2 Frequency Domain Features

Frequency domain features are related to changes in oscillatory activity. Such changes can be evoked by presentation of stimuli or by concentration of the user on a specific mental task. Since the phase of oscillatory activity is usually not time-locked to the presentation of stimuli or to actions of the user, time domain feature extraction techniques cannot be used. Instead, feature extraction techniques that are invariant to the exact temporal evolution of signals have to be used.

The most commonly used frequency domain features are related to changes in the amplitude of oscillatory activity. For example in systems based on motor imagery, the bandpower in the mu and beta frequency bands at electrodes located over the sensorimotor cortex is used as a feature. In the case of SSVEPs, band power in the harmonics of the visual stimulation frequency at occipital electrodes can be used as a feature. To estimate band power, different methods have been used. These include Welch's method (Lalor *et al.*, 2005), adaptive autoregressive models (Schlögl *et al.*, 2005), and Morlet wavelets (Lemm *et al.*, 2004).

A second type of frequency domain features is related to the synchronization between signals from different brain regions. Synchronization of signals from different brain regions might indicate that these regions communicate. This permits to discriminate cognitive tasks involving communication between different brain regions. The use of synchronization features in combination with band power features was explored by Gysels and Celka (2004) in a three-class BCI based on the cognitive tasks "left hand movement", "right hand movement", and "composition of words". Brunner *et al.* (2006) used synchronization features in combination with band power features in a four-class BCI based on the cognitive tasks "left hand movement", "right hand movement", "foot movement", and "tongue movement". In both studies, classification with acceptable accuracy was possible with synchronization features alone. Combining synchronization and band power features led to classification accuracy that was superior to that obtained with only synchronization or band power.

2.4.3 Spatial Domain Features

The feature extraction techniques described so far all work with univariate time series, i.e. data from only one electrode is used (an exception are synchronization features, extracted from bivariate time series). In many systems however, data from more than one electrode is available. Hence, the features extracted from several electrodes have to be combined in an efficient way. Finding efficient combinations of features from more than one electrode is the goal of spatial feature extraction methods.

The probably simplest way for performing spatial feature extraction is to use only electrodes that carry useful information for discrimination of a given set of cognitive tasks. The reasoning behind such an approach is that changes in band power, P300 peaks, or other features do not occur uniformly at all electrodes but are usually stronger at electrodes over brain regions implied in the

respective cognitive task. Electrodes can be selected manually or by using an algorithm that automatically selects an optimal electrode subset. Due to its simplicity the former approach has been used in almost all types of BCIs. The latter, more complex approach has been used for classification of data recorded with a sensorimotor rhythm paradigm (Lal *et al.*, 2004), for classification of P300 data (Rakotomamonjy *et al.*, 2005), and in this thesis (cf. Chapter 5).

A spatial feature extraction method that can be used in addition to electrode selection, consists in applying spatial filtering algorithms before further processing takes place. Spatial filtering corresponds to building linear combinations of the signals measured at several electrodes. Denoting by $\mathbf{s}(t) \in \mathbb{R}^E$ the signal from E electrodes at time t, spatial filtering can be expressed as follows:

$$\hat{\mathbf{s}}(t) = \mathbf{C}\mathbf{s}(t). \tag{2.1}$$

Here the $F \times E$ matrix $\mathbf{C}$ contains the coefficients for F spatial filters and the vector $\hat{\mathbf{s}}(t) \in \mathbb{R}^F$ contains the spatially filtered signals at time t.

To determine the filter coefficients different methods can be used. For example for motor imagery based BCIs, it has been shown that spatial filtering with a Laplacian filter can increase performance (McFarland *et al.*, 1997). Simple Laplacian filters can be built by subtracting the mean signal of the surrounding electrodes from the signal of each electrode. Applying a Laplacian filter corresponds to spatial high-pass filtering, focal activity which is characteristic for motor imagery tasks is thus enhanced.

In other methods for spatial feature extraction, filter coefficients are computed from a set of training data. An algorithm which is very popular in the area of motor imagery based BCI systems is the common spatial patterns (CSP) algorithm (Ramoser *et al.*, 2000). The CSP algorithm determines spatial filters that maximize the temporal variance of data recorded under one condition and minimize the temporal variance of data recorded under a second condition. The success of CSP stems from the fact that temporal variance, i.e. power, in the mu and beta frequency bands is an important feature for the classification of EEG signals recorded during motor imagery. Note that the CSP algorithm uses labeled training data and hence overfitting can occur when a large number of electrodes is used in conjunction with a small amount of training data (see Chapter 3 for a description of the overfitting phenomenon).

Another method for computing the coefficients of spatial filters from training data is independent component analysis (ICA). In ICA algorithms it is assumed that a set of multichannel signals $\mathbf{s}$(t) is generated by linearly mixing a set of source signals $\mathbf{x}$(t):

$$\mathbf{s}(t) = \mathbf{M}\mathbf{x}(t). \tag{2.2}$$

The goal is to compute a matrix $\mathbf{F}$ that allows one to reconstruct the source signals $\mathbf{x}$ by multiplying $\mathbf{s}$ with $\mathbf{F}$. To achieve this without having information about $\mathbf{M}$, one assumes that the source signals are statistically independent. The ICA algorithm thus computes $\mathbf{F}$ such that the signals $\mathbf{s}$(t) multiplied with $\mathbf{F}$ are maximally independent. In the case of EEG signals, the idea underlying the application of ICA is that the signals measured on the scalp are a linear and instantaneous mixture of signals from independent sources in the cortex, deeper brain structures, and noise (Makeig *et al.*, 1996). ICA has been mainly used in P300-based BCIs as a feature extraction method. In such systems ICA is used to separate multichannel EEG into several components, corresponding to sources in

the brain or noise, for example from eye blinks. By retaining only components that have a P300 like spatial distribution or show P300 like waveforms, the signal to noise ratio can be improved. Consequently, classification can be performed with improved accuracy.

Summary

The goal of applying feature extraction algorithms is to transform raw neurophysiologic signals into a representation suitable for subsequent classification. To this end, a priori knowledge about the characteristics of signals produced in different paradigms is employed. This knowledge can concern the characteristics of signals in the temporal, the frequency, or the spatial domain. Note that while the three types of domains have been discussed separately above, it is also possible to combine features from several domains. An example for this is spatial feature extraction which is often preceded by bandpass filtering. Another example is the combination of temporal and frequency domain features as proposed by Dornhege *et al.* (2003). Such combinations of features have the potential to increase classification accuracy.

Note that feature extraction is only the first step in the mapping from neurophysiologic signals to brain states. The second step is to classify the features. Algorithms for classification and for learning of classification rules are described in detail in Chapter 3. For the moment lets us assume that we have an algorithm at hand that can perform classification. Such an algorithm can be used to build BCI applications, which are described next.

2.5 Applications

In theory any device that can be connected to a computer or to a microcontroller could be controlled with a BCI. In practice however, the set of devices and applications that can be controlled with a BCI is limited. To understand this, one has to consider that the amount of information which can be transmitted with present day BCI systems is limited. The typical information transfer rate achievable with an EEG based BCI is about 20 to 40 bits/min. As an additional obstacle most present day BCI systems function only in synchronous mode. In synchronous mode, communication is possible only during predefined time intervals. This means the system tells the user when it is ready to receive the next command and limits severely the possible type of applications. In asynchronous mode users can send commands whenever they want, see for example the system described by Borisoff *et al.* (2006). Some of the applications possible with current BCIs are described below.

2.5.1 Spelling Devices

Spelling devices allow severely disabled users to communicate with their environment by sequentially selecting symbols from the alphabet. One of the first spelling devices mentioned in the BCI literature is the P300 speller (Farwell and Donchin, 1988) (see also Chapter 4). A system based on SCPs was described by Birbaumer *et al.* (1999). In their system the alphabet is split into two halves and subjects can select one halve by producing positive or negative SCPs. The selected halve is then again split into two halves and this process is repeated recursively until only one symbol remains. An advanced version of this system in which the relative frequency of letters in natural

language is taken into account is presented by Perelmouter and Birbaumer (2000). Systems based on sensorimotor rhythms are described by Scherer *et al.* (2004) and Blankertz *et al.* (2007).

2.5.2 Environment Control

Environment control systems allow to control electrical appliances with a BCI. Gao *et al.* (2003) describe a proof-of-concept environment control system based on SSVEPs. Bayliss (2003) describes the control of a virtual apartment based on the P300. To our knowledge none of the two aforementioned systems is asynchronous. Development of asynchronous BCI systems is probably the most important research topic to advance the area of environment control systems.

2.5.3 Wheelchair Control

Disabled subjects are almost always bound to wheelchairs. If control over some muscles remains, these can be used to steer a wheelchair. For example systems exist that allow to steer a wheelchair with only a joystick or with head movements. If no control over muscles remains, a BCI can potentially be used to steer a wheelchair. Because steering a wheelchair is a complex task and because wheelchair control has to be extremely reliable, the possible movements of the wheelchair are strongly constrained in current prototype systems. In the system presented by Rebsamen *et al.* (2006) the wheelchair is constrained to move along paths predefined in software joining registered locations, and a P300-based interface is used to select the desired location. In the system presented by Millan *et al.* (2004) a miniature robot can be guided through a labyrinth, based on oscillatory brain activity recorded with the EEG. Control of the robot is simplified by implementing a wall following behavior on the robot and allowing for turns only if there is an open doorway.

2.5.4 Neuromotor Prostheses

The idea underlying research on neuromotor prostheses is to use a BCI for controlling movement of limbs and to restore motor function in tetraplegics or amputees. Different types of neuromotor prostheses can be envisioned depending on the information transfer rate a BCI provides. If neuronal ensemble activity is used as control signal, high information transfer rates are achieved and 3D robotic arms can be controlled (Taylor *et al.*, 2002). If an EEG based BCI is used, only simple control tasks can be accomplished. For example in the system described by Pfurtscheller *et al.* (2005) sensorimotor rhythms were used to control functional electric stimulation of hand muscles and so to restore grasp function in a tetraplegic patient.

2.5.5 Gaming and Virtual Reality

Besides the applications targeted towards disabled subjects, prototypes of gaming and virtual reality applications have been described in the literature. Examples for such applications are the control of a spaceship with oscillatory brain activity (Garcia, 2004), the control of an animated character in an immersive 3D gaming environment with SSVEPs (Lalor *et al.*, 2005), and the control of walking in a virtual reality environment with sensorimotor rhythms (Leeb *et al.*, 2006).

Summary

Several application scenarios exist in which a BCI could be useful. However, so far no commercially available BCI application has emerged. This is possibly due to the fact that current technology does not allow to build BCI systems which can work in asynchronous mode and provide high information transfer rates. A possible approach to circumvent the problem of limited information transfer rates is to build intelligence into the application, i.e. to reduce the information needed by the application by cleverly restraining the number of commands possible in a given situation. Examples for applications in which such a strategy has been implemented are the advanced SCP based spelling device and the wheelchair control applications described above. Other problems, such as the restriction to asynchronous mode still have to be solved before practical BCI applications will appear.

2.6 Conclusion

The number of publications on BCI systems has grown quickly during the past years, and a considerable variety of prototypes can be found in the literature. Systems differ in the measurement technology used to acquire brain signals, in the neurophysiologic signals that are used, in the signal processing and machine learning algorithms, and in the target application. Despite the large number of approaches and despite results demonstrating the feasibility of communication and control with a BCI, none of the systems is commercially available and ready for daily use by disabled subjects.

However, it is probable that such systems will appear during the next years. Large advances could probably be made if new, noninvasive measurement technologies allowing for a detailed view into the brain would appear. Moreover, many studies investigate only isolated aspects of BCI systems such as for example the use of a new measurement technology or new application scenarios. Systematic studies of complete systems, investigating the dependencies between different components of a BCI systems, are largely missing and would probably serve to advance the field.

Review of Supervised Machine Learning for Brain-Computer Interfaces

3

3.1 Introduction

In a BCI complex neurophysiologic signals have to be translated into simple commands for a computer or other devices. The most straightforward approach to map signals to commands is probably to look at the distribution of a small number of simple features of the signals and to manually specify a translation rule. This method has indeed been used in early prototypes of BCIs. In the work described by Wolpaw *et al.* (1991), subjects could move a cursor up and down by modifying their mu-rhythm amplitude. To translate mu-rhythm amplitude into cursor movements, different voltage ranges were fixed manually by an operator, based on the characteristics of previously recorded signals. However, as noted by Wolpaw *et al.*, even if only one feature is used, it is difficult for a human to specify an optimal mapping between signals and commands. If more features are used, it quickly becomes impossible to manually design mappings. Moreover, neurophysiologic signals show a relatively large variance between subjects. This means that translation rules have to be specified for each new subject that wants to access a BCI.

A solution to these problems that is used in almost all BCI systems, is to first acquire labeled training data from a subject before it can use the system. Then, a computer is used to learn the mapping between signals and commands. Acquiring labeled training data means that the subject performs prescribed actions, while neurophysiologic signals are recorded. For example in a mu-rhythm BCI, the result of training can be a set of trials in which the subject has imagined left hand movement and another set of trials in which the subject has imagined right hand movement. After the training phase a supervised machine learning algorithm is used to learn the desired mapping of neurophysiologic signals into commands.

Below we review supervised machine learning algorithms that have been used for BCIs. In the first part of the chapter, we explain basic concepts of supervised machine learning in a nonproba-

bilistic framework and then describe probabilistic learning algorithms (Sections 3.2 and 3.3). The second part of the chapter contains a review and discussion of machine learning algorithms that have been used in BCI systems (Section 3.4).

Other reviews of machine learning methods for BCIs can be found in (Bashashati *et al.*, 2007; Lotte *et al.*, 2007; Vesin *et al.*, 2006). Good general introductions to machine learning are given in the books of Bishop (2006) and Hastie *et al.* (2001).

3.2 Basic Concepts

Algorithms that learn from a set of training examples how to map inputs to desired outputs are called supervised learning algorithms[1]. The training examples are pairs $(\mathbf{x}, y)$ of inputs $\mathbf{x} \in \mathcal{X}$ and desired outputs $y \in \mathcal{Y}$. In general $\mathcal{X}$ can be an arbitrary set, however often the inputs are vectors with real-valued entries computed with the help of a feature extraction method, i.e. $\mathcal{X} = \mathbb{R}^D$. The set of outputs $\mathcal{Y}$ can be an arbitrary set too, however one often uses $\mathcal{Y} = \{1 \dots K\}$, or $\mathcal{Y} = \mathbb{R}$. If $\mathcal{Y} = \{1 \dots K\}$ the outputs are qualitative measurements and the task to be solved is a classification task, i.e. inputs have to be mapped to one of K different classes. If $\mathcal{Y} = \mathbb{R}$ the outputs are quantitative measurements and the task to be solved is a regression task, i.e. inputs have to be mapped to real output values. In both cases the problem a learning algorithm has to solve is to choose, based on the training examples, a function $\mathrm{f} : \mathcal{X} \to \mathcal{Y}$ from a family of functions $\mathcal{F}$, such that new examples, not contained in the training set, are correctly mapped to the corresponding output. For practical reasons the family of functions $\mathcal{F}$ is usually indexed by a set of parameters $\boldsymbol{\theta}$, i.e. $y = \mathrm{f}(\mathbf{x}; \boldsymbol{\theta})$. Hence, the task of choosing a function is equivalent to choosing parameters $\boldsymbol{\theta}$.

To formalize the notion of learning from training data, it is convenient to assume that pairs of inputs and outputs are drawn independently and identically (i.i.d) from a probability distribution $p(\mathbf{x}, y)$. This assumption can be motivated by imagining a fixed deterministic relationship between inputs and outputs, together with i.i.d noise in the measurement of inputs and outputs. To evaluate the cost of using a specific function f for predicting outputs from inputs, a loss function $\mathrm{l} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_0^+$ that measures the cost of mapping an input vector to a specific output is introduced. The expected cost (or risk) R associated to a function f can be written as:

$$\mathrm{R}(\mathrm{f}) = \int \mathrm{l}(y, \mathrm{f}(\mathbf{x}))p(\mathbf{x}, y)\, d\mathbf{x}dy. \tag{3.1}$$

A simple example for a loss function is the $0/1$ loss

$$\mathrm{l}(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{if } y \neq \hat{y}, \end{cases} \tag{3.2}$$

the risk R is then the average number of classification errors. It can be shown that minimizing the $0/1$ loss is equivalent to predicting for each input the output with the maximal, i.e.:

$$\mathrm{f}(\mathbf{x}) = \arg\max_{\mathrm{y} \in \mathcal{Y}} \mathrm{p}(\mathrm{y}|\mathbf{x}). \tag{3.3}$$

[1] One can imagine that a teacher or supervisor indicates the desired output for each training example, hence the term supervised learning algorithm.

Another simple loss function, which is often used for regression tasks, is the squared error

$$\mathrm{l}(y, \hat{y}) = (y - \hat{y})^2, \tag{3.4}$$

the risk R is then the variance of the estimated outputs $\hat{y}$ around the true outputs y. It can be shown that minimizing the squared error loss is equivalent to predicting for each input the conditional expectation of the output:

$$\mathrm{f}(\mathbf{x}) = \int y p(y|\mathbf{x}) dy. \tag{3.5}$$

The problem in supervised machine learning is that the distribution p is unknown. Hence, simple solutions for minimizing the risk, such as those in Equations 3.3 and 3.5, cannot be used. Usually the only information we have about p is a set of training examples $(\mathbf{x}_i, y_i), i \in \{1 \dots N\}$. Therefore, it is impossible to directly search for a function f minimizing the expected risk R. A possible solution to this problem is to use empirical risk minimization, i.e. to use the empirical risk $\hat{\mathrm{R}}$, the average cost on the training set, as criterion for selecting a good function g:

$$\begin{aligned} \hat{\mathrm{R}}(\mathrm{f}) &= \frac{1}{N} \sum_{i=1}^{N} \mathrm{l}(y_i, \mathrm{f}(\mathbf{x}_i)) \\ \mathrm{g} &= \arg\min_{\mathrm{f} \in \mathcal{F}} \hat{\mathrm{R}}(\mathrm{f}). \end{aligned} \tag{3.6}$$

While using the empirical risk as a replacement for the expected risk intuitively seems to be a good idea, there are some severe problems associated to this approach. To understand this, let us assume for a moment that pairs of inputs and outputs are generated from a fixed linear function with slope a and intercept b and that i.i.d zero-mean Gaussian noise n_i corrupts the outputs:

$$y_i = a x_i + b + n_i. \tag{3.7}$$

Using for example the squared error loss function we can now easily use Equation 3.6 to fit functions f from a family of functions $\mathcal{F}$ to the training data. Let us consider a case in which we are given four training examples and in which $\mathcal{F}$ is chosen to be the family of polynomials of degree three.

$$\mathrm{f}(x; \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 \tag{3.8}$$

In this case we will always be able to find parameters θ that perfectly fit the training data. In other words, we can always find a polynomial of order three that goes through all the four training points (except for pathological cases in which two or more training examples have the same x value). However, the error made on examples not in the training set will be high, because a polynomial going through all training points tends to be a strongly oscillating function, while the function generating the data is linear (see Fig. 3.1).

The problem we just described is known as the *overfitting* problem in the machine learning literature. Overfitting means that there is a large difference between the risk R and the empirical risk $\hat{\mathrm{R}}$ and can occur due to several reasons. A first reason is, that not enough training examples are used. Clearly, in our example, if the training set would have been much larger, say $N = 20$, the danger of overfitting would have been strongly reduced (see Fig. 3.1). A second reason is, that

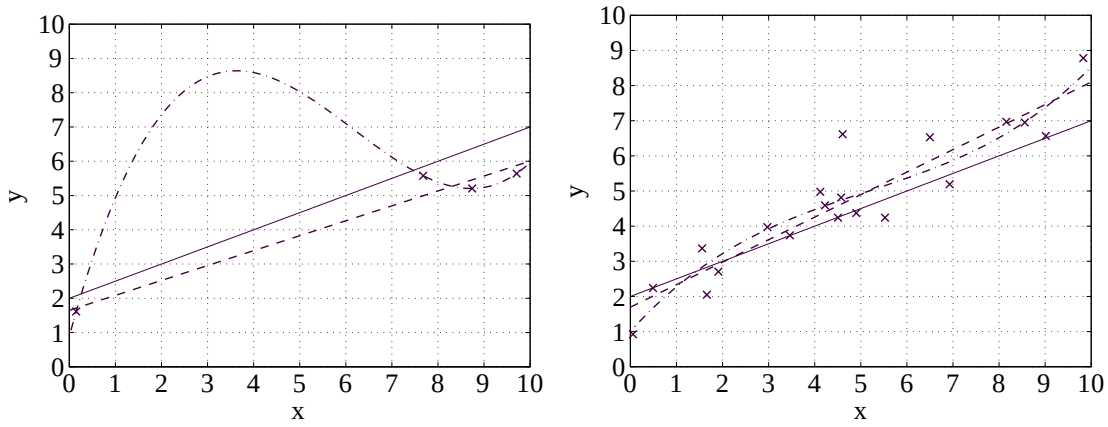

Figure 3.1 — Illustration of overfitting. Training examples (crosses) were created as follows. Values for the *x*-coordinate were drawn uniformly from the interval [0,10]. Values for the *y*-coordinate were created from a linear function with slope 0.5 and intercept 2, with additive Gaussian noise with mean zero and variance 1. On the left the result of fitting a linear function (dashed line) and a polynomial of degree three (dash-dotted line) to four training examples is shown. The polynomial overfits and strongly deviates from the underlying function (solid line). On the right the result of fitting a linear function and a polynomial of degree three to twenty examples is shown. The polynomial as well as the linear function fit the underlying function relatively well.

the family of functions $\mathcal{F}$ is too complex for the learning problem at hand. In our example, if we would have chosen $\mathcal{F}$ to be the family of linear functions, the danger of overfitting would have been reduced, even for small numbers of training examples (see Fig. 3.1). A third reason is noise in the training data. In our example, if there had been no noise added to the training examples and if we had used linear functions, we would have obtained R = 0, for all training sets of size $N \geq 2$.

While the problem of overfitting is controllable for low dimensional problems, for example by increasing the size of the training set or by simply plotting the fitted functions, the situation gets worse for high dimensional problems. This is because the number of training examples needed to sample the input space with a certain density grows exponentially with the dimensionality. Using N examples in a one-dimensional input space corresponds to using N^D training examples in a D-dimensional input space. This is known as the *curse of dimensionality* in machine learning literature (Bellman, 1961). Training data for typical problems with hundreds of input variables is thus almost always sparsely distributed in input space and the danger of overfitting is high.

An approach that is often taken to avoid overfitting is regularization. When regularization is used, instead of minimizing the empirical risk, a weighted sum of the empirical risk and a regularization term is minimized:

$$\mathrm{g} = \arg\min_{\mathrm{f}\in\mathcal{F}} \hat{\mathrm{R}}(\mathrm{f}) + \lambda\Omega(\mathrm{f}). \tag{3.9}$$

Here $\Omega : \mathcal{F} \to \mathbb{R}$ is a regularization functional, which penalizes complex functions f. The regularization parameter $\lambda \in [0, \infty)$ allows to choose how strongly complex functions are penalized. While many different regularization functionals can be used, complex most often is translated as nonsmooth, i.e. by using regularization one avoids to choose functions that vary much in small neighborhoods of the input space $\mathcal{X}$. The assumption underlying regularization is then that the function from which the data is generated is smooth in some sense, i.e. similar inputs give sim-

ilar outputs. If this assumption is true, regularization leads to better generalization - the risk of overfitting is reduced. Another, more pragmatic, motivation for penalizing nonsmooth functions is that training data is sparse. Clearly, it makes no sense to fit a complicated function if only a small number of training examples is available.

In practice often the squared L_2 norm of the parameters of the function is used for regularization. For example, when fitting a polynomial to training data (as described above), the parameters θ are the coefficients of the polynomials, and we can use

$$\Omega(\mathrm{f}) = \|\boldsymbol{\theta}\|^2. \tag{3.10}$$

This has the effect of penalizing polynomials that oscillate much, i.e. nonsmooth polynomials are penalized, and our learning algorithm will tend to fit smooth functions to the data.

To make regularization work it is necessary to carefully choose the regularization parameter λ. Choosing λ too large will lead to underfitting, i.e. functions that are too smooth will be fitted to the training data. Choosing λ too small will lead to overfitting. A related problem is that of choosing a family of functions $\mathcal{F}$ which gives good results for a specific learning problem. Choosing λ as well as choosing $\mathcal{F}$ is known as *model selection* in the machine learning literature. A simple procedure for model selection often employed in practice is to use a so-called validation set. This means that only a part of the training data is used to compute the empirical risk and to fit functions. Actually, functions are fitted for several values of λ and for several choices of $\mathcal{F}$. The result is a set of functions indexed by λ and $\mathcal{F}$. The other part of the training data - the validation set - is then used to estimate the risk for each function in this set. Since the validation set has not been used for fitting functions, the empirical risk on the validation set is a realistic estimate of the expected risk R. Finally the best λ and $\mathcal{F}$ are chosen, and the whole training data is used to fit a function with the chosen parameters.

A refined version of using a validation set is to use cross-validation. In k-fold cross-validation the training set is split into k non-overlapping subsets of size N/k. Then $k-1$ subsets are used for fitting a set of functions with different choices of λ and $\mathcal{F}$. After training, the left out subset is used to estimate the expected risk of each of the functions. This process is repeated k times, i.e. each subset is left out once, and the λ and $\mathcal{F}$ resulting in the smallest average risk are chosen.

3.3 Probabilistic Approaches

The approach to supervised machine learning described in the previous section used loss functions, regularization functionals, and optimization methods in order to fit functions to a training dataset. Probabilities were only used for expressing the expected risk of a given function. In this section we describe approaches to supervised machine learning in which a probabilistic interpretation is given to loss functions and regularization functionals.

3.3.1 Maximum-Likelihood Estimation

The simplest probabilistic approach to supervised machine learning is maximum-likelihood (ML) estimation. The idea underlying ML estimation is that the data can be described with the help of

a parametric probability distribution. Hence, regression and classification tasks can be solved in the following way. First, a probability distribution selected from a parametric family of probability distributions is fitted to the training data. Then, this probability distribution is used to perform classification or regression.

Parametric probability distributions of the training data are often built by negating and exponentiating loss functions. When regarded as a function of the parameters $\boldsymbol{\theta}$, the probability of a training example is also called the likelihood function. A general example for a likelihood function is:

$$\mathrm{L}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = p(y|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z} \exp\left(-\mathrm{l}(y, \mathrm{f}(\mathbf{x}; \boldsymbol{\theta}))\right). \tag{3.11}$$

Here l is a loss function as described in Section 3.2 and Z is a suitable normalization constant which assures that $p(y|\mathbf{x}, \boldsymbol{\theta})$ is a valid probability distribution. A more specific example is the likelihood function corresponding to the squared-error loss, which can be expressed with the help of a Gaussian probability distribution:

$$\mathrm{L}(y|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - \mathrm{f}(\mathbf{x}; \boldsymbol{\theta}))^2\right). \tag{3.12}$$

So far, we have only described likelihood functions for single training examples. To express the likelihood of a whole training set, in ML estimation it is almost always assumed that the training examples are i.i.d. Denoting by $\mathbf{X}$ all the input vectors in the training set and by $\mathbf{y}$ all the outputs in the training set, the likelihood can be written as:

$$\mathrm{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \boldsymbol{\theta}). \tag{3.13}$$

Remember that our goal is to fit a probability distribution to the training data. This is done by finding a set of parameters such that the probability of the training data is maximized. To maximize the probability of the training data, it is convenient to take the logarithm of the likelihood:

$$\log\left(\mathrm{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\right) = \sum_{i=1}^{N} \log\left(p(y_i|\mathbf{x}_i, \boldsymbol{\theta})\right). \tag{3.14}$$

The ML estimate of the parameters is then equal to the parameters that maximize the log-likelihood:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \log\left(\mathrm{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\right). \tag{3.15}$$

In simple cases, for example when the Gaussian likelihood function from Equation 3.12 is used in conjunction with functions that are linear in their parameters, closed form solutions for maximizing the log-likelihood can be derived. In more complex cases, a general approach to maximize the log-likelihood is to take derivatives with respect to $\boldsymbol{\theta}$ and to use gradient descent or other optimization methods.

Once a probability distribution has been inferred from the training data it can be used for classification or regression. In a regression task one can for example use the mean of $p(y|\mathbf{x}, \boldsymbol{\theta}_{\mathrm{ML}})$. In a classification task one can take decisions by using the following rule:

$$\hat{y} = \arg\max_{\mathrm{y} \in \mathcal{Y}} \mathrm{p}(\mathrm{y}|\mathbf{x}, \boldsymbol{\theta}_{\mathrm{ML}}). \tag{3.16}$$

Note that in the above description of the ML approach we have implicitly assumed that a conditional distribution of outputs given inputs is used to model the data. Similarly, in Section 3.2 we have fitted functions that take a feature vector as input and directly give a class label or regression target as output. This is called the *discriminative* approach because the focus is on models that allow to discriminate different classes or outputs. The discriminative approach makes sense because often the only information needed to solve a classification or regression task is the conditional probability of the outputs given the inputs. However, it is also possible to use parametric models of the joint distribution of input and outputs to solve supervised learning tasks. Using a joint distribution of inputs and outputs is known as the *generative* approach and is described next.

In the generative approach an approximation of $p(\mathbf{x}, y)$ is inferred from the training data. Generative models are almost always used for classification problems and work as follows. First, a parametric family of probability distributions $p(\mathbf{x}, y|\boldsymbol{\theta})$ is defined. Then, using the examples in the training set, parameters $\boldsymbol{\theta}$ are fitted using ML. Using the assumption of i.i.d training examples, the likelihood L can be expressed as follows:

$$\mathrm{L}(\mathbf{X}, \mathbf{y}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\mathbf{x}_i, y_i|\boldsymbol{\theta}). \tag{3.17}$$

As in the discriminative approach, the probability of the training data can be maximized by maximizing the log-likelihood. In simple cases, for example when $p(\mathbf{x}, y|\boldsymbol{\theta})$ is Gaussian, closed form solutions exist. In more complex cases, one can compute derivatives and use an optimization method, or use the so-called expectation-maximization (EM) algorithm. The latter option is especially interesting if the training data has missing values, for example missing entries in feature vectors or missing class labels (see (Bishop, 2006) for a detailed description of the EM algorithm).

Once the parameters $\boldsymbol{\theta}_{\mathrm{ML}}$ have been fitted to the training data, class labels y for new inputs $\mathbf{x}$ can be predicted using Bayes rule:

$$p(y|\mathbf{x}, \boldsymbol{\theta}_{\mathrm{ML}}) = \frac{p(\mathbf{y}, k|\boldsymbol{\theta}_{\mathrm{ML}})}{\sum_{y \in \mathcal{Y}} p(\mathbf{x}, y|\boldsymbol{\theta}_{\mathrm{ML}})}. \tag{3.18}$$

3.3.2 Maximum A Posteriori Estimation

ML estimation, as described in the previous section, is very similar to empirical risk minimization (cf. Equation 3.6). In fact, when the likelihood function is built by negating and exponentiating a loss function, maximizing the log-likelihood is equivalent to empirical risk minimization. A consequence of this is that ML estimation suffers from the same problem as empirical risk minimization: overfitting.

An approach that can be used to avoid overfitting in probabilistic models is maximum a posteriori (MAP) estimation. As in ML estimation, in MAP estimation a likelihood function is used to measure how well a set of parameters fits the training data. Unlike in ML estimation, in MAP estimation the values parameters can take are restricted by specifying a prior distribution over the parameters. Using Bayes rule the prior and the likelihood are combined, and a posterior distribution over parameters is obtained. The MAP estimate of parameters is the parameter setting that is most probable according to the posterior distribution and can be used for prediction in the same way as parameters derived with the help of ML.

Similar to the construction of likelihood functions, priors for use in MAP estimation can be built by negating and exponentiating regularization functionals. A general example for a prior is:

$$p(\boldsymbol{\theta}) = \frac{1}{Z} \exp\left(-\lambda \Omega(\mathrm{f}(.;\boldsymbol{\theta}))\right). \tag{3.19}$$

Here Z is a normalization constant that ensures that $p(\boldsymbol{\theta})$ is a valid probability distribution, Ω is a regularization functional, and $\lambda \in [0, \infty)$ is a hyperparameter which controls how strongly the parameters are regularized. The prior distribution expresses our beliefs about the form of the function f that generated the training data. Building a prior distribution with the help of a regularization functional which penalizes complex functions is equivalent to saying that, a priori, we believe the training data was generated by a smooth, non-complex function.

Using as example a discriminative model, the posterior probability $p(\boldsymbol{\theta}|\mathbf{x}, y)$ of the parameters after observing one training example $(\mathbf{x}, y)$ can be expressed using Bayes rule:

$$p(\boldsymbol{\theta}|\mathbf{x}, y) = \frac{p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}}. \tag{3.20}$$

The posterior probability of parameters after observing more than one training example can be conveniently expressed by using the likelihood function from Equation 3.13:

$$p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}) = \frac{\mathrm{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int \mathrm{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}}. \tag{3.21}$$

MAP estimation now consists of finding parameters that maximize $p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y})$.Since the denominator of Equation 3.21 does not depend on $\boldsymbol{\theta}$ it is sufficient to maximize $\mathrm{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) p(\boldsymbol{\theta})$.

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} \log\left(\mathrm{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\right) + \log\left(\mathrm{p}(\boldsymbol{\theta})\right). \tag{3.22}$$

As for ML, in simple cases, the maximization of the posterior probability can be achieved in closed form. In more complex cases gradient descent or other optimization methods have to be used.

Note that MAP estimation includes ML estimation as a special case. In fact, MAP estimation is equivalent to ML estimation if a flat, constant prior $p(\boldsymbol{\theta}) = c$ is used. Note also the close resemblance of Equation 3.22 and Equation 3.9, which shows that MAP estimation can be seen as a probabilistic version of regularized empirical risk minimization.

While the MAP approach was demonstrated for the case of discriminative models it can also be applied to generative models. However, we do not further describe this case here as it is very similar to the case of discriminative models.

3.3.3 Bayesian Estimation

Bayesian estimation is similar to MAP estimation in that a posterior distribution over parameters is estimated from prior beliefs and training data. However, whereas in the MAP approach a point estimate of the parameters is used for making predictions, in the Bayesian approach one integrates over the parameters in the posterior distribution to make predictions.

Taking the example of a discriminative model, the distribution used for predictions in the MAP approach is $p(y|\mathbf{x}, \boldsymbol{\theta}_{\text{MAP}})$. In the Bayesian approach the distribution used for predictions is:

$$p(y|\mathbf{x}, \mathbf{X}, \mathbf{y}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y})\, d\boldsymbol{\theta}. \tag{3.23}$$

Using the Bayesian approach to prediction has the advantage that the a posteriori uncertainty in the parameters is taken into account. Compared to the MAP and ML approaches, Bayesian methods will thus in general estimate more accurately the uncertainty in predictions, especially if the training data carries not enough information to obtain precise estimates of the model parameters.

A second aspect that distinguishes the Bayesian approach from non-probabilistic approaches, as well as from ML and MAP estimation is model selection (cf. Section 3.2). In non-Bayesian approaches often cross-validation is used to perform model selection, i.e. to select regularization parameters or a family of functions appropriate to a given problem. Using cross-validation can be problematic because the potentially time-consuming fitting of parameters has to be performed several times, and hence the overall time needed for training can be long. Moreover, regularization parameters and other possibly continuous hyperparameters have to be discretized in order to perform cross-validation and it is often unclear to which precision hyperparameters should be discretized. The Bayesian approach to model selection is to compute the probability of a model given the data. The main advantage of Bayesian model selection is that each model has to be fitted only once to the training set, whereas in k-fold cross-validation each model has to be fitted k times. A further advantage is that discretization of hyperparameters is unnecessary.

Using again the discriminative approach as example and denoting models by $\mathcal{M}_i, i \in 1 \dots M$, the probability of a model given the data can be expressed using Bayes rule:

$$p(\mathcal{M}_i|\mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathcal{M}_i) p(\mathcal{M}_i)}{\sum_{i=1}^{M} p(\mathbf{y}|\mathbf{X}, \mathcal{M}_i) p(\mathcal{M}_i)}. \tag{3.24}$$

Here $p(\mathcal{M}_i)$ denotes our prior belief that model i is the correct model. Model selection is performed by selecting the model that is a posteriori maximally probable. Several concepts need to be refined to better understand Bayesian model selection. First, we need to state more precisely what is meant by "model". In general in the context of Bayesian model selection, a model represents a probability distribution over training datasets. For example, in the discriminative approach a model is a conditional distribution of outputs given inputs. This distribution is formed by combining prior beliefs and a likelihood function:

$$p(\mathbf{y}|\mathbf{X}, \mathcal{M}) = \int \mathrm{L}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}, \mathcal{M}) p(\theta|\mathcal{M})\, d\boldsymbol{\theta}. \tag{3.25}$$

Models can be formed by choosing different likelihood functions or prior distributions. Different choices for likelihood functions can be motivated by knowledge about the structure of the problem to be solved and from assumptions about the distribution of noise in the training data. Different choices for the prior distribution can for example correspond to different values for the regularization parameter λ or to different choices for the regularization functional Ω (cf. Equation 3.19).

In practice often a flat prior $p(\mathcal{M}_i) = c$ is used because for many problems it is difficult to decide a priori which models are probable. With a flat prior, Bayesian model selection is equivalent to

ML estimation at the level of models. Note that ML estimation at the level of models (or hyperparameters) is also known as type II ML estimation in statistical literature. Because Bayesian model selection is equivalent to ML estimation it is in theory also vulnerable to overfitting. However, often only a small number of different models is used, or a small number of hyperparameters is estimated. Hence, in practice the danger of overfitting is small.

3.4 Algorithms for BCI Systems

We now turn our attention to the practical implementation of the concepts mentioned in the previous sections. In particular, we review some examples of supervised machine learning algorithms that have been used in BCI systems and highlight advantages and disadvantages of the different algorithms. For the description of the algorithms we assume that one of the feature extraction methods from Section 2.4 has been used to transform raw neurophysiologic signals into feature vectors $\mathbf{x} \in \mathbb{R}^D$.

3.4.1 Support Vector Machines

An example for a learning algorithm that is often used in BCI systems is the so-called support vector machine (SVM). In the following we will briefly describe some basic concepts underlying the SVM. A more detailed description of the SVM and related algorithms can be found in Müller *et al.* (2001), an extensive description of the application of an SVM in a BCI is given in Garcia (2004).

To understand how the SVM works it is instructive to first consider the case in which all the training examples can be separated by a hyperplane, i.e. the case in which the training data is linearly separable. In this case the SVM chooses a hyperplane that maximizes the minimal Euclidean distance between the hyperplane and the training examples. In the SVM literature this distance is called the margin. Intuitively, by maximizing the distance between training examples and the hyperplane the probability that future feature vectors fall on the wrong side of the hyperplane is kept small. Denoting class labels as $y_i \in \{-1, 1\}$, feature vectors as $\mathbf{x}_i \in \mathbb{R}^D$, and parameterizing the optimal hyperplane by $\mathbf{w} \in \mathbb{R}^D$, $b \in \mathbb{R}$, it can be shown that maximizing the margin is equivalent to solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) \geq 1 \quad \text{for } i \in \{1 \dots N\}. \end{aligned} \tag{3.26}$$

Here b is a bias variable, and N is the number of training examples. It can be shown, that the margin corresponds to the quantity $1/\|\mathbf{w}\|$, thus a maximization of the margin is achieved by minimizing $\|\mathbf{w}\|^2$. A geometrical interpretation of this optimization problem in the two-dimensional case is shown in Figure 3.2.

If the data are not linearly separable, Problem 3.26 is infeasible, i.e. no solution that respects all the constraints exists. To deal with non-separable datasets the constraints are relaxed by introducing

slack variables ξ_i and a regularization constant C:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{for } i \in \{1 \dots N\} \\ & \xi_i \geq 0 \qquad\qquad\qquad\quad \text{for } i \in \{1 \dots N\}. \end{aligned} \tag{3.27}$$

If $C = \infty$ this problem is equivalent to problem 3.26. However if C is small enough some training examples are allowed to lie inside the margin or even on the wrong side of the hyperplane. To obtain good generalization performance it is important to test different values for C and to choose an optimal value, for example via cross-validation.

It can be shown that the optimization problem solved by the SVM can also be expressed as the minimization of the sum of a loss function and a regularization functional (Hastie *et al.*, 2001):

$$\min_{\mathrm{f}\in\mathcal{F}} \sum_{i=1}^{N} \max(0, 1 - y_i \mathrm{f}(\mathbf{x}_i)) + \lambda\|\mathrm{f}\|^2. \tag{3.28}$$

The function $\max(0, 1 - y\mathrm{f}(\mathbf{x}))$ is called the hinge loss and gives zero penalty to training examples for which $y\mathrm{f}(\mathbf{x}) \geq 1$. Training examples for which $y\mathrm{f}(\mathbf{x}) < 1$ receive a penalty equal to $1 - y\mathrm{f}(\mathbf{x})$. The function f is in general of the form:

$$\mathrm{f}(\mathbf{x};\theta) = \sum_{i=1}^{N} y_i \theta_i \mathrm{k}(\mathbf{x}, \mathbf{x}_i) + \theta_0. \tag{3.29}$$

Here k is a kernel function, which allows to implement nonlinear mappings between inputs and outputs. If $\mathrm{k}(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}^\mathsf{T}\mathbf{x}_i$ the linear SVM is obtained. In the nonlinear case the Gaussian kernel $\mathrm{k}(\mathbf{x}, \mathbf{x}_i) = \exp(-\|\mathbf{x}-\mathbf{x}_i\|^2/\sigma^2)$ and the polynomial kernel $\mathrm{k}(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^\mathsf{T}\mathbf{x}_i + c)^d$ are widely used kernel functions.

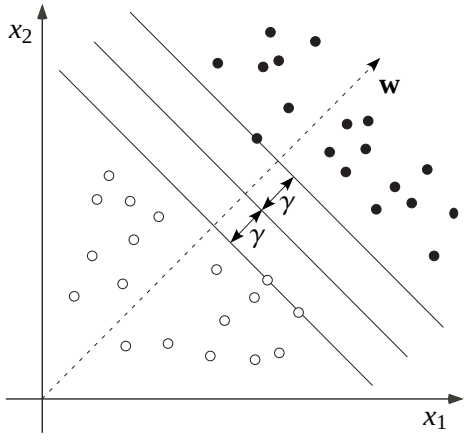

Figure 3.2 — Linear SVM in two dimensions for a separable dataset. The figure shows the margin $\gamma = 1/\|\mathbf{w}\|$ and the weight vector $\mathbf{w}$. The three points on the margin are called support vectors and fully define the solution, i.e. the solution does not change if the other points are moved while staying on the same side of the margin.

In BCI research the SVM has been successfully applied to data recorded with different paradigms. Examples include the classification of P300 data (Kaper *et al.*, 2004; Thulasidas *et al.*, 2006), motor imagery data (Schlögl *et al.*, 2005), and data from other cognitive tasks (Garrett *et al.*, 2003). Though very good classification accuracies have been achieved in the previously mentioned studies, several problems exist that hinder the application of SVMs in practical BCI systems. A first problem is linked to the optimization problems that have to be solved when training SVMs. Solving these problems can be very time consuming, especially when a large number of training examples is used. To reduce the computational costs of training SVMs, optimized algorithms have been developed (Platt, 1999). Even if optimized solvers are used computational costs remain relatively high, because cross-validation has to be used to select optimal regularization and kernel parameters and so multiple SVM instances have to be trained. Adapting a BCI that employs an SVM to a new user can thus be a cumbersome process, requiring expert knowledge and a relatively large amount of time.

A second issue is that the loss function used in the SVM is designed for problems in which only binary yes/no outputs are needed. The problem with binary yes/no outputs is that no information is given about the confidence the system has in those outputs. We will show in later chapters of this thesis that a classifier which provides confidence levels, for example in the form of class probabilities, is of great advantage when building a BCI system.

3.4.2 Generative Models

A basic generative approach to classification that is sometimes used in BCI systems is to use Gaussian densities for the class-conditional distributions of feature vectors. Gaussian probability distributions for vectors $\mathbf{x} \in \mathbb{R}^D$ are parameterized by a mean vector $\mathbf{m} \in \mathbb{R}^D$ and a covariance matrix $\mathbf{\Sigma} \in R^{D \times D}$:

$$p(\mathbf{x}|\mathbf{m}, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathsf{T}} \mathbf{\Sigma}^{-1} (\mathbf{x}-\mathbf{m})\right). \tag{3.30}$$

Here $|\mathbf{\Sigma}|$ denotes the determinant of the covariance matrix. To build a classifier with the help of this model, a Gaussian density is fitted to the training examples from each class. This results in conditional probability distributions for all classes. Using Bayes rule the conditional probability distributions can be used for classification:

$$\begin{aligned} p(k|\mathbf{x}) &= \frac{p(\mathbf{x}|\mathbf{m}_k, \mathbf{\Sigma}_k) p(k)}{\sum_{k \in \mathcal{Y}} p(\mathbf{x}|\mathbf{m}_k, \mathbf{\Sigma}_k) p(k)} \\ \hat{k} &= \arg\max_{k \in \mathcal{Y}} p(k|\mathbf{x}). \end{aligned} \tag{3.31}$$

Here $\mathbf{m}_k$ and $\mathbf{\Sigma}_k$ denote the parameters for class k. The prior probability $p(k)$ indicates the a priori probability for class k.

A method that is very often used to fit the parameters of a generative model is ML estimation. For the case of the Gaussian distribution well-known closed form solutions exist for the mean vectors and covariance matrices. Denoting by C_k the set of indices of training examples belonging to class k the ML estimate for the mean of class k is:

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{i \in C_k} \mathbf{x}_i. \tag{3.32}$$

The ML estimate for the covariance of class k is:

$$\mathbf{\Sigma}_k = \frac{1}{N_k} \sum_{i \in C_k} (\mathbf{x}_i - \mathbf{m}_k)(\mathbf{x}_i - \mathbf{m}_k)^\mathsf{T}. \tag{3.33}$$

While the Gaussian distributions are particularly popular for generative models it is also possible to use other types of parametric distributions. If no closed form solutions exists for the parameters of a given distribution, gradient descent with the log-likelihood as objective function can be used for parameter optimization.

Generative algorithms have been used less frequently in BCI research than discriminative methods. Generative algorithms based on Gaussian distributions have been applied with success to the classification of motor imagery data (Lemm *et al.*, 2004; Vidaurre *et al.*, 2006) and the classification of other cognitive tasks (Curran *et al.*, 2004; Keirn and Aunon, 1990). A potential advantage of using the generative approach in a BCI system is that a priori knowledge about neurophysiologic signals can be modeled relatively easy (see for example (Chiappa, 2006)). Further advantages are that generative approaches can readily be used for multi-class problems, that generative approaches can easily deal with missing data, and that a probabilistic output is given. A potential disadvantage is that in generative approaches often too many parameters have to be learned. In fact, in generative approaches the joint distribution of input vectors and outputs is modeled, while for classification tasks it is sufficient to model decision boundaries between classes. In other words, modeling the joint distribution of input vectors and outputs often implies modeling structures that are not important for classification.

3.4.3 Bayesian Algorithms

Bayesian techniques have been used relatively rarely in the area of BCI systems. However, the few examples in which Bayesian techniques have been used show that with their help systems can be built that offer functionality which goes beyond that of many other systems.

A basic example for the use of Bayesian techniques in a BCI system can be found in the study of Roberts and Penny (2000). In the system presented by Roberts and Penny an autoregressive (AR) model was used to extract features from EEG data recorded while the subject performed either mental arithmetic or imagined hand movements. The coefficients of the AR model were classified with the help of linear logistic regression, which is a method for two-class classification problems. In logistic regression models class probabilities are computed as follows:

$$p(y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^\mathsf{T}\mathbf{x})}. \tag{3.34}$$

Here $\mathbf{w}$ denotes the parameters of the classifier, and the probability p(y=-1|$\mathbf{x}$,$\mathbf{w}$) can be easily computed by using $1 - p(y = 1|\mathbf{x}, \mathbf{w})$. The likelihood function corresponding to the logistic regression model is

$$\mathrm{L}(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{N} p(y_i = 1|\mathbf{x}, \mathbf{w})^{t_i} p(y_i = -1|\mathbf{x}, \mathbf{w})^{1-t_i}, \tag{3.35}$$

where

$$t_i = \frac{y_i + 1}{2}. \tag{3.36}$$

In order to infer the model parameters, Roberts and Penny used an isotropic Gaussian prior with regularization parameter α

$$p(\mathbf{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{\frac{D}{2}} \exp(-\frac{\alpha}{2}\mathbf{w}^\mathsf{T}\mathbf{w}), \tag{3.37}$$

where D is the order of the AR model used for feature extraction, i.e. the dimensionality of the feature vectors. The posterior distribution resulting from the combination of the logistic regression likelihood and a Gaussian prior cannot be expressed in closed form and hence was approximated with a Laplace approximation (see (Bishop, 2006) for more information about the Laplace method). The regularization parameter α was estimated with the help of type II ML estimation (see Section 3.3.3 and (Roberts and Penny, 2000) for further details).

The probabilities over classes computed with Bayesian logistic regression where exploited in two ways in the system of Roberts and Penny. First, probabilities obtained from several consecutive EEG segments were used to obtain temporally smoothed estimates of class probability. Second, a reject-class was introduced in order to reject EEG segments for which no sufficiently certain decision could be taken. EEG segments were assigned to the reject-class, whenever the maximum class probability was smaller than a threshold $d \in [\frac{1}{2}, 1]$, i.e. whenever:

$$\max_{k\in\{-1,1\}} p(y = k|\mathbf{x}, \mathbf{w}) < d. \tag{3.38}$$

As was shown by Roberts and Penny, the temporal smoothing, as well as the use of a reject-class lead to significantly increased classification accuracy when compared to a system working without these features.

Other interesting examples for the use of Bayesian methodology in BCI systems can be found in the work of Sykacek *et al.* (2003). In this work the authors present two algorithms for classification of EEG data recorded during the performance of different cognitive tasks. The innovative aspect of the first algorithm is that it takes into account uncertainty in the features derived from neurophysiologic signals. This is different from the standard approaches to supervised machine learning in BCIs in which features are regarded as fixed values. As has been shown by Sykacek *et al.*, treating features as latent variables results in higher classification accuracies than treating features as fixed values. An important drawback of this algorithm is however that Monte Carlo techniques have to be used and that computational complexity is high.

The innovative aspect of the second algorithm presented by Sykacek *et al.* is that it is adaptive. This means the algorithm is capable to react to nonstationarities in the relation between neurophysiological signals and the underlying cognitive tasks. The adaptivity is achieved by treating the parameters of the classifiers as state variables in a first order Markov process. An update of the classifier parameters after observing data $(\mathbf{x}_t, y_t)$ at time t is expressed as follows:

$$p(\boldsymbol{\theta}_t|y_t, \mathbf{D}_{t-1}) = \int\int \frac{p(y_t|\mathbf{x}_t, \boldsymbol{\theta}_t)}{p(y_t)} p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \lambda) p(\boldsymbol{\theta}_{t-1}|\mathbf{D}_{t-1}) p(\lambda) d\,\boldsymbol{\theta}_{t-1} d\,\lambda. \tag{3.39}$$

Here $\mathbf{D}_{t-1}$ denotes the data observed up to time $t-1$ and λ serves to automatically control the speed of adaptation. The conditional distribution $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}, \lambda)$ is multivariate Gaussian with mean $\boldsymbol{\theta}_{t-1}$ and covariance $\lambda\mathbf{I}$. An experimental comparison of the adaptive classifier with an otherwise equivalent but static classifier showed that the adaptive classifier often outperformed the static classifier (Sykacek *et al.*, 2003).

In summary, the main advantage of the Bayesian approach is that it allows to build functionality into BCI systems, which is difficult to obtain with other approaches. Examples for such functionality are the automatic estimation of regularization parameters, the rejection of data that cannot be classified with certainty, the consideration of uncertainty in features, and the adaptation to temporal nonstationarities. A potential difficulty with the Bayesian approach is that often no closed form solutions exist for the integrals which are at the basis of Bayesian inference and prediction. A possible solution to this problem is to use sampling techniques such as Monte Carlo sampling. This approach is however not advisable in BCI systems due to its high computational complexity. Another possible solution, which is more suited for the use in BCI systems, is to use deterministic approximation schemes, such as the Laplace method or variational inference (see (Bishop, 2006) for more details about such methods).

3.5 Conclusion

In this chapter we have given a brief introduction to supervised machine learning methods for BCI systems. In the first part of the chapter we have reviewed non-probabilistic and probabilistic approaches to supervised learning and have described basic concepts, such as loss functions, risk, overfitting, regularization, model selection, and cross-validation. In the second part of the chapter we have reviewed some examples of supervised learning algorithms that have been used in BCI systems. During the discussion of the individual algorithms it became apparent that algorithms which are to be used in practical BCI systems ideally should fulfill the following requirements. First, algorithms should be robust with respect to outliers. This is important because neurophysiologic signals can contain many outliers and artifacts, caused for example by eye-blinks and muscle activity. Second, algorithms should be of low computational complexity during inference and prediction. Low computational complexity during inference reduces the time needed to setup a BCI system. Low computational complexity during prediction is crucial because in BCI systems data should be processed in realtime. Third, algorithms should provide confidence levels for their predictions or, equivalently, probabilistic outputs. This is important because probabilistic outputs provide a natural basis to combine information obtained from different sources and to use decision theory when taking decisions. As we will see in later chapters of this thesis, combining information as well as taking decisions in a principled manner allow to build advanced BCIs.

The Bayesian approach to supervised machine learning allows one to build algorithms that fulfill many of the requirements described above. Nevertheless, Bayesian techniques have been used only relatively rarely in the area of BCIs. In this thesis we will present Bayesian algorithms that can learn classifiers quickly, robustly, and fully automatically. Moreover, with these algorithms, electrode configurations can be adapted to specific users and information from several data segments can be aggregated. Before describing the details of our algorithms, we review in the next chapter the specific type of BCI used in this thesis, namely BCIs based on the P300 evoked potential.

Review of P300-Based Brain-Computer Interfaces

4

4.1 Introduction

After the general discussion of BCI systems in Chapter 2 and the review of supervised machine learning in Chapter 3 we now give a more detailed review of the type of system used in this thesis, that is to say P300-based BCIs. First, in Section 4.2 we describe the P300 from a neuroscientific point of view, i.e. we list conditions under which the P300 can be evoked and factors that influence the characteristics of the P300. Then, in Section 4.3 the basic idea underlying P300-based BCIs is introduced and several systems implementing this idea are described. In Section 4.4 the algorithmic aspects of P300-based BCIs are discussed. Finally, in Section 4.5 criteria for evaluating the different systems and algorithms are described. The chapter is summarized in Section 4.6.

4.2 The P300 Event-Related Potential

Event-related potentials (ERPs) can be divided into two classes. Exogenous ERPs are the result of early, automatic processing of stimuli and have a latency, amplitude, and topographic distribution that depends mainly on the physical stimulus characteristics. Endogenous ERPs are the result of later, more conscious processing of stimuli and have characteristics that depend mainly on the stimulus context, i.e. on the task the subject was given and on the attention the subject pays to the stimuli. An endogenous ERP that has gained much attention in the neuroscientific and medical research communities is the P300 (see Fig. 4.1). The P300 is an interesting and fruitful research topic because it can be reliably measured and because the characteristics of the P300 waveform, such as for example amplitude and latency, can be influenced by various factors. Since the discovery of the P300 by Sutton *et al.* (1965) many studies have tried to uncover the psychological and neurophysiological meaning of the P300 by varying the way stimuli are presented and by observing the

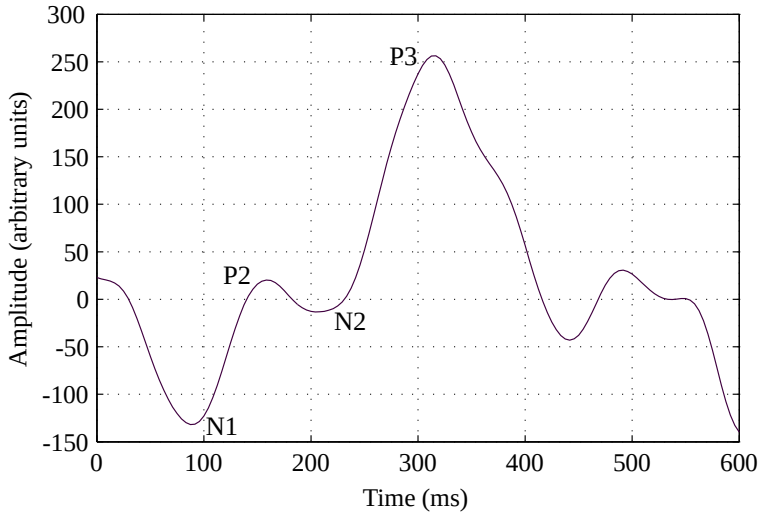

Figure 4.1 — Typical P300 wave. The P300 (or P3) is a positive deflection in the EEG, which appears approximately 300 ms after the presentation of a rare or surprising stimulus. A series of negative and positive components (N1, P2, N2) precedes the P3. While the P3 reflects high-level processing of stimuli, the earlier components reflect low-level, automatic processing of stimuli.

corresponding changes in the waveform of the P300. Other studies have linked the characteristic of the P300 to subject specific factors such as gender, age, or brain diseases, for example Alzheimer or schizophrenia. As it is impossible to review all these studies in detail, the following discussion is restricted to points that are important for the use of the P300 in a BCI. Readers who are keen to learn more about the P300 are referred to the reviews in (Donchin, 1981), (Hruby and Marsalek, 2003), and (Nieuwenhuis *et al.*, 2005).

To evoke the P300 different stimulus modalities and paradigms can be used. Regarding the stimulus modality, auditory, visual, tactile, gustatory, or olfactory stimuli can be employed. However, for practical reasons, often auditory or visual stimuli are used. Mainly two paradigms are employed, the oddball paradigm and the three-stimulus paradigm. In the oddball paradigm two different stimuli are used, a target (or oddball) stimulus and a nontarget stimulus. The two stimuli are presented in a random sequence and the target stimulus appears only rarely. Subjects are instructed to respond to each occurrence of the target stimulus and to ignore the nontarget stimuli. For example subjects can be instructed to react with a button press to each 1000 Hz tone in a random sequence of 1000 Hz and 2000 Hz tones.

The three-stimulus paradigm is a modified oddball paradigm in which a so-called distracter stimulus appears infrequently in the sequence of target and nontarget stimuli (Courchesne *et al.*, 1975). The distracter stimulus is usually not mentioned when giving instructions to the subjects and so it surprises subjects when it first appears in a sequence. To increase the effect of surprise, several unique distracter stimuli are used and each distracter stimulus is presented only once. The distracter stimuli are perceptually different from the target and nontarget stimuli. For example dog-barks or other environmental sounds can be used in an oddball sequence consisting of 1000 Hz and 2000 Hz tones.

Different types of P300 can be observed in the two paradigms described above. In the classical

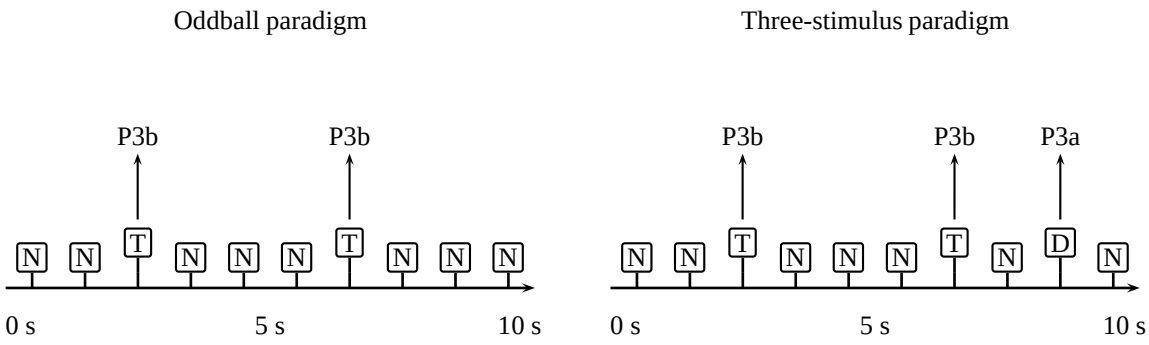

Figure 4.2 — Paradigms for evoking the P300. Left: In the oddball paradigm a sequence of target (T) and nontarget (N) stimuli is presented in random order. The probability for target stimuli is low, and subjects are instructed to react to the targets, either by a button press or by silently counting the targets. Each target stimulus evokes a P3b. Right: In the three-stimulus paradigm distracter stimuli are added to the sequence of target and nontarget stimuli. A P3a is evoked by surprising distracter stimuli.

oddball paradigm, target stimuli evoke the so-called P3b. The P3b has a latency of about 300-500 ms and can be observed mostly over centro-parietal brain regions. The P3b appears only if subjects pay attention to stimuli and disappears if subjects do not pay attention to stimuli. When subjects do not pay attention to stimuli, the target stimuli in the oddball paradigm evoke a different type of P300 - the so-called P3a (Squires *et al.*, 1975). The P3a has a latency of about 200-400 ms and can be observed mostly over fronto-central brain regions. In the three-stimulus paradigm the target stimuli also evoke a P3b. The distracter stimuli however evoke a P3a (Courchesne *et al.*, 1975). The relation between the different paradigms and the P3a and P3b is summarized in Fig. 4.2.

In addition to the dependence on different experimental paradigms, the P300 is also influenced by many other factors. The dependence of the P300 on these factors shows that the P300 is not a static, fixed phenomenon but rather an inherently variable response of the brain, occurring in situations in which novel or improbable and task-relevant stimuli have to be processed. Some important factors influencing the P300 are listed below.

- *Target Probability*
  The P3b peak amplitude is inversely related to the probability of the evoking stimulus. High amplitude P3b waves are evoked when the probability of the target stimulus is low. Low amplitude P3b waves are evoked when the probability of the target stimulus is high. In practice, the probability for target stimuli is usually set to values around 10% in order to reliably evoke the P300. In addition to the effect of global target probability, the amplitude of the P3b is also affected by local target probability. This means that amplitude is high when many nontarget stimuli precede a target stimulus and that amplitude is low if a small number of nontarget stimuli precedes a target stimulus (Squires *et al.*, 1976).

- *Interstimulus Interval*
  The amplitude of the P3b wave is positively correlated to the interstimulus interval (ISI), i.e. to the amount of time between two consecutive stimuli. Long ISIs lead to high amplitudes, short ISIs lead to smaller amplitudes.

- *Habituation*
  The amplitude of the P3a habituates. After presentation of many distracter stimuli subjects get used to these stimuli and P3a amplitude decreases (Courchesne, 1978). The amplitude of the P3b is mostly unaffected by long-term repetition of stimuli.

- *Attention*
  The amplitude of the P3b wave depends on how much attention subjects pay to stimuli and on how concentrated subjects are. In fact, the P3b wave completely disappears if subjects are not actively engaged in an oddball task. The P3a wave on the contrary remains mostly unaffected by changes in attention and can be observed even if subjects completely ignore the stimuli.

- *Task Difficulty*
  The latency of the P3b increases and the amplitude decreases with increasing task difficulty. For example target tones being very different from nontarget tones yield higher P3b amplitudes than target tones being only a little different from nontarget tones (Polich, 1987). For the P3a the effect of task difficulty is different from the effects for the P3b. Increasing the difficulty of discrimination between target and nontarget tones in a three-stimulus paradigm will lead to increased P3a amplitudes. In addition to stimulus novelty the P3a thus also seems to be related to perceptual discrimination difficulty between target and nontarget stimuli. The P3b amplitude decreases in such a setup as expected (Hagen *et al.*, 2006).

The paradigms used for evoking the P3a and P3b, together with the factors influencing the shape of the P3a and P3b, allow to draw conclusions about the psychological and physiological meaning of these ERPs. In general the P3a seems to be related to frontal lobe function and is evoked by stimuli that require attention and subsequent processing. In particular, it has been proposed that the P3a is a part of the so-called orienting response, i.e. the response of the human body to novel, surprising or potentially threatening situations, consisting of rapid changes in heart rate, skin conductance, and other physiological parameters (Courchesne *et al.*, 1975; Knight, 1996). The P3b is thought to be related to processes for context updating, processes for updating "models of the environment", and to stimulus evaluation. In contrast to overt, immediate responses to a stimulus these processes (and the P3b) are thought to be part of high-level, metacontrol processing (Donchin and Coles, 1988). Note however, that the context updating model of Donchin and Coles has been criticized by Verleger (1988) who promotes a theory in which the P300 is a sign of context closure. In this theory the P300 is linked to expected events, instead of unexpected events as in the theory of Donchin and Coles. In summary, a conclusive theory about the role of P3a and P3b in human information processing has yet to be established.

4.3 P300-Based BCI Systems

The basic idea underlying P300-based BCI systems is to use an oddball-like paradigm and to let the user decide which stimulus plays the role of the target stimulus. As the P300 (P3b) occurs only if a subject voluntarily reacts to a target stimulus, the target chosen by the user can be automatically

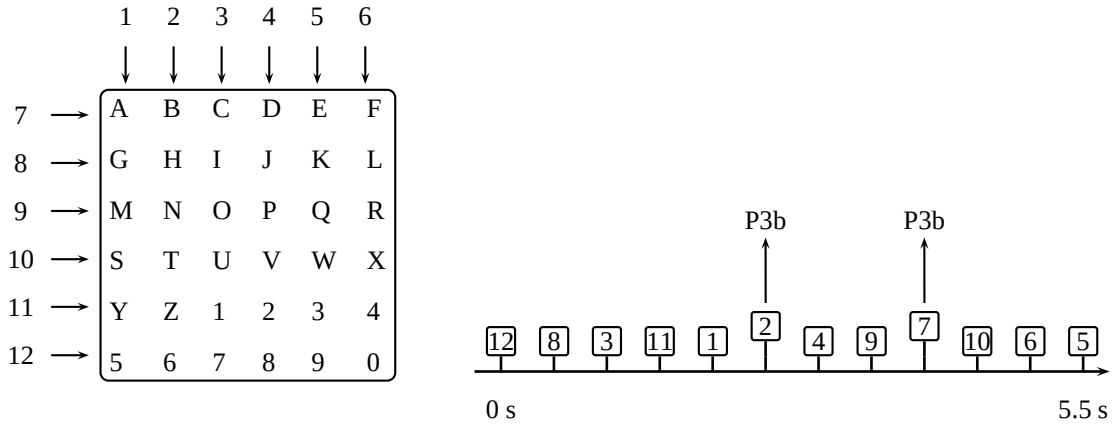

Figure 4.3 — Working principle of the P300 speller. Left: Example for a symbol matrix that can be used in the P300 speller. Flashes of rows or columns are used as stimuli. The stimuli are numbered from 1 to 12. Right: A random stimulus sequence. If the user concentrates for example on the letter 'B', a P3b will be evoked for stimuli 2 and 7.

inferred from the EEG recorded during stimulus presentation. More specifically, the sequence of events in a P300-based BCI is usually as follows. First, the user decides on a command he wants to execute with the help of the BCI. Then, stimuli are presented and the user concentrates on the stimulus associated to the desired command. After stimulus presentation the recorded EEG is analyzed with the help of a classification algorithm (see Section 4.4). The goal of this analysis is to infer which stimulus was chosen as target by the user. If the analysis is successful the command associated to the chosen stimulus is executed by the BCI system. Below we present several systems that implement this idea.

4.3.1 P300 Speller

The first P300-based BCI has been presented by Farwell and Donchin (1988). In their work a 6×6 matrix containing the letters of the alphabet and some other symbols was displayed on a computer screen. Rows and columns of the matrix were flashed in random order, and subjects could choose a symbol from the matrix by counting how often it was flashed. Flashes of the row or column containing the desired symbol constituted target stimuli and evoked a P300 while all other flashes of rows and columns constituted nontarget stimuli and did not evoke a P300. To infer which symbol the user wanted to select, it was thus sufficient to find out which flashes evoked a P300. The principle underlying the P300 speller is depicted in Fig. 4.3.

Since the work of Farwell and Donchin several researchers have proposed extensions and modifications of the basic P300 speller paradigm. Allison and Pineda (2003) tested the impact of different matrix sizes on the amplitude and latency of the P300. They presented matrices of size 4×4, 8×8, and 12×12 to their subjects. Instead of single symbols the entries in the matrix were digrams, i.e. pairs of letters. The outcome of their study was that P300 latency decreased and amplitude increased as matrix size was increased. Note that this is in line with the relation between target probability and P300 amplitude described in Section 4.2: the smaller the target probability,

the higher the P300 amplitude. Unfortunately in the study of Allison and Pineda classification of the P300 signals and thus the automatic detection of the symbol the user wanted so select was not attempted. The impact of different matrix sizes on the communication speed achievable with the P300 speller thus remained unclear.

Another modification of the basic P300 speller, the so-called single display paradigm, was proposed by Guan *et al.* (2004). In their system, instead of flashing whole rows and columns of the symbol matrix, single symbols were used as stimuli. This has the effect of reducing the probability for the target stimulus. In the basic P300 speller paradigm the target probability is $1/6 \approx 0.16$, while in the single display paradigm it is $1/36 \approx 0.03$. In the experiments performed by Guan *et al.* the lower target probability led to higher P300 amplitudes and better classification accuracy than in the basic P300 speller.

Many other studies were concerned with classification algorithms for the P300 speller and payed less attention to stimulus display aspects. These studies are described in Section 4.4.

4.3.2 Virtual Apartment

A departure from the P300 speller paradigm was initiated by Bayliss (2003) who tested if the P300 could be evoked in a virtual reality environment. In the system presented by Bayliss, subjects viewed a virtual apartment alternatively on a monitor or through a head-mounted display. Control of several items in the virtual apartment, for example switching on/off a lamp, was possible by concentrating on small spheres that were flashing in random order over the controllable items. The outcome of the study was that only small differences existed between the P300 waves recorded in the monitor and head-mounted display conditions. It was thus shown that virtual reality, which allows for complex, yet controllable experimental environments, is an interesting alternative to other, simpler P300 BCI paradigms.

4.3.3 Cursor Control

Yet another P300 BCI paradigm was presented by Polikoff *et al.* (1995). The idea behind the system described by Polikoff *et al.* is to allow users to control a two-dimensional cursor with the help of the P300. To implement this idea a fixation cross with target arms in the north, east, south, and west directions was displayed on a monitor. At the end of each arm small crosses were displayed and temporarily replaced by asterisks. The replacement of crosses occurred in random order, and to move the cursor in a given direction subjects were instructed to count the number of asterisks appearing at the corresponding target arm. While in the study of Polikoff *et al.* actual cursor movement was not implemented, an offline analysis showed that cursor control with the help of the P300 was in principle possible.

4.3.4 Systems for Disabled Subjects

The cursor control paradigm was further explored by Piccione *et al.* (2006). Flashing arrows were displayed in the peripheral area of a screen and subjects could move a cursor by concentrating on one of the arrows. Piccione *et al.* tested their system with five severely handicapped and seven

able-bodied subjects. The outcome of their study was that handicapped as well as able-bodied subjects were able to control cursor movement with their P300 signals. The communication speed achieved by the severely handicapped subjects was significantly lower than that of the able-bodied subjects. Nevertheless, the study of Piccione *et al.* (2006) was one of the first studies showing that P300-based communication is possible for severely handicapped subjects.

Another study testing P300 based communication with severely disabled subjects was presented by Sellers and Donchin (2006). A paradigm similar to the P300 speller was used by Sellers and Donchin (2006), however the matrix size was reduced to 2×2. The motivation for reducing the matrix size was to simplify use of the system for disabled subjects who might have visual deficits and thus might not be able to concentrate on a small item on a screen. Sellers and Donchin also tested auditory stimuli and combinations of visual and auditory stimuli. The results obtained in the study showed that communication with the help of the P300 was possible in the auditory, the visual, and in the combined auditory-visual modality. Furthermore it was shown that communication was possible for the handicapped as well as for the able-bodied subjects.

4.4 Algorithms for P300-Based BCI Systems

Clearly, in all of the systems described above, algorithms are necessary that can infer the command a user wants to execute from the EEG recorded during stimulus presentation. The input for these algorithms is the EEG recorded during presentation of stimuli, together with the sequence and timing of stimuli. The required output is the identity of the stimulus that was chosen by the user, i.e. the identity of the target stimulus. To compute this output in all algorithms described in the literature the same general approach is employed. First, for each presentation of a stimulus a short EEG segment, a so-called single trial, is extracted. Then the single trials are classified with the help of a (non-probabilistic) machine learning algorithm. The outcome of the classification is a score that indicates for each single trial if a P300 is present or not. Finally the scores from all single trials are aggregated in order to form a decision about the identity of the target stimulus. In the following we first describe algorithms that have been used to aggregate information from several single trials (see Section 4.4.1). Then, in Section 4.4.2 we describe the machine learning methods that have been used to classify single trials.

4.4.1 Algorithms for Aggregating Information from Single Trials

In the simplest type of algorithms the information from single trials is directly used, without aggregating information from several stimuli, i.e. the EEG recorded after each stimulus presentation is immediately translated into a command (see Fig. 4.4.).

For example in the cursor control system described by Piccione *et al.* the following three steps are repeated until the system is stopped by an operator:

1. One of the four arrows in the peripheral area of the screen is randomly chosen and flashed.

2. The EEG segment recorded during the flash of the arrow is analyzed with a classifier. The output of the classifier is a score that indicates if a P300 is present in the analyzed segment.

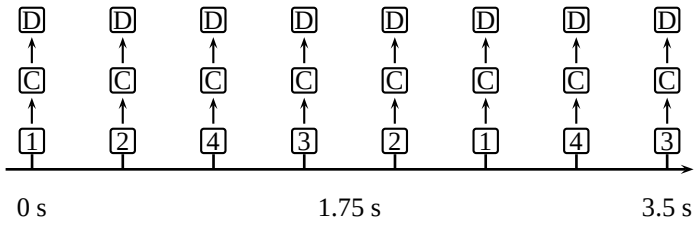

Figure 4.4 — Immediate translation of EEG into commands. Four different stimuli are presented in random order with an ISI of 500 ms (1,2,3,4). The EEG segment corresponding to each stimulus presentation is classified (C); the output of the classifier is a score indicating how similar the EEG segment is to a P300. The classifier scores are immediately used to take decisions (D), i.e. if the classifier score indicates that a P300 is present, the command associated to the stimulus that evoked the P300 is executed.

3. If the classifier output is larger than a preset threshold, the cursor moves into the direction of the arrow chosen in step 1. Otherwise the cursor remains still.

This algorithm has the advantage that the user almost immediately obtains feedback from the system. A disadvantage is however that wrong decisions will be taken relatively often. This is the case because the EEG is a noisy signal and consequently the classifier output for single trials also contains noise.

To allow for more complex application scenarios, a method that was first described by Farwell and Donchin (1988) is often used (see Fig. 4.5). In this method stimulus presentation also has to be started by an operator and stops after all stimuli have been presented a certain number of times (in a random sequence). To infer a command from the EEG, first the classifier-outputs corresponding to multiple presentations of one stimulus are summed. Then, the command associated to the stimulus with the maximal summed classifier score is executed. The main advantage of this approach is that summing the classifier scores obtained from multiple presentations of a stimulus reduces noise. The danger of executing an unwanted command is thus greatly reduced. In the original P300 speller system of Farwell and Donchin for example, flashing each row and column of the symbol matrix fifteen times allowed for perfect classification. The disadvantage of repeating each stimulus several times is that sending a command takes more time than when decisions are taken immediately. For example, if an ISI of 500 ms is used in the P300 speller system and each row and column is flashed fifteen times, selecting one character takes $15 \times 12 \times 500$ ms = 90 s. An additional disadvantage is that the number of stimulus presentations has to be fixed a priori. This is problematic because the system cannot take into account fluctuations in the signal-to-noise ratio of the EEG. These fluctuations can arise for example from changes in the level of concentration of the user or changes in the electrode-skin connection.

An algorithm that is able to dynamically adapt to fluctuations in the signal-to-noise ratio has been described by Serby *et al.* (2005). In the system of Serby *et al.* stimulus presentation is started by an operator and stops as soon as "enough" data has been acquired or after a fixed maximal number of stimuli has been presented. Enough data here means that the system can take a reliable decision, i.e. the system presents more stimuli if it is unsure which command the user wants to send and stops stimulus presentation as soon as it is sure about the desired command (see Fig. 4.6).

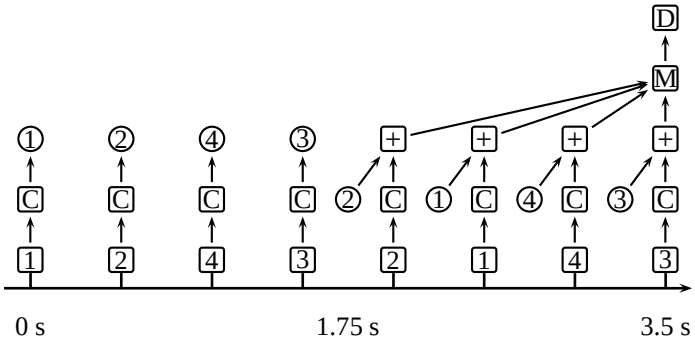

Figure 4.5 — Translation of EEG into commands after a fixed number of stimulus presentations (symbols in squares represent operations, symbols in circles represent variables). Four different stimuli are presented in random order with an ISI of 500 ms (1,2,3,4). The EEG segment corresponding to each stimulus presentation is classified (C). The classifier scores from the second block of stimuli are summed with the scores from the first block (+). The maximum of the summed scores is computed (M) and the command associated to the stimulus with the largest summed score is executed (D). In this example each stimulus is presented twice, however different numbers of stimulus presentations can be used to optimize performance for a given user.

As has been shown by Serby *et al.*, adapting the number of stimulus presentation to the signal-to-noise ratio significantly improves the speed of communication achievable with a P300-based BCI. However, instead of fixing the number of stimulus presentations a priori as in the system of Farwell and Donchin, now a criterion has to be chosen that allows one to decide how much data the system requires to take a reliable decision. In the system of Serby *et al.* a thresholding technique was used to decide if more stimuli have to be presented or not but no details about the algorithm for computing optimal thresholds were given.

4.4.2 Classification Algorithms

All of the methods for aggregating information from single trials, presented in the previous section, depend on algorithms that can transform a EEG segment into a score which indicates if a P300 is present or not. Discrimination of P300 and non-P300 EEG segments is a surprisingly problematic task because the amplitude of the P300 wave is relatively small when compared to the background EEG activity and because the latency, topography, and amplitude of the P300 differ from subject to subject. The approach that is usually taken in P300 BCIs to solve these problems is to use supervised machine learning algorithms. This means that first in one or several training sessions a training dataset is acquired that contains many examples of P300 and non-P300 EEG segments from a specific user. Then, a classifier that can solve the discrimination task is learned from the training dataset.

A straightforward classification algorithm that has been used with good success in several P300 BCIs consists of the following steps. First, a subset of electrodes positioned at the locations on the scalp where one expects strong P300 amplitudes is selected. Then the raw signals from these electrodes are bandpass filtered and downsampled. Finally the filtered and downsampled signals from the selected electrode subset are concatenated into feature vectors and fed into a machine learning algorithm.

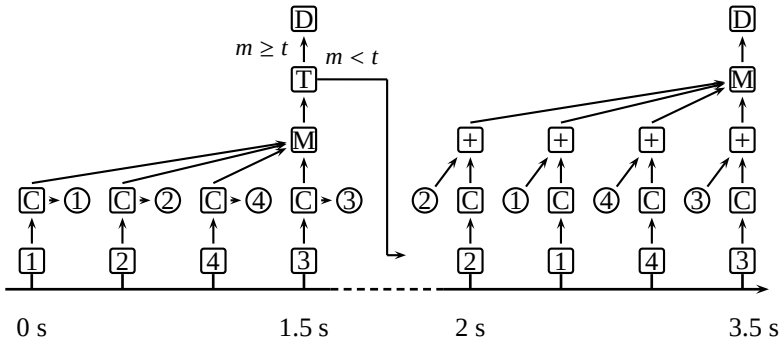

Figure 4.6 — Decision after a variable number of blocks (symbols in squares represent operations, symbols in circles represent variables). Four different stimuli are presented in random order with an ISI of 500 ms (1,2,3,4). The EEG segment corresponding to each stimulus presentation is classified (C); the output of the classifier is a score indicating how similar the EEG segment is to a P300. After the first block of stimulus presentations, the maximum of the classifier scores is computed (M). If the maximum is larger than a certain threshold (T), a decision is taken (D), i.e. the systems executes the command associated to the stimulus with the largest score. If the maximum is smaller than the threshold, an additional block of stimuli is presented. The classifier scores from the second block of stimuli are summed with the scores from the first block (+). The maximum of the summed scores is computed and the command associated to the stimulus with the largest summed score is executed.

This general approach was used in the algorithms described by Kaper *et al.* (2004) and Thulasidas *et al.* (2006). In the method described by Kaper *et al.* (2004) a ten electrode configuration consisting of the midline electrodes, the parietal-occipital electrodes PO7, P08, P3, P4 and the central electrodes C3, C4 was used. A support vector machine (SVM) with Gaussian kernels was used for classification. The method described by Kaper *et al.* (2004) was one of the winning entries for the P300 dataset from the BCI competition 2003 (Blankertz *et al.*, 2004). The algorithm described by Kaper *et al.* (2004) was also employed in another study. In this study (Kaper and Ritter, 2004), classifiers were trained from a pool of data from several subjects and then tested with data from new, unseen subjects. This is different from the usual approach in which training data from only one subject is used. The results described by Kaper and Ritter (2004) showed that generalizing to new subjects without subject-specific training data is in principle possible, however significantly lower classification accuracies than in the standard approach were achieved.

In the method described by Thulasidas *et al.* a set of 25 central and parietal electrodes was used. In addition to the downsampled signals also estimates of the time-derivatives of the signals were used. According to Thulasidas *et al.* the use of the time-derivatives improves classification accuracy. A SVM with a Gaussian kernel was used for classification. The method described by Thulasidas *et al.* was tested with several P300 speller datasets and showed very good performance. Unfortunately the method was not tested with publicly available datasets and thus a direct comparison with the method of Kaper *et al.* is impossible.

An alternative to manually fixing parameters for feature extraction (e.g. filter settings and subset of electrodes) is to let an algorithm select the optimal features from a set of predefined features. This idea is implemented in the stepwise discriminant analysis (SWDA) algorithm and was used in the studies described by Farwell and Donchin (1988), Donchin *et al.* (2000), and Sellers and

Donchin (2006). In these studies filter settings and a subset of electrodes were fixed, however SWDA was used to select timepoints relevant for P300 classification within EEG segments. In a recent comparison of classification methods for the P300 speller, SWDA turned out to be one of the best methods in terms of classification performance and in terms of effort needed for implementation of the method (Krusienski *et al.*, 2006).

The principle of automatically selecting features was also used by Bostanov (2004). In the algorithm of Bostanov an overcomplete dictionary of continuous wavelets was used to transform the raw EEG signals into the time-scale space. During training a t-test was used to identify points in time-scale space at which the difference between the mean wavelet coefficients from P300 segments and non-P300 segments is high and at which at the same time the variance around these means is small. The wavelet coefficients with the best t-test results were fed into an linear discriminant analysis (LDA) classifier for classification. The method of Bostanov (2004) was tested with slow cortical potentials (SCP) and P300 datasets in the BCI competition 2003 and was among the winning entries for both datasets.

Still another algorithm which used the principle of automatic feature selection was presented by Rakotomamonjy *et al.* (2005). In this algorithm recursive feature selection with the SVM was used to find an optimal subset of electrodes. To further improve performance several SVM classifiers were learned from different subsets of the training data. This approach is based on the assumption that the distribution of the P300 and non-P300 segments in the training set is variable and thus cannot be appropriately modeled by a single classifier. The method of Rakotomamonjy *et al.* (2005) was tested with very good results on the 2003 BCI competition P300 dataset and was the winning entry for the 2004 BCI competition P300 dataset.

In addition to the algorithms based on filtering for feature extraction and the algorithms based on feature selection a third group of algorithms can be identified. The algorithms in this group use independent component analysis (ICA) for spatial feature extraction (Piccione *et al.*, 2006; Serby *et al.*, 2005; Xu *et al.*, 2004). The first step in all algorithms using ICA is to compute independent components from the training data. Then the components that present well the P300 are selected. This can be either done manually, i.e. by inspecting the data (Serby *et al.*, 2005), or by defining criteria that allow to automatically select P300 like components (Piccione *et al.*, 2006; Xu *et al.*, 2004). When the algorithm is applied to new data, the data is projected on the retained independent components and then classified. Different types of classifiers were used in combination with ICA as feature extraction method. Xu *et al.* proposed to use LDA, Serby *et al.* tested matched filters and a maximum-likelihood based classifier, and Piccione *et al.* used a neural network. While the methods of Serby *et al.* and Piccione *et al.* were only tested on proprietary datasets, the algorithm described by Xu *et al.* was tested on the 2003 P300 competition datasets and was one of the winning entries in this competition.

4.5 Evaluation of Systems and Algorithms

In the previous sections we have seen that to build a P300-based BCI a lot of different approaches can be taken. Different approaches exist for stimulus presentation, for the aggregation of information from single trials, and for discriminating P300 from non-P300 segments. As a consequence,

it is virtually impossible to find a sensible metric with which all the different systems can be compared and evaluated. Nevertheless, it is possible to describe the properties of P300-based systems that should enter in a metric for comparison and evaluation. In the following we mention some of these properties, concentrating especially on aspects related to the practicality and suitability for daily use of BCI systems.

A certainly very important aspect of any BCI system is the achievable speed of communication. In a P300-based BCI the speed of communication depends on the ISI, the number of different stimuli, the classification accuracy, and the control flow algorithm. To abstract from all these factors it is useful to use the information transfer rate (also known as bitrate, or capacity) as a metric for the speed of communication. Roughly speaking the bitrate measures the number of bits that can be transferred from a user to the system in a given amount of time. It is also commonly used to evaluate other, non-P300 BCI systems. The bitrate b in bits/min has also been used to characterize other types of BCI systems and can be computed according to the following equation (Wolpaw *et al.*, 2002):

$$\mathrm{b}(N, p, t) = \left(\log_2(N) + p\log_2(p) + (1-p)\log_2\left(\frac{1-p}{N-1}\right)\right)\frac{60}{t}. \tag{4.1}$$

Here N denotes the number of different commands a user can send, p denotes the probability that a command is correctly recognized by the system, and t is the time in seconds that is needed to send one command. Note that according to the noisy-channel coding theorem, the bitrate is an upper limit on the number of bits that can be transmitted, given the characteristics of the transmission channel (MacKay, 2003). This limit can only be attained if optimal encoding and decoding algorithms are used. Since in a BCI the encoding has to be performed by the user and since optimal encoding algorithms are relatively complex, the bitrate is mostly of theoretical value.

Other than the bitrate, several important characteristics concern the overall practical usability of a BCI system, particularly with regard to usage of a system by severely handicapped users. Clearly, any system for handicapped users should be adapted to their often limited cognitive abilities. For example, using a large number of different stimuli, possibly in combination with very short ISIs might strongly limit the usability for subjects with visual impairments. In other words, the number of different stimuli, the size of the stimuli, and the ISI should be adapted to a user. In general any BCI system targeted for use by handicapped subjects should also be tested by such subjects.

A further point influencing the practicality of a given system is the time and effort needed to setup and adapt the system for a new user. Clearly, systems that use only few electrodes take less time for setup and are more user friendly than systems with many electrodes. However, if too few electrodes are used not all features that are necessary for accurate classification can be captured and communication speed decreases. A good tradeoff between time needed for setup and classification accuracy thus has to be found. Moreover, it is important to minimize the time and effort needed to train a classifier for a specific user. If a system is to be accepted by end users, lengthy training sessions should be avoided. Therefore, the amount of training data that is necessary to achieve a certain communication speed is probably as important a characteristic as the communication speed itself.

In addition it is important to limit the amount of user intervention necessary during setup of a system. Several of the prototype systems described in the previous section rely on the intervention

of a trained technician during setup, for example to select ICA components or to choose hyperparameters for classification. This is clearly undesirable and a practical system should be able to adapt to new users according to a simple and fully automatic protocol.

4.6 Conclusion

In this chapter we have reviewed P300-based BCI systems. The P300 is an endogenous ERP, which appears approximately 300 ms after the presentation of rare, task-relevant or surprising stimuli. While to analyze the psychological and physiological aspects of the P300 different experimental paradigms can be used, in P300-based BCIs usually a variant of the classical oddball paradigm is employed. In fact, the user can select one of several commands by concentrating on the stimulus associated to the command. This basic principle has been used in different designs, targeted toward different application scenarios. Besides developing new application scenarios much research in the area of P300-based BCIs has also concentrated on the development and refinement of algorithms for inferring the command a user wants to send. These algorithms usually consist of two modules, one module controls stimulus presentation and the aggregation of information obtained from several single trials. The second module has the task of transforming single trials into scores that indicate if a P300 is present or not. Much research has been dedicated to the latter type of algorithms, i.e. to feature extraction methods and supervised machine learning algorithms for discriminating target trials from nontarget trials.

However, only little research has concentrated on the problem of optimally integrating information from several trials. In fact, the most interesting scheme for aggregating information from several trials, namely the scheme in which decisions are taken adaptively (cf. Fig. 4.6), has been used in only one study (Serby *et al.*, 2005). Moreover, in this study no details have been given about the exact implementation of this scheme. Another observation that can be made when reviewing the supervised machine learning methods for P300-based BCIs is that when training classifiers for a specific subject usually only data from that subject is used. The only exception is the study of Kaper and Ritter (2004) in which SVM classifiers trained on data from a pool of subjects were used to classify data from new, unseen subjects. A last observation is that all classification algorithms, exclusively use features related to the P300 to perform classification. Side information for example from bigram probabilities in a speller application or information from other phenomena than the P300 is difficult to integrate in existing algorithms.

In the next chapter we present Bayesian classification algorithms that can be used in a P300-based BCI. As we will see, these algorithms allows us to remove some of the above mentioned limitations of classification algorithms for P300-based BCIs.

5 Bayesian Algorithms for EEG Classification

5.1 Introduction

In this chapter we describe Bayesian machine learning algorithms that are well suited for BCI systems using EEG measurements. The algorithms are using a two-stage procedure in which first a probability distribution over discriminant directions is inferred from training data using a Bayesian approach. Then heuristics are used to estimate class probabilities for new input vectors from the distribution over discriminant directions.

Before describing the Bayesian algorithms for EEG classification we discuss simpler, related algorithms, namely least squares regression and Fisher's discriminant analysis (FDA). This discussion can be found in Section 5.2, where in particular it is shown that FDA is a special case of least squares regression. After the introductory material, an algorithm is discussed that we have termed Bayesian linear discriminant analysis (BDA) (see Section 5.3). BDA is especially interesting for use in BCIs because it is robust to noise in the training data and because it can learn classifiers quickly and without intervention of expert users. The technical basis for BDA is the so-called evidence framework for Bayesian regression (MacKay, 1992).

In Section 5.4 we show how a technique that is known as automatic relevance determination (ARD) in the machine learning literature (MacKay, 1995; Tipping, 2001) can be used to perform electrode selection in our BCI application. The resulting algorithm is termed sparse Bayesian linear discriminant analysis (SBDA) and is a simple extension of BDA.

Finally, in Section 5.5 we describe algorithms that allow to compute class probabilities from the distribution over discriminant directions learned with BDA or SBDA. We also show how these probabilities can be used to build a straightforward implementation of a P300-based BCI in which the number of presented stimuli is dynamically adapted such that a preset, approximate bound on the probability of classification errors is not exceeded. A summary of the chapter is given in Section 5.6.

5.2 From Least Squares Regression to Fisher's Discriminant

5.2.1 Least Squares Regression

Regression analysis is arguably one of the most often used tools in science and engineering. While many different linear and nonlinear methods for regression analysis exist, here we concentrate on methods that are relevant for the developments in later sections of this chapter, namely on linear regression and ridge regression. Other, more extended reviews of regression can for example be found in (Hastie *et al.*, 2001) or in (Bishop, 2006).

In linear regression one is given a training set of target values $t_i \in \mathbb{R}, i \in \{1 \dots N\}$ and corresponding input vectors $\mathbf{x}_i \in \mathbb{R}^{D+1}, i \in \{1 \dots N\}$. The goal is to find a weight vector $\mathbf{w} \in \mathbb{R}^{D+1}$ that can be used to map the input vectors to target values. To this end, the sum of squared errors between regression targets and mapped input vectors is minimized:

$$\mathrm{J}(\mathbf{w}) = \sum_{i=1}^{N} (t_i - \mathbf{w}^\mathsf{T}\mathbf{x}_i)^2. \tag{5.1}$$

To unclutter the notation, we assume here that the $(D+1)$st entry in the input vectors is equal to 1 for all i. Consequently, the $(D+1)$st entry of the weight vector is equivalent to the bias value as is usually done in regression.

By setting the derivative of J with respect to $\mathbf{w}$ to zero it can be shown that the weights and bias that minimize the sum of squared errors on the training set satisfy the following equation:

$$\mathbf{w} = (\mathbf{X}\mathbf{X}^\mathsf{T})^{-1}\mathbf{X}\mathbf{t}, \tag{5.2}$$

where

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_1 & t_2 & \cdots & t_N \end{bmatrix}^\mathsf{T}. \tag{5.3}$$

If the dimensionality D of the input vectors is nearly as big as the number of training examples N, one can observe the effect of overfitting. This means that on the training set a near to perfect fit is achieved, however, for pairs of input vectors and target values that are not in the training set typically large errors will be observed (see Chapter 3 for a discussion of overfitting). If the dimensionality is bigger than the number of training examples, the matrix $\mathbf{X}\mathbf{X}^\mathsf{T}$ becomes singular and cannot be inverted. One approach to avoid overfitting and singular matrices is regularization. The canonical approach to regularized regression is called ridge regression (Hoerl and Kennard, 1970). In ridge regression a modified objective function is used:

$$\mathrm{J}(\mathbf{w}) = \sum_{i=1}^{N} (t_i - \mathbf{w}^\mathsf{T}\mathbf{x}_i)^2 + \lambda \mathbf{w}^\mathsf{T}\mathbf{I}'\mathbf{w}. \tag{5.4}$$

Here $\mathbf{I}'$ is an identity matrix in which the $(D+1)$st diagonal element is set to zero. The regularization term $\lambda\mathbf{w}^\mathsf{T}\mathbf{I}'\mathbf{w}$ has the effect of shrinking the optimal solution for the first D weights towards the origin, while leaving the solution for the bias unconstrained. For a correct choice of the hyperparameter λ weight vectors with a large norm and overfitting are thus avoided. To choose λ one can use cross-validation or other model selection methods. As for regression without regularization, one

can set the derivative of J to zero in order to find the solution that minimizes the objective function. The solution to the ridge regression problem satisfies the following equation:

$$\mathbf{w} = (\mathbf{X}\mathbf{X}^\mathsf{T} + \lambda \mathbf{I}')^{-1}\mathbf{X}\mathbf{t}. \tag{5.5}$$

Once the weights and the bias have been estimated from training data, either by using regression or ridge regression, the target values $\hat{t}$ for new input vectors $\hat{\mathbf{x}}$ are computed as follows:

$$\hat{t} = \mathbf{w}^\mathsf{T}\hat{\mathbf{x}}. \tag{5.6}$$

5.2.2 Fisher's Discriminant

While in regression the goal is to map input vectors to target values, the goal in FDA is to compute a discriminant vector that separates two or more classes as well as possible (Fisher, 1936). Here we consider only the two-class case. We are given a set of input vectors $\mathbf{x}_i \in \mathbb{R}^D, i \in \{1 \dots N\}$ and corresponding class-labels $y_i \in \{-1, 1\}$[1]. We denote by N_1 the number of training examples from class 1 (i.e. examples for which $y_i = 1$), by C_1 the set containing the indices of the training examples belonging to class 1, and use analogous definitions for N_2, C_2. The objective function for computing a discriminant vector $\mathbf{w} \in \mathbb{R}^D$ then is

$$\mathrm{J}(\mathbf{w}) = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}, \tag{5.7}$$

where

$$\mu_k = \frac{1}{N_k}\sum_{i \in C_k} \mathbf{w}^\mathsf{T}\mathbf{x}_i, \quad \sigma_k^2 = \sum_{i \in C_k} (\mathbf{w}^\mathsf{T}\mathbf{x}_i - \mu_k)^2. \tag{5.8}$$

In FDA the objective function is maximized. This amounts to searching for discriminant vectors that result in a large distance between the projected means and small variance around the projected means (small within-class variance). To compute directly the optimal discriminant vector for a training dataset, matrix equations for the quantities $(\mu_1 - \mu_2)^2$ and $\sigma_1^2 + \sigma_2^2$ can be used. To this end, we first define the class means $\mathbf{m}_k$ for $k \in \{1, 2\}$.

$$\mathbf{m}_k = \frac{1}{N_k}\sum_{i \in C_k} \mathbf{x}_i \tag{5.9}$$

Now we can define the between-class scatter matrix $\mathbf{S}_B$ and the within-class scatter matrix $\mathbf{S}_W$.

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\mathsf{T} \tag{5.10}$$

$$\mathbf{S}_W = \sum_{k=1}^{2}\sum_{i \in C_k} (\mathbf{x}_i - \mathbf{m}_k)(\mathbf{x}_i - \mathbf{m}_k)^\mathsf{T} \tag{5.11}$$

With the help of these two matrices the objective function for FDA can be written as a Rayleigh quotient:

$$\mathrm{J}(\mathbf{w}) = \frac{\mathbf{w}^\mathsf{T}\mathbf{S}_B\mathbf{w}}{\mathbf{w}^\mathsf{T}\mathbf{S}_W\mathbf{w}}. \tag{5.12}$$

[1] In this section, we slightly change our notation and denote by $\mathbf{x}_i$ input vectors without appended ones and by $\mathbf{w}$ weight vectors without appended bias value.

By computing the derivative of J and setting it to zero, one can show that the optimal solution for $\mathbf{w}$ satisfies the following equation:

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2). \tag{5.13}$$

The discriminant vector $\mathbf{w}$ is thus equal to the difference between the class means, scaled by the inverse of the sum of the within-class scatter matrices. The effect of scaling by $\mathbf{S}_W^{-1}$ is that discriminant directions $\mathbf{w}$ with small within-class variance are preferred, whereas directions with large within-class variance are penalized. This is important whenever the within-class scatter is anisotropic. The concept of FDA is illustrated in Fig. 5.1.

As in regression analysis, we run into problems when the number of training examples becomes small compared to the dimensionality of the input vectors. If the dimensionality of the input vectors is nearly as big as the number of training examples overfitting occurs. If the number of training examples is smaller than the dimensionality of the input vectors, the within-class scatter matrix becomes singular and cannot be inverted. Several solutions to these problems exists. A solution that is similar to the ridge regression approach is to add a multiple of the identity matrix to the within-class scatter matrix. The objective function then reads

$$\mathrm{J}(\mathbf{w}) = \frac{\mathbf{w}^\mathsf{T}\mathbf{S}_B\mathbf{w}}{\mathbf{w}^\mathsf{T}(\mathbf{S}_B + \lambda\mathbf{I})\,\mathbf{w}}. \tag{5.14}$$

The solution is then given by

$$\mathbf{w} \propto (\mathbf{S}_W + \lambda\mathbf{I})^{-1}(\mathbf{m}_1 - \mathbf{m}_2). \tag{5.15}$$

Another possible solution, which has the advantage that no hyperparameters have to be specified, is to replace the inverse $\mathbf{S}_W^{-1}$ by the Moore-Penrose pseudo-inverse $\mathbf{S}_W^{\dagger}$ (Tian *et al.*, 1988). The solution for $\mathbf{w}$ then reads:

$$\mathbf{w} \propto \mathbf{S}_W^{\dagger}(\mathbf{m}_1 - \mathbf{m}_2). \tag{5.16}$$

Note that the discriminant vector $\mathbf{w}$ alone cannot be used to perform classification. This is because $\mathbf{w}$ only defines a one-dimensional projection of the feature vectors in which classes are maximally separated. In order to use FDA for classification, additionally a bias value b has to be inferred. This can be done for example by fitting one-dimensional Gaussian distributions to the projections of the classes.

After inference of the discriminant vector $\mathbf{w}$ and bias b new feature vectors $\hat{\mathbf{x}}$ can be mapped to outputs as follows:

$$\mathrm{f}(\hat{\mathbf{x}}; \mathbf{w}, b) = \mathbf{w}^\mathsf{T}\hat{\mathbf{x}} + b. \tag{5.17}$$

Since the output of FDA is a continuous value it can for example be used to control a one-dimensional cursor in a BCI. Another option is to convert the output of FDA into class labels by using the sign of f:

$$\hat{y} = \begin{cases} 1 & \text{if } \mathrm{f}(\hat{\mathbf{x}}; \mathbf{w}, b) \geq 0 \\ -1 & \text{if } \mathrm{f}(\hat{\mathbf{x}}; \mathbf{w}, b) < 0. \end{cases} \tag{5.18}$$

In BCI research FDA has been successfully applied in different scenarios. Examples include the use of FDA for classification of data from motor imagery experiments (Blankertz *et al.*, 2002;

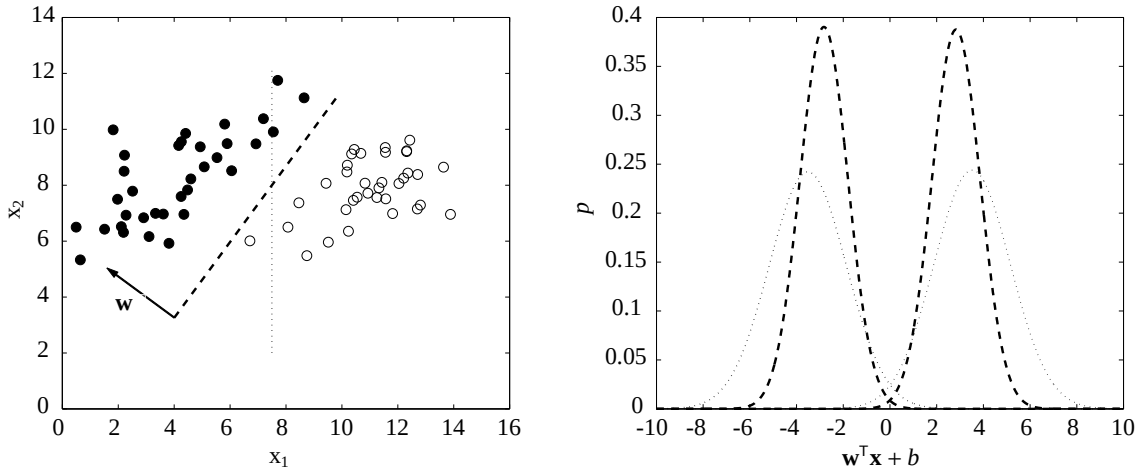

Figure 5.1 — Illustration of FDA. The left panel shows examples drawn from two two-dimensional Gaussian distributions with identical covariance but different means. Also shown are the direction of $\mathbf{w}$ computed with FDA and the corresponding discriminating hyperplane (dashed line). The thin dotted line shows the discriminating hyperplane that is obtained if $\mathbf{w} \propto \mathbf{m}_1 - \mathbf{m}_2$. The right panel shows the distributions of the projected data. The weight vector $\mathbf{w}$ computed with FDA leads to a smaller overlap between classes than a weight vector equal to the difference of the class means.

Pfurtscheller and Neuper, 2001), the use of FDA for classification of data from P300 and slow cortical potentials (SCP) experiments (Bostanov, 2004; Kaper, 2006), and the use of FDA for classification of data from steady-state visual evoked potentials (SSVEP) experiments (Lalor *et al.*, 2005).

The main advantages of FDA are its computational and conceptual simplicity. More specifically, FDA is computationally efficient for situations in which the number of features D is small, and the number of training examples N is large. This is the case because the only complex operation required for FDA is the inversion of the within-class scatter matrix, which scales as $O(D^3)$. Situations in which $D \leq N$ are relatively often found in BCI applications.

Note that in BCI applications often plain FDA, i.e. FDA without regularization is used. This is problematic because data from BCI experiments often contains outliers, resulting for example from eyeblinks or muscle activity, and hence there is an increased tendency for overfitting. A possible remedy to this problem is to use a regularized version of FDA. However, regularized FDA is surprisingly seldom used in the context of BCI. An exception is the work by (Blankertz *et al.*, 2002), in which a non=probabilistic, regularized version of FDA is compared with other classification methods. A Bayesian version of regularized FDA, in which regularization parameters are estimated with Bayesian model selection, is described in Section 5.3 of the present chapter.

5.2.3 Relation between Regression and Fisher's Discriminant

A deeper understanding of least squares regression and FDA can be obtained by investigating the connection between the two methods. It turns out that FDA is equivalent to linear regression with target values representing (modified) class-labels. This fact will be used in the next section in order

to motivate an algorithm for BDA. We show below that by setting target values t_i for training examples in class 1 to N/N_1 and to $-N/N_2$ for class 2, regularized linear regression is equivalent to regularized FDA[1]. The proof is adapted from the proof given in (Duda *et al.*, 2001) but a little more general (it also considers the case of regularized regression and regularized FDA, instead of considering only regression and FDA).

To show the relation between the two methods, we first write down the matrix equations for ridge regression with target values $t_i = N/N_1$ for $i \in C_1$ and $t_i = -N/N_2$ for $i \in C_2$[2].

$$\begin{bmatrix} (\mathbf{X}\mathbf{X}^\mathsf{T} + \lambda\mathbf{I}) & N_1\mathbf{m}_1 + N_2\mathbf{m}_2 \\ N_1\mathbf{m}_1^\mathsf{T} + N_2\mathbf{m}_2^\mathsf{T} & N \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} = \begin{bmatrix} N(\mathbf{m}_1 - \mathbf{m}_2) \\ 0 \end{bmatrix} \tag{5.19}$$

This can be interpreted as a set of two equations, one for b and one for $\mathbf{w}$. Solving for b we obtain

$$b = -\frac{1}{N}(N_1\mathbf{m}_1 + N_2\mathbf{m}_2)^\mathsf{T}\mathbf{w}. \tag{5.20}$$

Inserting the solution for b in the equation for $\mathbf{w}$ we obtain

$$\left(\mathbf{X}\mathbf{X}^\mathsf{T} + \lambda\mathbf{I} - \frac{N_1^2}{N}\mathbf{m}_1\mathbf{m}_1^\mathsf{T} - \frac{N_1N_2}{N}\mathbf{m}_1\mathbf{m}_2^\mathsf{T} - \frac{N_2N_1}{N}\mathbf{m}_2\mathbf{m}_1^\mathsf{T} - \frac{N_2^2}{N}\mathbf{m_2}\mathbf{m_2}^\mathsf{T}\right)\mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2). \tag{5.21}$$

Using the identities

$$\mathbf{S}_w = \mathbf{X}\mathbf{X}^\mathsf{T} - N_1\mathbf{m}_1\mathbf{m}_1^\mathsf{T} - N_2\mathbf{m}_2\mathbf{m}_2^\mathsf{T}, \text{ and } \quad N_1 - \frac{N_1^2}{N_1 + N_2} = \frac{N_1N_2}{N_1 + N_2} \tag{5.22}$$

it follows that

$$\left(\mathbf{S}_w + \lambda\mathbf{I} + \frac{N_1N_2}{N}(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\mathsf{T}\right)\mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2). \tag{5.23}$$

Since $(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\mathsf{T}\mathbf{w}$ is always in the direction of $(\mathbf{m}_1 - \mathbf{m}_2)$ we can write

$$\mathbf{w} \propto (\mathbf{S}_W + \lambda\mathbf{I})^{-1}(\mathbf{m}_1 - \mathbf{m}_2). \tag{5.24}$$

The discriminant vector obtained by performing ridge regression to target values $\pm N/N_k$ is thus in the same direction as the one obtained by regularized FDA. By setting λ to zero, we also see that the discriminant vector obtained by performing regression is in the same direction as the one obtained by FDA.

5.3 Bayesian Discriminant Analysis

Given the connection between ridge regression and regularized FDA, we are now ready to describe BDA. In short, BDA is equivalent to performing Bayesian regression and setting target values to N/N_1 for examples from class 1 and to $-N/N_2$ for examples from class 2. BDA is relatively robust to noise in the training data because regularization is used during learning. Additionally,

[1] Actually, it is sufficient to use any two distinct values for the target values of class 1 and 2. The advantage of using N/N_1 and $-N/N_2$ is that the proof becomes a little bit simpler.

[2] As in the previous section, in this section $\mathbf{w}$ is a D-dimensional vector without appended bias value. Accordingly the matrix $\mathbf{X}$ has dimensions $D \times N$ and b denotes the bias value.

the regularization parameters are estimated automatically and quickly, without the need for time-consuming cross-valdiation. These facts make BDA an interesting alternative to FDA which is popular for BCI applications.

The basis for BDA is the evidence framework for Bayesian regression, which was first introduced to the machine learning community by MacKay (1992). A good description of Bayesian regression and the evidence framework is given in the book of Bishop (2006). The evidence framework was used in a variety of contexts, for example for Bayesian regression (MacKay, 1992), for the development of Bayesian neural network algorithms (MacKay, 1995), for the estimation of regularization parameters and kernel parameters in support vector machines (Kwok, 2000), for the estimation of regularization parameters and kernel parameters in least squares support vector machines (Van Gestel *et al.*, 2002), and for a Bayesian implementation of FDA in which the class means are treated as latent parameters (Centeno and Lawrence, 2006).

It has to be noted that the developments in this section are almost equivalent to the work on Bayesian regression presented by MacKay and are also similar to the work of Van Gestel *et al.*. One difference to the neural networks and kernel methods presented by MacKay and by Van Gestel *et al.* is that here only linear discriminants are considered. This simplification is motivated by the observation that for EEG classification often linear discriminants are sufficient (Müller *et al.*, 2003). The main aim of this section is thus not to introduce a new machine learning algorithm but rather to give a simple introduction to linear Bayesian regression and to show the relation to BDA. An earlier version of the material presented in this section can be found in (Hoffmann *et al.*, 2004).

5.3.1 Prior, Posterior, and Predictive Distribution

The basic idea behind Bayesian regression and consequently behind BDA is to interpret the objective function for ridge regression as the exponent of a probability distribution from the exponential family. We can write

$$p(\mathbf{w}) = \frac{1}{Z}\exp(-\mathrm{J}(\mathbf{w})), \tag{5.25}$$

where Z is a suitable normalization constant and J is the objective function for ridge regression, from equation 5.4. The distribution for $\mathbf{w}$ can also be written as the product of two distributions. One distribution is associated to the sum of squared errors and the other distribution is associated to the regularization term.

$$p(\mathbf{w}) = \frac{1}{Z}\exp(-\sum_{i=1}^{N}(t_i - \mathbf{w}^\mathsf{T}\mathbf{x}_i)^2)\exp(-\lambda\mathbf{w}^\mathsf{T}\mathbf{I}'\mathbf{w}) \tag{5.26}$$

Under the assumption that regression targets and input vectors are linearly related with additive white Gaussian noise, the first term on the right hand side of the above equation can be identified as the likelihood function for $\mathbf{w}$. Denoting by β the inverse variance of the noise and by $\mathbf{D}$ the pair $\{\mathbf{X}, \mathbf{t}\}$, the proper, normalized expression for the likelihood function is

$$p(\mathbf{D}|\beta, \mathbf{w}) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}}\exp(-\frac{\beta}{2}\|\mathbf{X}^\mathsf{T}\mathbf{w} - \mathbf{t}\|^2). \tag{5.27}$$

The second term from the right hand side of equation 5.26 can be interpreted as the prior distribution for **w**. In Bayesian analysis the prior distribution is used to specify the prior belief we have about the values of **w**. The expression for the normalized prior distribution is:

$$p(\mathbf{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{\frac{D}{2}} \left(\frac{\epsilon}{2\pi}\right)^{\frac{1}{2}} \exp(-\frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{I}'(\alpha)\mathbf{w}), \tag{5.28}$$

where $\mathbf{I}'(\alpha)$ is a $D+1$ dimensional, diagonal matrix:

$$\mathbf{I}'(\alpha) = \begin{bmatrix} \alpha & 0 & \dots & 0 \\ 0 & \alpha & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \epsilon \end{bmatrix}. \tag{5.29}$$

The prior for the weights is thus an isotropic, zero-mean Gaussian distribution with variance $\frac{1}{\alpha}$. The effect of using a zero-mean Gaussian prior for the weights is similar to the effect of applying regularization in ridge regression and FDA. The estimates for **w** are shrunk towards the origin and overfitting is avoided. The prior for the bias, which is the last entry in **w** is a zero-mean univariate Gaussian with variance $\frac{1}{\epsilon}$. Setting ϵ to a very small value, the prior for the bias is practically flat, which expresses that we do not make assumptions about the location of the discriminating hyperplane.

Given likelihood and prior, the posterior distribution of **w** can be computed using Bayes rule:

$$p(\mathbf{w}|\beta, \alpha, \mathbf{D}) = \frac{p(\mathbf{D}|\beta, \mathbf{w})p(\mathbf{w}|\alpha)}{\int p(\mathbf{D}|\beta, \mathbf{w})p(\mathbf{w}|\alpha)\, d\mathbf{w}}. \tag{5.30}$$

Since both prior and likelihood are Gaussian, the posterior is also Gaussian and its parameters can be derived from likelihood and prior by completing the square (cf. Bishop (2006)). The mean **m** and covariance **C** of the posterior satisfy the following equations.

$$\mathbf{m} = \beta(\beta\mathbf{X}\mathbf{X}^\mathsf{T} + \mathbf{I}'(\alpha))^{-1}\mathbf{X}\mathbf{t} \tag{5.31}$$

$$\mathbf{C} = (\beta\mathbf{X}\mathbf{X}^\mathsf{T} + \mathbf{I}'(\alpha))^{-1} \tag{5.32}$$

By multiplying the likelihood function for a new input vector $\hat{\mathbf{x}}$ with the posterior distribution and integrating over **w** we obtain the predictive distribution, i.e. the probability distribution over regression targets given the input vector.

$$p(\hat{t}|\beta, \alpha, \mathbf{x}, \mathbf{D}) = \int p(\hat{t}|\beta, \hat{\mathbf{x}}, \mathbf{w})p(\mathbf{w}|\beta, \alpha, \mathbf{D})\, d\mathbf{w} \tag{5.33}$$

The predictive distribution is again Gaussian and can be characterized by its mean μ and its variance σ^2.

$$\mu = \mathbf{m}^\mathsf{T}\hat{\mathbf{x}} \tag{5.34}$$

$$\sigma^2 = \frac{1}{\beta} + \hat{\mathbf{x}}^\mathsf{T}\mathbf{C}\hat{\mathbf{x}} \tag{5.35}$$

How the predictive distribution can be used for classification and specifically how it can be used for classification of sequences of EEG trials in the framework of a P300 BCI will be discussed in section 5.5.

5.3.2 Estimation of Hyperparameters

Both the posterior distribution of $\mathbf{w}$ and the predictive distribution depend on the hyperparameters α and β. The strict Bayesian approach to eliminate this dependence would be to compute the posterior distribution of the hyperparameters and to integrate out the hyperparameters. Assuming we have already computed the posterior $p(\beta, \alpha|\mathbf{D})$ of the hyperparameters, the posterior of $\mathbf{w}$ could then be expressed as follows:

$$p(\mathbf{w}|\mathbf{D}) = \int p(\mathbf{w}|\beta, \alpha, \mathbf{D}) p(\beta, \alpha|\mathbf{D})\, d\beta\, d\alpha. \tag{5.36}$$

The problem with this approach is that except for trivial cases no closed-form solution is available for the posterior distribution of $\mathbf{w}$. The solution used in the evidence framework is to assume that the posterior over the hyperparameters is unimodal and sharply peaked (MacKay, 1992). The posterior of $\mathbf{w}$ can then be approximated as

$$p(\mathbf{w}|\mathbf{D}) \approx p(\mathbf{w}|\hat{\beta}, \hat{\alpha}, \mathbf{D}), \tag{5.37}$$

where $\hat{\beta}, \hat{\alpha}$ are maximum a posteriori (MAP) estimates of the hyperparameters. Moreover, in the evidence framework usually a flat prior $p(\beta, \alpha) = c$ is used, hence the MAP estimates are equal to maximum-likelihood (ML) estimates. Estimating hyperparameters with maximum-likelihood is also known as type-II maximum-likelihood in the statistics literature (Berger, 1988).

To compute $\hat{\beta}, \hat{\alpha}$ we write down the likelihood for the hyperparameters. The likelihood $p(\mathbf{D}|\beta, \alpha)$ is the normalizing integral from equation 5.30.

$$p(\mathbf{D}|\beta, \alpha) = \int p(\mathbf{D}|\beta, \mathbf{w}) p(\mathbf{w}|\alpha)\, d\mathbf{w} \tag{5.38}$$

The quantity $p(\mathbf{D}|\beta, \alpha)$ is also known as the evidence, or the marginal likelihood, and corresponds to the probability of the data given the hyperparameters β and α. The integral in equation 5.38 can be solved by noting that everything is Gaussian and using standard expressions for Gaussian integrals. After computing the integral, it is convenient to use the logarithm of the likelihood function for further analysis.

$$\begin{aligned} \log\left(p(\mathbf{D}|\beta, \alpha)\right) = & \frac{D}{2}\log(\alpha) + \frac{1}{2}\log(\epsilon) + \frac{N}{2}\log(\beta) - \frac{N}{2}\log(2\pi) + \frac{1}{2}\log(\det(\mathbf{C})) \\ & - \frac{\beta}{2}\|\mathbf{X}^\top\mathbf{m} - \mathbf{t}\|^2 - \frac{1}{2}\mathbf{m}^\top\mathbf{I}'(\alpha)\mathbf{m} \end{aligned} \tag{5.39}$$

To maximize the log-likelihood, partial derivatives with respect to α and β are taken and equated to zero. To compute the derivative with respect to α and β, the following identity for the derivative of the logarithm of the determinant of a matrix $\mathbf{A}$ is useful (cf. Bishop (2006); MacKay (1992)):

$$\frac{\partial}{\partial x}\log\det\mathbf{A} = \operatorname{tr}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial x}\right). \tag{5.40}$$

Using this identity we obtain

$$\frac{\partial \log\left(p(\mathbf{D}|\beta, \alpha)\right)}{\partial \alpha} = \frac{D}{2\alpha} - \frac{1}{2}\sum_{i=1}^{D} c_{ii} - \frac{1}{2}\sum_{i=1}^{D} m_i^2, \tag{5.41}$$

where the c_{ii} are the values on the diagonal of $\mathbf{C}$ and the m_i are the elements of $\mathbf{m}$. Taking the derivative with respect to β yields

$$\frac{\partial \log\left(p(\mathbf{D}|\beta,\alpha)\right)}{\partial\beta} = \frac{N}{2\beta} - \frac{1}{2}\mathrm{tr}(\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{C}) - \frac{1}{2}\|\mathbf{X}^\mathsf{T}\mathbf{m} - \mathbf{t}\|^2. \tag{5.42}$$

Setting the derivatives to zero and solving for α and β we obtain the update equations:

$$\alpha = \frac{D}{\sum_{i=1}^{D} c_{ii} + m_i^2} \tag{5.43}$$

$$\beta = \frac{N}{\mathrm{tr}(\mathbf{X}\mathbf{X}^\mathsf{T}\mathbf{C}) + \|\mathbf{X}^\mathsf{T}\mathbf{m} - \mathbf{t}\|^2}. \tag{5.44}$$

The partial derivatives for α and β depend on the posterior mean $\mathbf{m}$ which itself depends on α and β. Equations 5.43 and 5.44 thus represent implicit solutions for the hyperparameters. Thus, to maximize the log-likelihood an iterative scheme is used in which first $\mathbf{C}$ and $\mathbf{m}$ are computed for a given setting of the hyperparameters and then the hyperparameters are updated according to equations 5.43 and 5.44. After a few iterations the values for the hyperparameters converge to the maximum-likelihood solution. More specifically, for the EEG datasets we tested, hyperparameter optimization typically converged after twenty to fifty iterations.

5.4 Sparse Bayesian Discriminant Analysis

5.4.1 Electrode Selection via Automatic Relevance Determination

Having discussed a basic version of BDA in the previous section, we now turn our attention to an extension of BDA that allows to perform feature selection. Feature selection is a strategy that is often used in machine learning to reduce the dimensionality of a given learning problem and to enhance classification accuracy. Feature selection also reduces the computational cost of classification algorithms and allows to gain insights into the structure of learning problems by examining the selected features. In particular, we will use the feature selection capabilities of the algorithm presented in this section to perform electrode selection. As we have seen during the review of classification algorithms for P300-based BCIs, electrode selection has led to good results and thus is interesting to investigate.

To implement electrode selection, we make use of a method that is known as automatic relevance determination (ARD) in the neural networks literature or as the relevance vector machine in the area of kernel methods (MacKay, 1995; Tipping, 2001). The idea underlying the ARD method is to associate a hyperparameter α_i to each feature instead of using one α for all features (as in the basic BDA method). The effect of this modification to the basic BDA algorithm is that the relevance of each feature can be determined separately via the optimization of the hyperparameter α_i. As we will see, the α_i corresponding to irrelevant features will take very large values and hence irrelevant features will be effectively switched off.

To write down the equations for electrode selection via ARD, we first need to state more precisely the structure of the feature vectors $\mathbf{x}_i$. Let us assume that we want to classify EEG trials containing data from N_e electrodes and N_s temporal samples. Let us assume further that the feature

vectors are built by vertically concatenating the signals from all electrodes. The feature vectors then have dimension $D = N_e N_s$ and the first N_s entries of a feature vector correspond to samples from the first electrode, entries $N_s + 1$ to $2N_s$ correspond to samples from the second electrode, and so forth. With this structure of feature vectors the prior distribution used for electrode selection can be expressed as follows (cf. Equation 5.28):

$$p(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N_e} \left(\frac{\alpha_i}{2\pi}\right)^{\frac{N_s}{2}} \left(\frac{\epsilon}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{w}^\mathsf{T}\mathbf{I}'(\boldsymbol{\alpha})\mathbf{w}\right). \tag{5.45}$$

Here $\mathbf{I}'$ is a $D + 1$ dimensional, diagonal matrix with the following diagonal:

$$\mathrm{diag}(\mathbf{I}') = [\alpha_1 \ldots \alpha_1 \ \alpha_2 \ldots \alpha_2 \quad \ldots \quad \alpha_{\mathrm{N_e}} \ldots \alpha_{\mathrm{N_e}} \ \epsilon]^\mathsf{T}. \tag{5.46}$$

The prior for the weights is thus an axis-aligned, zero mean Gaussian distribution with variance $\frac{1}{\alpha_i}$ for the weights corresponding to electrode i. The effect of this prior is that weights are shrunk towards the origin, however in contrast to the prior specified in Equation 5.28 the shrinkage factor can now be determined separately for each electrode. As before, ϵ is set to a small value in order to leave the bias value unconstrained.

The posterior distribution and the predictive distribution resulting from the use of a prior as specified in Equation 5.45 can be calculated as in the case of an isotropic prior, i.e. with the help of Equations 5.30 and 5.34 which are not repeated here. What slightly changes with respect to the use of an isotropic prior is the expression for the likelihood of the hyperparameters and consequently the expressions that are necessary to optimize the hyperparameters. The likelihood of the hyperparameters can be expressed as follows:

$$\begin{aligned}\log\left(p(\mathbf{D}|\beta,\boldsymbol{\alpha})\right) =& \frac{N_s}{2}\sum_{i=1}^{N_e}\log(\alpha_i) + \frac{1}{2}\log(\epsilon) + \frac{N}{2}\log(\beta) - \frac{N}{2}\log(2\pi) + \frac{1}{2}\log(\det(\mathbf{C})) \\ & - \frac{\beta}{2}\|\mathbf{X}^\mathsf{T}\mathbf{m} - \mathbf{t}\|^2 - \frac{1}{2}\mathbf{m}^\mathsf{T}\mathbf{I}'(\boldsymbol{\alpha})\mathbf{m}.\end{aligned} \tag{5.47}$$

The partial derivative of the hyperparameter likelihood with respect to α_i is:

$$\frac{\partial \log\left(p(\mathbf{D}|\beta,\boldsymbol{\alpha})\right)}{\partial \alpha_i} = \frac{N_s}{2\alpha_i} - \frac{1}{2}\sum_{j=k_i}^{k_i+N_s} c_{jj} - \frac{1}{2}\sum_{j=k_i}^{k_i+N_s} m_j^2, \tag{5.48}$$

where the summation is over the posterior parameters c_{jj}, m_j corresponding to α_i, i.e. $k_i = (i - 1)N_s + 1$. The partial derivative with respect to β is the same as in the case of an isotropic prior, i.e. Equation 5.42 can be used. Setting the derivative with respect to α_i to zero yields the following update equation:

$$\alpha_i = \frac{N_s}{\sum_{j=k_i}^{k_i+N_s} c_{jj} + m_j^2}. \tag{5.49}$$

To optimize the hyperparameters it is sufficient to sequentially update all the α_i and β until convergence. The result of this optimization is a set of optimal values for the hyperparameters α_i and β. Small α_i are equivalent to a large prior variance and allow large values for the weights corresponding to electrode i. Large α_i are equivalent to small prior variance and allow only small

values for the weights corresponding to electrode i. In other words the α_i can be used to rank electrodes by their importance for classification: electrodes with small α_i are more important than electrodes with large α_i.

A straightforward strategy to choose a subset of electrodes from the ranking given by the α_i is to specify a threshold τ and to retain only electrodes for which $\alpha_i \leq \tau$. Actually it is advantageous to apply the threshold already during optimization of hyperparameters, i.e. to remove rows and columns from the matrices $\mathbf{X}\mathbf{X}^\mathsf{T}$ and $\mathbf{I}'$ as soon as the α_i corresponding to these rows and columns start to take values larger than τ. The effect of this is that the inversion of the posterior precision matrix which is necessary during each iteration of the optimization procedure is speeded up and hence the optimization converges faster. The choice of τ is relatively uncritical because the α_i of irrelevant electrodes tend to take on very large values during optimization and hence it is sufficient to simply choose a τ that is large compared to the scale of the data. We chose $\tau = 10^8$ during the experiments with the SBDA algorithm. The results of these experiments are described in Chapter 7.

5.4.2 Automatic Relevance Determination and Backward Selection

A task that cannot be directly solved with the method for electrode selection described above, is to select a predetermined number of electrodes. This is the case because the number of electrodes retained by the SBDA algorithm depends on the threshold τ and on the dataset at hand. To allow for selection of electrode subsets with predetermined size, we use a strategy that is similar to what is known as backward selection in the feature selection literature. In this strategy first ARD is applied to the initial electrode set, i.e. the updates from Equations 5.49 and 5.44 are executed until the changes of the α_i and β become sufficiently small. Typically, during this first run of ARD some α_i take very large values and thus some electrodes are removed. If the desired number of electrodes is attained after the first run of ARD, the algorithm is stopped. Otherwise, the electrode with the largest α_i is removed and ARD is run again on the remaining electrodes. The motivation for removing the electrode with the largest α_i is that the weights corresponding to this electrode are constrained to be small and thus the electrode is unimportant. Running ARD after removal of the electrode with the largest α_i can either result in some small adjustments to the α_i and β or can lead to removal of further electrodes as the corresponding α_i take values larger than the threshold. The strategy of alternately removing the electrode with the largest α_i and running ARD on the subset of remaining electrodes is repeated until the desired number of electrodes is attained.

5.5 Classifying Single Trials and Sequences of Trials

The result of running the BDA or SBDA algorithms are maximum-likelihood values for the hyperparameters β and α and a posterior distribution for the weights and the bias value. The posterior distribution can be used to compute the predictive distribution of target values given a new input vector. However, what is ultimately needed are class probabilities, i.e. the probabilities that $p(\hat{y} = 1|\hat{\mathbf{x}})$ and $p(\hat{y} = -1|\hat{\mathbf{x}}) = 1 - p(\hat{y} = 1|\hat{\mathbf{x}})$. Furthermore, in a P300-based BCI we not only need to classify data from single EEG trials but also need to aggregate classification results from several single trials into a final decision. As explained in Chapter 4, aggregation of classification results is

often used in P300-based BCIs in order to improve classification accuracy.

In the next section we first describe the steps that are necessary to perform single trial classification with the help of BDA or SBDA. In the following section we then describe how classification results can be combined, i.e. how sequences of trials can be classified.

5.5.1 Single Trials

To obtain class probabilities for single trials we make use of the predictive distribution given by the BDA and SBDA algorithms. As shown in equations 5.33 and 5.34, the predictive distribution is a Gaussian distribution describing how probable different target values $\hat{t}$ are, given a new input vector $\hat{\mathbf{x}}$. Recalling that during training the target values N/N_1 and $-N/N_2$ were used for examples from class 1 and 2, a canonical way to compute the probability for class 1 is:

$$p(\hat{y} = 1|\beta, \alpha, \hat{\mathbf{x}}, \mathbf{D}) = \frac{p(\hat{t} = \frac{N}{N_1}|\beta, \alpha, \hat{\mathbf{x}}, \mathbf{D})}{p(\hat{t} = \frac{N}{N_1}|\beta, \alpha, \hat{\mathbf{x}}, \mathbf{D}) + p(\hat{t} = -\frac{N}{N_2}|\beta, \alpha, \hat{\mathbf{x}}, \mathbf{D})}. \tag{5.50}$$

Note that the accuracy of the above approach depends on the accuracy of the predictive distributions. In other words, using this approach is equivalent to assuming that $p(\hat{t} = \frac{N}{N_1}|\beta, \alpha, \hat{\mathbf{x}}, \mathbf{D})$ takes large values for examples $\hat{\mathbf{x}}$ from class 1 and that $p(\hat{t} = \frac{-N}{N_2}|\beta, \alpha, \hat{\mathbf{x}}, \mathbf{D})$ takes large values for examples from class 2.

For the datasets with which we performed tests, we found that the predictive probabilities were slightly inaccurate. In particular, we observed that the means of the predictive distributions were biased towards zero, when compared to the target values used during training. This is probably an effect of using a zero mean Gaussian prior for regularization and is difficult to avoid if one wants to avoid overfitting. Furthermore, investigation of the regression residuals showed that these were larger for examples from class 1 than from class 2. This can be explained if one takes into account that in P300 datasets there are typically less target trials (class 1) than nontarget trials (class 2). This imbalance typically leads to larger errors for the minority class, which are however not taken into account in the predictive distribution.

In the following we present two approaches that allow to deal with the aforementioned problems and yield relatively accurate class probabilities. An experimental comparison of the quality of the two approaches can be found in Chapter 7.

Van Gestel's Method

The first approach is used for least-squares support vector machines in the work of Van Gestel *et al.* (2002). Van Gestel *et al.* use the following equation for computing the probability of a new input vector $\hat{\mathbf{x}}$, given the class label y and parameters $\mathbf{w}$ and β_*[1]:

$$p(\hat{\mathbf{x}}|y = 1, \beta_*, \mathbf{w}) = \left(\frac{\beta_*}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\beta_*}{2}(\mathbf{w}^\mathsf{T}(\hat{\mathbf{x}} - \mathbf{c}_1))^2\right). \tag{5.51}$$

Here $\mathbf{c}_1$ is the mean of the training examples of class 1. The class-conditional probability for $\hat{\mathbf{x}}$ thus depends on the difference between class mean $\mathbf{c}$ and example $\hat{\mathbf{x}}$ and on the angle between this

[1] An analogous expression is used for class 2, i.e. for $y = -1$.

difference and the discriminant direction $\mathbf{w}$. The precision β_* corresponds roughly to the inverse variance of the projected differences between class means and training examples and is computed as follows in the approach of Van Gestel *et al.*:

$$\beta_* = \frac{N - \gamma_{\text{eff}}}{\sum_{i \in C_1} (\mathbf{m}^\mathsf{T}(\mathbf{x}_i - \mathbf{c}_1))^2 + \sum_{i \in C_2} (\mathbf{m}^\mathsf{T}(\mathbf{x}_i - \mathbf{c}_2))^2}. \tag{5.52}$$

Here $\mathbf{m}$ is the mode of the posterior distribution of $\mathbf{w}$ (cf. Equation 5.31) and γ_{eff} is the effective number of parameters (see (Bishop, 2006; MacKay, 1992; Van Gestel *et al.*, 2002) for an explanation).

To remove the dependence on the discriminant direction, Van Gestel *et al.* integrate over the posterior distribution of $\mathbf{w}$.

$$p(\hat{\mathbf{x}}|y = 1, \beta_*) = \int p(\hat{\mathbf{x}}|y = 1, \beta_*, \mathbf{w}) p(\mathbf{w}|\beta, \alpha, \mathbf{D}) \, d\mathbf{w} \tag{5.53}$$

This integral can be solved by noting that under the posterior distribution $p(\mathbf{w}|\beta, \alpha, \mathbf{D})$, the expression $\mathbf{w}^\mathsf{T}(\hat{\mathbf{x}} - \mathbf{c}_1)$ corresponds to a univariate Gaussian with mean $\mathbf{m}^\mathsf{T}(\hat{\mathbf{x}} - \mathbf{c}_1)$ and variance $(\hat{\mathbf{x}} - \mathbf{c}_1)^\mathsf{T}\mathbf{C}(\hat{\mathbf{x}} - \mathbf{c_1})$. Using additionally the fact that a convolution of two Gaussians is another Gaussian, with variance equal to the sum of the variances of the two original Gaussians and mean equal to the sum of the two means, the probability for $\hat{\mathbf{x}}$ then is

$$p(\hat{\mathbf{x}}|y = 1, \beta_*) = \frac{1}{\sqrt{2\pi(\beta_*^{-1} + \sigma_1^2)}} \exp\left(-\frac{(\mathbf{m}^\mathsf{T}(\hat{\mathbf{x}} - \mathbf{c}_1))^2}{2(\beta_*^{-1} + \sigma_1^2)} \right), \tag{5.54}$$

with

$$\sigma_1^2 = (\hat{\mathbf{x}} - \mathbf{c}_1)^\mathsf{T}\mathbf{C}(\hat{\mathbf{x}} - \mathbf{c_1}). \tag{5.55}$$

Now using Bayes' rule the probabilities for the class labels can be computed as

$$p(y|\hat{\mathbf{x}}, \beta_*) = \frac{p(\hat{\mathbf{x}}|y, \beta_*) p(y)}{\sum_{y \in \mathcal{Y}} p(\hat{\mathbf{x}}|y, \beta_*) p(y)}, \tag{5.56}$$

where $\mathcal{Y} = \{-1, 1\}$ and the $p(y)$ allow to take into account prior class probabilities.

A Leave-One-Out Approach

While Van Gestel's method is sound and yields good results, other possibilities exist for the computation of class probabilities. One such possibility is the leave-one-out approach presented in the following. An important difference to Van Gestel's method is that the contribution to the variance from the posterior uncertainty in the parameters, i.e. Equation 5.55 is completely ignored in the leave-one-out approach. The motivation for this is that for reasonably large training sets this contribution is very small when compared to β_* (Qazaz *et al.*, 1996). Hence, the solution of Equation 5.55 for each test example can be avoided and time can be saved during prediction. Moreover the mean projection of the classes is computed with a leave-one-out method instead of using simply the training examples as in Van Gestel's method. The potential advantage of this is that overfitting effects due to the use of the same training examples for the computation of discriminant directions

as well as for the mean projections are avoided. The disadvantage is that a considerable amount of computational complexity is added to the training stage of classifiers by the leave-one-out approach.

The leave-one-out approach uses univariate generative Gaussian models to capture the between-class and within-class variation of the mean of the predictive distribution. Denoting by $\mathbf{m}$ the mean of the posterior distribution (Equation 5.31), by $\hat{\mathbf{x}}$ the new input vector we want to classify, by $\hat{t} = \mathbf{m}^\mathsf{T}\hat{\mathbf{x}}$ the mean of the predictive distribution, and by μ_1,μ_2 and σ_1^2,σ_2^2 the parameters of the Gaussians, the class probability can be expressed as follows:

$$p(y|\hat{t},\mu_1,\mu_2,\sigma_1^2,\sigma_2^2) = \frac{p(\hat{t}|\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,y)p(y)}{\sum_{y\in\mathcal{Y}} p(\hat{t}|\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,y)p(y)}. \tag{5.57}$$

Here $p(\hat{t}|\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,y)$ are univariate Gaussian probabilities, i.e. :

$$p(\hat{t}|\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,y=1) = \frac{1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{1}{2\sigma_1^2}(\hat{t}-\mu_1)^2\right) \tag{5.58}$$

$$p(\hat{t}|\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,y=-1) = \frac{1}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{1}{2\sigma_2^2}(\hat{t}-\mu_2)^2\right) \tag{5.59}$$

The accuracy of class probabilities computed with this approach depends of course on realistic estimates for the means μ_1, μ_2 and variances σ_1^2, σ_2^2. One possibility to obtain such estimates would be to simply use the training examples, i.e. one could use:

$$\mu_k = \frac{1}{N_k}\sum_{i\in C_k}\mathbf{m}^\mathsf{T}\mathbf{x}_i \tag{5.60}$$

$$\sigma_k^2 = \frac{1}{N_k}\sum_{i\in C_k}(\mathbf{m}^\mathsf{T}\mathbf{x}_i-\mu_k)^2, \tag{5.61}$$

where N_k denotes the number of training examples in class k, and C_k denotes the set containing the indices of the training examples belonging to class k. This is however not advisable because using the training set for computing the posterior mean and for computing the variance σ_k^2 of the projected training examples typically leads to overly optimistic (too small) estimates for the variances.

To obtain realistic estimates for σ_k^2, we employed a leave-one-out procedure, in which each training example is removed once from the training set. Denoting the mean of the posterior distribution computed without training example i by $\mathbf{m}_{\setminus i}$, we can compute class conditional estimates of the mean and variance of the mean of the predictive distribution by the following equations:

$$\mu_k = \frac{1}{N_k}\sum_{i\in C_k}\mathbf{m}_{\setminus i}^\mathsf{T}\mathbf{x}_i \tag{5.62}$$

$$\sigma_k^2 = \frac{1}{N_k}\sum_{i\in C_k}(\mathbf{m}_{\setminus i}^\mathsf{T}\mathbf{x}_i-\mu_k)^2. \tag{5.63}$$

Note that to exactly compute the posterior mean $\mathbf{m}_{\setminus i}$ we need to compute the hyperparameters $\beta_{\setminus i}$ and $\alpha_{\setminus i}$ and additionally the covariance matrix $\mathbf{C}_{\setminus i}$. Since this has to be done once for each training example, a naive implementation of the leave-one-out procedure would be computationally very demanding. To reduce the computational complexity we assume that the changes in the

hyperparameters resulting from the removal of one training example are negligible, i.e. $\beta_{\setminus i} \approx \beta$ and $\alpha_{\setminus i} \approx \alpha$. Furthermore, to compute the covariance $\mathbf{C}_{\setminus i}$ we use the Woodbury identity (see for example Golub and Van Loan (1996)), which allows to quickly compute rank-1 updates (or downdates) of the covariance matrix $\mathbf{C}$.

$$\mathbf{C}_{\setminus i} = \mathbf{C} - \frac{\mathbf{C}\mathbf{x}_i\mathbf{x}_i^T\mathbf{C}}{-\frac{1}{\beta} + \mathbf{x}_i^T\mathbf{C}\mathbf{x}_i} \tag{5.64}$$

5.5.2 Sequences of Trials

Given the algorithms for computing class probabilities presented in the previous section, the classification of sequences of trials is relatively straightforward. Assuming independence of single trials, the probability for a sequence of class labels given a sequence of input vectors can in general be expressed as follows:

$$p(\hat{\mathbf{y}} = y_1 \dots y_T | \hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_T) = \frac{\prod_{i=1}^{T} p(\hat{y} = y_t | \hat{\mathbf{x}}_t)}{\sum_{\mathbf{l} \in \mathcal{L}} \prod_{i=1}^{T} p(\hat{y} = l_t | \hat{\mathbf{x}}_t)} \quad \text{for } \hat{\mathbf{y}} \in \mathcal{L}. \tag{5.65}$$

Here $\mathbf{l} = l_1 \dots l_T$ is a sequence of labels of length T, $\mathcal{L}$ is the set of all possible class label sequences of length T, and we have omitted the conditioning on μ_1, μ_2, σ_1^2, and σ_2^2. The reader might object that the computation of the denominator in the above equation becomes computationally infeasible for large T because $|\mathcal{L}| = 2^T$. The objection is correct, however it is not relevant for the application we are envisaging here, namely P300-based BCIs. In P300-based BCIs the number of possible label sequences is equal to the number of different stimuli and hence the denominator can be computed easily.

To see how the probabilities for sequences of class labels can be used in a P300-based BCI, let us recall the scheme for aggregating information from a variable number of trials which was reviewed in Chapter 4 on page 50. In this scheme stimuli are presented blockwise until a reliable decision about the target stimulus can be taken. The advantage of this scheme over the other decision schemes presented in Chapter 4 is that the number of stimuli can be dynamically adapted to the performance of the user and the noise level in the signals. To implement the scheme, Serby *et al.* (2005) used nonprobabilistic classifiers and combined information from several trials by averaging classifier outputs. Decisions were taken when the averaged classifier outputs exceeded a threshold. Thresholds depended on the number of averaged classifier outputs and were computed with a method that is not specified in detail in the paper of Serby *et al.* (2005).

The probabilities for sequences of class labels allow us to implement the scheme of Serby *et al.* in a more straightforward way. Taking as example the four-stimuli setup depicted in Fig. 5.2, the probability that the user was concentrating on stimulus 1 can be easily computed. In particular after the first block of stimulus presentations this probability can be obtained by using Equation 5.65 with

$$\hat{\mathbf{y}} = \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad \text{and} \quad \mathcal{L} = \left\{ \begin{bmatrix} 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \\ 1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \\ -1 \\ 1 \end{bmatrix} \right\}. \tag{5.66}$$

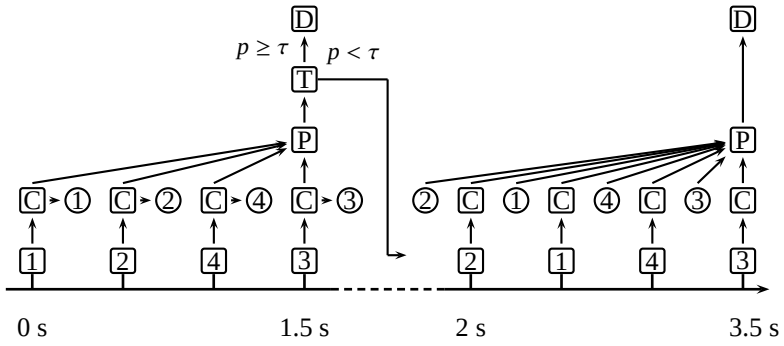

Figure 5.2 — Decision after a variable number of blocks using a probabilistic approach (symbols in squares represent operations, symbols in circles represent variables). Four different stimuli are presented in random order with an interstimulus interval of 500 ms (1,2,3,4). The EEG segment corresponding to each stimulus presentation is classified (C); the output of the classifier is a probability indicating how similar the EEG segment is to a P300. After the first block of stimulus presentations, the classifier outputs are combined in order to compute for each stimulus the probability that the user was concentrating on it (P). If the maximum of these probabilities is larger than a certain threshold a decision is taken (D), i.e. the systems executes the command associated to the stimulus with the largest probability. If the maximum is smaller than the threshold a second block of stimuli is presented. The classifier outputs from the second block of stimuli are combined with the outputs from the first block (P). The command associated to the stimulus with the largest probability is executed (D).

The probabilities for the other stimuli can be computed by accordingly changing $\hat{\mathbf{y}}$. Adaptive decisions are taken by comparing the maximum of the probabilities for all four stimuli to a threshold. If the maximum of the probabilities is smaller than the threshold a new block of stimuli is presented and the probabilities are recomputed using all blocks presented so far. If the maximum of the probabilities is larger than the threshold the system decides that the user was concentrating on the stimulus corresponding to the maximal probability (see Fig. 5.2). Compared to the implementation of the adaptive decision scheme described by Serby *et al.* our implementation has the advantage that only one threshold has to be specified and that this threshold has an intuitive meaning. Setting the threshold to a value τ means that one accepts approximately $100(1-\tau)\%$ wrong decisions.

Note however, that using a fixed threshold to decide when to stop sampling data is not necessarily optimal. In fact, the problem of deciding for one of several hypotheses after evaluating a variable number of samples is relatively complex and is known as sequential analysis or sequential hypothesis testing in the decision theory literature. A decision theory based analysis of the threshold procedure we used can be found in (Draglia *et al.*, 1999). Note also, that the approach we presented for classification of sequences of trials is certainly not the only possible one. While the advantages of our approach are that it is simple and straightforward and leads to good results, other possible approaches exist. In particular, it would be interesting to test methods for combining results obtained from multiple classifiers (Kittler *et al.*, 1998). For example the majority voting rule could be used in the context presented here, by first finding for each stimulus block the stimulus with the maximal score. Then, a decision about the target stimulus could be taken by finding the stimulus that has the maximal score in the majority of blocks.

Apart from using probabilities for taking adaptive decisions, other, extended, applications can

be envisaged. For example in the P300 speller system, it might be interesting to combine the probabilities computed from the EEG with probabilities computed from a language model. This approach could also be combined with adaptive decisions, i.e. if the language model strongly reduces the a priori uncertainty for the next symbol to be spelled, the number of stimuli would be small. If the language model is unsure about the next symbol the number of stimuli would be large. Another area in which the probabilistic approach to classification might be useful is the area of asynchronous BCI systems. More generally, every BCI application in which a priori probabilities for commands can be computed could profit from the probabilistic approach to classification presented in the last sections.

5.6 Conclusion

In this chapter we have discussed algorithms for learning classifiers from training data and for performing classification of new data not used during training. The algorithms we discussed can be seen as linear versions of the well-known relevance vector machine (Tipping, 2001) and are based on the evidence framework presented originally by MacKay (1992). Starting from least squares regression and FDA we have described BDA, an algorithm which is obtained by applying the evidence framework to the discriminant analysis scenario. When compared to other algorithms that have been used in BCI systems, the main advantages of BDA are that a regularized discriminant is computed and that regularization constants are estimated automatically. Compared to simple algorithms such as FDA, which work without regularization, the advantage is that high-dimensional data can be used for training without the danger of overfitting. Compared to regularized algorithms such as for example regularized FDA or the support vector machine (SVM) the advantage is that regularization constants are estimated automatically. Time-consuming and cumbersome cross-validation procedures are thus not necessary. As an extension of the basic BDA algorithm we have described the SBDA algorithm which is based on a technique known as automatic relevance determination and can be employed to perform electrode selection when learning classifiers from EEG datasets.

After the description of the inference algorithms BDA and SBDA we have described probabilistic approaches for the classification of single trials and for the classification of sequences of trials. The first approach for probabilistic single trial classification we described, is the approach used in the work of Van Gestel *et al.* on Bayesian least-squares support vector machines. The main idea underlying the second approach is to use a leave-one-out procedure to estimate Gaussian probability models for the one-dimensional projections of feature vectors.

Building on the class probabilities for single trials, we have presented a probabilistic approach to classification of sequences of trials. As we have seen, this can be used for a straightforward implementation of a P300-based BCI system, in which the number of stimuli is automatically adapted to the noise level of the signals and to the performance of the user.

In the next two chapters the algorithms that were described in the present chapter will be tested with different datasets. In Chapter 6 we describe a BCI system which was used to record P300 datasets from several disabled and able-bodied subjects. These datasets are then used to compare BDA with FDA and to test different (static) electrode configurations. In Chapter 7 we use the data from disabled and able-bodied subjects to test electrode selection with SBDA and the adaptive

decision scheme. Additionally, in order to test how the algorithms presented here compare to the state-of-the-art, experiments are performed with BCI competition datasets in Chapter 7.

An Efficient Brain-Computer Interface for Disabled Subjects

6

6.1 Introduction

In this chapter we present an efficient BCI for disabled subjects. The system is based on the P300 evoked potential and is tested with data from five disabled and four able-bodied subjects. Except for some minor modifications in Sections 6.4 and 6.5, the material in this chapter is identical to that presented in (Hoffmann *et al.*, 2007).

The chapter starts in Section 6.2 with a brief review on BCIs for disabled subjects. Then, in Section 6.3 the materials and methods used for recording and analyzing data are discussed. In Section 6.4 results are presented. An important result is that the classification accuracy and bitrate achieved for the disabled subjects are significantly beyond those previously reported in the literature. Additional results concern the classification accuracy and bitrate achievable with Bayesian linear discriminant analysis (BDA) and Fisher's discriminant analysis (FDA) and a comparison of different, static electrode configurations. In Section 6.5 the results are discussed and reasons for the good classification accuracy and bitrate achieved for disabled subjects are sought. The chapter is summarized in Section 6.6.

6.2 Related Work

One of the earliest systems that used the EEG and was tested with disabled subjects was described by Birbaumer *et al.* (1999). In their pioneering work, Birbaumer *et al.* showed that patients suffering from amyotrophic lateral sclerosis (ALS) can use a BCI to control a spelling device and communicate with their environment. The system relied on the fact that patients were able to learn voluntary regulation of slow cortical potentials (SCP), i.e. voltage shifts of the cerebral cortex which occur in the frequency range 1-2 Hz (cf. Chapter 2, page 16). Drawbacks of the system were that it usu-

ally took several months of patient training before the subjects could control the system and that communication was relatively slow.

Parallel to the work of Birbaumer *et al.* BCI systems were developed that used changes in brain activity correlated to motor imagery (Pfurtscheller and Neuper, 2001). While these systems were for a long time tested exclusively with able-bodied and quadriplegic subjects, recently tests have been performed with ALS patients and other disabled subjects. Positive results have been obtained by Kübler *et al.* (2005) who showed that ALS patients can learn to control motor imagery based BCI systems. However, as for the system based on SCP, users were trained over several months and communication was relatively slow. Negative results have been obtained by Hill *et al.* (2006), who tested a motor imagery based BCI with several completely locked-in patients and could not obtain signals that were suitable for communication. One possible reason for the different results is the fact that in the study of Kübler *et al.* the patients were not completely locked-in whereas the patients in the study of Hill *et al.* were completely locked in. Furthermore, in the study of Kübler *et al.* several training sessions were used whereas in the work of Hill *et al.* only one, relatively long training session was used. In summary, it has thus been shown that motor imagery based systems can be used by disabled subjects, however positive evidence is limited to cases in which subjects were not completely locked-in and followed a long training protocol.

Recently, two studies have been published in which P300-based BCI systems were tested with disabled subjects. Piccione *et al.* (2006) tested a 2D cursor control system with five disabled and seven able-bodied subjects. For cursor control, a four-choice P300 paradigm was used. Subjects had to concentrate on one of four arrows flashing every 2.5 s in random order in the peripheral area of a computer screen. Signals were recorded from one electrooculogram electrode and four EEG electrodes, preprocessed with independent component analysis and classified with a neural network. The results described by Piccione *et al.* showed that the P300 is a viable control-signal for disabled subjects. However, the average communication speed obtained in their study was relatively low when compared to state-of-the-art systems, as for example the systems described by Kaper *et al.* (2004); Thulasidas *et al.* (2006). This was the case for the disabled subjects, as well as for able-bodied subjects and can probably be ascribed to the use of signals from only few electrodes, the small number of different stimuli, and long interstimulus intervals (ISIs).

Sellers and Donchin (2006) also used a four-choice paradigm and tested their system with three subjects suffering from ALS and three able-bodied subjects. In their study four stimuli ('YES', 'NO', 'PASS', 'END') were presented every 1.4 s in random order, either in the visual modality, in the auditory modality, or in a combined auditory-visual modality. Signals from three electrodes were classified with a stepwise linear discriminant algorithm. The research of Sellers and Donchin showed that P300 based communication is possible for subjects suffering from ALS. The research also showed that communication is possible in the visual, auditory, and combined auditory-visual modality. However, as in the work of Piccione *et al.*, the achieved classification accuracy and communication rate were low when compared to state-of-the-art results. This can again be ascribed to the small number of electrodes, the small number of different stimuli, and long ISIs.

Figure 6.1 — The display used for evoking the P300. Images were flashed, one at a time, by changing the overall brightness of images.

6.3 Materials and Methods

6.3.1 Experimental Setup

Users were facing a laptop screen on which six images were displayed (see Fig. 6.1). The images showed a television, a telephone, a lamp, a door, a window, and a radio. The images were selected according to an application scenario in which users can control electrical appliances via a BCI system. The application scenario served however only as an example and was not pursued in further detail.

The images were flashed in random sequences, one image at a time. Each flash of an image lasted for 100 ms and during the following 300 ms none of the images was flashed, i.e. the ISI was 400 ms. The EEG was recorded at 2048 Hz sampling rate from thirty-two electrodes placed at the standard positions of the 10-20 international system. A Biosemi Active Two amplifier was used for amplification and analog to digital conversion of the EEG signals. Signal processing and machine learning algorithms were implemented with MATLAB. The stimulus display and the online access to the EEG signals were implemented as dynamic link libraries (DLLs) in C. The DLLs were accessed from MATLAB via a MEX interface.

6.3.2 Subjects

The system was tested with five disabled and four able-bodied subjects. The disabled subjects were all wheelchair-bound but had varying communication and limb muscle control abilities (see Table 6.1). Subjects 1 and 2 were able to perform simple, slow movements with their arms and hands but were unable to control other extremities. Spoken communication with subjects 1 and 2 was possible, although both subjects suffered from mild dysarthria. Subject 3 was able to perform restricted movements with his left hand but was unable to move his arms or other extremities.

	S1	S2	S3	S4	S5
Diagnosis	Cerebral palsy	Multiple sclerosis	Late-stage amyotrophic lateral sclerosis	Traumatic brain and spinal-cord injury, C4 level	Post-anoxic encephalopathy
Age	56	51	47	33	43
Age at illness onset	0 (perinatal)	37	39	27	37
Sex	M	M	M	F	M
Speech production	Mild dysarthria	Mild dysarthria	Severe dysarthria	Mild dysarthria	Severe hypophony
Limb muscle control	Weak	Weak	Very weak	Weak	Very weak
Respiration control	Normal	Normal	Weak	Normal	Normal
Voluntary eye movement	Normal	Mild nystagmus	Normal	Normal	Balint's syndrome

Table 6.1 — Subjects from which data was recorded in the study of the environment control system.

Spoken communication with subject 3 was impossible. However the patient was able to answer yes/no questions with eye blinks. Subject 4 had very little control over arm and hand movements. Spoken communication was possible with subject 4, although a mild dysarthria existed. Subject 5 was only able to perform extremely slow and relatively uncontrolled movements with hands and arms. Due to a severe hypophony and large fluctuations in the level of alertness, communication with subject 5 was very difficult. Subjects 6 to 9 were PhD students recruited from our laboratory (all male, age 30 ± 2.3). None of subjects 6 to 9 had known neurological deficits.

6.3.3 Experimental Schedule

Each subject completed four recording sessions. The first two sessions were performed on one day and the last two sessions on another day. For all subjects the time between the first and the last session was less than two weeks. Each of the sessions consisted of six runs, one run for each of the six images. The following protocol was used in each of the runs.

1. Subjects were asked to count silently how often a prescribed image was flashed (For example: "Now please count how often the image with the television is flashed").

2. The six images were displayed on the screen and a warning tone was issued.

3. Four seconds after the warning tone, a random sequence of flashes was started and the EEG was recorded. The sequence of flashes was block-randomized, this means that after six flashes each image was flashed once, after twelve flashes each image was flashed twice, etc.. The number of blocks was chosen randomly between 20 and 25. On average 22.5 blocks of six

flashes were displayed in one run, i.e. one run consisted on average of 22.5 target (P300) trials and $22.5 \cdot 5 = 112.5$ nontarget (non P300) trials.

4. In the second, third, and fourth session the target image was inferred from the EEG with a simple classifier. The classifier was trained from the data recorded in session one, one to two, and one to three, respectively. The algorithm described in Hoffmann *et al.* (2006) was used for preprocessing and the algorithm described in Hoffmann *et al.* (2004) was used for classification. At the end of each run the image inferred by the classification algorithm was flashed five times to give feedback to the user. The feedback served to keep the users interested and concentrated during the training sessions.

5. After each run subjects were asked what their counting result was. This was done in order to monitor performance of the subjects.

The duration of one run was approximately one minute and the duration of one session including setup of electrodes and short breaks between runs was approximately 30 minutes. One session comprised on average 810 trials, and the whole data for one subject consisted on average of 3240 trials.

6.3.4 Offline Analysis

The impact of different electrode configurations and machine learning algorithms on classification accuracy was tested in an offline procedure. For each subject four-fold cross-validation was used to estimate average classification accuracy. More specifically, the data from three recording sessions were used to train a classifier and the data from the left-out session was used for validation. This procedure was repeated four times so each session served once for validation.

Preprocessing

Before learning a classification function and before validation, several preprocessing operations were applied to the data. The preprocessing operations were applied in the order stated below.

1. *Referencing*
   The average signal from the two mastoid electrodes was used for referencing.

2. *Filtering*
   A 6th order forward-backward Butterworth bandpass filter was used to filter the data. Cutoff frequencies were set to 1.0 Hz and 12.0 Hz. The MATLAB function *butter* was used to compute the filter coefficients and the function *filtfilt* was used for filtering.

3. *Downsampling*
   The EEG was downsampled from 2048 Hz to 32 Hz by selecting each 64th sample from the bandpass-filtered data[1].

[1]Note that a more robust approach to downsampling would be to use averaging with a window size of 64 samples. Although, due to the preceding lowpass filtering, the improvement of averaging compared to selecting each 64th sample is probably small, averaging should be used in future versions of the system presented here.

4. *Single Trial Extraction*
   Single trials of duration 1000 ms were extracted from the data. Single trials started at stimulus onset, i.e. at the beginning of the intensification of an image, and ended 1000 ms after stimulus onset. Due to the ISI of 400 ms, the last 600 ms of each trial were overlapping with the first 600 ms of the following trial.

5. *Windsorizing*
   Eye blinks, eye movement, muscle activity, or subject movement can cause large amplitude outliers in the EEG. To reduce the effects of such outliers, the data from each electrode were windsorized. For the samples from each electrode the 10th percentile and the 90th percentile were computed. Amplitude values lying below the 10th percentile or above the 90th percentile were then replaced by the 10th percentile or the 90th percentile, respectively.

6. *Scaling*
   The samples from each electrode were scaled to the interval $[-1, 1]$. Scaling constants were computed for each electrode from all trials in the training set and then applied to the validation data. Note that scaling was *not* done on a trial by trial basis. Instead the same scaling constants were used for all trials. This is important since scaling each trial individually could potentially destroy important amplitude information characterizing the P300.

7. *Electrode Selection*
   Four static electrode configurations with different numbers of electrodes were tested. The electrode configurations are shown in Fig. 6.2.

8. *Feature Vector Construction*
   The samples from the selected electrodes were concatenated into feature vectors. The dimensionality of the feature vectors was $N_e \times N_s$, where N_e denotes the number of electrodes and N_s denotes the number of temporal samples in one trial. Due to the trial duration of 1000 ms and the downsampling to 32 Hz, N_s always equaled 32. Depending on the electrode configuration, N_e equaled four, eight, sixteen, or thirty-two.

Machine Learning and Classification

Classifiers and the percentile values used for windsorizing were trained on the data from three sessions and validated on the left-out fourth session. Training datasets contained 405 target trials and 2025 nontarget trials and validation datasets consisted of 135 target and 675 nontarget trials (these are average values cf. Section 6.3.3). BDA was used to learn classifiers (cf. Chapter 5, page 60). To compare the performance of BDA with a standard algorithm, in a second set of experiments classifiers were computed with FDA. In particular the version of FDA based on the Moore-Penrose pseudoinverse of the within-class scatter matrix was used (cf. Chapter 5, page 58). Both algorithms were fully automatic, i.e. no user intervention was required to adjust hyperparameters, and the computation of classifiers took less than one minute on a standard PC.

After the classifiers had been trained, they were applied to validation data in the following way. For each run in the validation session, the single trials corresponding to the first twenty blocks of

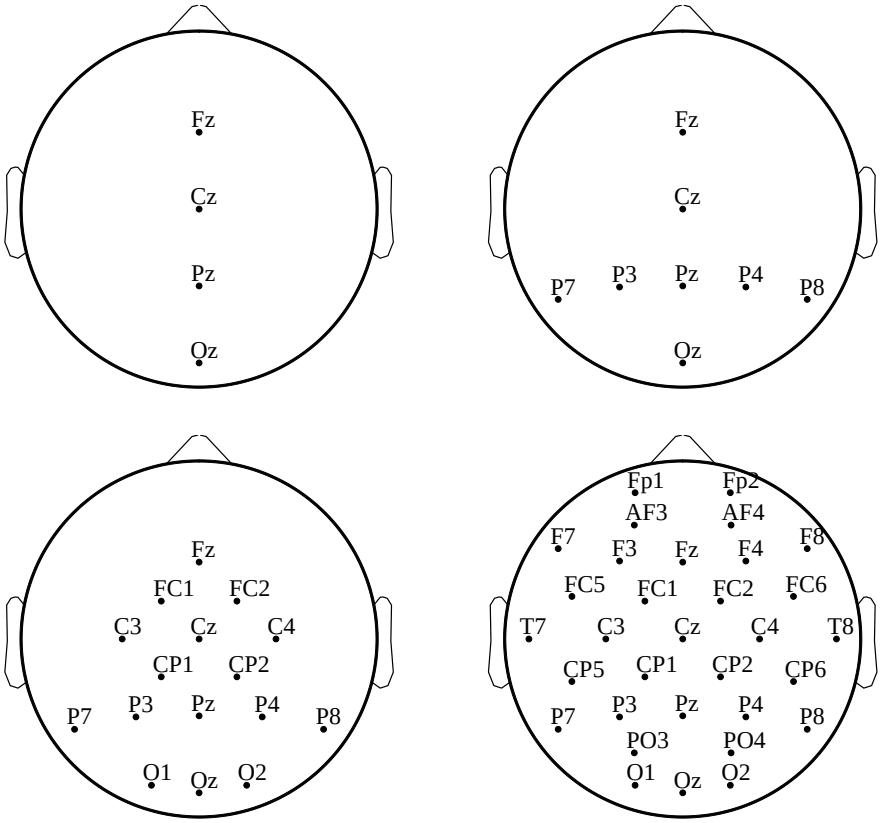

Figure 6.2 — Electrode configurations used in the experiments. From top left to bottom right: Configuration I (four electrodes), configuration II (eight electrodes), configuration III (sixteen electrodes), and configuration IV (thirty-two electrodes).

flashes were extracted using the preprocessing operations. Then the single trials were classified. This resulted in twenty blocks of classifier outputs. Each block consisted of six classifier outputs, one output for each image on the display. To decide which image the user was concentrating on, the classifier outputs were summed over blocks for each image and then the image with the maximum summed classifier output was selected[1]. Different tradeoffs between the time needed to take a decision and the classification accuracy were simulated by varying the number of summed classifier outputs, i.e. the number of blocks. The performance measures used for comparing classifiers are described in more detail in the next section.

6.4 Results

6.4.1 Performance Measures

To illustrate and compare the results obtained for different subjects, classifiers, and electrode configurations we have used the following performance measures. All performance measures are based

[1]This decision scheme is described in more detail in Chapter 4 and is depicted in Fig. 4.5 on page 49.

on the cross-validation procedure presented in the previous section.

- *Classification Accuracy Graphs*
  Classification accuracy graphs illustrate the dependence of classification accuracy on the amount of aggregated data. This is best understood by considering the following notation. Let $t^r_{sb} \in \mathbb{R}$ denote the classifier output corresponding to the presentation of stimulus s, in block b, during run r. The identity $\mathrm{ts}(r, B)$ of the target stimulus in run r, taking into account data from B blocks of stimulus presentations, is then computed as follows:

$$\mathrm{ts}(r, B) = \arg\max_s \sum_{b=1}^{B} t^r_{sb}. \tag{6.1}$$

  The classification accuracy $\mathrm{ac}(B)$ as a function of the number of blocks can be expressed as:

$$\mathrm{ac}(B) = \frac{1}{R} \sum_{r=1}^{R} \mathrm{I}\left(\mathrm{ts}(r, B) = \mathrm{gt}(r)\right). \tag{6.2}$$

  Here R denotes the number of runs in the validation set, I denotes the indicator function, and $\mathrm{gt}(r)$ denotes the groundtruth identity of the target stimulus in run r. The classification accuracy as a function of the number of blocks can easily be converted into graphs depending on time by noting that each block has a duration of 6×400 ms = 2.4 s.

- *Bitrate Graphs and Maximum Bitrate*
  The dependence of communication speed (the bitrate) on the amount of aggregated data was computed by applying the definition of Wolpaw *et al.* (2002) to the classification accuracy graphs. Maximum bitrates were computed by finding the maximum of the bitrate graphs.

  Note that in the bitrate definition of Wolpaw *et al.* (cf. Chapter 4, page 51) it is assumed that the user communicates an infinite amount of data and that he encodes the data he wants to transmit in an optimal way, such that eventual communication errors can be corrected by a decoding algorithm. In the environment control application discussed here the amount of transmitted data is limited and no encoding takes place. Hence, the bitrates depicted in the graphs on page 84 are only actually achievable at points where the classification accuracy is 100%. When the classification accuracy is lower than 100% an optimal encoding procedure would be necessary to actually achieve the depicted bitrates. Despite this drawback of the bitrate definition of Wolpaw *et al.* we have nevertheless used it as it is widely used for comparisons between different types of BCI systems.

- *Per Block Accuracy (PBA)*
  The principal performance measure used for comparing classifiers is what we have termed "per block accuracy". The motivation for introducing this performance measure is that it is difficult to find a sensible metric for comparing graphs of accuracy or bitrate. Maximum bitrate is also unsuitable as a performance measure because it depends mainly on the data recorded during the first few stimulus presentations and thus might have high variance. The PBA is computed from all blocks of EEG trials seen during cross-validation and hence should

be more reliable. To define this performance measure more precisely, let us introduce the following notation.

$$\tilde{\text{ts}}(r, b) = \arg\max_s t^r_{sb}. \tag{6.3}$$

As before t^r_{sb} denotes the classifier output for stimulus s, in block b, during run r. Therefore, $\tilde{\text{ts}}(r, b)$ denotes the identity of the target stimulus computed only from block b in run r. The PBA is then computed as:

$$\text{pba} = \frac{1}{RB} \sum_{r=1}^{R} \sum_{b=1}^{B} \text{I}\left(\tilde{\text{ts}}(r, b) = \text{gt}(r)\right). \tag{6.4}$$

Here, by a slight abuse of notation, B denotes the total number of blocks in each run.

6.4.2 General Observations

Graphs of classification accuracy and bitrate are shown in Fig. 6.3. Electrode configuration (II) in conjunction with BDA as classification method was used for these graphs[1]. The maximum bitrates for all possible combinations of electrode configuration and classification algorithm are listed in Table 6.2. The PBAs for all possible combinations of electrode configuration and classification algorithm are listed in Table 6.3.

Data for subject 5 are not included in Fig. 6.3 , Table 6.2, and Table 6.3 because classification accuracies above chance level could not be obtained. During the experiments a speech therapist helped to communicate with subject 5. However, it was not clear if the subject understood the instructions given before the experiments. Furthermore, the level of alertness of the subject fluctuated strongly and rapidly during experiments.

All of the subjects, except for subjects 6 and 9, achieved an average classification accuracy of 100% after 12 or more blocks of stimulus presentations were averaged (i.e. after 28.8 s). Subject 6 reported that he accidentally concentrated on the wrong stimulus during one run in session 1. This explains the lower average classification accuracy for this subject. In all other runs the average classification accuracy after more than 12 blocks was 100% for subject 6. The somewhat lower performance for subject 9 is restricted to session 4, i.e. in sessions 1 to 3 subject 9 always reached 100% classification accuracy. The reason for the lower performance in session 4 might be fatigue.

The best performance was achieved by subject 8. Subject 8 was highly concentrated and motivated during the experiments. It is known that motivation and arousal in general increase P300 amplitude (Carrillo-de-la Pena and Cadaveira, 2000). One possible explanation for the very good performance of subject 8 might thus be the fact that the subject was very motivated.

6.4.3 Differences between Disabled and Able-bodied Subjects

The differences that can be observed between disabled and able-bodied subjects depend on the performance measure used. If maximum classification accuracy is used as performance measure, no

[1]Electrode configuration (II) was chosen for plotting because it represents a good tradeoff between classification performance and practical applicability of a BCI system. To keep the plots uncluttered, the curves for FDA, which for electrode configuration (II) are very similar to those of BDA, are not shown.

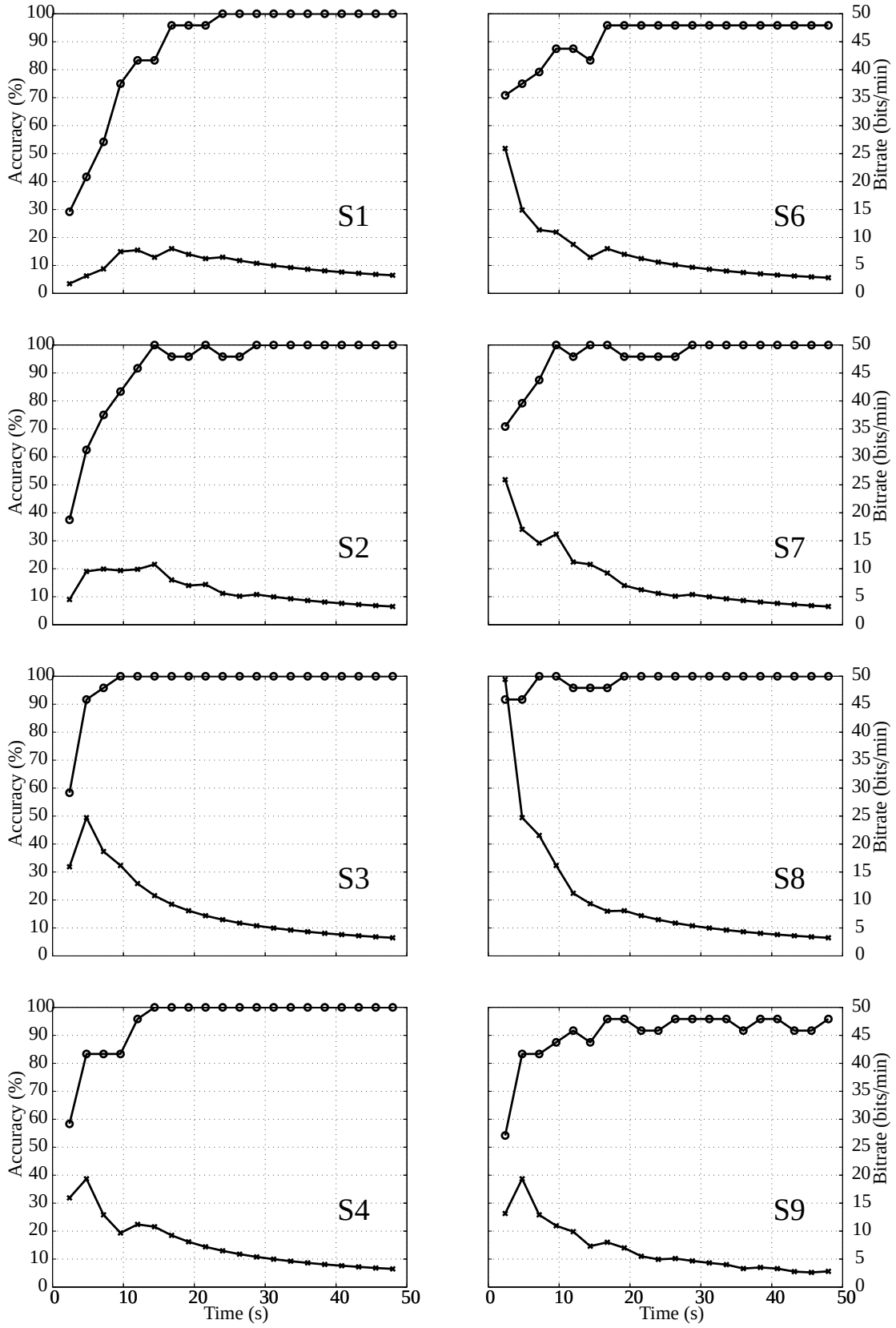

Figure 6.3 — Classification accuracy and bitrate plotted vs. time. The panels show the classification accuracy obtained with BDA and the eight electrode configuration, averaged over four sessions (circles) and the corresponding bitrate (crosses), for disabled subjects (S1-S4) and able-bodied subjects (S6-S9).

differences can be found between able-bodied and disabled subjects. This is shown for classification with BDA and the eight electrodes configuration in Fig. 6.3. The same behavior was found for the other combinations of classifier and electrode configuration (not shown). If bitrate is used as performance measure, differences between disabled and able-bodied subjects can readily be observed. Able-bodied subjects achieved higher maximum bitrates than disabled subjects. This was the case for all combinations of classifier and electrode configuration (see Table 6.2). Differences between disabled and able-bodied subjects were also found in the PBA (see Table 6.3). This indicates that the smaller performance of disabled subjects is not restricted to the first few stimulus presentations but persist also for stimuli presented later during a run.

6.4.4 Electrode Configurations and Classification Methods

Using different electrode configurations in conjunction with BDA and FDA yielded the results shown in Fig. 6.4 and Table 6.3. For BDA one can observe that increasing the number of electrodes always led to an increase in performance. The largest improvements were obtained by using eight instead of four electrodes and by using sixteen instead of eight electrodes. The increase in performance obtained by using thirty-two electrodes was relatively small. For FDA the performance was not directly related to the number of electrodes. Using eight electrodes led to a strong increase in performance over the four electrode configuration. A further small improvement was obtained by using sixteen electrodes. The thirty-two electrodes configuration, however, led to performance below that of the eight electrode configuration. Concerning the relative performance of FDA and BDA it can be seen that BDA always outperformed FDA.

6.4.5 Averaged Waveforms

Detecting the target image from a sequence of EEG trials relies on differences between the waveforms of target and nontarget trials. To visualize these differences the averaged waveforms at elec-

	Disabled				Able-bodied				Average		
	S1	S2	S3	S4	S6	S7	S8	S9	S1-S4	S6-S9	All
FDA-04	6	7	24	13	22	19	44	8	13±9	23±15	18±12
-08	7	13	28	17	22	19	56	13	16±9	28±19	22±15
-16	5	6	28	19	17	22	50	15	15±11	26±16	20±14
-32	7	6	19	15	13	19	39	13	12±7	21±12	16±10
BDA-04	9	7	22	15	26	22	39	17	13±7	26±9	20±10
-08	9	11	25	19	26	22	50	19	16±8	29±14	23±13
-16	8	11	25	22	26	39	56	22	16±8	36±15	26±15
-32	13	11	22	30	34	39	65	17	19±9	39±29	29±18

Table 6.2 — Maximum average bitrate per minute (rounded to integer values). Bitrates were computed from average accuracy curves and are shown for all combinations of classification algorithm and electrode configuration. Mean bitrate and standard deviations were computed for disabled subjects (S1-S4), able-bodied subjects (S6-S9), and all subjects.

	Disabled				Able-bodied				Average		
	S1	S2	S3	S4	S6	S7	S8	S9	S1-S4	S6-S9	All
FDA-04	41	37	63	40	48	56	66	45	45±12	54±10	49±10
-08	42	46	67	56	55	58	78	45	53±11	59±14	56±12
-16	41	39	67	56	58	63	77	49	49±13	62±12	56±13
-32	42	37	62	55	53	57	74	44	49±11	52±13	53±12
BDA-04	43	43	68	43	54	56	69	49	49±13	57±8	53±11
-08	46	53	71	63	60	61	80	51	49±11	63±12	60±11
-16	47	53	75	68	68	71	85	55	61±13	70±13	65±13
-32	57	51	76	70	70	72	87	58	64±12	72±12	68±12

Table 6.3 — Per block accuracy (PBA) in percent for all subjects. Shown are the mean PBA for each subject, the mean and standard deviations for disabled subjects (S1-S4), able-bodied subjects (S6-S9), and all subjects. All numbers were rounded to integer values to increase readability of the table.

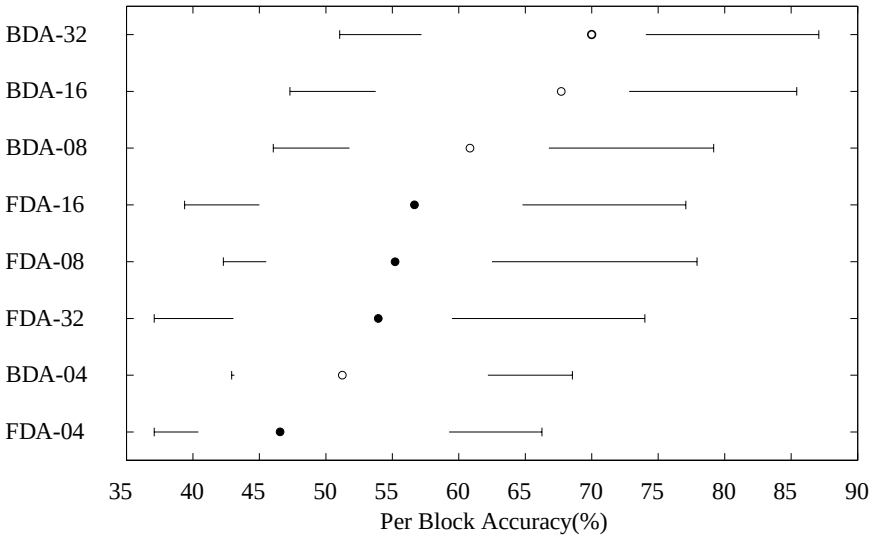

Figure 6.4 — Boxplots of the per block accuracy (PBA). Each boxplot summarizes the data from subjects S1-S4 and subjects S6-S9. The leftmost vertical lines indicate the minimal PBA among subjects and the rightmost vertical lines indicate the maximal PBA among subjects. Circles represent the median PBA among subjects (filled circles were used for FDA, empty circles were used for BDA). White space around the circles indicates the interquartile range of the PBA.

trode Pz are plotted in Fig. 6.5[1]. As expected, disabled subjects and able-bodied subjects show a P300-like peak in the target condition which is not present in the nontarget condition. The latency of the P300 is higher for the disabled subjects (around 500 ms) when compared to the one from able-bodied subjects (around 300 ms). The amplitude at the P300 peak is smaller for the disabled

[1]Electrode Pz was chosen for plotting because it typically shows the largest P300 amplitude.

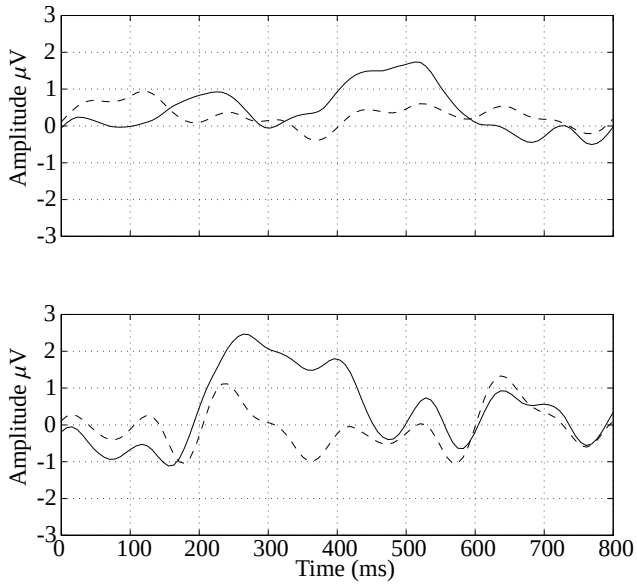

Figure 6.5 — Top: Average waveforms at electrode Pz for disabled subjects (S1-S4). Bottom: Average waveforms at electrode Pz for able-bodied subjects (S6-S9). Shown are the average responses to target stimuli (solid line) and nontarget stimuli (dashed line) from all four sessions. A prestimulus interval of 100 ms was used for baseline correction of single trials.

subjects (around 1.5 μV) than for the able-bodied subjects (around 2 μV).

6.5 Discussion

6.5.1 Differences to Other Studies

Compared to other P300-based BCI systems for disabled users, the classification accuracy and bitrate obtained in the current study are relatively high. In the work of Sellers and Donchin (2006) the best classification accuracy for the able-bodied subjects was on average 85% and the best classification accuracy for the ALS patients was on average 72% (values taken from Table 3 in Sellers and Donchin (2006)). In the present study the best classification accuracy for the able-bodied subjects was on average close to 100% and the best classification accuracy for disabled subjects was on average 100% (see Fig. 6.3). Bitrates in bits/min were not reported in the study of Sellers and Donchin.

In the work of Piccione *et al.* (2006) the definitions for bitrate and classification accuracy are different from those used in this thesis. Therefore a direct comparison with the system of Piccione *et al.* is impossible. However, given the number of stimuli (four) and the ISI (2.5 s) used in the system of Piccione *et al.*, the maximal possible bitrate according to the definition in Equation 4.1 can be computed. This bitrate is 12 bits/min. In the present study the average bitrate obtained with electrode configuration (II) was 15.9 bits/min for the disabled subjects and 29.3 bits/min for the able-bodied subjects.

Due to differences in experimental paradigms and subject populations the classification accu-

racy and bitrate obtained in the two studies described above cannot be compared directly to those obtained in the present study. Nevertheless, several factors that might have caused the differences can be identified. These factors are described below.

- *Number of Choices*
  In the present study a six-choice paradigm was used, whereas in the experiments of Sellers and Donchin and Piccione *et al.* four-choice paradigms were used. As a consequence the target stimulus occurred with a probability of 0.25 in the experiments of Sellers and Donchin and Piccione *et al.*, whereas in the present work it occurred with a probability of 0.16. Smaller target probabilities correspond to higher P300 amplitudes (Duncan-Johnson and Donchin, 1977), thus the P300 in our system might have been easier to detect.

  In general, when designing a P300-based BCI, one has to take into account that disabled subjects might suffer from visual impairments. Systems such as the P300 speller in which users have to focus on a relatively small area of the display might thus not be appropriate for disabled subjects. Reducing the number of choices enlarges the area occupied by one item on the screen and thus facilitates concentration on one item. This might be particularly important for subjects who have little remaining control over their eye-movements. Such subjects might use covert shifts of visual attention (Posner and Petersen, 1990) to control a P300-based BCI, which should be easier when a small number of large items is used.

- *Interstimulus Interval*
  Several factors have to be kept in mind when choosing an ISI for a P300-based BCI system. Regarding classification accuracy, longer ISIs theoretically yield better results. This should be the case because longer ISIs (within some limits) cause larger P300 amplitude. On the other hand, a consequence of long ISIs is a longer overall duration of runs. Disabled subjects might have difficulties to stay concentrated during long runs and thus P300 amplitude and classification accuracy might actually decrease for longer ISIs.

  Regarding bitrate, the factors described above have to be considered together with the fact that for a given classification accuracy higher bitrates are obtained with shorter ISIs. Additionally one has to consider that if the ISI is made too short, subjects with cognitive deficits might have problems to detect all target stimuli and classification accuracy might decrease.

  Given the complex interrelationship of several factors an optimal ISI for P300-based BCIs can only be determined experimentally. Here we have shown that an ISI of 400 ms yields good results. Sellers and Donchin have used an ISI of 1.4 s, and Piccione *et al.* have used an ISI of 2.5 s. The results obtained in their studies seem to indicate that these ISIs are too long.

6.5.2 Visual Evoked Potentials

In the literature on P300-based BCI systems it is almost always assumed that the only factor allowing to discriminate target trials from nontarget trials is the P300 (see Kaper *et al.* (2004) for an exception). However, for systems using visual stimuli this assumption might be too limited. To

understand this let us consider that in the system presented here and in any other visual P300 BCI, users can use one of (at least) two strategies to select an item displayed on the screen. In the first strategy, users gaze at a neutral position on the screen (for example the center of the screen) and use covert shifts of attention to concentrate on the flashes of the desired target item. For this strategy, the assumption that the P300 is the main factor for discrimination of targets from nontargets is probably correct. In the second strategy, users employ eye movements to gaze at the desired item and to foveate this item. For this strategy, it is probable that the visual potentials evoked by the target item are different from those evoked by nontarget items. The target item is at the center of the visual field and influences a relatively large part of visual cortex whereas peripheral nontarget items influence a smaller part of visual cortex. Hence, the visual evoked potentials (VEPs) corresponding to target flashes can be expected to have a larger amplitude than the VEPs corresponding to nontarget flashes. In the second strategy, discrimination of targets from nontargets might thus be based on the P300 *and* on differences in the VEPs.

For the system presented here, the plots of the average waveforms in the target and nontarget conditions (cf. Fig. 6.5) provide evidence that the P300 plays an important role for the classification of targets and nontargets. However, the possibility that the classification accuracy depends partly on the ability to perform eye movements and to focus on an item cannot be excluded. Further research is necessary to elucidate the role of P300 and VEPs in P300-based BCI systems.

6.5.3 Electrode Configurations

The electrode configuration used in a BCI determines the suitability of the system for daily use. Clearly, systems that use only few electrodes take less time for setup and are more user friendly than systems with many electrodes. However, if too few electrodes are used not all features that are necessary for accurate classification can be captured and communication speed decreases.

For P300-based BCI systems different electrode configurations have been described in the literature. Good results have been reported using only three or four midline electrodes (Fz, Cz, Pz, Oz) (Piccione *et al.*, 2006; Sellers and Donchin, 2006; Serby *et al.*, 2005). Krusienski *et al.* (2006) described an eight electrode configuration consisting of the midline electrodes and the four parietal-occipital electrodes PO7, PO8, P3, and P4. Kaper *et al.* (2004) employed a ten electrode configuration consisting of the midline electrodes, the parietal-occipital electrodes PO7, P08, P3, P4 and the central electrodes C3, C4. Thulasidas *et al.* (2006) used a set of 25 central and parietal electrodes.

Here we have tested different electrode configurations, consisting of four, eight, sixteen, and thirty-two electrodes, in combination with the BDA and FDA classification algorithms. The results show that for both algorithms a significant increase in classification accuracy can be obtained by augmenting the set of four midline electrodes with the parietal electrodes P7, P3, P4, and P8. For most of the subjects, inspection of the average waveforms at the parietal electrodes showed that in target trials there was a negative peak with a latency of about 200 ms which was weaker in the nontarget condition. This N200-like component probably is responsible for the increase of classification accuracy when the parietal electrodes are included. Further research is needed to clarify the possible functional significance of this component.

With the BDA algorithm a further increase in classification accuracy could be obtained by using

the configurations consisting of sixteen or thirty-two electrodes. With the FDA algorithm, classification decreased when more than sixteen electrodes were used. This probably happened because the FDA algorithm is unable to deal with training data sets in which the number of features is large compared to the number of training examples.

In summary, regardless of the classification algorithm that is used, the eight electrode configuration represents a good compromise between suitability for daily use and classification accuracy and seems to capture most of the important features for P300 classification.

6.5.4 Machine Learning Algorithms

Many of the characteristics of a BCI system depend critically on the employed machine learning algorithm. Important characteristics that are influenced by the machine learning algorithm are classification accuracy and communication speed, as well as the amount of time and user intervention necessary for setting up a classifier from training data.

A simple and efficient algorithm that has relatively often been used in P300-based and other BCI systems is FDA (Bostanov, 2004; Kaper, 2006; Pfurtscheller and Neuper, 2001). In a comparison of classification techniques (Krusienski *et al.*, 2006) for P300-based BCIs, FDA was among the best methods in terms of classification accuracy and ease of use. However, using FDA becomes impossible when the number of features becomes large, relative to the number of training examples. This is known as the small sample size problem. The small sample size problem occurs because the between-class scatter matrix used in FDA becomes singular when the number of features becomes large. In the present study the solution to this problem was to use the Moore-Penrose pseudoinverse of the between-class scatter matrix (cf. Chapter 5, page 58). This allows to use FDA, even if the number of features is high. However, with this approach the performance deteriorated when the number of electrodes was increased.

In BDA, the small sample size problem, and more generally the problem of overfitting are solved by using regularization. Through a Bayesian analysis, the degree of regularization can be automatically estimated from training data without the need for user intervention or time consuming cross-validation. With the datasets used in this work, the BDA algorithm is superior to FDA in terms of classification accuracy and bitrates, especially if the number of features is large.

In summary, BDA offers good classification accuracy and does not constrain the practical applicability of a BCI system and is thus an interesting alternative to FDA.

6.6 Conclusion

In this chapter an efficient P300-based BCI system for disabled subjects was presented. It was shown that high classification accuracies and bitrates can be obtained for severely disabled subjects. Due to the use of the P300, only a small amount of training was required to achieve good classification accuracy.

Concerning the relative performance of disabled and able-bodied subjects we have seen that the data from able-bodied subjects can be classified with higher accuracy. Nevertheless, by integrating information from many stimulus presentations it was possible to achieve communication

without errors also for the disabled subjects. A comparison between the machine learning algorithms FDA and BDA revealed that BDA clearly outperforms FDA. This was especially the case when high-dimensional feature vectors, resulting from the usage of many electrodes, were employed. Concerning the performance of different electrode configurations we concluded that the eight electrode configuration represents a good compromise between practicality and achievable classification accuracy.

In the next chapter experiments with the sparse Bayesian linear discriminant analysis (SBDA) algorithm, which allows to adapt electrode configurations to specific subjects, will be presented. Moreover, experiments conducted with the adaptive stopping algorithm will be presented.

Experiments with Bayesian Algorithms for EEG Classification

7

7.1 Introduction

In this chapter we describe experiments with Bayesian linear discriminant analysis (BDA) and sparse Bayesian linear discriminant analysis (SBDA), the adaptive stopping algorithm, and different approaches for computing class probabilities from the output of BDA and SBDA. The theory underlying these algorithms and methods is described in Chapter 5.

We start in Section 7.2 with a report about the classification accuracy that can be obtained with SBDA and with a comparison of SBDA and BDA. Furthermore, we report on the electrodes that are selected by SBDA, and compare the automatically selected electrodes to the predefined electrode subsets proposed in Chapter 6. Then, BDA and SBDA are applied to P300 datasets from past BCI competitions. We show that both algorithms lead to classification accuracies that are competitive with the state-of-the-art. In Section 7.3, experiments with the adaptive stopping algorithm and with the algorithms for computing class probabilities are described. The adaptive stopping algorithm dynamically adapts to the level of uncertainty in the signals by varying the amount of data used for taking decisions. We show that the adaptive stopping algorithm allows to obtain higher communication speed than decision schemes in which a fixed amount of data is used. For the computation of class probabilities it is shown that the leave-one-out approach performs slightly better than Van Gestel's method, this comes however at the cost of increased training time. The chapter is summarized in Section 7.4.

7.2 Sparse Bayesian Discriminant Analysis

7.2.1 Results with Proprietary Datasets

Comparison with BDA

Four versions of SBDA were compared with BDA. In the version that we refer to as SBDA-32, the SBDA algorithm was used to select an optimal number of electrodes from all thirty-two electrodes. In the versions that we refer to as SBDA-16, SBDA-08, and SBDA-04, the number of selected electrodes was predetermined to be sixteen, eight, and four, respectively, and SBDA was used to select electrode subsets of that size. The datasets used for the experiments, as well as the preprocessing methods and the cross-validation procedure were the same as those used in Chapter 6. As performance measure we used per block accuracy (PBA), as defined in Chapter 6, on page 82.

The results obtained by running SBDA, together with the results obtained by running BDA, are summarized in Fig. 7.1.

Detailed results for each subject and for different groups of subjects are provided in Table 7.1. As can be seen in Fig. 7.1, SBDA in general outperformed BDA. The largest improvements were obtained when the number of electrodes was small. In particular the improvement obtained by using SBDA-04 instead of BDA-04 was about 8% in the median PBA. The improvement obtained for the configuration consisting of eight electrodes was about 6%. For the configurations consisting

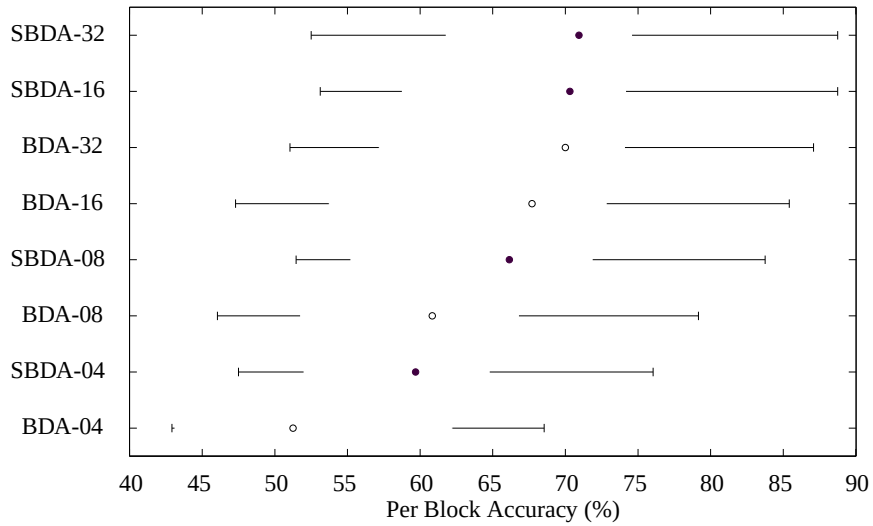

Figure 7.1 — Boxplots of per block accuracy (PBA) for BDA and SBDA. Each boxplot summarizes the data from subjects S1-S4 and subjects S6-S9 (cf. Chapter 6, page 78). The leftmost vertical lines indicate the minimal PBA among subjects. The rightmost vertical lines indicate the maximal PBA among subjects. Circles represent the median PBA among subjects (filled circles were used for SBDA, empty circles were used for BDA). White space around the circles indicates the interquartile range of the PBA.

	Disabled				Able-bodied				Average		
	S1	S2	S3	S4	S6	S7	S8	S9	S1-S4	S6-S9	All
BDA-04	43	43	68	43	54	56	69	49	49±13	57±8	53±11
-08	46	53	71	63	60	61	80	51	58±11	63±12	60±11
-16	47	53	75	68	68	71	85	55	61±13	70±13	65±13
-32	57	51	76	70	70	72	87	58	64±12	72±12	68±12
SBDA-04	52	48	65	57	64	62	76	52	56±8	63±10	60±9
-08	54	51	73	63	70	71	84	57	60±10	70±11	65±11
-16	56	53	74	71	69	74	89	62	64±11	73±11	69±11
-32	62	53	75	72	70	74	89	62	65±10	74±11	69±11

Table 7.1 — Per block accuracy (PBA) in percent for all subjects. Shown are the mean PBA for each subject, the mean and standard deviations for disabled subjects (S1-S4), able-bodied subjects (S6-S9), and all subjects. All numbers were rounded to integer values to improve readability of the table.

of sixteen and thirty-two electrodes, the differences between SBDA and BDA were smaller. Intuitively, this can be explained by assuming that for each subject a small set of electrodes is critical for obtaining good results, while electrodes not in this set interfere only little. For the static configurations consisting of sixteen or thirty-two electrodes, the probability that the critical electrodes are included is relatively high. However, for smaller numbers of electrodes the probability that the important electrodes are included in a predetermined subset becomes smaller. The automatic adaptation of electrode subsets to a given subject thus is important whenever one wants to use only a small number of electrodes.

Looking at Table 7.1, which contains the detailed results, we can make several additional observations. Regarding the improvements in classification accuracy, we can see that for nearly all subjects SBDA yielded better results than BDA. The biggest increases in classification accuracy were 14% and 10% and were obtained for subjects S4, S6, and S7. However, for subject S2 and especially for subject S3, SBDA led in some cases to a decrease of between 1% and 3% in classification accuracy. Possible reasons for the fact that SBDA decreased accuracy for some subjects while it improved accuracy for other subjects will be given during the following discussion of electrode rankings.

Electrode Rankings

To rank electrodes by their importance, we ran SBDA on the whole data from each subject and restricted the number of retained electrodes to one. In other words, we used SBDA to sequentially remove all electrodes from the initial configuration consisting of thirty-two electrodes. Electrodes were then ranked in the order in which they were removed. Electrodes that were removed first received low rankings, while electrodes that were retained even in small electrode subsets received high rankings. The results of this procedure are shown in Table 7.2.

A first insight that can be gleaned from this table is related to the varying performance of SBDA among subjects, which was mentioned above. As can be seen, subjects S4, S6, and S7, for which the biggest increases in performance were obtained, are subjects for which many of the electrodes from

Rank	S1	S2	S3	S4	S6	S7	S8	S9	Avg.
1	Fp2	O1	O2	P7	Cz	O2	Pz	Pz	O2
2	Fp1	P4	P3	FC2	P8	PO4	O2	O2	P7
3	P7	Pz	Fz	Cz	CP2	Pz	P7	O1	O1
4	Fz	P8	O1	O1	O2	P8	Oz	Oz	Pz
5	AF4	P7	Cz	Pz	P7	PO3	FC2	Fp1	Cz
6	F3	P3	P7	CP1	CP1	C3	P3	P8	CP1
7	Cz	AF3	PO3	O2	AF3	C4	PO4	FC2	Oz
8	Oz	Oz	Fp2	CP2	Oz	O1	Fp2	Cz	FC2
9	AF3	Fz	CP1	Fz	Fp2	P3	Cz	PO3	Fp2
10	FC1	CP1	CP2	CP6	F4	FC6	CP6	C4	P8
11	O2	FP1	Oz	CP5	FC2	FC5	FC6	P3	P3
12	FC5	CP2	Pz	P4	Fp1	P7	CP1	C3	Fz
13	PO4	FC2	FC1	Fp2	O1	FC1	P4	CP1	CP2
14	FC6	Fp2	C3	C3	FC5	CP1	P8	CP6	Fp1
15	F4	O2	FC2	AF3	Fz	FC2	C4	AF3	AF3
16	O1	F7	FC5	FC1	C4	F4	O1	P7	C3
17	CP1	Cz	CP5	Fp1	F8	CP5	F4	F4	PO4
18	C3	F8	P8	Oz	FC6	P4	CP2	Fz	P4
19	CP2	AF4	CP6	T7	F7	CP2	CP5	F8	CP6
20	P3	FC6	PO4	T8	F3	AF3	Fz	T8	C4
21	P4	C3	F7	F8	PO4	Fp2	F8	T7	PO3
22	F7	F4	F3	P3	T7	F7	Fp1	FC6	FC6
23	F8	CP6	Fp1	C4	T8	F3	AF4	FC5	F4
24	**CP6**	**FC1**	F8	F7	CP6	AF4	F3	FC1	FC5
25	Pz	PO3	C4	PO4	C3	T7	PO3	AF4	FC1
26	T8	CP5	T8	FC5	**Pz**	**T8**	T8	**F7**	F8
27	T7	FC5	**T7**	**F3**	PO3	CP6	**T7**	P4	F7
28	FC2	PO4	P4	P8	P4	Oz	C3	Fp2	F3
29	P8	F3	AF3	AF4	AF4	Fz	FC1	PO4	CP5
30	PO3	C4	F4	PO3	P3	Fp1	FC5	CP2	AF4
31	C4	T7	FC6	F4	FC1	Cz	AF3	F3	T7
32	CP5	T8	AF4	FC6	CP5	F8	F7	CP5	T8

Table 7.2 — Electrodes as ranked by the SBDA algorithm. Each column contains a ranking of electrodes from most important to least important for one subject. The last column contains an average ranking computed from the rankings of all subjects. Electrode names printed in bold font indicate the size of the electrode subsets selected by SBDA. To show that the results are physiologically plausible, electrodes at which large P300 amplitudes are expected are highlighted (these are the electrodes from the eight electrode configuration proposed in Chapter 6). On average 69% of the highlighted electrodes are ranked among the first sixteen electrodes.

the static electrode configurations received low ranks. Hence, big increases in performance when using SBDA instead of BDA might be expected for subjects who do not match the static electrode configurations. Small differences in performance might be expected for subjects who match well the static electrode configurations. Furthermore, we can see from Table 7.2 that the electrodes

selected by SBDA correspond roughly to those at which we expect large P300 amplitudes. The results returned by SBDA are thus physiologically plausible. In addition to electrodes related to the P300, also the occipital electrodes O1 and O2 received consistently high rankings. This indicates that besides the P300 also visual evoked potentials (VEPs) are important to classify EEG trials as belonging to target or nontarget stimuli.

7.2.2 Results with BCI Competition Datasets

To test how BDA and SBDA compare to state-of-the-art classification algorithms, we performed experiments with the P300 datasets from the BCI competitions 2003 and 2004 (Blankertz *et al.*, 2004, 2006a). All the competition datasets consist of a training set and a test set. Training set as well as test set contain data recorded with the P300 speller paradigm (cf. Chapter 4, page 45). The goal in the competition was to train a classifier on the training set and to predict the symbols in the test set. We simulated the competition conditions and used only the training set to determine classifier parameters and hyperparameters.

Preprocessing

Before learning classifiers and before performing classification, the data were preprocessed with methods that were similar or equal to those used by the competition winners. This was done because our goal was to do a fair comparison of BDA, SBDA, and other state-of-the-art machine learning techniques for P300 datasets. In other words the preprocessing methods described in Chapter 6 were not used because it would not have been possible to differentiate between the contribution of preprocessing to classification performance and the contribution of machine learning to classification performance.

For the 2003 dataset an approach similar to the one described by Kaper *et al.* (2004) was used. The following preprocessing steps were used.

1. *Filtering*
   The data were bandpass filtered between 0.5 Hz and 30 Hz with a 6-th order forward-backward Butterworth filter.

2. *Scaling*
   The data were scaled to the interval $[-1, 1]$. As already discussed in Chapter 6 on page 79, the scaling was performed for the whole training and validation set and not on a trial by trial basis.

3. *Single Trial Extraction*
   Single trials of length 600 ms, starting at stimulus onset, were extracted from the data.

4. *Downsampling*
   The data were downsampled from 240 Hz sampling rate to 60 Hz sampling rate by selecting each 4th sample. This is the only done difference to the preprocessing proposed by Kaper *et al.*. The downsampling was performed to reduce the dimension of feature vectors and thus to make learning of classifiers computationally more efficient.

5. *Electrode Selection*
   The electrode subset described by Kaper *et al.* was used. The subset consists of the ten electrodes Fz, Cz, Pz, Oz, C3, C4, P3, P4, PO7, and PO8. Additionally the electrode configurations from Chapter 6 and electrode selection with SBDA were tested.

6. *Feature Vector Construction*
   The samples from the selected electrodes were concatenated into feature vectors. The dimensionality of the feature vectors was 360 for the electrode configuration proposed by Kaper *et al.* and varied accordingly for other electrode configurations.

For the 2004 dataset the approach described by Rakotomamonjy *et al.* (2005) has been used. More specifically, the preprocessed data were downloaded from (Rakotomamonjy, 2007) and fed into BDA and SBDA. Hence, the preprocessing was exactly identical to the one employed by Rakotomamonjy *et al.*. The preprocessing steps used in the method of Rakotomamonjy *et al.* were as follows.

1. *Single Trial Extraction*
   Single trials of length 667 ms, starting at stimulus onset, were extracted from the data.

2. *Filtering*
   The single trials were filtered with an 8-th order Chebyshev type I filter with cutoff frequencies of 0.1 Hz and 20 Hz.

3. *Decimation*
   The data were downsampled to a sampling rate of 20 Hz. This was done with the MATLAB function *decimate* which involves an additional low pass filtering step.

4. *Electrode Selection*
   The electrode subsets described in Chapter 6 were used. Additionally electrode selection with SBDA was tested.

5. *Feature Vector Construction*
   The samples from the selected electrodes were concatenated into feature vectors. The dimensionality of the feature vectors was 896 for the full electrode configuration and varied accordingly for other electrode configurations.

Results

The results obtained on the competition data, together with the results of the competition winners are summarized in Table 7.3. For the data from the BCI competition 2003, perfect classification after five and fifteen blocks was obtained with the BDA-64, BDA-32, BDA-16, and for the electrode configuration used by Kaper *et al.* (2004), i.e. for BDA-10. For the data from the BCI competition 2004, 74.5% classification accuracy after five blocks and 97.5% after fifteen blocks were obtained with SBDA-64. Very good results were also obtained with SBDA-32. This shows that in terms of classification accuracy BDA and SBDA are competitive with the algorithms of the competition winners.

	2003		2004 A		2004 B	
	5	15	5	15	5	15
BDA-04	71	84	30	62	44	72
-08	97	100	41	81	51	83
-10	100	100	-	-	-	-
-16	100	100	46	87	64	89
-32	100	100	56	98	78	94
-64	100	100	66	98	80	97
SBDA-04	74	87	37	79	70	87
-08	97	100	43	90	75	96
-16	97	100	52	94	79	96
-32	97	100	66	99	80	97
-64	97	100	67	98	82	97
Winner 2003	100	100	-	-	-	-
Winner 2004	-	-	73.5	96.5	73.5	96.5

Table 7.3 — Classification results for the P300 datasets from BCI competitions 2003 and 2004. Shown is the percentage of correctly predicted symbols on the test set after five blocks of stimulus presentations and after fifteen blocks of stimulus presentations. The last two rows contain the results of the competition winners according to Blankertz *et al.* (2004, 2006a). Note that for the BCI competition 2004 only averaged results for datasets A and B are available. The average results are displayed for dataset A as well for dataset B.

At the same time the computational complexity of BDA and SBDA is quite low. Learning a classifier from the competition data with an unoptimized MATLAB implementation of BDA took on average four minutes on a PC with a 3.4 GHz processor and 1 GB of RAM[1]. Computing parameters of the univariate Gaussians for estimation of class probabilities, with the leave-one-out procedure described in Chapter 5, took ten minutes. In total, the setup of a classifier that gives probabilistic outputs thus took fourteen minutes. The amount of time needed for setting up classifiers with an unoptimized version of SBDA-64 was only slightly higher than that needed for BDA.

To get an estimate of the computational complexity of support vector machine (SVM) based solutions, proposed by the winners of the BCI competitions 2003 (Kaper *et al.*, 2004) and 2004 (Rakotomamonjy *et al.*, 2005), we used LIBSVM. LIBSVM is an optimized, state-of-the-art implementation of the SVM (Chang and Lin, 2001). Linear ν-SVMs were trained on the competition training sets by performing a ten-fold cross-validation with ten different values for ν. This procedure took on average 4.5 hours on the PC used for testing BDA and SBDA. The time needed to train an SVM-based classifier is thus much higher than the time needed for training BDA. The reason for this is that training an SVM is computationally complex. In addition, regularization parameters and kernel parameters have to be estimated via cross-validation.

A further advantage of BDA and SBDA when compared to the SVM is that these algorithms allow for simple and fast estimation of class probabilities (via the procedures presented in Chapter 5). Experiments with the adaptive stopping algorithm, which uses these class probabilities, are described next.

[1]The time required for setting up a classifier varied according to the different sizes of the competition training sets. All runtimes reported here are averages.

7.3 Adaptive Stopping

Up to now all experiments were based on a scheme in which an a priori fixed number of stimuli is presented before classification of the recorded EEG signals is attempted. In this scheme, a small number of stimuli corresponded to low classification accuracy but also to fast decisions. A large number of stimuli corresponded to high classification accuracy at the cost of slower decisions. Now, instead of fixing the number of stimulus presentations a priori, we use thresholds on the probability of misclassification to automatically adapt the number of stimulus presentations to the uncertainty in the recorded signals. More specifically, in the following we report results that were obtained with the algorithm for adaptive stopping and with probabilities computed with the leave-one out approach (cf. Chapter 5, page 66)[1]. A comparison of the probabilities computed with the leave-one-out approach and with Van Gestel's method can be found at the end of this section.

7.3.1 Results with Proprietary Datasets

Classification Accuracy and Bitrates

To get a first impression of the performance the adaptive stopping algorithm offers, we used BDA-08 in an experiment with the datasets, preprocessing methods, and cross-validation procedure described in Chapter 6. In this experiment, a set of six probabilities $p_1 \dots p_6$ was computed after each block of stimulus presentations from all data recorded so far. In other words, after the first block of stimulus presentations, the probabilities were computed from six EEG trials, after the second block of stimulus presentations the probabilities were computed from twelve EEG trials, and so forth. The $p_1 \dots p_6$ corresponded to the probability of occurrence of six mutually exclusive events. The event linked to p_1 was "The subject concentrated on stimulus 1" and the events corresponding to $p_2 \dots p_6$ were defined accordingly. After each block of stimulus presentations the maximum of $p_1 \dots p_6$ was compared to a preset threshold. If the maximum p_m was larger than the threshold the system decided that the user was concentrating on stimulus m. If the maximum was lower than the threshold the next block of stimulus presentations was evaluated.

Decisions in which m corresponded to the ground-truth target stimulus were counted as correct. Decisions in which m was different from the ground-truth were counted as wrong. The accuracies, i.e. the percentage of correct decisions, for the thresholds 0.15, 0.4, 0.65, 0.9, 0.95, and 0.99 are plotted for each subject in Fig. 7.2[2]. The accuracies are plotted at the stopping times corresponding to the thresholds. The stopping time for a threshold is the average time that was needed until the maximum of $p_1 \dots p_6$ first exceeded the threshold. For comparison purposes also the accuracy and bitrate obtained without adaptive stopping are plotted in Fig. 7.2.

Adaptive stopping led to improved classification accuracies and bitrates for almost all subjects. In terms of classification accuracy the largest improvements were around 15% (subject S4). In terms of maximal bitrate the largest improvements were around 5 bits/min (subjects S2, S4, S7).

[1] The leave-one-out approach was chosen because it yielded slightly better results than Van Gestel's method.

[2] Sometimes, the large thresholds were not exceeded even after the maximum of 20 stimulus blocks. In this case decisions were taken based on probabilities computed from all 20 stimulus blocks and the stopping time was adjusted accordingly.

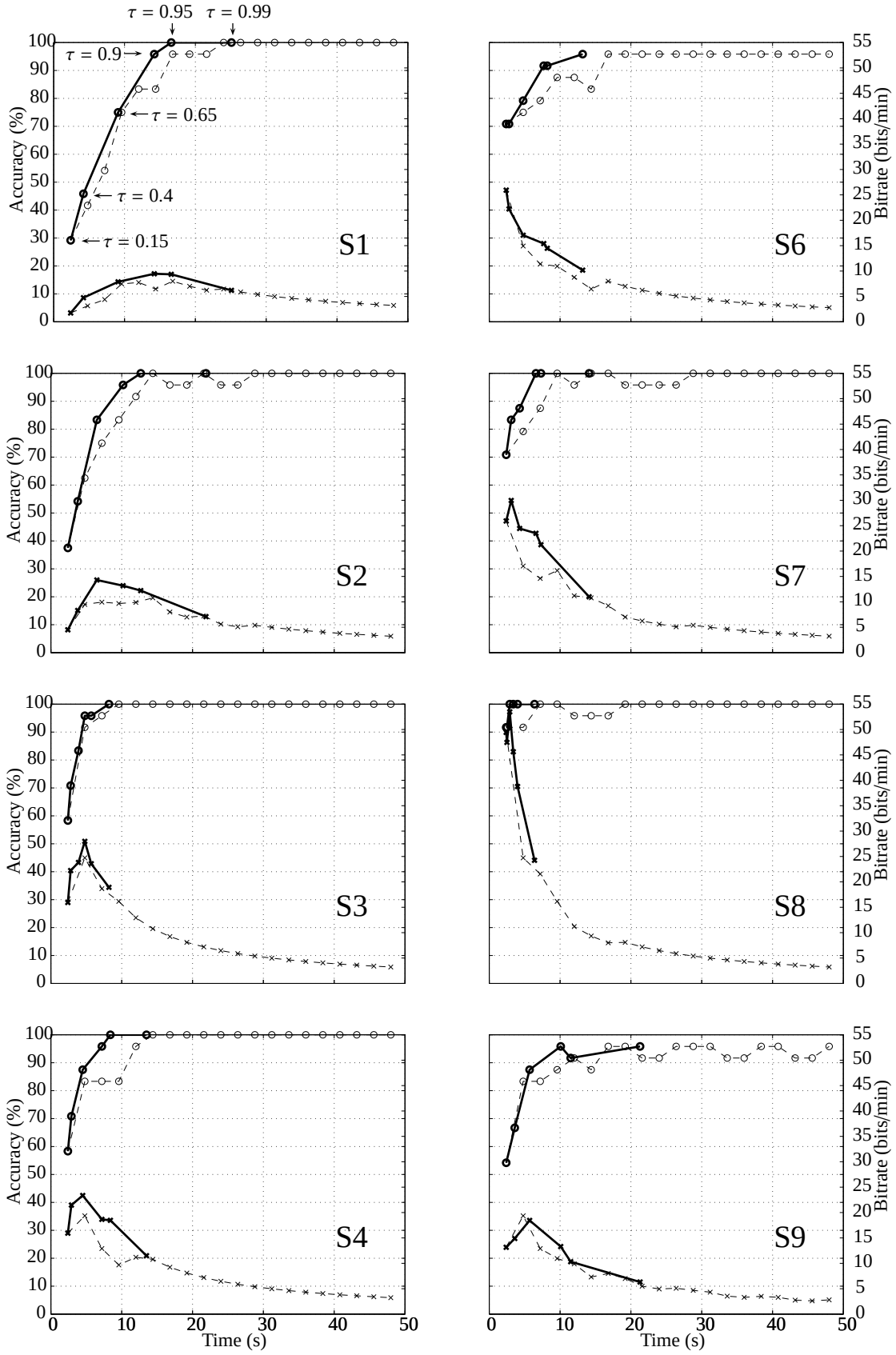

Figure 7.2 — Comparison of classification accuracy and bitrate obtainable with BDA-08 with and without adaptive stopping. Thick, solid lines represent results obtained with adaptive stopping. Thin, dashed lines represent results obtained without adaptive stopping. The panels show the classification accuracy, averaged over four sessions (circles) and the corresponding bitrate (crosses), for disabled subjects (S1-S4) and able-bodied subjects (S6-S9).

Regarding the relation between threshold and resulting classification accuracy, one can see that classification accuracy was almost always larger than the threshold. In other words, for a threshold equal to 0.15 the classification accuracy was bigger than 15%, for a threshold equal to 0.4 the classification accuracy was bigger than 40%, and so forth. This is a desirable behavior as it allows to choose thresholds based on the percentage of errors one is ready to accept. Exceptions to this behavior can be found in the performance curves for subjects S6, and S9. For these subjects, classification accuracy resulting from thresholds equal to 0.95 and 0.99 is lower than it should be. For subject S6, this can be explained by the fact that the training and test data contain some mislabeled trials (cf. Chapter 6, page 83). For subject S9, the cause for the lower than predicted classification accuracy is at present unknown.

Calibration and Refinement

From the relation between thresholds and resulting classification accuracy it seems that $p_1 \dots p_6$ computed with the leave-one-out approach are *calibrated*, i.e. it seems that $p_1 \dots p_6$ are realistic estimates of the probability of occurrence of the corresponding events. However, this cannot be checked thoroughly based on the results in Fig. 7.2 alone. To check how well the probability estimates were calibrated, we applied the following reasoning: If an event is predicted on N occasions with a probability of p it should occur about pN times. Hence, for all blocks of stimulus presentations tested during cross-validation, the probabilities $p_1 \dots p_6$ were computed and sorted into bins. Then, for each bin the number of events that actually occurred was computed. The resulting plot is shown in Fig. 7.3. As can be seen from this plot, the predicted probabilities coincide relatively well with the observed (true) probabilities. A possible reason for the small deviations between true and predicted probabilities is that some of the model assumptions are wrong. In particular, the assumptions of Gaussian predictive distributions and independence of trials should be checked in order to further improve the probability estimates.

The calibratedness of $p_1 \dots p_6$ is an important feature of the adaptive stopping algorithm, as it allows for an intuitive adjustment of the threshold values. In addition,

calibrated probabilities are important whenever probabilistic estimates from several systems or applications have to be combined. In the BCI context this would for example be the case if one wanted to build a spelling system in which different probabilities for different symbols are taken into account.

Besides being calibrated, probabilistic classifier outputs and probability estimates derived from these outputs should also be *refined*. Speaking abstractly, this means that probability estimates for a certain event should allow to discriminate between occasions on which the event occurs and other occasions. Translated to the BCI scenario considered here, this means that the probability estimates $p_1 \dots p_6$ should allow to decide on which stimulus the user is concentrating. The difference between refinement and calibration can be understood by noting that perfect calibration could be easily achieved by predicting $p_i = \frac{1}{6}$ for $i \in \{1 \dots 6\}$. More generally, perfect calibration can always be achieved by predicting the long run probability of an event. Such probability estimates are however not useful, as they do not allow to discriminate between occasions on which the event in question occurs and other occasions.

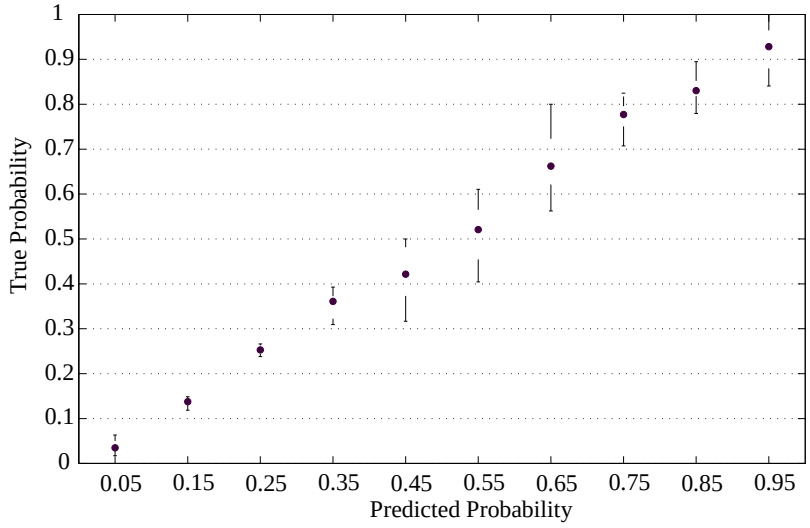

Figure 7.3 — Boxplots of true probabilities versus predicted probabilities (the leave-one-out approach was used to compute class probabilities). Each boxplot summarizes the data from subjects S1-S4 and subjects S6-S9. The uppermost horizontal lines indicate the maximal true probability among subjects. The lowermost horizontal lines indicate the minimal true probability among subjects. Circles represent the median true probability. White space around the circles indicates the interquartile range of the true probabilities.

A tool that allows to measure calibration as well as refinement is the so-called Brier score (Brier, 1950; DeGroot and Fienberg, 1983). The Brier score has been routinely used for evaluation of probabilistic forecast in economics and meteorology and has more recently also been introduced in machine learning (Cohen and Goldszmidt, 2004). The Brier score P is computed as follows:

$$P = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{E} (p_{ij} - e_{ij})^2. \tag{7.1}$$

Here N is the number of observations, E is the number of events, and p_{ij} is the predicted probability that event j occurs in observation i. The e_{ij} indicate if events actually occur, i.e. $e_{ij} = 1$ if event j occurs in observation i and $e_{ij} = 0$ if event j does not occur in observation i. In the BCI scenario considered here, N is equal to the number of stimulus blocks and E is equal to the number of stimuli, i.e. E is equal to six. The average Brier scores for all subjects and for all classifier configurations are shown in Table 7.4. The Brier scores follow closely the behavior that was observed when testing the PBA (cf. Table 7.1). In particular, classifiers using a large number of electrodes achieved better Brier scores than classifiers using a small number of electrodes. Furthermore, the scores achieved with SBDA were in general better than those achieved with BDA. This means that among the algorithms we tested, the algorithms that were preferable for classification were also preferable for producing probabilistic forecasts. This is not self-evident, as the tasks of discrimination and assigning class probabilities are different tasks.

	Disabled				Able-bodied				Average		
	S1	S2	S3	S4	S6	S7	S8	S9	S1-S4	S6-S9	All
BDA-04	70	72	46	71	63	59	43	67	65±12	58±10	61±11
-08	67	62	39	50	54	51	28	64	55±13	49±15	52±13
-16	65	62	35	46	47	40	21	58	52±14	42±15	47±15
-32	57	62	33	41	44	40	19	56	48±14	39±15	44±14
SBDA-04	63	68	45	55	56	50	35	62	58±10	51±12	54±11
-08	59	62	38	48	47	41	24	57	52±11	42±14	47±13
-16	57	61	36	41	43	37	18	54	49±12	38±15	43±14
-32	57	61	34	39	43	36	17	53	48±13	37±15	43±14

Table 7.4 — Brier scores for all subjects. Shown are the mean score for each subject, the mean score and standard deviation for disabled subjects (S1-S4), able-bodied subjects (S6-S9), and all subjects. All values were multiplied by 100 and rounded to integer values to increase readability of the table. The best possible score is 0 and the worst possible score is 200.

Comparison of Van Gestel's Method and the Leave-One-Out Approach

Having discussed results that can be achieved with the adaptive stopping algorithm and having introduced the Brier score as a tool for evaluating estimates of probabilities, we now present a comparison of the quality of the probabilities computed with Van Gestel's method and the leave-one-out approach. For this comparison we used class probabilities instead of the probabilities for different stimuli. This was done in order to avoid masking of differences between the two approaches by the combination scheme for class probabilities. The Brier scores obtained with the two methods are shown in Fig. 7.4. As one can see the Brier scores for the class probabilities are better than the scores in Table 7.4, which can intuitively be explained by the fact that computing probabilities for two different events is a much simpler task than computing probabilities for six different events. Moreover, it can be seen from Fig. 7.4 that the leave-one-out approach yielded slightly better scores than Van Gestel's method, especially so for subjects and classification methods with very good Brier scores.

Given that the difference between the two methods is only small, computational complexity is the most important factor to choose between the two methods. The leave-one-out approach adds complexity to the training phase of classifiers but is very efficient for computing class probabilities for new feature vectors. Hence, it should be chosen for applications in which an increased training time can be afforded and for which it is important to compute class probabilities for test examples very fast. Van Gestel's method does add only little complexity to the training phase but is less efficient for the computation of class probabilities. Hence, it can be used for applications in which the training time should be as short as possible and in which slightly slower computation of class probabilities for test examples can be accepted.

7.3.2 Results with BCI Competition Datasets

We now turn our attention to experiments performed with the BCI competition datasets and adaptive stopping. The goal in these experiments was to test if the good results achieved with the SBDA

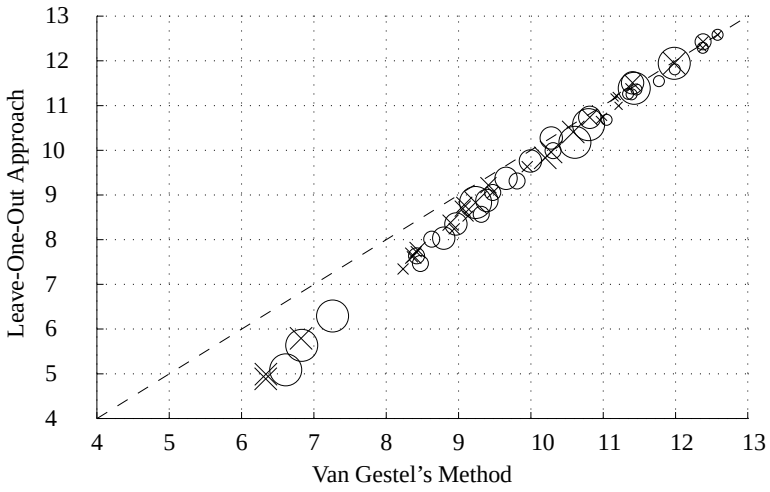

Figure 7.4 — Comparison of Brier scores obtained with Van Gestel's method and the leave one-out approach (cf. Chapter 5, page 66). The leave-one-out approach performs slightly better than Van Gestel's method. Big circles correspond to scores for BDA-32, intermediate circles correspond to scores for BDA-16 and BDA-08, small circles correspond to scores for BDA-04. Crosses correspond to scores obtained with SBDA, size of the crosses indicates the number of channels as for BDA. All scores were multiplied by 100, so the best possible score is 0 and the worst possible score is 200.

algorithm on the competition datasets (cf. Table 7.3) could be pushed even further by using adaptive stopping. We used thresholds equal to 0.02, 0.15, 0.4, 0.65, 0.9, 0.95, and 0.99. The results of these experiments are shown in Fig. 7.5. As can be seen, it was not possible to improve the maximal classification accuracy with the adaptive stopping algorithm. However, the maximal classification accuracy was obtained earlier than by the competition winners. For the 2003 dataset the maximal accuracy was obtained on average after 7.5 s, whereas the competition winners needed 10.5 s. For the 2004 datasets the maximal accuracy was obtained after 25 s, instead of 31.5 s for the competition winners. Using SBDA-64 in conjunction with adaptive stopping thus led to a slight improvement over the results obtained by the competition winners.

7.4 Conclusion

In this chapter we have reported the results of experiments conducted with SBDA and the adaptive stopping algorithm. Concerning SBDA, one of the main results was that electrode selection improves classification accuracy over the BDA algorithm, which does not perform electrode selection. Moreover, the electrode subsets selected by SBDA were physiologically plausible, i.e. the selected electrodes coincided mostly with electrodes at which large P300 amplitudes were expected. Experiments were also conducted with publicly available datasets from past BCI competitions. These experiments showed that SBDA and also BDA allow to achieve results that are in terms of clas-

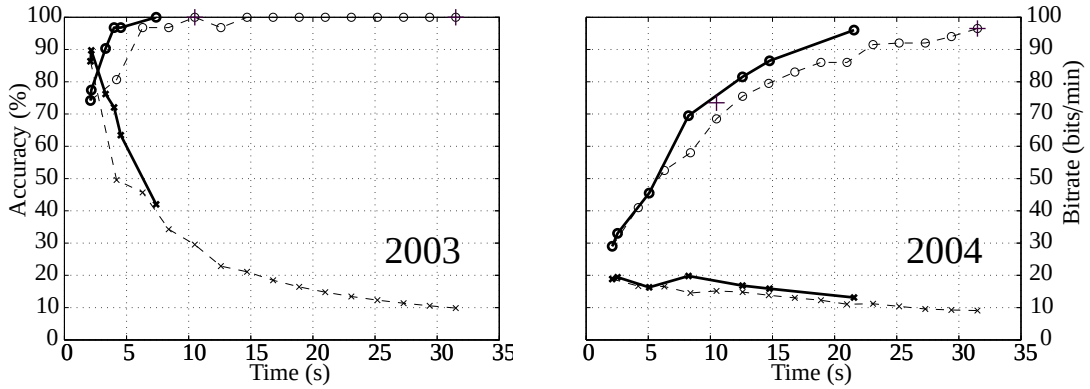

Figure 7.5 — Classification accuracy and bitrate obtained on the BCI competition datasets 2003 and 2004 with SBDA-64. Shown are the accuracy and bitrate obtained with adaptive stopping (thick solid lines) and without adaptive stopping (thin dashed lines). The results of the competition winners are indicated by crosses.

sification accuracy as good as those of the competition winners. In addition, the computational complexity of learning classifiers with SBDA and BDA is significantly lower than that of the SVM-based solutions proposed by the competition winners.

Experiments conducted with the adaptive stopping algorithm showed that automatically adapting the amount of data is preferable to decision schemes in which a fixed amount of data is used. In particular, in the setting of the BCI presented in Chapter 6, adaptive stopping allowed to improve the communication speed while maintaining the classification accuracy. An investigation of the class probabilities used in the adaptive stopping algorithm showed that these probabilities are approximately calibrated. This is important for selecting stopping thresholds in the adaptive stopping algorithm and for interfacing with other probabilistic applications or systems. The comparison of the leave-one-out approach and Van Gestel's method for computing class probabilities showed that the leave-one-out approach yields slightly more precise class probabilities. This comes however at the cost of increased computational complexity during the training phase.

Using adaptive stopping with the BCI competition datasets showed that adaptive stopping allows to improve upon the results of the competition winners. In particular, adaptive stopping permitted to reduce the number of stimuli that are necessary for obtaining the maximal classification accuracy.

8 Conclusion

8.1 Summary

In this thesis two closely related aspects of research on BCI systems were discussed. These aspects were the development of machine learning algorithms suitable for application in BCIs and the development and analysis of BCIs for disabled subjects. In the following, conclusions that can be drawn from this research will be summarized.

Machine Learning Algorithms

In many publications on machine learning for BCI, the only performance measure that is employed is the accuracy with which electroencephalogram (EEG) trials, or data from other sensors, can be classified. The viewpoint taken in this thesis is that besides classification accuracy also other properties are important for characterizing good algorithms. These properties are related to the applicability of algorithms in systems suitable for daily use by disabled users.

A property that is important when adapting a system to a new user, is that classifiers can be learned quickly, robustly, and without intervention from expert users. The Bayesian linear discriminant analysis (BDA) algorithm presented in this thesis possesses this property. In BDA, hyperparameters are estimated quickly and automatically with a type-II maximum-likelihood formalism. An extension of the algorithm allows to adapt electrode configurations to the peculiarities of a given user. Adapting electrode configurations reduces the number of electrodes necessary for successful operation of a BCI. Moreover, it was shown that using user-specific electrode configurations increases classification accuracy as compared to using static, preselected electrode configurations. To test the algorithms proposed in this thesis, P300 data recorded from disabled and able-bodied subjects and data from past BCI competitions were used. The experiments with data from the BCI competitions demonstrated that the proposed algorithms are competitive with state-of-the-art algo-

rithms for EEG classification. More specifically, it was shown that the classification accuracy of the algorithms proposed in this thesis is equivalent to that of state-of-the-art solutions using the support vector machine (SVM). In addition, the time needed for setting up classifiers is significantly lower than that needed by solutions using the SVM.

When applying a classifier to new data it is important that classification can be done quickly and without the need for computationally complex algorithms. This is ensured for the algorithms proposed in this thesis because everything is linear. Hence, classifier outputs can be computed quickly. Equally important is the form the classifier outputs take. We proposed to use probabilistic outputs computed with the help of a leave-one-out scheme. It was shown how in the context of P300-based BCIs probabilistic outputs can be used to adapt the amount of data such that a preset, approximate bound on the probability of misclassifications is not exceeded. Concerning the evaluation of machine learning algorithms with probabilistic outputs, we proposed to use Brier scores. Brier scores measure calibration and refinement of probability estimates, and have been used with success to evaluate algorithms and methods in areas other than BCI, such as for example economical or meteorological forecasting.

Brain-Computer Interfaces for Disabled Subjects

To test the algorithms proposed in this thesis, data from disabled and able-bodied subjects was recorded. Validating algorithms with data from disabled subjects is crucial, simply because disabled subjects are the target population for BCI systems. Recording data also from able-bodied subjects allowed to perform comparisons between disabled and able-bodied subjects in terms of the recorded signals themselves and in terms of achievable BCI performance. Maybe most importantly, interaction with disabled persons is crucial to get a feeling for factors that make BCIs suitable or unsuitable for disabled subjects.

The BCI used for recording data was based on the well-known P300 paradigm. The main difference between the proposed system and other P300-based systems, such as for example the P300 speller, is that the number of items on the screen is smaller and that the interstimulus interval (ISI) is longer. These modifications were motivated by the hope to simplify usage of the system for disabled persons with cognitive deficits. Compared to other P300-based systems for disabled subjects, the classification accuracy achieved with the system presented in this thesis is significantly higher.

8.2 Perspectives

8.2.1 Short Term Perspectives

Machine Learning Algorithms

Based on the research in this thesis, extensions in the following areas immediately come into mind:

- *Filtering Methods*
  For the filtering of the EEG data a forward-backward scheme using Butterworth filters was employed that is being applied in one step to the integrality of the ingoing signal. This choice

was made for reasons of simplicity, however in order to enable realtime classification in online BCI systems, the filter should be realized either by using a windowed implementation of the forward-backward scheme (Djokic *et al.*, 1998; Kurosu *et al.*, 2003), or by using a traditional causal filtering scheme, for instance based on finite impulse response (FIR) filters.

- *Handling of Outliers*
  To deal with outliers, a windsorizing approach was used in this thesis. The advantages of this approach are that it is conceptually and computationally simple. Clear disadvantages are however, that the proportion of outliers has to be specified manually and that it is assumed that all electrodes have the same proportion of outliers. Moreover, it would be desirable to have a method enabling adaptation to changing proportions of outliers. A method that seems promising for dealing with such problems is to model EEG samples as a mixture of two or more univariate Gaussians with zero mean and different variances. Gaussians with small variance would then account for normal samples, whereas Gaussians with large variance would account for outliers. Parameters of such a model can be learned and adapted with the expectation-maximization (EM) algorithm (cf. Aitkin and Wilson (1980)).

- *Electrode Selection*
  Concerning electrode selection, improvements may be possible in the algorithm for selecting a predetermined number of electrodes. The approach proposed in Chapter 5 was to alternate automatic relevance determination (ARD) and backward selection until the desired number of electrodes is attained. A clear drawback of this procedure is that electrodes cannot reenter the selected subset once they have been removed. This is problematic because already removed electrodes might become important again as other electrodes are removed. A possible solution to this problem, which is however not explored in this thesis, would be to use combinations of ARD and more sophisticated procedures for feature selection, such as for example floating search methods (Pudil *et al.*, 1994). Another possible solution that might be worth investigating is to use a parameterized hyperprior for the α_i. In particular a hyperprior of the following form might be of interest (cf. (Schmolck and Everson, 2007)):

$$p(\boldsymbol{\alpha}|c,\beta) \propto \exp\left(-c\sum_{i=1}^{N_e}(1+\frac{1}{\beta}\alpha_i)^{-1}\right). \tag{8.1}$$

  Here the notation from Chapter 5 has been used. With such a hyperprior the number of selected electrodes can be indirectly controlled by changing the sparsity constant c.

- *Loss Functions*
  In this thesis, a Gaussian likelihood function has been used for learning classifiers from training data. Using a Gaussian likelihood is equivalent to using a squared error loss function. In other words, classification functions were learned that map training examples as close as possible to their class labels. The advantage of using the Gaussian likelihood is that in combination with a Gaussian prior, closed form expressions exist for the posterior. The only complex operation necessary for computing the posterior is the inversion of the correlation matrix of the feature vectors.

An alternative loss function that is often proposed for classification problems in the machine learning literature is the logistic loss (cf. Bishop (2006)). The logistic loss might be more suitable for classification problems than the squared error which is rather used in the case of regression problems. Moreover, the logistic loss might be advantageous because it is less sensitive to outliers than the squared error loss. A disadvantage of the logistic loss is however, that no closed form expressions exist for the posterior distribution. Hence, optimization algorithms have to be used to compute the mode of the posterior and in a full Bayesian setting additionally approximation schemes have to be used to approximate the posterior. This makes algorithms for logistic regression computationally and conceptually relatively complex. It remains to be checked if the additional complexity is payed off by improved classification accuracy or other advantages.

Brain-Computer Interfaces for Disabled Subjects

Concerning the BCI for disabled users presented in this thesis, the following extensions might be of interest.

- *Development of Dialog Systems*
  The BCI for disabled users presented in this thesis allows only for extremely simple interactions with the environment. For example the system could be used to switch on/off a particular set of devices. To overcome this restriction it would be interesting to develop a dialog-system allowing a disabled user to perform everyday tasks and basic communication with other persons. Dialog and communication systems for disabled users are a research area on its own and it seems that using results from this area might be a fruitful approach to build better BCIs. An example for a dialog- and environment-control system for disabled users that might be adapted to a BCI environment is the ScriptTalk system described by Dye *et al.* (1998). In the ScriptTalk system a set of useful scenarios for disabled persons (e.g. a visit to the doctor, calling help via the telephone, etc.) is predefined. Each scenario has a corresponding script which allows to select from a sequence of communication steps that are typically performed in that scenario. Scenarios and communication options can be chosen via a graphical user interface.

  Two techniques are used in the ScriptTalk system to speed up communication. Firstly the communication options are limited to a prescribed set of reasonable size and secondly words and sentences are predicted from the structure of communication scenarios and from past communication. To transfer these techniques to a BCI for disabled persons, an algorithmic framework should be developed in which a dialog-based communication can be performed and which allows prediction of items that will be communicated next.

- *Extended Testing*
  The system presented in this thesis was tested with five disabled subjects, out of which four achieved good results. Clearly, more extended tests are necessary to build an optimal BCI for disabled users. Acquisition of a large database with EEG records from disabled persons

would allow to precisely define which cognitive abilities are necessary to control a BCI and which cognitive disabilities might limit the use of BCIs.

8.2.2 Longer Term Perspectives

Using a BCI Without Training

In almost all current BCI systems, subjects first have to go through a training phase, in which they concentrate on prescribed mental tasks or prescribed stimuli. After the training phase a classifier is learned and used to classify new, unseen data. A drawback of this setup is that for many disabled users a long training phase is an insurmountable obstacle due to cognitive impairments and concentration problems. Another problem is caused by the fact that patterns of cerebral activity are constantly changing, and hence new training sessions have to be performed periodically to adapt classification rules.

To overcome these problems, we propose to develop learning algorithms, with which subjects can immediately start using a BCI, without training. The basic idea to achieve this goal is to profit from data that was recorded from other subjects while using the same BCI system. To build a classifier from a large database of EEG records from different subjects, a mixture of experts model could be used. Roughly speaking, training a mixture of experts model corresponds to a clustering of data and to simultaneously learning classifiers for each of the clusters. After training, test instances are first assigned to one of the clusters by a so-called gating function, and then classified by the expert responsible for the cluster (Jordan and Jacobs, 1994). In the BCI context it can be hoped that the learning stage detects subgroups of subjects with highly similar EEG signals and that new subjects are automatically assigned to the correct subgroup during application of the mixture of experts of model.

Asynchronous P300 BCI

One significant limitation of the P300-based BCI presented in this thesis and of many other BCI systems is that they only work in synchronous mode. This means that users can only communicate via the BCI at time instants predetermined by the system. A possible solution to this problem is to develop asynchronous BCI systems. Asynchronous BCI systems can detect autonomously that a user is trying to communicate via the BCI and hence allow for more realistic application scenarios than synchronous systems. To build an asynchronous P300 BCI several steps have to be foreseen. First, experimental protocols and evaluation criteria for asynchronous BCI systems should be defined. Second, algorithms that can detect if the user wants to communicate via the BCI or is engaged in other activity have to be developed. Such algorithms could possibly make use of features other than the P300. For example visual evoked potentials (VEPs) could be used to detect that a user is concentrating on the stimulus display and hence wants to communicate via the BCI.

Operant Conditioning of the P300

In systems employing slow cortical potentials or motor imagery, the use of feedback and operant conditioning is very common. It was shown that classification accuracy significantly increases as

subjects learn how to modulate their brain activity (Kübler *et al.*, 2001). In evoked potential BCIs, feedback and operant conditioning are currently not used, probably because evoked potentials are natural responses of the brain and are sufficiently robust for accurate classification. However, the possibility exists that feedback training could increase classification accuracy for systems based on evoked potentials. Such an increase in classification accuracy would be especially helpful for disabled subjects with a low base classification accuracy.

Evidence for the hypothesis that subjects can learn to control their evoked potentials is given in several papers. Sommer and Schweinberger (1992) and Miltner *et al.* (1986) describe experiments in which subjects learned, with the help of feedback, to increase or decrease the amplitude of their P300 evoked potentials. In a related experiment stimulus presentation was stopped after a P300 was evoked and subjects were asked to classify their brain response as small, medium or large (Sommer and Matt, 1990). Averaging of the evoked potentials, according to the classification given by the subjects, showed that P300 amplitude corresponded to the classification category. Subjects were thus aware of the amplitude of the P300 they were producing. This awareness could be important for learning to produce strong P300s.

One of the main challenges in building a P300 BCI system that uses operant conditioning is probably the development of classification algorithms that can give results quickly after stimulus presentation. Another challenge lies in the development of a feedback display which does not evoke unwanted EEG activity that could be confounded with activity relevant for classification.

Bibliography

M. Aitkin, G. T. Wilson (1980). Mixture models, outliers, and the EM algorithm. *Technometrics* **22**(3):325–331.

B. Z. Allison, J. A. Pineda (2003). ERPs evoked by different matrix sizes: Implications for a brain-computer interface (BCI) system. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**(2):110–113.

A. Bashashati, M. Fatourechi, R. K. Ward, G. E. Birch (2007). A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals. *Journal of Neural Engineering* **4**(2):R32–R57.

J. D. Bayliss (2003). Use of the evoked P3 component for control in a virtual apartment. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**(2):113–116.

R. Bellman (1961). *Adaptive Control Processes*. Princeton University Press.

H. Berger (1929). Über das Elektrenkephalogramm des Menschen. *Archiv für Psychiatrie und Nervenkrankheiten* **87**:527–570.

J. O. Berger (1988). *Statistical Decision Theory and Bayesian Analysis*. Springer.

N. Birbaumer, L. G. Cohen (2007). Brain-computer interfaces: Communication and restoration of movement in paralysis. *The Journal of Physiology* **579**(3):621–636.

N. Birbaumer *et al.* (1999). A spelling device for the paralysed. *Nature* **398**:297–298.

C. M. Bishop (2006). *Pattern Recognition and Machine Learning*. Springer.

B. Blankertz, G. Curio, K. R. Müller (2002). Classifying single trial EEG: Towards brain-computer interfacing. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, vol. 14.

B. Blankertz *et al.* (2004). The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering* **51**(6):1044–1051.

B. Blankertz *et al.* (2006a). The BCI Competition III: Validating alternative approaches to actual BCI problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2):153–159.

B. Blankertz *et al.* (2006b). The Berlin brain-computer interface: EEG-based communication without subject training. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2):147–152.

B. Blankertz *et al.* (2007). A note on brain actuated spelling with the Berlin brain-computer interface. In *Proceedings of International Conference on Human-Computer Interaction*.

J. Borisoff, S. Mason, G. Birch (2006). Brain interface research for asynchronous control applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2):160–164.

V. Bostanov (2004). Feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram. *IEEE Transactions on Biomedical Engineering* **51**(6):1057–1061.

G. W. Brier (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review* **78**:1–3.

C. Brunner *et al.* (2006). Online control of a brain-computer interface using phase synchronization. *IEEE Transactions on Biomedical Engineering* **53**(12):2501–2506.

M. T. Carrillo-de-la Pena, F. Cadaveira (2000). The effect of motivational instructions on P300 amplitude. *Neurophysiologie Clinique/Clinical Neurophysiology* **30**(4):232–239.

T. Centeno, N. Lawrence (2006). Optimising kernel parameters and regularisation coefficients for non-linear discriminant analysis. *Journal of Machine Learning Research* **7**:455–491.

C.-C. Chang, C.-J. Lin (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

S. Chiappa (2006). *Analysis and classification of EEG signals using probabilistic models for brain computer interfaces*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Switzerland.

I. Cohen, M. Goldszmidt (2004). Properties and benefits of calibrated classifiers. In *Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*, Springer.

E. Courchesne (1978). Changes in P3 waves with event repetition: Long-term effects on scalp distribution and amplitude. *Electroencephalography and Clinical Neurophysiology* **45**(6):754–766.

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

S. Coyle, T. Ward, C. Markham, G. McDarby (2004). On the suitability of near-infrared (NIR) systems for next-generation brain-computer interfaces. *Physiological Measurement* **25**(4):815–822.

E. Curran *et al.* (2004). Cognitive tasks for driving a brain-computer interfacing system: A pilot study. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **12**(1):48–54.

M. H. DeGroot, S. E. Fienberg (1983). The comparison and evaluation of forecasters. *The Statistician* **32**(1/2):12–22.

B. Djokic, M. Popovic, M. Lutovac (1998). A new improvement to the Powell and Chau linear phase IIR filters. *IEEE Transactions on Signal Processing* **46**(6):1685–1688.

E. Donchin (1981). Surprise!..surprise? *Psychophysiology* **18**:494 – 513.

E. Donchin, M. G. H. Coles (1988). Is the P300 component a manifestation of context updating? *Behavioral and Brain Sciences* **11**:357–374.

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

G. Dornhege, B. Blankertz, G. Curio, K. R. Müller (2003). Combining features for BCI. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, vol. 15.

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

V. Draglia, A. Tartakovsky, V. Veeravalli (1999). Multihypothesis sequential probability ratio tests Part I: Asymptotic optimality. *IEEE Transactions on Information Theory* **45**(7):2448–2461.

R. O. Duda, P. E. Hart, D. G. Stork (2001). *Pattern Classification*. Wiley Interscience.

C. Duncan-Johnson, E. Donchin (1977). On quantifying surprise. The variation of event related potentials with subjective probability. *Psychophysiology* **14**(5):456–467.

R. Dye *et al.* (1998). A script-based AAC system for transactional interaction. *Natural Language Engineering* **4**:57–71.

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

R. A. Fisher (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics* **7**:179–188.

X. Gao, D. Xu, M. Cheng, S. Gao (2003). A BCI-Based environmental controller for the motion-disabled. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**(2):137–140.

G. N. Garcia (2004). *Direct Brain-Computer Communication through Scalp Recorded EEG Signals*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Switzerland.

D. Garrett, D. Peterson, C. Anderson, M. Thaut (2003). Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**(2):141–144.

G. Golub, C. Van Loan (1996). *Matrix Computations*. John Hopkins University Press.

B. Graimann, J. Huggins, S. Levine, G. Pfurtscheller (2004). Toward a direct brain interface based on human subdural recordings and wavelet-packet analysis. *IEEE Transactions on Biomedical Engineering* **51**(6):954–962.

C. Guan, M. Thulasidas, J. Wu (2004). High performance P300 speller for brain-computer interface. In *Proceedings of IEEE International Workshop on Biomedical Circuits and Systems*.

E. Gysels, P. Celka (2004). Phase synchronization for the recognition of mental tasks in a brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **12**(4):406–415.

G. Hagen, J. R. Gatherwright, B. A. Lopez, J. Polich (2006). P3a from visual stimuli: Task difficulty effects. *International Journal of Psychophysiology* **59**:8 – 14.

T. Hastie, R. Tibshirani, J. Friedman (2001). *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*. Springer.

C. S. Herrmann (2001). Human EEG responses to 1-100 Hz flicker: Resonance phenomena in visual cortex and their potential correlation to cognitive phenomena. *Experimental Brain Research* **V137**(3):346–353.

N. Hill *et al.* (2006). Classifying EEG and ECoG signals without subject training for fast BCI implementation: Comparison of nonparalyzed and completely paralyzed subjects. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2):183–186.

T. Hinterberger *et al.* (2003). Brain areas activated in fMRI during self-regulation of slow cortical potentials (SCPs). *Experimental Brain Research* **V152**(1):113–122.

L. Hochberg *et al.* (2006). Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* **442**(7099):164–171.

A. E. Hoerl, R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **12**(1):55–67.

U. Hoffmann, G. N. Garcia, J.-M. Vesin, T. Ebrahimi (2004). Application of the evidence framework to brain-computer interfaces. In *Proceedings of the IEEE Engineering in Medicine and Biology Conference*.

U. Hoffmann, J.-M. Vesin, T. Ebrahimi (2006). Spatial filters for the classification of event-related potentials. In *Proceedings of the 14th European Symposium on Artificial Neural Networks (ESANN)*.

U. Hoffmann, J.-M. Vesin, T. Ebrahimi, K. Diserens (2007). An efficient P300-based brain-computer interface for disabled subjects. *Journal of Neuroscience Methods* Accepted for publication.

T. Hruby, P. Marsalek (2003). Event-related potentials - The P3 wave. *Acta Neurobiologiae Experimentalis* **63**(1):55–63.

M. Jordan, R. Jacobs (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* **6**:181–214.

M. Kaper (2006). *P300-based brain-computer interfacing*. Ph.D. thesis, University of Bielefeld, Germany.

M. Kaper, H. Ritter (2004). Generalizing to new subjects in brain-computer interfacing. In *Proceedings of the IEEE EMBS International Conference*.

M. Kaper *et al.* (2004). Support vector machines for the P300 speller paradigm. *IEEE Transactions on Biomedical Engineering* **51**(6):1073–1076.

L. Kauhanen *et al.* (2006). EEG and MEG brain-computer interface for tetraplegic patients. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2):190–193.

Z. Keirn, J. Aunon (1990). A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering* **37**(12):1209–1214.

J. Kittler, M. Hatef, R. Duin, J. Matas (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(3):226–239.

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

D. J. Krusienski *et al.* (2006). A comparison of classification techniques for the P300 speller. *Journal of Neural Engineering* **3**(4):299–305.

A. Kübler *et al.* (2001). Brain-computer communication: Self-regulation of slow cortical potentials for verbal communication. *Archives of Physical Medicine and Rehabilitation* **82**(11):1533–1539.

A. Kübler *et al.* (2005). Patients with ALS can use sensorimotor rhythms to operate a brain computer interface. *Neurology* **64**:1775 – 1777.

A. Kurosu, S. Miyase, S. Tomiyama, T. Takebe (2003). A technique to truncate IIR filter impulse response and its application to real-time implementation of linear-phase IIR filters. *IEEE Transactions on Signal Processing* **51**(5):1284–1292.

J. T.-Y. Kwok (2000). The evidence framework applied to support vector machines. *IEEE Transactions on Neural Networks* **11**(5):1162–1173.

T. Lal *et al.* (2004). Support vector channel selection in BCI. *IEEE Transactions on Biomedical Engineering* **51**(6):1003–1010.

T. Lal *et al.* (2005). A brain-computer interface with online feedback based on magnetoencephalography. In *Proceedings of the International Conference on Machine Learning*, pp. 465–472.

E. Lalor *et al.* (2005). Steady-state VEP-based brain-computer interface control in an immersive 3D gaming environment. *EURASIP Journal on Applied Signal Processing* **2005**(19):3156–3164.

M. A. Lebedev, M. A. Nicolelis (2006). Brain-machine interfaces: Past, present and future. *Trends in Neurosciences* **29**(9):536–546.

R. Leeb *et al.* (2006). Walking by thinking: The brainwaves are crucial, not the muscles! *Presence: Teleoperators and Virtual Environments* **15**(5):500–514.

S. Lemm, C. Schäfer, G. Curio (2004). Probabilistic modeling of sensorimotor mu-rhythms for classification of imaginary hand movements. *IEEE Transactions on Biomedical Engineering* **51**(6):1077–80.

E. Leuthardt *et al.* (2006). Electrocorticography-based brain computer interface - The Seattle experience. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2):194–198.

F. Lotte *et al.* (2007). A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering* **4**(2):R1–R13.

D. J. C. MacKay (1992). Bayesian interpolation. *Neural Computation* **4**(3):415–447.

D. J. C. MacKay (1995). Probable networks and plausible predictions – a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems* **6**:469–505.

D. J. C. MacKay (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

S. Makeig, A. Bell, T. Jung, T. Sejnowski (1996). Independent component analysis of electroencephalographic data. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, vol. 8.

J. Martin (1991). *Principles of neural science*, chap. The collective electrical behavior of cortical neurons: The electroencephalogram and the mechanisms of epilepsy, pp. 777–790. Elsevier.

S. Mason *et al.* (2007). A comprehensive survey of brain interface technology designs. *Annals of Biomedical Engineering* **35**(2):137–169.

D. McFarland, L. McCane, S. David, J. Wolpaw (1997). Spatial filter selection for EEG- based communication. *Electroencephalography and Clinical Neurophysiology* **103**(3):386–394.

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

J. Millan, F. Renkens, J. Mourino, W. Gerstner (2004). Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering* **51**(6):1026–1033.

W. Miltner, W. Larbig, C. Braun (1986). Biofeedback of visual evoked potentials. *The International Journal of Neuroscience* **29**(3-4):291–303.

K. Müller *et al.* (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks* **12**(2):181–201.

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

M. A. L. Nicolelis *et al.* (2003). Chronic, multisite, multielectrode recordings in macaque monkeys. *Proceedings of the National Academy of Sciences (PNAS)* **100**(19):11041–11046.

S. Nieuwenhuis, G. Aston-Jones, J. Cohen (2005). Decision making, the P3, and the locus coeruleus-norepinephrine system. *Psychological Bulletin* **131**(4):510–532.

J. Perelmouter, N. Birbaumer (2000). A binary spelling interface with random errors. *IEEE Transactions on Rehabilitation Engineering* **8**(2):227–232.

G. Pfurtscheller, G. R. Müller-Putz, J. Pfurtscheller, R. Rupp (2005). EEG-Based asynchronous BCI controls functional electrical stimulation in a tetraplegic patient. *EURASIP Journal on Applied Signal Processing* **2005**(19):3152–3155.

G. Pfurtscheller, C. Neuper (2001). Motor imagery and direct brain-computer communication. *Proceedings of the IEEE* **89**(7):1123–1134.

F. Piccione *et al.* (2006). P300-based brain computer interface: Reliability and performance in healthy and paralysed participants. *Clinical Neurophysiology* **117**(3):531–537.

J. Platt (1999). Using sparseness and analytic QP to speed training of support vector machines. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, vol. 11.

J. Polich (1987). Task difficulty, probability, and inter-stimulus interval as determinants of P300 from auditory stimuli. *Electroencephalography and Clinical Neurophysiology* **68**:311 – 320.

J. Polikoff, H. Bunnell, W. Borkowski (1995). Toward a P300-based computer interface. In *Proceedings of the RESNA '95 Annual Conference*.

V. Polikov, P. Tresco, W. Reichert (2005). Response of brain tissue to chronically implanted neural electrodes. *Journal of Neuroscience Methods* **148**(1):1–18.

M. I. Posner, S. E. Petersen (1990). The attention system of the human brain. *Annual Review of Neuroscience* **13**(1):25–42.

P. Pudil, J. Novovičová, J. Kittler (1994). Floating search methods in feature selection. *Pattern Recognition Letters* **15**(11):1119–1125.

C. S. Qazaz, C. K. I. Williams, C. M. Bishop (1996). *Mathematics of Neural Networks: Models, Algorithms and Applications*, chap. An Upper Bound on the Bayesian Error Bars for Generalized Linear Regression. Kluwer.

A. Rakotomamonjy (2007). Ensemble of SVMs for BCI III P300 speller competition. Website: `http://asi.insa-rouen.fr/~arakotom/code/bciindex.html`.

A. Rakotomamonjy, V. Guigue, G. Mallet, V. Alvarado (2005). Ensemble of SVMs for improving brain-computer interface P300 speller performances. In *Proceedings of International Conference on Neural Networks*.

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

N. Ramsey, M. Van De Heuvel, K. Kho, F. Leijten (2006). Towards human BCI applications based on cognitive brain systems: An investigation of neural signals recorded from the dorsolateral prefrontal cortex. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2):214–217.

B. Rebsamen *et al.* (2006). A brain-controlled wheelchair based on P300 and path guidance. In *Proceedings of IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*.

S. Roberts, W. Penny (2000). Real-time brain-computer interfacing: A preliminary study using Bayesian learning. *Medical and Biological Engineering and Computing* **38**(1):56–61.

R. Scherer *et al.* (2004). An asynchronously controlled EEG-based virtual keyboard: Improvement of the spelling rate. *IEEE Transactions on Biomedical Engineering* **51**:979–984.

A. Schlögl, F. Lee, H. Bischof, G. Pfurtscheller (2005). Characterization of four-class motor imagery EEG data for the BCI-competition 2005. *Journal of Neural Engineering* **2**(4):L14–L22.

A. Schmolck, R. Everson (2007). Smooth relevance vector machine: a smoothness prior extension of the RVM. *Machine Learning* **68**(2):107–135.

E. Sellers, E. Donchin (2006). A P300-based brain-computer interface: Initial tests by ALS patients. *Clinical Neurophysiology* **117**(3):538–548.

H. Serby, E. Yom-Tov, G. Inbar (2005). An improved P300-based brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **13**(1):89–98.

M. D. Serruya *et al.* (2002). Instant neural control of a movement signal. *Nature* **416**:141–142.

R. Sitaram *et al.* (2007). Temporal classification of multichannel near-infrared spectroscopy signals of motor imagery for developing a brain-computer interface. *Neuroimage* **34**(4):1416–1427.

W. Sommer, J. Matt (1990). Awareness of P300-related cognitive processes: A signal detection approach. *Psychophysiology* **27**(5):575–585.

W. Sommer, S. Schweinberger (1992). Operant conditioning of P300. *Biological Psychology* **33**(1):37–49.

K. C. Squires, C. Wickens, N. K. Squires, E. Donchin (1976). The effect of stimulus sequence on the waveform of the cortical event-related potential. *Science* **193**(4258):1142–1146.

N. K. Squires, K. C. Squires, S. A. Hillyard (1975). Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man. *Electroencephalography and Clinical Neurophysiology* **38**:387 – 401.

R. Srinivasan (1999). Methods to improve the spatial resolution of EEG. *International Journal of Bioelectromagnetism* **1**:102–111.

U. Strehl *et al.* (2006). Self-regulation of slow cortical potentials: A new treatment for children with attention-deficit/hyperactivity disorder. *Pediatrics* **118**(5):1530–1540.

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

P. Sykacek *et al.* (2003). Probabilistic methods in BCI research. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**(2):192–194.

D. Taylor, S. Tillery, A. Schwartz (2002). Direct cortical control of 3D neuroprosthetic devices. *Science* **296**(5574):1829–1832.

M. Thulasidas, C. Guan, J. Wu (2006). Robust classification of EEG signal for brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(1):24–29.

Q. Tian, Y. Fainman, S. H. Lee (1988). Comparison of statistical pattern-recognition algorithms for hybrid processing. II. Eigenvector-based algorithm. *Journal of the Optical Society of America A* **5**:1670–1682.

M. Tipping (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* **1**(3):211–244.

T. Van Gestel *et al.* (2002). Bayesian framework for least-squares support vector machine classifiers, gaussian processes, and kernel fisher discriminant analysis. *Neural Computation* **14**(5):1115–1147.

T. Vaughan, J. Wolpaw (2006). The third international meeting on brain-computer interface technology: Making a difference. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**:126–127.

R. Verleger (1988). Event-related potentials and memory: A critique of the context updating hypothesis and an alternative interpretation of P3. *Behavioral and Brain Sciences* **11**:343–356.

J.-M. Vesin, U. Hoffmann, T. Ebrahimi (2006). *Wiley Encyclopedia of Biomedical Engineering*, chap. Human-Brain Interface: Signal Processing and Machine-Learning. Wiley.

C. Vidaurre *et al.* (2006). A fully on-line adaptive BCI. *IEEE Transactions on Biomedical Engineering* **53**(6):1214–1219.

N. Weiskopf *et al.* (2004). Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). *IEEE Transactions on Biomedical Engineering* **51**(6):966–970.

J. Wilson *et al.* (2006). ECoG factors underlying multimodal control of a brain-computer interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **14**(2):246–250.

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

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

J. R. Wolpaw *et al.* (2002). Brain-computer interfaces for communication and control. *Clinical Neurophysiology* **113**(6):767–791.

N. Xu *et al.* (2004). BCI competition 2003 – Data Set IIb: Enhancing P300 wave detection using ICA-based subspace projections for BCI applications. *IEEE Transactions on Biomedical Engineering* **51**(6):1067–1072.

S.-S. Yoo *et al.* (2004). Brain-computer interface using fMRI:Spatial navigation by thoughts. *Neuroreport* **15**(10):1591–1595.

Curriculum Vitae

Ulrich Hoffmann

Avenue Vinet 8, 1004 Lausanne, Switzerland
Tel.: +41 76 222 1831
Email: ulrich.hoffmann@epfl.ch

Personal

Date of birth: October 18, 1974

Nationality: German

Civil Status: Single

Education

2003 - 2007: **Ph.D.**, Signal Processing Institute, School of Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

1994 - 2001: **Diploma**, Informatics, with minor in medicine, University of Tübingen, Germany

Professional Experience

2003 - 2007: **Research and Teaching Assistant**, Signal Processing Institute, School of Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Assisted students in courses, supervised students for semester projects
Preparation of grant applications

2001 - 2002: **Consultant and Programmer**, Innovate GmbH, Wildberg, Germany
Teaching and consulting in the areas Java and SAP databases
Development of Java-based webinterfaces

1996 - 2000: **Assistant Programmer**, Data Center, University of Tübingen, Germany
Administration of UNIX systems, programming in C

Publications

Journal Papers

- U. Hoffmann, J.M. Vesin, T. Ebrahimi, and K. Diserens, "**An Efficient P300-based Brain-Computer Interface for Disabled Subjects**", accepted to *Journal of Neuroscience Methods*, 2007
- D. Studer, U. Hoffmann, and T. Koenig, "**From EEG Dependency Multichannel Matching Pursuit to Sparse Topographic Decomposition**", *Journal of Neuroscience Methods*, 153, pp. 261-275, 2006

Book Chapters

- J.M. Vesin, U. Hoffmann, and T. Ebrahimi, "**Human Brain Interface: Signal Processing and Machine Learning**", *Wiley Encyclopedia of Biomedical Engineering*, 2006

Conference Papers

- U. Hoffmann, J.M. Vesin, and T. Ebrahimi, "**Spatial Filters for the Classification of Event-Related Potentials**", *European Symposium on Artificial Neural Networks*, 2006
- U. Hoffmann, G.N. Garcia, J.M. Vesin, K. Diserens, and T. Ebrahimi, "**A Boosting Approach to P300 Detection with Application to Brain-Computer Interfaces**", *IEEE EMBS Conference on Neural Engineering*, 2005
- U. Hoffmann, G.N. Garcia, J.M. Vesin, and T. Ebrahimi, "**Application of the Evidence Framework to Brain-Computer Interfaces**", *IEEE EMBS Conference*, 2004
- W. Stürzl, U. Hoffmann, H.A. Mallot,"**Vergence Control and Disparity Estimation with Energy Neurons: Theory and Implementation**", *International Conference on Artificial Neural Networks*, 2002

Grant Applications

First author of three successful grant applications:

Towards Second Generation Brain-Computer Interfaces, Swiss National Science Foundation, Research Project, Duration 36 months

A Mobile EEG Acquisition System, Ecole Polytechnique Fédérale de Lausanne (EPFL), Funding for Equipment

Machine Learning and Signal Processing for Brain-Computer Interfaces, Swiss National Science Foundation, Research Project, Duration 18 months