Determining User State and Mental Task Demand From Electroencephalographic Data

Diplomarbeit

Matthias Honal

University of Karlsruhe

Supervisors:
Dr. Tanja Schultz
Prof. Dr. Alexander Waibel

November 2005

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Karlsruhe, den 30.11.2005

Contents

List of Figures

List of Tables

Abstract

Information about people's current activity (their user state) and their mental task demand can be used for multiple purposes in meeting, lecture or office scenarios.

Depending on the current user state and the level of task demand mobile communication devices such as cell-phones can configure themselves in a way that they notify their owner of an incoming event (e.g. a phone call) only, if this does not disturb him for instance. Furthermore information about user state and task demand of an audience can be used to provide feedback to a speaker about his talk.

In this thesis a system is proposed which determines user state and task demand using electroencephalographic data (EEG data). EEG is recorded using either 16 scalp electrodes from a standard recording device which is usually used for clinical purposes, or a headband with only four electrodes over the pre-frontal and frontal cortex, which is much more comfortable to wear. The recorded data is then passed to a computer where features are extracted which represent the frequency content of the signals, features are preprocessed and finally passed to an artificial neural network or to a Support Vector Machine which predict user state and task demand.

For the discrimination of the user states resting, listening, perceiving a presentation, reading an article in a magazine, summarizing the read article and performing arithmetic operations classification accuracies of 94.9% in session and subject dependent experiements, 58.9% in subject independent experiments and 62.7% in subject dependent but session independent experiments could be obtained. For the prediction of low and high task demand during the perception of a presentation accuracies of 92.2% in session and subject dependent experiements, 80.0% in subject independent experiments and 87.1% in subject dependent but session independent experiments were achieved. While all these experiments were obtained in offline scenarios, where data had been collected long before the system was trained and tested, also a prototype system has been developed which demonstrates the feasibility of user state identification and task demand assessment in real time.

Acknowledgments

At this place I would like to thank the many people who helped me to complete this thesis. First of all, all the volunteers must be mentioned who allowed me to collect their EEG data, particularly the students from Carnegie Mellon University and the people from the second floor of the ”Insterburg” in Karlsruhe. Next I would like to thank Dana Burlan who helped me during the data collection and who sewed in the electrodes in the EEG-recording-headband with skillful hands.

Very special thanks go to my wife Laura who often assisted me during the data collections, who volunteered for many data collections herself, who helped me to produce a demo video, who proofread this manuscript and last but not least who was always there for me in times when I found it difficult to carry on with this thesis.

Finally I would like to thank my supervisor Tanja Schultz for all the valuable discussions and the comments to my work which significantly improved its quality.

Chapter 1

Introduction

In modern office or meeting environments people interact with each other in various ways: (mobile) electronic devices such as cell phones, PDAs and laptops are used to communicate via speech or text but also face-to-face meetings take place frequently nowadays.

Information about user state, i.e. the current activity of an individual and mental task demand, i.e. the amount of mental resources required to execute the current activity, can provide important information about the individuals needs and wishes concerning interaction. This information could be exploited for example by communication devices which select the kinds of notifications a user receives appropriately according to the user's current state and mental task demand level. On the other hand information about the predominant state and the task demand level of his audience can represent an interesting feedback for a speaker about a talk.

We refer to the current activity of an individual as *user state* in this work to emphasize that we see a major purpose of gathering such activity information to improve the interfaces between communication devices and their *users*. To characterize the predominant activity of an audience during a meeting or a lecture, the term ”audience state” or ”participant state” would probably be more appropriate. Note however that technically speaking a listener of a talk can also be seen as a ”user” who obtains information from a ”device”, namely the speaker, via an ”interface”, which is the talk of the speaker[1]. Therefore and for reasons of clarity only the term user state is used in this work to characterize an individual's current activity. Note furthermore that the more general term task demand instead of mental task demand is used in the following for simplicity, although we are exclusively concerned with mental and not with physical task demand in this work.

1.1 Goal

The major goal of the research work described in this thesis is the development of a system which automatically identifies the user state and estimates the task demand level of an individual from his electrical brain activity, i.e. his electroencephalogram (EEG). In particular such a system shall be able to determine user state and task demand level in meeting, lecture or office scenarios such that the obtained information can be used to improve the interaction with and via electronic communication devices and face-to-face interaction between individuals. (Examples illustrating the relevance of user

[1] This point of view is even more justified when the human speaker is replaced by a computer system providing one listener or an audience with information about a specific topic which is not a too unrealistic vision in times of modern dialog and information retrieval systems.

state and task demand information for applications in the mentioned scenarios are given in the next section.)

Several subgoals emerge from this major goal:

- **Robustness:** Users must be allowed to talk and to move freely during EEG recording in an office or meeting environment which introduces usually large artifacts in the data. During clinical EEG recording data artifacts are reduced by requiring the patients not to talk and to remain immobile in a fixed position during the whole recording which is clearly not acceptable for the applications we are aiming at. Therefore the proposed system must be able to cope with all kinds of artifacts introduced in the data by moving and talking, i.e. it must be robust towards these artifacts.

- **Acceptability:** The sensors for EEG recording should disturb the user as little as possible and the user must find his outer appearance still acceptable while wearing them. During clinical EEG recording usually more than 20 scalp electrodes are placed all over the head. Prior to the recording intensive preparation of the patient is required to assure good data quality: electrode positions must be determined exactly, the scalp is cleaned using alcohol and conductive paste in the patients hair is required to establish a good conductivity between skin and electrode. The use of electrodes at positions all over the head and intensive user preparation is clearly not possible for the scenarios we are considering here. Ideally very few electrodes should be used which the user can attach quickly himself. Furthermore conductive paste must not get into contact with the user's hair.

- **Real time behavior:** It must be possible to determine user state and task demand information in real time, to allow for immediate reactions when these parameters change.

- **Realistic scenarios:** The user states to be considered here must be typical for real world meeting, lecture or office scenarios. Typical states are reading, typing, listening to a talk, perceiving an audio-visual presentation, talking or resting for example. Also task demand variations must be measured during realistic activities such as listening to a talk. As a mid-term goal the developed system shall be used and evaluated in real meetings, lectures or office work places.

Note that the research work described in this thesis is just a first attempt towards user state identification and task demand estimation in meeting or office scenarios. Therefore the aspects enumerated above are realized only to a certain extend in the proposed system (see also chapter 7). Results are promising however and we are confident that some further research and development will allow to fulfill the afore mentioned goals even better.

1.2 Motivation

The most important motivation for the development of a system which determines user state and task demand is to allow for more intelligent interaction between individuals and their communication devices and to make communication between individuals more effective. Concrete examples which illustrate how the obtained information can be used for these purposes are presented in the following.

1.2.1 Human-Machine Interaction

Modern communication devices can be configured in various ways. In particular, different kinds of notifications (e.g. audio, visual or tactile notifications) can be chosen to announce different events or communication requests of different (groups of) communication partners. It will be illustrated in the following examples that in presence of different user states and/or task demand levels, different configurations of communication devices may be appropriate. To change a device configuration, usually manual interaction is required which is often inconvenient. Therefore important communication requests are often missed or they disturb users while they are busy with more important activities.

An essential benefit of automatic user state identification and task demand estimation is that intelligent user interfaces can be constructed, which change their configurations themselves according to the current user state and/or task demand level. Thus they require the user's attention only when this does not divert him from doing something more important. The following examples illustrate possible applications:

- Consider the meeting scenario shown in figure 1.1. Let us assume that each participant possesses a user state identification device which determines his current user state as shown in the picture. This information can now be used for example to configure the participants' cell phones appropriately. Let us first consider the speaker: he might not want to be distracted by any kinds of notifications since this would confuse him during talking. Therefore his cell phone switches itself off to be completely quiet while he is talking. The two persons who are listening might want to have configured their cell phones in a way that the devices notify their owners only of very important events such as calls from important business partners or of their boss. The meeting participants who are reading or resting might be open for all kinds of notifications since they are not directly involved in the meeting at the moment. Therefore their cell phones configure themselves in a way that their owners are notified (by a non-audible, e.g. a vibrating alert) of all incoming events since they have enough time to look at the display and to decide themselves how they want to proceed with the current event.

  Information about the current task demand level might be useful here as well. It could be used for example to find out whether a user is highly concentrated during listening or whether it is easy for him to follow a talk. In the latter case a disturbance by some communication device might be more acceptable for him since here the risk of loosing the thread is smaller.

  Besides cell phones also other devices could configure themselves appropriately using user state and task demand information. A person who is taking minutes or reading might be disturbed more by a text message or a chat request popping up on his laptop screen than someone who is resting, since he is simply not interested in the current subject of the meeting.

  Note that regardless of the device which uses the collected information, it is essential that *the user himself* is able to define its behavior for each user state and task demand level, i.e. he must not be forced to accept a preset device configuration, where he can not decide anymore himself how the device behaves in different situations.

- Typical activities during office work are reading, typing or the development of own concepts and ideas which sometimes may require even more or less complex mathematical reasoning. Finally there are also intermediate resting periods where no such action is taken.

  Let us assume again that for some person who is working in an office, user state and task demand can be identified by some device. Similarly to the meeting scenario described above the obtained information could be used here as well to configure communication devices appropriately. Possible preferences might be for example to allow no, or only very important phone calls during complex reasoning and to disable chat requests when complex reasoning coincides with typing or reading something on the computer screen. Information about the task demand

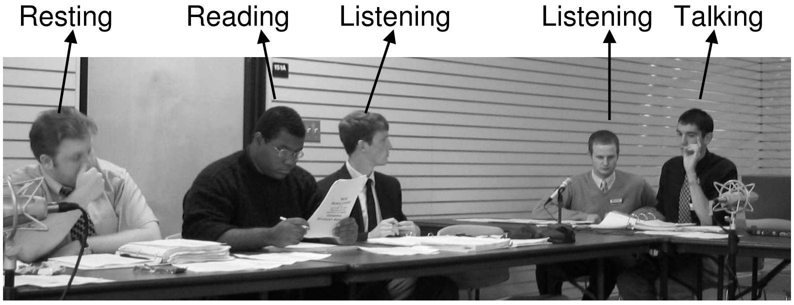


Figure 1.1: A typical meeting. Picture taken from [Student Government, University of Maine, 2005]

level might help here to make more fine grained decisions: communication requests might be less disturbing while reading an article which is easy to understand in contrast to another article with a more difficult contents for example. Also during routine tasks (e.g. reading or writing e-mails) some people might want to constrain the potential communication requests they are willing to receive since they want to get their work done quickly. On the other hand they would like to be notified of all kinds of communication requests (also of those which have previously been rejected) when they are idle, such as during resting periods. Consequently it might be desirable in an office environment to have intelligent devices.

1.2.2 Enhance Face-to-Face Communication

User state and task demand can provide important feedback about how perceived in a face-to-face communication. This is illustrated in the following two examples.

- During a meeting or a lecture it is usually very difficult for the speaker to tell whether his talk is too easy or too difficult for his audience or whether he is talking about a subject of common interest or not. Information about the predominant user state of his audience and its average task demand could help the speaker to find out how his talk is perceived. Low task demand could be seen as a hint to proceed faster while high task demand (or even overload) might indicate on the other hand that the current topic must be explained more clearly. This could be particular helpful in a lecturing or teaching scenario.

  User state information could be used here to find out whether the audience is interested at all in the current topic. This might not be the case if the predominant user state is something else than listening. Furthermore, user state information is important to retrieve out the precise reasons for high task demand: While a user's task demand is constantly high, he can either listen attentively to the presentation and watch the slides (user state ”audiovisual perception”) or that he can read a paper which requires a lot of mental effort.

- Another interesting application in this context is to improve an information retrieval and dialog system which provides multi modal information about a selected topic to a user. If the system detects high task demand or even overload it might use this as an indicator that the user can't keep up anymore with the current presentation and thus it could come up with other explanations. Note that also explicit interaction between listener and system is possible and perhaps even desirable in such situations. However the information how familiar the user is with different aspects of a certain topic, which can be perhaps inferred from the course of his task demand, might help to select the most appropriate explanations.

- Now let us consider two people who cannot communicate in a common language and therefore use a speech translation device to talk to each other. In this case, it is usually difficult to tell whether the translation of what was just said is correct, i.e. if the dialog partner understood the meaning of the utterance as it was originally intended in the source language. If now the confusion of the dialog partner could be measured when he hears the translated utterance, this information could be used as an indicator for the translation quality. Note that the parameters task demand and user state are not used here explicitly, however it might be a plausible hypothesis that the degree of confusion is correlated with the degree of task demand, since a confused user will mobilize more mental resources to determine the meaning of an utterance he could not understand at once.

  This hypothesis is supported by findings from [Applied Anthropology Institute, 2001] and [Defayolle et al., 1971] who report that mental confusion is often associated with the state of ”extreme alertness” which can be identified by distinct patterns in the EEG. (For the relation between alertness and task demand please refer to section 2.3.2.) In these studies a very high degree of confusion, leading often to disorganized behavior, has been considered however. Therefore it remains to be investigated whether EEG correlates of confusion can also be detected in the above described scenario.

1.2.3 Measure Usability

Probably task demand information can be used as well for the evaluation of the usability of interfaces. Using the underlying hypothesis that an electronic device (e.g. a cell phone, a PDA, a radio, an air condition or a navigation system in a car) which is difficult to operate requires higher mental resources during operation, task demand could be used directly as an indicator for usability. Thus EEG based task demand estimation could be used as a tool in ergonomics.

Although a few years ago it has been proposed to use EEG for usability assessment [Beer et al., 2003], to our current knowledge no research has been conducted yet which provides evidence for or against the above hypothesis. Only the relations between EEG and attitude, satisfaction or acceptance in the domain of user interface design have been investigated [Nielsen, 1993].

Note that for many aforementioned applications EEG data is certainly not the only source of information. Other physiological parameters such as the electrocardiogram (the ECG, i.e. the electrical heart activity) or the electromyogram (the EMG, i.e. the electrical muscular activity), methods like eye tracking or head tracking and other techniques from computer vision or speech and language processing can certainly provide useful information for these applications as well. Physiological data and particularly EEG data seem however to be complementary in many aspects to these other modalities which makes their investigation particularly interesting here. A fusion of different modalities should be an important goal towards the development of intelligent user interfaces and tools which enhance communication between individuals.

1.3 Ethical Considerations

When user state or mental task demand are determined from individuals in every day situations, very personal and sensitive data is collected. Thus it is important to handle this data very carefully since it might easily be used to offend peoples privacy. The performance of employees or their actions could be easily tracked using the recorded data. User state information might be used for example to find out how much time people spend on other than work tasks during their office hours; information about mental task demand during the execution of a specific task could indicate how well an applicant is suited for a certain position.

The research work reported in this thesis has clearly not been done to encourage the development of applications aiming at these or similar purposes.

Rather then to control the user, the collected information should be used to give the user better control over his environment. As mentioned above, one reason to determine user state and task demand is to enhance the user's ability to interact with electronic devices and to enable these devices to behave according to the users wishes without requiring explicit input. However this can only be granted if the user has complete control over the collected information and over the devices using it. That means in particular that the user must be able to choose desired device configurations for a given user state and/or task demand level *himself* and he must not be forced to accept any preset configuration. The latter might for example lead to a situation where in order to increase the productivity of an employee, phone calls, text messages etc. from his family or friends are blocked while he is not in the user state of resting. This would take him away from controlling the use of the collected data which is clearly not desirable.

In this context it is also important to point out the difference between this work and the DARPA Augmented Cognition (AugCog) program where important goals are the identification of cognitive states and the assessment of mental task demand [Schmorrow and Kruse, 2002]. In the AugCog program the information about cognitive states shall be used to increase operator performance by presenting him information appropriately according to his current state and thus to avoid cognitive overload. That implies that the operator has to allow the recording of his cognitive state continuously in order to be able to fulfill the task assigned to him. He has no choice how data is presented to him, i.e. which view of the real data he can see. This might arouse the feeling of the operator that he is degraded to a kind of ”computer” himself which has to fulfill a certain task without having the control over the whole situation. While this may be appropriate for some military applications, it is not acceptable for an individual in a office or meeting environment. Here everyone must be able to control himself how and when which kind of information is presented to him, i.e. he must be able to configure the devices around him himself as mentioned above. Additionally it is important that users should wear EEG devices only voluntarily and they should always be able to switch them off.

Furthermore the user must know exactly what happens with the collected data and it must be his own decision whether he wants to share the data with others, e.g. with a speaker to give him feedback about his talk, or not. In the meeting or lecture scenario described above, it is important that for privacy reasons the speaker can only see the *predominant* user state and the *average* task demand of his audience but that the data of particular individuals remains hidden to him. Otherwise such data might be used for example to find out which students do not listen to a lecture or which employees do not pay attention during a presentation of their boss.

Finally EEG devices for user state and task demand recordings should be preferably personal devices used only by their owner, since this allows him to record information about his mental state and his task demand whenever he wants and use it for the purpose he wants to use it. Also for hygienic reasons users may not want to wear an EEG device which has been previously used by someone else (unless it has been cleaned thoroughly which is time consuming).

1.4 Contributions

In this research work several state-of-the-art techniques from the domains of machine learning and signal processing are applied to the problems of user state identification and task demand estimation from EEG data. The selection of appropriate methods, their combination, their adaptation to the specific requirements of EEG data and finally the careful evaluation and discussion of their performance represent important scientific contributions of this research.

As already mentioned, the research work presented here is to be seen as a first attempt to construct a system for user state identification and task demand estimation using EEG data in every day meeting, lecture or office scenarios. To date, no publications could be found concerning the application of EEG data in these scenarios. Particularly the attempt to reach the goals formulated in section 1.1 distinguishes this work from other research which is concerned with computational processing of EEG data.

Also two more practical contributions shall be pointed out here which address the goals formulated in section 1.1:

- To illustrate the feasibility recording devices which are more comfortable to wear than standard clinical equipment, a headband with four build-in electrodes has been developed (see figure 5.2). This can be seen as a first step towards recording devices which are more acceptable to wear in every day situations.

- Furthermore a prototype system has constructed which demonstrates that the identification of realistic user states is possible in real time and in a scenario which has little in common with the well defined laboratory conditions during clinical EEG recording (see section 6.1.10). Note that the same user states and the same recording conditions were also used for the data collection conducted in this work (see chapter 5).

1.5 Overview

Figure 1.2 gives an overview of the developed system and illustrates the tasks to be accomplished to derive a hypothesis for the current user state and task demand level from the raw EEG data. For each of the task one or more possible methods are investigated.

As a first step, artifacts introduced in the raw EEG data (50Hz or 60Hz AC noise, muscular activity, eye movements etc.) must be eliminated. We concentrate here on eye activity related artifacts and attempt to use independent component analysis (ICA) for their detection and removal. From the artifact free data feature vectors are extracted representing the frequency content of the data using a short time Fourier transform (STFT). Then averaging over a history of k feature vectors is performed and features are normalized to reduce natural fluctuations and unwanted variability in the data. Since the feature space has a very high dimension, the benefit of feature reduction methods is investigated. Finally artificial neural networks (ANNs) and support vector machines (SVMs) are used as classification techniques for user state identification. These classification techniques are also applicable for the estimation of different task demand levels. However since task demand is an ordinally scaled parameter, also variants of ANNs and SVMs for regression estimation and a simple linear model (OLS-Regression) are applied for this purpose.

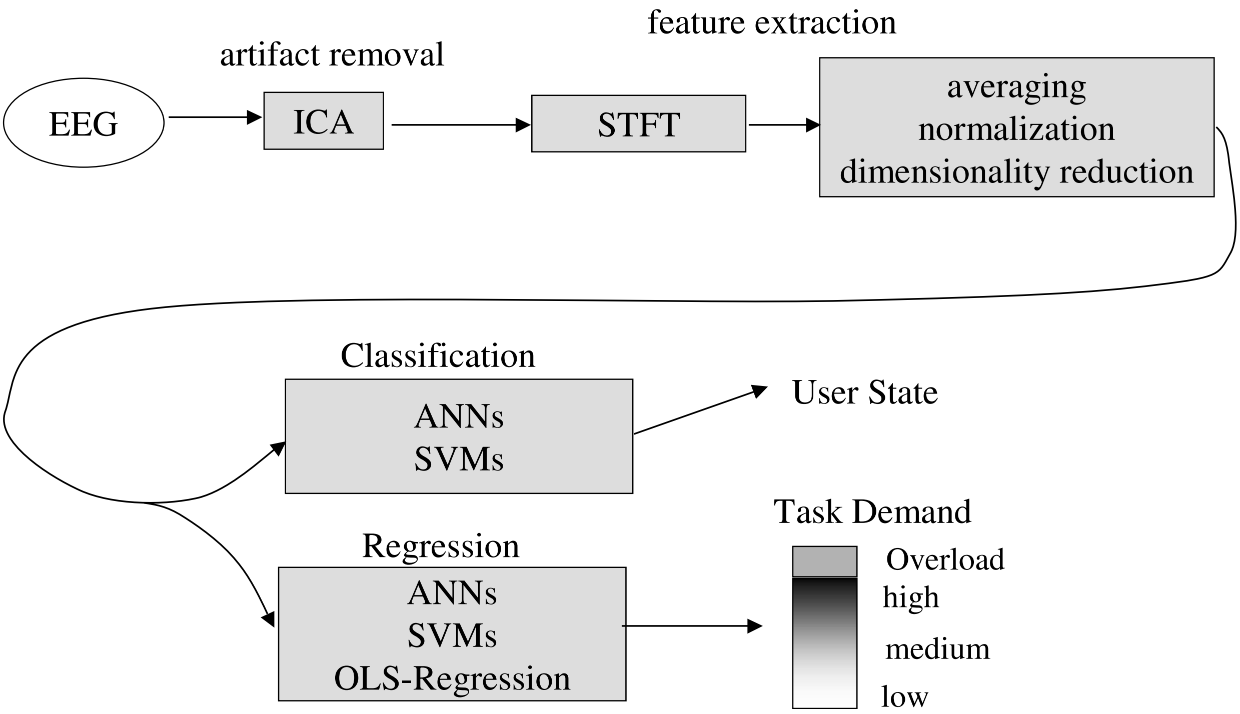


Figure 1.2: Overview of the system for user state identification and task demand estimation.

The remainder of this thesis is organized as follows. In chapter 2 basics of brain anatomy and physiology are presented, an overview over several techniques for monitoring brain activity (with special emphasis on the EEG) is provided and the neural correlates of different user states and of mental task demand are explained briefly. Other research dealing with the computational processing of brain activity is reviewed in chapter 3. Chapter 4 describes the methods used in the developed system. The data collection is presented in chapter 5 and the results from the experiments conducted for this work are summarized in chapter 6. Conclusion and directions for future work are given in chapter 7.

Chapter 2

Bio-Medical Background

Our ability to observe the activity of the living brain is very limited. Current techniques for monitoring brain activity can only provide information about an extremely small fraction of those processes which are responsible for our actions, our thinking and also our consciousness. To give the reader an impression of the huge complexity of the brain, section 2.1 reviews briefly basics of brain anatomy and physiology. A special emphasis is put on those processes causing the EEG. In section 2.3 neural correlates and in particular EEG correlates of different mental states and of varying task demand levels are explained which represent the bio-medical foundation for the research presented here. Finally some none-invasive techniques for monitoring brain activity are introduced in section 2.2 and it will become clear that due to limitations in spatial or temporal resolution of these techniques, to date only very incomplete information about ongoing processes in the brain can be visualized.

2.1 Anatomy and Physiology of the Brain

The human's central nervous system consists of the spinal cord and the brain. One of its tasks is to process and to integrate incoming sensory stimuli which are received via peripheral nerves (afferences) and to give impulses back to actuators, e.g. to muscles or glands (efferences) which cause automatic or voluntary action. Furthermore the central nervous system, particularly the brain, is responsible for higher integrative abilities such as thinking, learning, production and understanding of speech, memory, emotion etc. Finally vegetative functions such as respiration and the cardio-vascular system are controlled by the central nervous system.

2.1.1 Brain Anatomy

Anatomically five basic parts of the brain can be distinguished [Faller, 1995] as shown in figure 2.1:

Cerebrum: The cerebrum which is located directly under the skull surface is the largest part of the brain. Its main functions are the initiation of complex movement, speech and language understanding and production, memory and reasoning. Brain monitoring techniques which make use of sensors placed on the scalp mainly record activity from the outermost part of the cerebrum, the cortex. More inside the cerebrum the basal ganglions can be found which consist of a number of nuclei controlling the extend and the direction of slow movements. Also the thalamus is located here which directs sensory information to appropriate parts of the cortex. A more detailed explanation of the cortex anatomy is given below, since this becomes particularly important in later sections of this work.

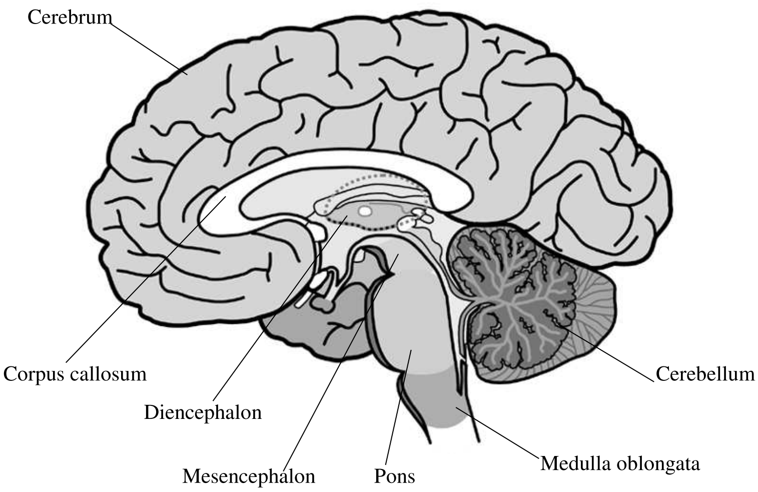


Figure 2.1: Different anatomical parts of the human brain, with modifications from [Scientific Learning Cooperation, 1999]

Diencephalon: One important function of the diencephalon is the forwarding of sensory information to other brain areas. Besides that, it contains the hypothalamus which controls the body temperature, the water balance and the ingestion to assure the state of homeostasis for the body, i.e. ”good working conditions” for all body cells.

Cerebellum: The coordination of all kinds of movements is done in the cerebellum. Therefore it cooperates closely with structures from the cerebrum (e.g. the basal ganglions). Cerebellum and cerebrum are connected via the Pons.

Mesencephalon: The largest part of the reticular system (the formatio reticularis) is located here. It controls vigilance and the sleep-wake rhythm.

Medulla oblongata: The medulla oblongata connects the brain with the spinal cord. Respiration and the cardiovascular system are controlled by that part of the central nervous system. Furthermore a huge number of peripheral nerves pass through the medulla oblongata.

Compared to the brains of other mammals, the human brain has the largest and best developed cortex. Neural processes related to abilities like complex reasoning, speech and language etc. which distinguish humans from other mammals take place in that part of the brain.

The cortex consists of two hemispheres which are connected via a beam called corpus callosum (see figure 2.1). Each hemisphere is dominant for particular abilities. For right handed persons the right hemisphere is activated more during the recognition of geometric patterns, spatial orientation, the use non verbal memory and the recognition of non-verbal noises while more activity in the left hemisphere can be observed during the recognition of letters and words, the use verbal memory and auditory perception of words and language. Note however that this hemispheric asymmetry is usually not very pronounced and in cases of injuries one hemisphere is often able to fulfill tasks for which the other one was previously dominant [Schmidt and Thews, 1997].

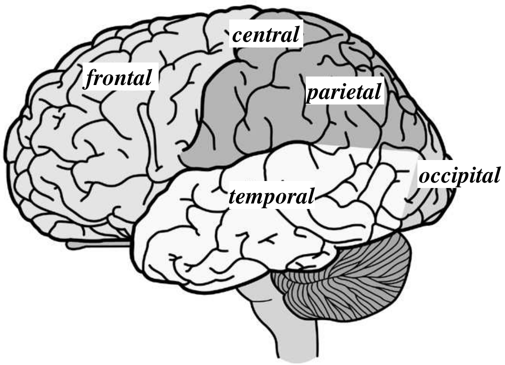


Figure 2.2: Different cortex lobes, with modifications from [Scientific Learning Cooperation, 1999].

Each hemisphere is partitioned in five anatomically well defined regions, the so called lobes as depicted in figure 2.2. The functions of particular lobes are explained in section 2.3.1.

2.1.2 Brain physiology

The basic unit for information processing in the brain is the neuron. As shown in figure 2.3 a neuron consists of dendrites which collect information from other neurons, the soma where this information is processed and the axon which transfers processed information to other neurons. In the following signals which are directed to a neuron are referred to as *afferences*, signals which flow away from the neuron via the axon are referred to as *efferences*. Note that this terminology is used as well for the neural information flow in the whole body: Afferences denote here all kinds of signals which are directed to the brain, efferences are signals which flow away from the brain.

The *frequency* of the electrical impulses is used to code information in the axon, i.e. the higher the impulse frequency, the stronger the intensity of the transmitted signal. The point where the end of an axon gets in contact with a dendrite or with the soma of a neighboring neuron is called synapse. At the synapse a chemical reaction takes place which allows the flow of certain types of ions in and out of the post-synaptic neuron. The rate of the ion flow is controlled by the impulse frequency in the (pre-synaptic) axon. Thus the frequency coding of information in the axon is changed to an *amplitude* coding in the post-synaptic neuron. For certain types of axons positive ions (Na^+ or Ca^{2+}) move *in* the post-synaptic neuron which causes an excitatory post-synaptic potential (EPSP), for other types positive ions (K^+) move *out* of the post-synaptic neuron which causes an inhibitory post-synaptic potential (IPSP). In the soma all these potentials are summed up (EPSPs are positive, IPSPs are negative) and if the sum exceeds a certain threshold an impulse is given on the axon (of the post-synaptic neuron) with a frequency which is proportional to the sum of all EPSPs and IPSPs (figure 2.4). Note that at that point amplitude coding of information is converted back to frequency coding.

In the rest of this section the relation between the neural information transfer and the EEG is outlined following [Zschocke, 1995].

The ion flow related to EPSPs and IPSPs causes potential fluctuations in the extra cellular space which are commonly referred to as cortical field potentials. These potential fluctuations exhibit a dipole structure which is explained here only for an EPSP. (The explanation for IPSPs is analogous):

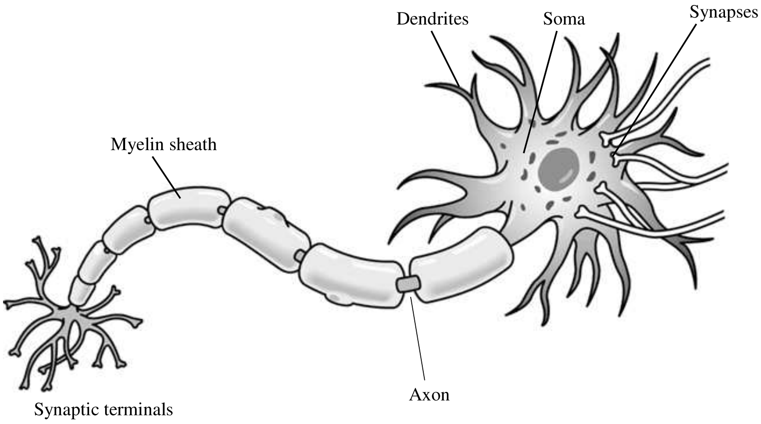


Figure 2.3: Main components of a neuron. At the synapses information between neighboring neurons is exchanged. In this figure the axon is myelinated as in most cases which is a mean to speed up the signal transfer.

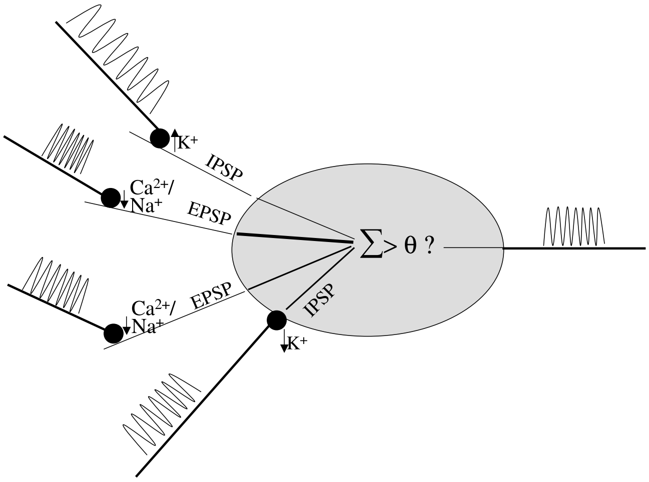


Figure 2.4: Information transfer between neurons. The thickness of the lines in the soma symbolizes the signal amplitude. See text for explanation.

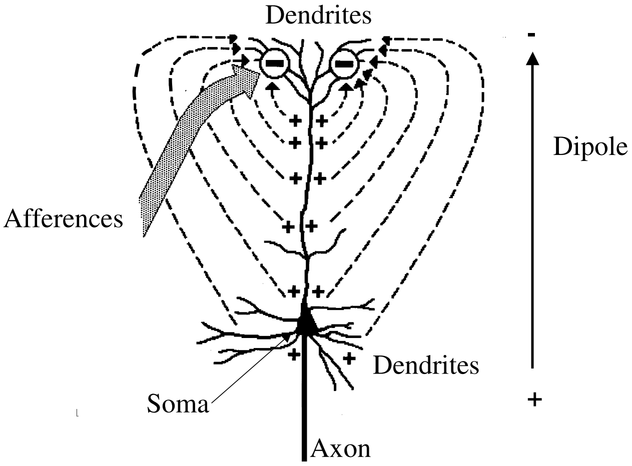


Figure 2.5: Dipole structure of cortical field potentials for a single neuron (a pyramid cell). With modifications from [Zschocke, 1995]

An EPSP is caused when at the subsynaptic membrane a larger quantity of positive ions moves in the post-synaptic neuron as explained above. This makes the outer part of this membrane segment appear more *negative* (because of the lack of positive ions) than all other parts of the membrane of the same neuron where the number of positive ions can even increase due to capacitive effects. Thus we have a small negative pole at the subsynaptic membrane segment and a relatively large positive pole at all other membrane segments as shown in figure 2.5 for the example of a pyramid cell. Pyramid cells are mainly located in the human cortex and they play an essential role for the EEG. An important characteristic of this type of cells is that they have often some very long dendrites which point to the outmost parts of the cortex.

If now a few thousands of neurons are excited synchronously *and* the corresponding dipoles point in the same direction, potential differences between the point on the scalp above these neurons and a constant reference point can be registered which are commonly referred to as EEG (figure 2.6). Note that not only cortical mechanisms, but also the thalamus (a collection of nuclei located in the cerebrum) is responsible for the synchronization of neural activity in the cortex such that potential fluctuations in the EEG can be observed. The thalamus in turn is largely influenced by the formatio reticularis which has an important function in the control of vigilance. More information about the role of the thalamus and the formatio reticularis concerning the brain activity related to different mental states and to different levels of task demand is provided in section 2.3.

Only very view neurons of the brain actually contribute to the EEG. Pyramid cells are the main sources of cortical field potentials. But only $\frac{1}{3}$ of all pyramid cells in the cortex have influence on the EEG because they have an orientation which is perpendicular to the brain surface. Only for these cells a synchronous activation can cause a superposition of dipoles whose sum is large enough so that potential fluctuations can be measured at the scalp. Such a dipole configuration is called then an *open dipole field* (the first of the four examples in figure 2.6). A *closed dipole field* which has no effect on the EEG is generated when different dipoles neutralize each other. This can happen for an ensemble of pyramid cells which receive afferences from opposite directions (the fifth example in figure 2.6), or for non-pyramid cells where one neuron receives afferences from multiple directions. Star cells which connect pyramid cells in the cortex in various ways (figure 2.7) are an example for that.

From this brief summary of the physiological processes underlying brain activity and in particular

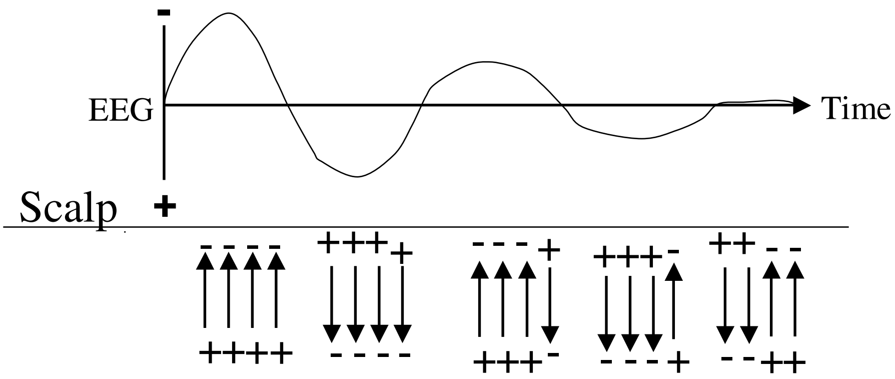


Figure 2.6: Synchronized neuron activity causing potential differences which can be measured at the scalp. The dipoles are depicted by the arrows. Note that by convention positive potential differences correspond to amplitudes with *downward* orientation and negative potential differences to amplitudes with *upward* orientation.

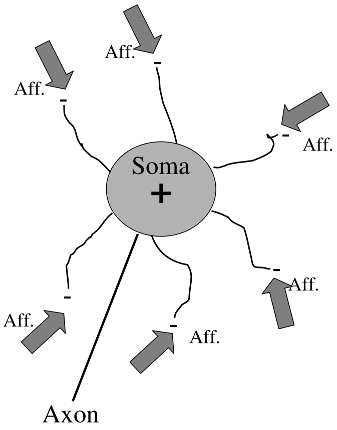


Figure 2.7: A closed dipole field generated by a star cell. Aff.: Afferences causing a negative pole at the outer end of the dendrites.

of those processes generating the EEG, we conclude that most brain activity remains hidden to this monitoring technique. Therefore EEG signals are certainly not suitable to make fine grained inferences about neural processes in the brain. In section 2.3 it will be explained however, that nevertheless a lot of information about mental states and mental task demand can be extracted from the potential differences which are measured using scalp electrodes.

2.2 Monitoring Brain Activity

After in the previous section the mechanisms related to brain activity have been explained, several techniques for monitoring this activity shall be reviewed in this section. The following monitoring techniques are commonly applied in the medical domain:

- Electroencephalography (EEG)
- Magnetoencephalogtaphy (MEG)
- Functional Magnetic Resonance Imaging (fMRI)
- Functional Near-Infrared Spectroscopy (fNIRS)
- Single Photon Emission Tomography (SPECT)
- Proton Emission Tomography (PET)

They can be classified using several characteristics:

- Intrusiveness
- Spatial resolution
- Temporal resolution
- Physiological parameter which is monitored
- Resources required for operation of the monitoring device
- Applicability as portable device

In this section the key ideas of the different monitoring techniques are explained and compared according to the characteristics enumerated above. Since the experimental part of this work is focused on EEG, this technique is reviewed in greater detail, following mainly [Zschocke, 1995] and [Bolz and Urbaszek, 2002] if no other reference is mentioned explicitly. A good description of all other monitoring techniques can be found in [Dössel, 2000], except functional near infrared spectroscopy which is described for example in [Izzetoglu et al., 2004].

2.2.1 The Electroencephalogram (EEG)

The EEG which can be recorded at the scalp has amplitudes between $0\mu V$ and $80\mu V$ and a frequency range between 0Hz and 80Hz [Schmidt and Thews, 1997]. For several reasons the potential differences which can be measured between two points of the scalp are very different from those which could be measured when electrodes were implanted directly in the brain, i.e. when the activity of the potential generators could be measured directly:

1. A superposition of potentials generated in different areas of the cortex is measured using scalp electrodes since brain tissue and the liquor are conductive (volume conduction, figure 2.8).
2. The amplitude of the originally generated potential differences is attenuated because of the resistive properties of the tissue between the potential generators and the electrode (e.g. liquor, skin, bone of the skull).
3. Capacities caused by cell membranes and other inhomogeneities (e.g. liquor-skull, skull-skin) between potential generators and electrodes influence the amplitude of the EEG signals as a function of their frequency as sketched in figure 2.10.

A diagram of the resistive and capacitive elements between potential generators and electrodes is depicted in figure 2.9.

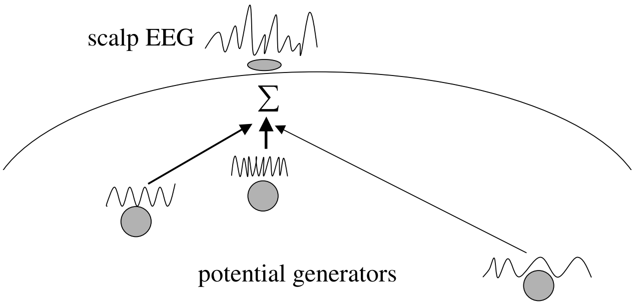


Figure 2.8: Volume conduction in the brain. Bold lines indicate a stronger impact on the measured signal, since the distance from the corresponding potential generator to the scalp electrode and thus the total resistance is smaller.

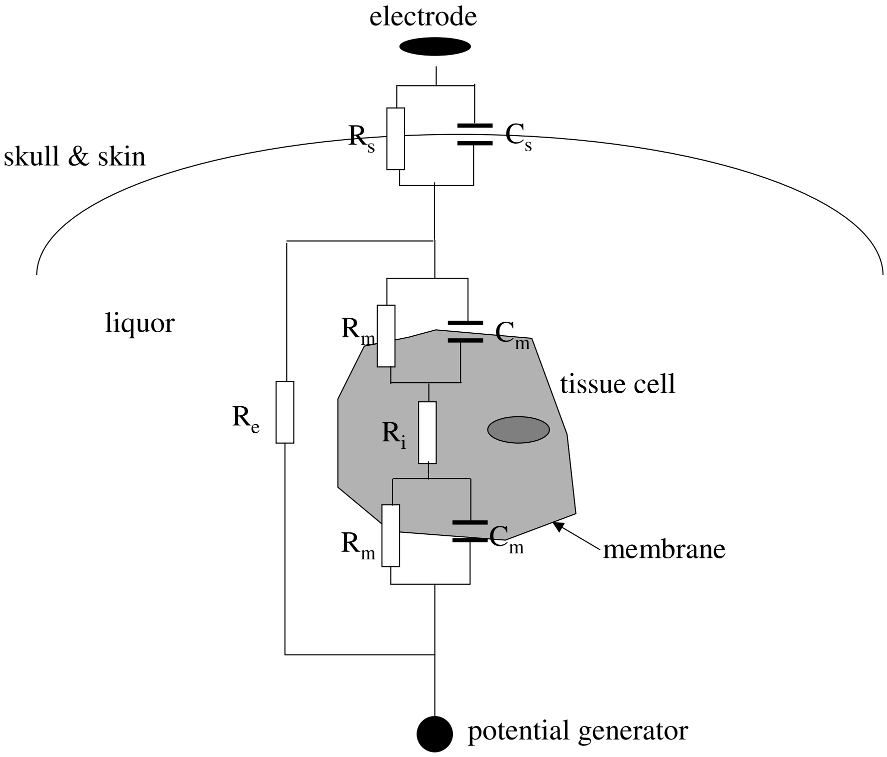


Figure 2.9: Resistive and capacitive elements between potential generators and scalp electrodes according to [Dössel, 2000] and [Bolz and Urbaszek, 2002]. For simplicity only one tissue cell is shown. R_e: resistance of the extra-cellular space (the liquor), R_i: resistance of the intra-cellular space, R_m: membrane resistance, C_m: membrane capacity, R_s: resistance of skin and skull, C_s: capacity representing all inhomogeneities at the boundary of liquor, skin and skull.

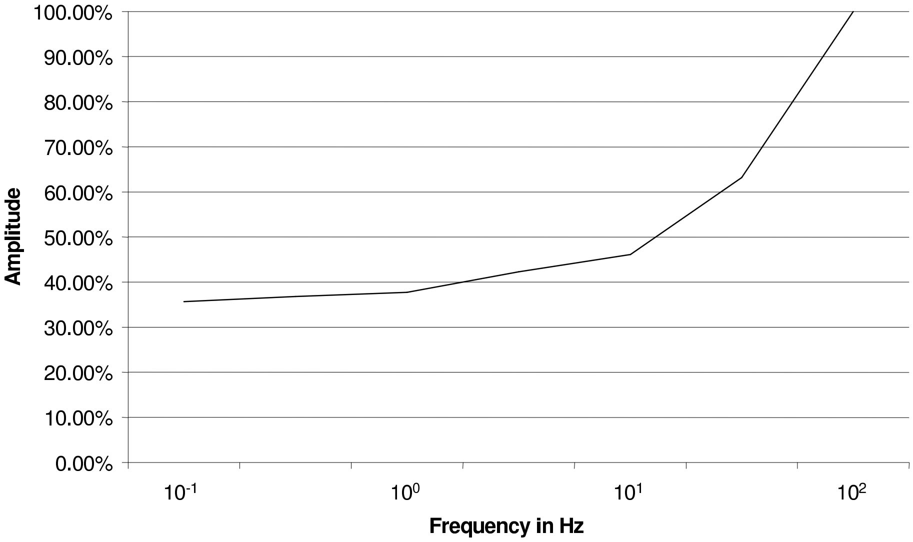


Figure 2.10: Influence of capacities caused by inhomogeneities between potential generators and electrodes on the EEG amplitude as a function of EEG frequency. Sketched using a few data points from [Meyer-Waarden, 1985]

2.2.1.1 Electrode Positions

The positions for EEG electrodes should be chosen in a way, that all cortex regions which might exhibit interesting EEG patterns are covered. For most applications this is usually the whole cortex. An internationally accepted standard for electrode placements is the 10-20 system[1]introduced in 1957 by the International EEG Federation [Jasper, 1958]. During all experiments conducted for this work electrodes were placed according to the 10-20 system.

Three anatomical reference points must be determined before the 10-20 system electrode positions can be found (figure 2.11):

Nasion: The onset of the nose on the skull, below the forehead.

Inion: The bony protuberance which marks the transition between skull and neck.

Pre-auricular reference point: Located before the cartilaginous protrusion of the acoustic meatus (the auditory canal).

Figure 2.12 shows the 19 electrode positions of the 10-20 system in their projection on the cortex. The name for a particular electrode position reflects the anatomical region of the cortex above which it is located. Fp stands for frontopolar, F for frontal, T for temporal, C for central P for parietal, O

[1] The reason for the name ”10-20 system”is that electrodes are placed at distances of 10% or 20% of the length of several connections between some reference points as described below.

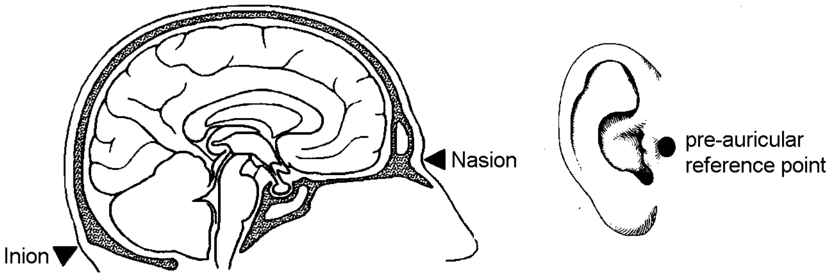


Figure 2.11: Anatomical reference points which represent the starting points for finding the electrode positions defined by the 10-20 system (with modifications from [Zschocke, 1995]).

for occipital and A for auricular. G denotes the ground electrode. Even numbers denote the right part of the head, odd numbers the left part. The following procedure can be used to find the electrode positions using the above defined anatomical references as starting points:

1. The connection NI from nasion to inion via the top of the head is measured. Beginning from the nasion, the point Fpz is placed after 10% of NI, the electrodes Fz, Cz, Pz and the point Oz are placed each after proceeding 20% of NI from the previous position. (No electrodes are located at the points Fpz and Oz but they are used to find the positions of other electrodes.)

2. The electrodes A1 and A2 are placed at the left and right ear lobes. The connection PA from one pre-auricular reference point to the other one via the electrode position Cz (which has been determined before) is measured. After 10% of PA above both reference points the electrodes T3 and T4 are placed. 20% of PA above T3 and T4 the electrodes C3 and C4 are placed.

3. The connection FO_1 between the point Fpz and Oz is measured via the position T3. Starting at Fpz the electrode Fp1 is placed after 10% of FO_1, the electrodes F7, T5 and O1 are placed each after proceeding 20% of FO_1 from the previous electrode. In an analogous way the connection FO_2 between Fpz and Oz via T4 is measured and the electodes Fp2, F8, T6 and O2 are placed accordingly.

4. The electrode F3 is placed at half the distance between F7 and Fz, the electrode P3 is placed at half distance between T5 and PZ. In the same fashion the electrodes F4 and P4 are placed on the other side of the head.

When recording EEG with commonly available EEG-caps or other standard recording devices, the procedure for determining the electrode positions needs not to be repeated prior to each data acquisition session. However the accuracy of electrode placements often suffers when flexible EEG-caps are used. Although usually different caps exist for different head circumferences, it can not be guaranteed that for each individual electrodes are placed *exactly* at the positions defined by the 10-20-system. Therefore in clinical EEG recordings the whole procedure as described above is usually carried out prior to each session which is extremely time consuming but which guarantees on the other hand exact electrode positions. For EEG recording in the context of user state identification and task demand assessment, it is important that electrodes can be attached very quickly. For that reason either flexible EEG-caps or other recording devices which are more comfortable to wear and even easier to attach (see also section 5.1) are suitable for that purpose.

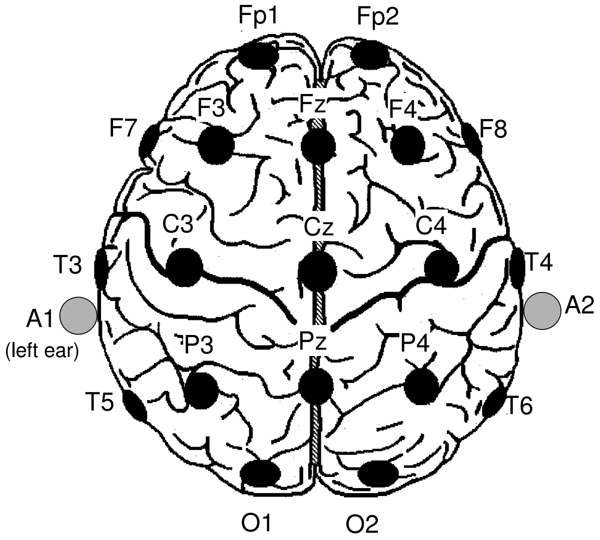


Figure 2.12: Electrode positions of the 10-20 system (with modifications from [Zschocke, 1995])

The 19 electrodes of the 10-20 system are sufficient for most clinical purposes. Due to improved amplifier technology, for special applications much more electrodes (up to 128) can be used, which are usually placed according to various non-standardized conventions in each EEG laboratory. Note that also extensions of the 10-20 system with a couple of more electrodes exist which are accepted as quasi-standards.

2.2.1.2 Electrode Montage

The term ”electrode montage” refers to the various ways in which the signals from different electrodes can be combined before amplification [Niedermeyer and da Silva, 1987]. Two generally different methods for connecting EEG electrodes to the amplifier can be distinguished: bipolar electrode montage and common-reference electrode montage. In both cases the EEG is measured *differentially*, i.e. the *difference* between the potential of two electrodes (or between one electrode and the averaged potential of a set of electrodes) is measured.

For a common-reference montage the EEG signals represent the difference of the potentials of each electrode and one or more reference points which all electrodes (or at least a subset of electrodes) have in common. There are generally two criteria for good reference points:

1. They should be far enough away from the cortex, so that no brain activity is captured with the reference electrode.

2. They should be at a position where artifacts introduced by other physiological processes are low.

Therefore a reference at the hand or at the legs, which would be ideal to fulfill the first criterion, is not reasonable, since large artifacts originating from the ECG would contaminate the signal in that case.

When only one reference point is used for all electrodes or for each subset of electrodes (natural reference), electrode A1 is often used as reference for the EEG electrodes over the left half of the cortex and electrode A2 for those over the right half. Another common choice is to use electrode Cz as reference for all other electrodes. As sets of reference electrodes whose potential is averaged before calculating the potential difference between EEG electrode and references the electrodes A1 and A2 or even all electrodes can be used (technical reference). Although the latter case has some advantages since artifacts introduced by one electrode from the reference set have a lower impact, it has the disadvantage of a spatial low pass. The activity of each electrode is mapped with a small amplitude to all other electrode channels because each electrode is part of the reference set.

The idea of bipolar electrode montages is that always the difference between neighboring electrodes is taken. This has the large disadvantage that not the ”real” EEG, i.e. an approximation of the dipole configuration below each electrode is measured but the difference between the dipole configurations below adjacent electrodes. In the medical domain bipolar recordings are interesting however to detect the specific location of particular processes: if one process takes place below two electrodes, there will be a zero signal for the recording of exactly these two electrodes. A special case of a bipolar electrode montage is the laplacian montage. Here all neighbors of one electrode (up to 8 electrodes) are included in the set of references. This is advantageous for some special clinical applications.

Finally ground electrodes are required to reduce noise from the AC power lines which is present in the whole body and thus also contaminates the measured signals. Intuitively speaking the ground electrode gives the noise signal the possibility to flow out of the body against a comparatively small resistance. Therefore the noise takes preferably the path via the ground electrode and not via other electrodes so that the measured signal is contaminated less. (It is also possible to give a signal on the ground electrode which is inverse to the noise signal, so that both signals annihilate. This is an appropriate mean for even more effective noise reduction.) In the 10-20 system the position for the ground electrode is close in front of the position Fz (see figure 2.12). However also other positions for ground electrodes are possible as long as they capture as little muscular activity as possible, since this would introduce artifacts in all electrode channels.

2.2.1.3 Electrodes

Electrodes are the most critical components which determine the EEG signal quality. It is their task to mediate between the ion based transport of electrical charge in the tissue and the electron based charge transport in copper wires which lead to the amplifier. In the tissue an electrolyte is responsible for the conduction of electricity. This is typically $NaCl$ which dissociates to Na^+ and Cl^- ions in a watery solution. The electrode however gets only in contact with the skin which is relatively dry and usually not permeable for the electrolyte in the body. Therefore it is essential to use an additional electrolyte (in form of electrode paste) which permeates the skin and thus establishes a connection between body electrolyte and the metal phase of the electrode. As a consequence the resistance and the capacity of the skin (R_s and C_s is figure 2.9) decrease by some orders of magnitude when using electrode paste. Only this makes EEG recording possible.

The interaction between metal phase and electrolyte phase mainly determines the properties of an electrode. Due to different electro-chemical properties of both phases electrons tend to move preferably from metal to the electrolyte or vice versa, until a certain potential difference between both phases is reached. This potential difference is called electrode potential or equilibrium potential since chemical and electrical potential differences between electrolyte and metal are in the equilibrium here. Because of the electrode potential polar water molecules (which are contained in the electrolyte) attach themselves at the boundary between both phases so that a double layer (ions and water molecules) is formed, the Helmholtz double layer. This double layer is not easily permeable for electrons and therefore it represents a capacity (figure 2.13). Nevertheless an excess of electrons at either side of the

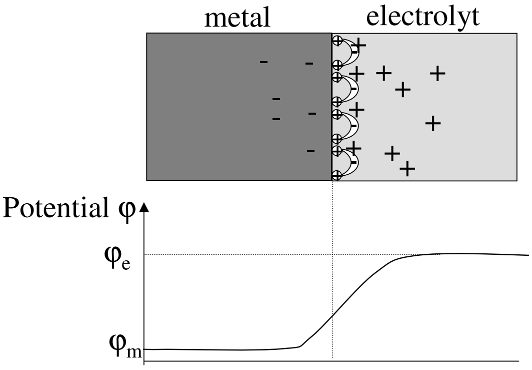


Figure 2.13: The Helmholtz double layer. Since the potential of the metal φ_m is smaller than the electrolyte potential φ_e, the hydrogen molecules of the water dipoles point the metal side.

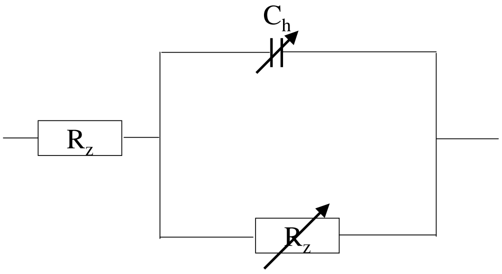


Figure 2.14: Electrical diagram of an electrode. C_h and R_f represent the capacitive and resistive parts of the impedance at the boundary between electrolyte and metal phase (i.e. at the Helmholtz double layer). They vary with different frequencies. R_z is the resistance of the electrolyte.

double layer leads besides a capacitive current also to a resistive current, i.e. a real transfer of electrons between both phases, resulting in the electrical diagram of the electrode as shown in figure 2.14. A DC source parallel to the capacity C_h and the resistance R_f, representing the electrode potential which varies with the current required for the measurement should be included in the chart as well. For simplicity it has been omitted here however since its influence can be modeled by changes of the resistance R_f, which is a valid assumption for the further explanations in this section. R_z represents the resistance of the electrolyte.

From the chart in figure 2.14 it can be seen that if R_f is high, most charge is transferred capacitively and as a consequence signals with lower frequencies are attenuated significantly due to the properties of a capacity. This implies that R_f should be kept as low as possible. Furthermore C_h and R_f vary with the frequency and with the current density of the measured AC signal as sketched in figure 2.15. It can be seen that especially for higher current densities impedances vary a lot with the signal frequency which largely distorts the original signal. (In particular R_f becomes higher when a higher current density is required for the measurement since only a limited amount of electrons can pass the double layer per unit of time.) Therefore the current density required for the measurement should be kept as low as possible which is one important design criterion for the amplifier.

We conclude that the ideal electrode should have a low R_f so that little of the charge exchange

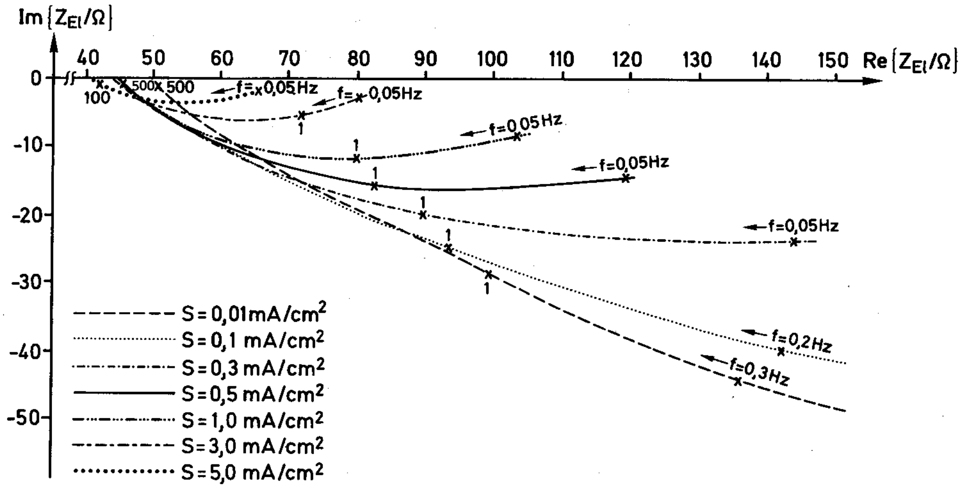


Figure 2.15: Behavior of the complex electrode impedance (the real part corresponds to the resistive component, the imaginary part to the capacitive component) for different current densities and frequencies (from [Meyer-Waarden, 1985]).

takes place capacitively and the amount of electrons which can pass the double layer at a time is high. This assures that the whole EEG frequency range is distorted only little by the properties of the electrode.

Two types of electrodes can be distinguished. Polarized (or reversible) electrodes and non-polarized (or irreversible) electrodes. Polarized electrodes are usually made of precious metals, mostly gold or of stainless steel. For the transfer of electrons between metal and electrolyte phase reductions and oxidations must take place which require an high excess of electrons at either side. That means that the resistance R_f is comparatively high and therefore the frequency range of the original signal is distorted largely by the electrode properties. Non-polarized electrodes which are made of silver (*Ag*) with a thin silver-chloride (*AgCl*) layer on top (*Ag*/*AgCl* electrodes) are state of the art since their resistive component is much lower. (Polarized electrodes are used only for special applications, mostly signal registration inside the body, where *AgCl* would be toxic.) The idea of non-polarized electrodes is that charge is never transfered directly between metal and electrolyte but that the *AgCl* serves as mediator. The chemical reactions related to the charge transfer require much less activation energy than the oxidation and the reduction in case of the polarized electrodes. Therefore a smaller excess of electrons at either side is required to initiate a real electron transfer, i.e. R_f is considerably lower compared to polarized electrodes and it stays almost constant for a large frequency range, also for higher current densities. The effect of that can be seen in figure 2.16: For the *Ag*/*AgCl* electrodes, the signal amplitudes are only attenuated significantly for frequencies below 1Hz since only for very low frequencies R_f becomes too large. When stainless steel (SS) electrodes are used R_f is already large for higher frequencies which can be seen in a stronger signal attenuation compared to *Ag*/*AgCl* electrodes.

The principle of non-polarized electrodes is depicted in figure 2.17. In case of an excess of Cl^- ions in the electrolyte, Ag^+ ions are drawn from the *AgCl*. The resulting lack of Ag^+ ions in the *AgCl* is covered by new silver ions which fall out the solid silver and leave a free electron there which then ”moves” to the amplifier. Too many K^+ ions in the electrolyte draw Cl^- ions from the *AgCl* so that the excess of free Ag^+ ions remains which are then integrated in the solid silver. During this integration process an electron is drawn from the current source.

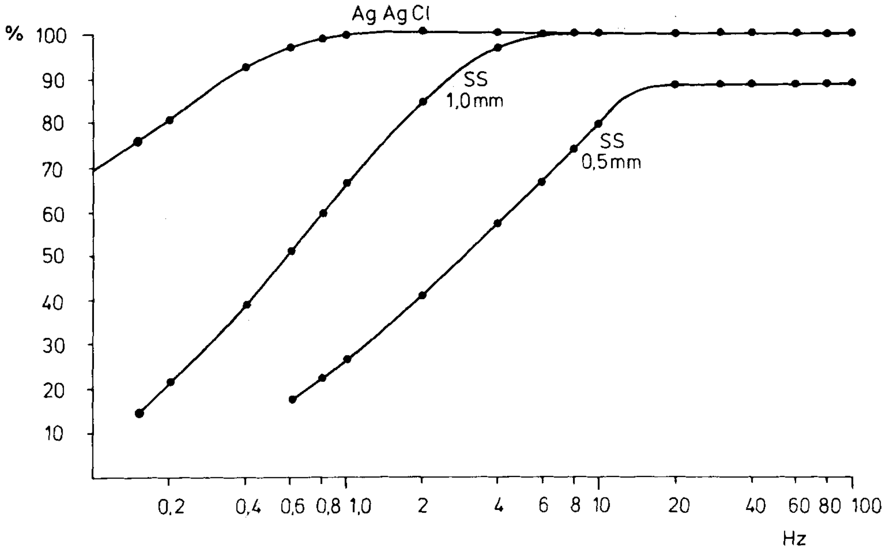


Figure 2.16: Signal attenuation for varying frequencies for *Ag*/*AgCl* electrodes and stainless steel electrodes with 1.0mm (SS 1.0mm) and 0.5mm (SS 0.5mm) tip size, which are usually applied for intracranial recordings. With modifications from [Niedermeyer and da Silva, 1987].

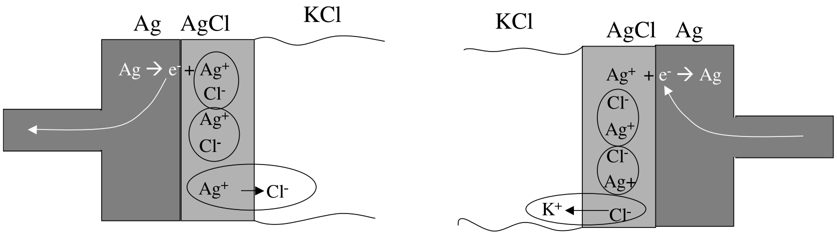


Figure 2.17: Principle of non-polarized *Ag*/*AgCl* electrodes with *KCl* electrolyte. Left: excess of Cl^- ions in the electrolyte, right: excess of K^+ ions in the electrolyte.

2.2.1.4 EEG Amplifiers

In section 2.2.1.2 is was mentioned that EEG signals are usually measured differentially, i.e. the amplifier outputs the difference between the signal from the electrode which captures the actual EEG (henceforth referred to as EEG electrode) and the signal from the reference electrode(s). The EEG electrode is usually connected to the positive input of the amplifier (therefore the corresponding signal is referred to as U_p in the following), the reference electrode is connected to the negative input (therefore its signal is referred to as U_n in the following). The reason for the application of this measurement technique is that it helps to eliminate the so called common mode signal U_{gl}. This is mainly a 50Hz or 60Hz signal which is injected capacitively or inductively in the body. Its amplitude is at least one order of magnitude higher than the signal of interest, however it has the same phase and amplitude at all positions of the body. Thus we have

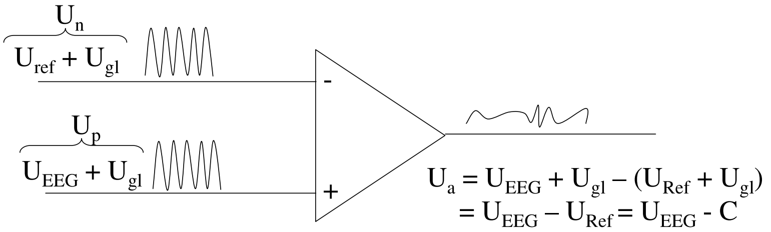

U_n
U_ref + U_gl
U_p
U_EEG + U_gl
-
+
U_a = U_EEG + U_gl – (U_Ref + U_gl)
= U_EEG – U_Ref = U_EEG - C

Figure 2.18: Principal of differential EEG measurement. See text for the explanation.

$$U_p = U_{EEG} + U_{gl} \text{ and } U_n = U_{Ref} + U_{gl}$$

where U_{EEG} is the signal from the EEG electrode without the common mode noise and U_{Ref} the signal from the reference electrode without the common mode noise. Using differential measurements the output signal U_a ideally does not contain the common mode noise U_{gl} anymore (figure 2.18). When it can be assumed furthermore that the signal U_{Ref} is mostly constant ($U_{Ref} = C$), the output of the amplifier U_a differs only by an offset from the potential fluctuations at the electrode of interest.

Due to tolerances during the manufacturing of the electronic components of the amplifier the common mode signal is never rejected completely. Therefore U_a usually evaluates to

$$\begin{aligned} U_a &= \alpha_1 U_p - \alpha_2 U_n \\ &= \alpha_1 U_{EEG} - \alpha_2 U_{Ref} + (\alpha_1 - \alpha_2) U_{gl} \end{aligned} \tag{2.1}$$

It can be seen that U_{gl} is not contained in U_a only if $\alpha_1 = \alpha_2$, i.e. if the signals U_p and U_n are amplified exactly equally which is never the case for real amplifiers. The common mode rejection ratio (CMMR) is an accepted measure to quantify the ability of the amplifier to reject the common mode signal:

$$CMMR = \frac{\text{amplification of the EEG signal}}{\text{amplification of the common mode signal}} \approx \frac{\overbrace{\frac{1}{2}(\alpha_1 + \alpha_2)}^{\alpha}}{|\alpha_1 - \alpha_2|}$$

The latter approximation is valid for small values of $\alpha_1 - \alpha_2$ compared to α_1 and α_2.

Acceptable CMMRs must be larger than 100000:1 (100 dB). For an amplification factor of $\alpha = 100$ this corresponds to a tolerance of $\frac{|\alpha_1 - \alpha_2|}{\alpha} = 0.1\%$. Note that the common mode signal is about 1000 times larger than the EEG signal. Therefore the signal to noise ratio is only 1 for a CMMR of 1000:1 (60 dB) while it is 100 for a CMMR of 100000:1.

A second important issue for the design of EEG amplifiers is that the input resistance has to be very high since this assures that the current density for the measurement is kept low. (The lower the resistance R in a circuit the higher the current I which flows for a certain potential difference U following Ohm's law $U = R \cdot I$.) A low current density guarantees that the original signal is distorted as little as possible by the electrode properties as explained in section 2.2.1.3.

Furthermore a high input resistance of the amplifier makes the impedances of skin and electrodes negligible, which is important to obtain a good common mode rejection: For two different measurement points these resistances are usually not symmetric. If they played a large role for the overall

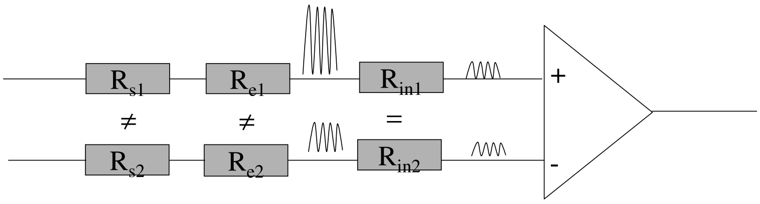


Figure 2.19: Principle of voltage adaptation. R_{s1} and R_{s2} are the skin impedances, R_{e1} and R_{e2} the electrode impedances and R_{in1} and R_{in2} the input impedances of the amplifier. The average signal amplitude for both channels becomes approximately equal, only after the input impedances.

impedance, a consequence of this asymmetry would be that the common mode signal would have different amplitudes for both channels which would decrease the CMMR since the common mode signal could not be eliminated by differential measurements. Only if the electrode and skin impedances are negligible low compared to the amplifier input resistances (which are equal for both channels) the voltage difference between both channels becomes sufficiently small. This principle, commonly known as voltage adaptation, is depicted in figure 2.19. Another advantage of voltage adaptation is that the influence of the frequency dependency of the skin and electrode resistances is reduced since they are too small to distort signal amplitudes significantly.

Since a high input resistance can not be combined with a high amplification for physical reasons, a resistance converter has to be used before amplification. The resistance converter has the following properties: It has a high input resistance, it already rejects partly the common mode signal and it allows the use of an amplifier with a low input resistance for further processing without modifying the frequency response of the signal.

Finally an EEG amplifier usually contains a low pass and a high pass filter. The high pass filter is necessary to eliminate offsets in the signal, mainly produced by varying impedances of the skin or the electrodes (e.g. due to sweating or too little electrode gel). It is usually placed *after* the actual amplifier, since the amplifier itself can also be the source of offsets due to manufacturing tolerances. A low pass filter is particularly important for digital EEG devices in order to avoid aliasing artifacts during sampling. But also high frequent artifacts originating from electronic devices in the surrounding or from muscular movement are eliminated using low pass filtering. Therefore the low pass filter is mostly placed *before* the actual amplifier. Figure 2.20 depicts all main components of a common EEG amplifier, consisting of the low pass and the high pass filter, the resistance converter and the actual amplifier. Note that a real EEG amplifier contains many additional components which prevent the flow of too high currents in the patient's body. Their description is however beyond the scope of this thesis.

2.2.1.5 Artifacts

Generally two categories of artifacts can be distinguished in EEG measurements: biological artifacts which are caused by the recorded subject and technical artifacts caused by the EEG recording device.

The sources of many biological artifacts are dipoles originating for example from muscular activity which are much stronger than the EEG related dipoles. A superposition of both types of dipoles causes artifacts in the signal which are often characterized by large peaks or fluctuations of a particular morphology (figure 2.21). Sometimes however they can hardly be distinguished from the actual EEG.

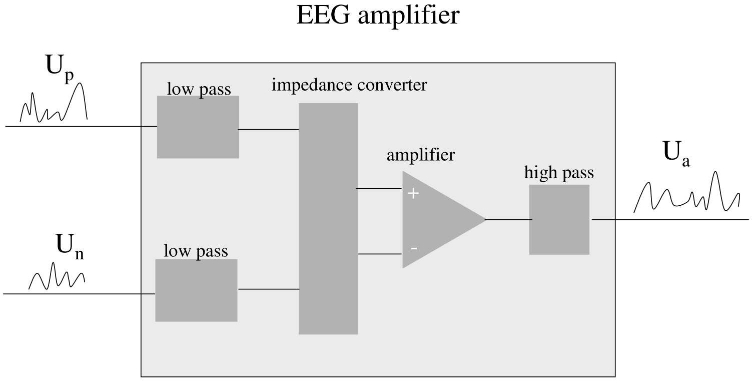


Figure 2.20: Main components of an EEG amplifier.

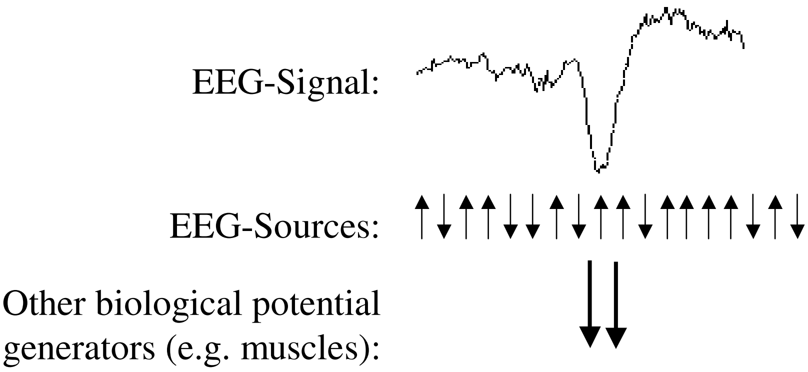


Figure 2.21: Generation mechanism of biological artifacts caused by large dipoles which are superimposed on the EEG related dipoles.

Other biological artifacts influence the contact between skin and electrode or the electrical properties of the medium between potential generators and electrodes.

The following processes or activities related to the recorded subject can cause biological artifacts.:

- The neurons in the retina generate dipoles. During vertical eye movements and also during eye blinks the eye moves up- and downwards which changes the direction of these dipoles. This introduces saw tooth shaped artifacts in the EEG signal. Artifacts introduced by horizontal eye movements are more rectangular shaped.
- The dipole caused by the electrical activity of the heart is one order of magnitude larger than the dipoles caused by neural brain activity. Peaks representing the QRS complex of the ECG signal can therefore often be seen in the EEG signal at frequencies of about 1Hz.
- Sensory neurons of the tongue also generate dipoles. Therefore tongue movements can be seen as slow and irregular fluctuations in the EEG.
- Potential differences caused by facial muscles introduce peaks in the EEG.

- If an electrode is placed directly above a blood vessel in the scalp, its pulse changes the electrical properties of the medium between cortex and electrode rhythmically. Furthermore it can cause small electrode shifts. Both effects result in resistance changes between potential generator and electrode and thus in periodic oscillations of the signal.
- The state of the subjects hair (greasy, treated with hairspray etc.) can influence resistance between skin an electrode and thus cause a change of amplitudes across sessions.
- A lot of psychological and physiological effects (e.g. the increase of physical or mental task demand) can influence the activity of perspiratory glands resulting in resistance changes between skin an electrode and therefore in changes of the EEG amplitude. Note that for the assessment for task demand such artifacts might possibly be even of advantage.
- Sweat or too much conductive paste can cause shortcuts between electrodes which leads to a strong decrease of signal amplitudes.

The following important technical artifacts can be distinguished:

- Broken or dirty electrodes can cause sudden large potential changes, since the Helmholtz double layer becomes unstable.
- Dirty or corroded contacts of cables or even broken cables can cause a decrease of the signal amplitude or even a zero signal.
- Badly attached ground electrodes or a too strong asymmetry between the impedances of EEG and reference electrodes (due to dirt or corrosion) can be detrimental for the CMMR.
- Depending on the design of the amplifier noise from high frequent electromagnetic fields (e.g. radio waves) can be mapped partly to lower frequencies of the signal which are displayed in the EEG.
- Persons next to the subject which is recorded can charge themselves electrostatically. Then electricity which is transfered inductively to the recorded subject with large voltages can cause artifacts of variable shape in the signal.

Note that technical artifacts and also some biological artifacts can be generally avoided when the recording equipment is operated correctly and the subject to be recorded is prepared carefully. Other artifacts are inevitable.

2.2.2 The Magnetoencephalogram (MEG)

Similar to the EEG the MEG also measures correlates of the neural brain activity. However no electrical potential differences but the magnetic flow caused by cortical field potentials is measured here. Since the magnetic field which corresponds to an electrical field is perpendicular to this electrical field, the MEG is most sensitive for dipoles with a tangential orientation to the surface of the head while the EEG is most sensitive for dipoles which are oriented radially from the center to the surface of the head.

The MEG has a high temporal resolution which is comparable to the EEG and its spatial resolution is even better. Furthermore MEG sensors require no direct contact with the skin in contrast to EEG electrodes. The most important drawback of the MEG is however that highly sensitive magnetometers

(SQUIDs, i.e. super conduction quantum inference devices) must be used for its registration, since the signals which can be measured at the scalp are extremely weak. SQUIDSs require a lot of space, they are expensive and they must be cooled near the absolute zero temperature to exhibit their superconducting properties. Therefore MEG recording with portable devices is impossible. Finally the MEG suffers from the same problem as the EEG that only activity from a very small fraction of neurons can be measured at the scalp (see section 2.1.2). Since EEG and MEG measure complementary signals as mentioned above, both modalities are often combined for clinical purposes.

2.2.3 Functional Near-Infrared Spectroscopy (fNIRS)

Functional Near-Infrared Spectroscopy (fNIRS) is a technique which can be used to monitor the concentration of oxygenated hemoglobin (Oxy-Hb) and deoxygenated hemoglobin (Deoxy-Hb) in the blood. Oxy-Hb binds oxygen molecules to transport them to regions where they are consumed while Deoxy-Hb does not bind any oxygen. Since neurons need more oxygen when more neural signals have to be processed, the information about concentration of both hemoglobin types allows inferences about brain activity: A larger concentration of Oxy-Hb in a certain area indicates a higher neural activity in this area.

The underlying principle of fNIRS is that biological tissues are comparatively transparent to near infrared light (wavelengths between 700 and 900 nm) but hemoglobin in the blood reflects this near infrared light at certain wavelengths. However light intensity is not only reflected but also absorbed and scattered by hemoglobin. Since the absorption and the scattering properties of Oxy-Hb and Deoxy-Hb are different, they reflect the same light impulses with different intensities. This information can be used to calculate the relative changes of the concentration of Oxy-Hb and Deoxy-Hb at a certain position compared to a baseline.

An apparatus for fNIRS usually consists of one or more near infrared light sources and a detector array. Both can easily be mounted on the forehead. Therefore fNIRS is well suited to construct portable devices for monitoring brain activity. Compared to EEG they have the advantage that no conductive paste between sensors and skin is needed, however sensors can only be placed at positions without hair. Furthermore the information provided by fNIRS is much less detailed than EEG data: For each sensor only an estimate of the current brain activity for the cortex region under this sensor is obtained. While this may be suitable for applications like assessment of task demand, EEG is indispensable for clinical applications since inferences about underlying neural processes can be made from the morphology and the frequency of the EEG signals. The temporal and spatial resolution of fNIRS is comparable to EEG.

2.2.4 Functional Magnetic Resonance Imaging (fMRI)

Standard magnetic resonance imaging measures the response of biological tissue when exposed to a strong magnetic field and electromagnetic high frequency pulses perpendicular to this field. For functional magnetic resonance imaging this technique is modified so that the hemodynamics related to neural activity in the brain can be measured.

Similar to fNIRS this monitoring technique makes use of differences between Oxy-Hb and Deoxy-Hb, which exhibit a different behavior in magnetic fields. For regions with a higher concentration of Deoxy-Hb the fMRI response (i.e. the contrast of the obtained image) is slightly weaker compared to regions with a higher Oxy-Hb concentration. This difference is extremely small so that in order to detect it, data obtained from several repetitions of stimuli causing a particular Oxy-Hb and Deoxy-Hb distribution must be averaged (which increases the signal to noise ratio). This method is also referred to as BOLD contrast imaging, i.e. Blood Oxygen Level Dependent contrast imaging.

The precise relationship between neural signals (e.g. EEG signals) and the BOLD response is under active research. In different studies a correlation between EEG-α activity and blood oxygenation has been found (see for example [Laufs et al., 2003] and [Martinez-Montes et al., 2004]). However models which relate other types of neural activity to brain metabolism do not exist yet to our current knowledge.

The temporal resolution of fMRI is much worse compared to the imaging techniques mentioned above, since it takes 2-5 seconds to produce a single image. However a much better spatial resolution can be obtained: slice images are produced in a way that three dimensional models of the brain activity can be constructed with voxels as small as three millimeters length on each side. Furthermore a lot of brain activity information from central brain regions can be visualized here which remains hidden to EEG, MEG or fNIRS.

An MRI device is very large and expensive and it is not suited at all to record people's brain activity in every-day situations. The magnetic pulses produce a lot of noise during recording and even slight motion causes artifacts which can hardly be corrected. For clinical purposes and the study of the brain's function fMRI is however well suited.

2.2.5 Single Photon Emission Computer Tomography (SPECT)

Single Photon Emission Computer Tomography (SPECT) also produces slice images of the brain, similar to fMRI. While fMRI detects however differences between oxygenated and deoxygenated hemoglobin in the blood, SPECT provides information about the amount of blood in particular brain regions and the temporal distribution of a certain blood quantity (or a certain drug) in the brain which has previously been marked with radioactive tracers (radiopharmaceuticals).

The key idea of SPECT is that radiopharmaceuticals emitting gamma rays are injected in a blood vessel prior to the examination. These radioactive tracers are transported then via cardiovascular system so that their concentration is particularly high at regions where more blood is required. Brain activity has been proved to be particularly high in regions with high blood flow [Schmidt and Thews, 1997], so that the gamma activity from a certain brain region can serve as indicator for activity in this region. Note that blood flow is highly correlated with the concentration of Oxy-Hb in the blood [Schmidt and Thews, 1997] which is assessed by fNIRS and fMRI.

Gamma activity is measured using a gamma camera rotating around the patient. Thus projections of the gamma activity for a number of angels (usually 100-200) are obtained. Methods which are commonly used for standard x-ray computer tomography (e.g. filtered back projection) are applied then to generate a slice image of the gamma activity.

The application of radiopharmaceuticals is harmful to the patient in a similar way as x-rays. Therefore SPECT can be used only for clinical applications. Furthermore the same restrictions as for fMRI concerning the space demands of the recording unit and the mobility of the patient apply here. The temporal resolution of SPECT is slightly better than that of fMRI (in each second one image can be obtained), the voxels of the recorded volume however have lengths of about 1cm at each side.

2.2.6 Positron Emission Tomography (PET)

For Positron Emission Tomography (PET) radiopharmaceuticals are used which emit positrons instead of gamma rays. Positrons are very unstable and almost immediately after emission (after a few mm path length) a positron collides with an electron resulting in the annihilation of both particles and the

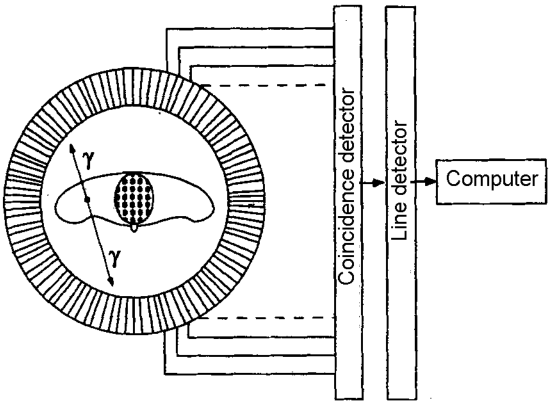


Figure 2.22: Principal of a PET system (with modifications from [Dössel, 2000]). The coincidence detector detects gamma quanta arriving almost simultaneously at opposite detectors. The line detector increments a counter for the line which connects the two detectors where the gamma quanta arrived. The values for all lines which can be interpreted as projection of the positron concentration are finally passed to a computer for reconstruction of a slice image.

emission of two gamma quanta which move in *exactly* opposite directions. A detector ring around the patient is used to detect such events, i.e. the arrival of two gamma quanta at detectors on opposite positions of the ring at the same time (figure 2.22). For reconstruction of a slice image from the detected events techniques similar to those used for SPECT or x-ray computer tomography are applied.

PET provides similar images as SPECT, however it has a better spatial resolution with voxels of about 3mm side length. However PET devices are much more expensive than SPECT devices and the production and application of radiopharmaceuticals is much more difficult. Their half-life period is usually only between 10min and 100min so that positron emitters must be applied almost immediately after their production.

2.2.7 Summary

Table 2.1 provides an overview over all described techniques for monitoring brain activity.

2.3 Physiological Correlates Mental States and Task Demand

2.3.1 Functional Cortex Divisions and and the Identification of Mental States in the EEG

Different sensory inputs are processed at different parts of the cortex. Furthermore particular cortex areas are involved in higher mental functions such as memory, control of attention, complex planing and reasoning. Figure 2.23 gives an overview over the functions of different cortex regions.

When sensory inputs are processed or when other mental processing takes place, the corresponding regions of the cortex are particularly active. We are interested now in the EEG-correlates of this activity. At this point the thalamus becomes important. Its main role is the preprocessing of peripheral sensory inputs and their forwarding to those cortex regions which are concerned with the final processing of these inputs. These are the primary sensory cortex regions and those regions which are

	EEG	MEG	fNIR	fMRI	SPECT	PET
Intrusiveness	medium: conductive paste is needed to assure good conductive properties between skin and electrodes	low: no contact between sensors and head	low: contact between sensors and head, however no gel in between is required	low: no contact between sensors and head	high: harmful substances are injected in the blood	high: harmful substances are injected in the blood
Spatial resolution	several centimeters, however at one electrode a superposition of signals from potential generators from all over the cortex is measured	electrical sources can be localized with a precision of a few millimeters, however superposition effects are observed as well	pixel side length of about 1 cm	voxel side length of about 3mm	voxel side length of 10mm - 15mm	voxel side length of 2mm - 5mm
Temporal resolution	events with time scales on the order of milliseconds can be resolved	events with time scales on the order of milliseconds can be resolved	a few seconds	2-5 seconds	1 second	1 second
Physiological parameter which is monitored	electrical activity of neurons	electrical activity of neurons	concentration of oxygenated and deoxygenated hemoglobin	concentration of oxygenated and deoxygenated hemoglobin	blood flow	blood flow
Resources required for operation of the monitoring device	little space and energy is required, little cost for recording devices and their application	huge space and energy (for cooling the SQUIDS) requirements, high cost for recording devices and application	little space and energy is required, little cost for recording devices and their application	huge space and energy (for superconducting coils) requirements, high cost for recording devices and their application	huge space requirements, high cost for recording device and medium cost for radiopharmaceuticals	huge space requirements, high cost for recording device and for radiopharmaceuticals
Applicability as portable device	yes	no	yes	no	no	no

Table 2.1: Overview over all reviewed brain monitoring techniques.

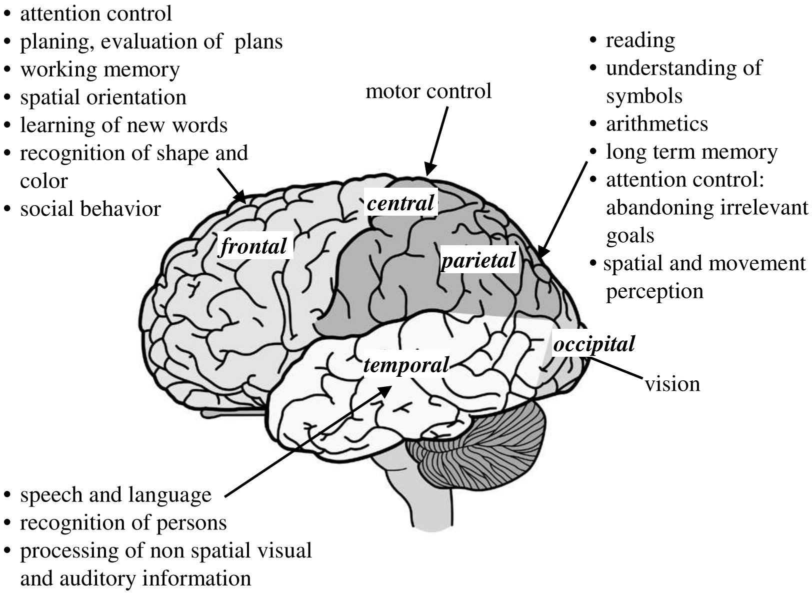


Figure 2.23: Functional cortex divisions according to [Schmidt and Thews, 1997] and [Dudel and Backhaus, 1996]. The cortex image is taken from [Scientific Learning Cooperation, 1999].

responsible for high level processing of the perceived information, e.g. the planning and initiation of responses.

The different nuclei of the thalamus can operate however in two different modes (figure 2.24), depending on the presence of sensory stimuli and on the current vigilance level:

Oscillation mode: When no or only weak sensory inputs are present for a particular nucleus and/or vigilance is low, impulses with a frequency between 8Hz and 13Hz are constantly sent to those cortex regions where this nucleus projects to (i.e. to which this nucleus sends afferences). These impulses cause a *synchronization* of neural activity in the corresponding cortex regions. Thus a rhythmic neural activity called α-rhythm is generated and in the EEG measured over these cortex regions, oscillations with a frequency of the thalamus impulses can be observed.

Transfer mode: In this mode sensory inputs are preprocessed and the preprocessed information instead of rhythmic impulses is sent to the cortex regions which are responsible for the final processing. This leads to a *desynchronization* of neural activity, i.e. to the attenuation or even the vanishing of α-activity and an increase of activity at other frequencies for the particular regions which can also be observed in the EEG.

Thus we can briefly summarize neural processes related to mental activity as follows: Given a sufficiently high vigilance level and the presence of sensory inputs, the thalamic nuclei processing these inputs switch from oscillation mode into transfer mode, i.e. they activate the corresponding cortex regions. Activation means desynchronization, so that in the EEG the power in the α-frequency band decreases for the activated regions and power in other frequency bands increases indicating that higher mental processes are taking place.

Note that mental activity can certainly take place as well without the presence of external sensory stimuli. Although the precise mechanisms of the cortex activation related to that kind of activity is still

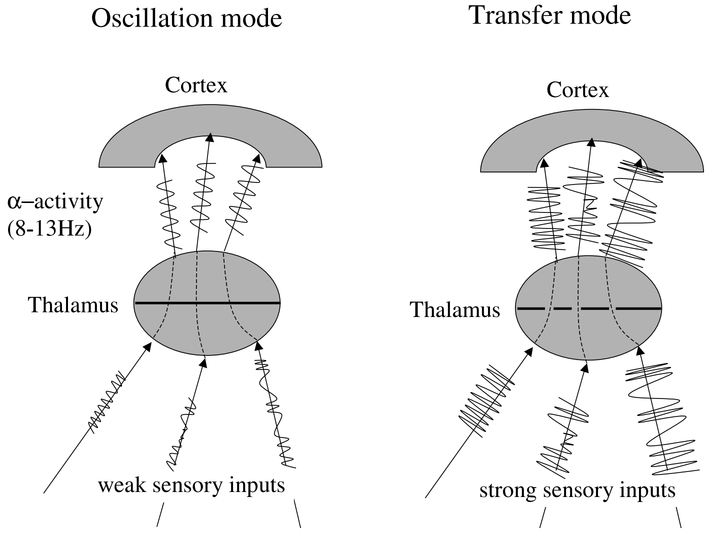


Figure 2.24: Oscillation mode (left) and transfer mode (right) of the thalamus.

subject of research, it can be taken for granted that the thalamus does not send in any cases impulses causing α-activity to the involved cortex regions. Possibly the stimuli from the cortex itself influence the thalamus indirectly via other functional entities of the brain in order to avoid that it switches to oscillation mode.

Furthermore it must be pointed out that the activation of a particular region, i.e. the desynchronization in this region never corresponds to an increase of total power in the EEG which is recorded with common amplifiers. Instead even a decrease in total power is usually observed since the amplitude of other EEG rhythms is much lower than the amplitude of the α-rhythm. The total power of the *real* EEG signal in the presence of sensory inputs, i.e. the total power of *all* potential fluctuations which can be measured at the scalp, is usually much higher than that of the *measured* EEG however, due to DC potentials which are characteristic for the processing of sensory inputs [Zschocke, 1995]. The measurement of the DC component of the signal is very difficult, since it is influenced strongly by properties of the skin and electrodes so that its fraction caused by neural activity can usually not be detected. Therefore common EEG amplifiers eliminate the DC component of the signal using a high pass filter.

Table 2.2 gives an overview over the most common EEG frequency bands according to [Schmidt and Thews, 1997]. Note that sometimes several sub bands of α- and β-activity are considered (see for example [Duta et al., 2004]). Furthermore γ-activity is seen as a subband of β-activity by some researchers ([Duta et al., 2004], [Niedermeyer and da Silva, 1987]) or it is simply referred to as 40Hz oscillation [Schmidt and Thews, 1997]. Finally a μ-rhythm between 8Hz and 13Hz which is particularly related to motor activity is often considered as well [Pineda et al., 2000]. Similarly to α-activity which vanishes in presence of *sensory stimuli*, μ-activity can be observed over the motor cortex (the central cortex lobe) only when no *motor activity* is executed, observed or imagined. Otherwise μ-activity vanishes.

Particularly the activity in γ-frequency band has been found to be related to higher mental processes. γ-activity is generated by cortical neurons themselves, i.e. no extra-cortical rhythm generators such as the thalamus are involved here. Thus a plausible hypothesis concerning the characteristics of different user states could be, that in regions which are activated during a particular state the energy of the α-frequency range decreases while the energy for higher frequency bands (around 40 Hz)

Name	Frequency range
δ	0.5Hz - 3.5Hz
θ	4Hz - 7Hz
α	8Hz - 13Hz
β	14Hz - 30Hz
γ	around 40Hz

Table 2.2: Different EEG frequency bands according to [Schmidt and Thews, 1997]

increases.

Figure 2.25 shows the spectrograms for 16 electrodes for recordings of the user states resting, listening to a talk and reading an article in a magazine. While spectrograms of single electrodes often do not differ much across all three or at least two of these user states, the information contained in the spectrograms of *all* electrodes seems sufficient to uniquely identify a particular state. This seems even possible when pooling frequencies into only two sub-bands, a lower and higher one. In some cases the differences between the spectrograms across states can be explained by the fact that regions which should be activated during a particular state (according to figure 2.23) show an increase of high frequencies and a decrease of low frequencies: Electrode P4 (which is located over the parietal cortex, see figure 2.12) shows for example much more high frequent activity during reading than during resting or listening. However this hypothesis is by far not true in all cases. Electrode T4 for example should capture activity from the areas in the temporal cortex which process speech and language. However, it shows more high frequent activity during resting than during the other states.

We conclude that for different mental states different kinds of potential fluctuations over the whole cortex can be observed, i.e. the frequency content of the signals measured at several electrodes differs. The combination of the spectrograms obtained for the signals from all these electrodes results in a pattern which can identify a user state uniquely. While it is probably difficult to explain the pattern corresponding to a particular state by a priori knowledge, the information that different patterns for different states exist, seems sufficient to attempt the application of statistical machine learning algorithms for their discrimination.

2.3.2 Task Demand, Alertness and Vigilance

In psychology and ergonomics the distinction between task demand (stress) and workload (strain) is very important. Mental task demand is defined as the degree to which an individual has to use his mental resources to fulfill a certain task while mental workload corresponds to the complexity of a task independent from the mental resources required by an individual for its execution [Bokranz and Landau, 1991].

Within this thesis we are concerned with the assessment of mental task demand, i.e. we are interested in the effect which a mental task has on an individual. This is often, but not necessarily correlated with the complexity of the task itself, i.e. with workload.

The relationship between the physiological correlates of task demand and workload (i.e. task complexity) has been analyzed in many research works, e.g. [Smith et al., 2001, Pleydell-Pearce et al., 2003, Berka et al., 2004, Izzetoglu et al., 2004]. (These works are reviewed in section 3.2 in greater detail.) For this purpose physiological data (e.g. EEG, EMG, ECG etc.) was collected during the execution of tasks which varied in difficulty (i.e. tasks with varying workload), and regression functions were trained to output indices representing the use of mental resources during task execution. In all cases data points from different *workload* levels were used as examples for different *task demand*

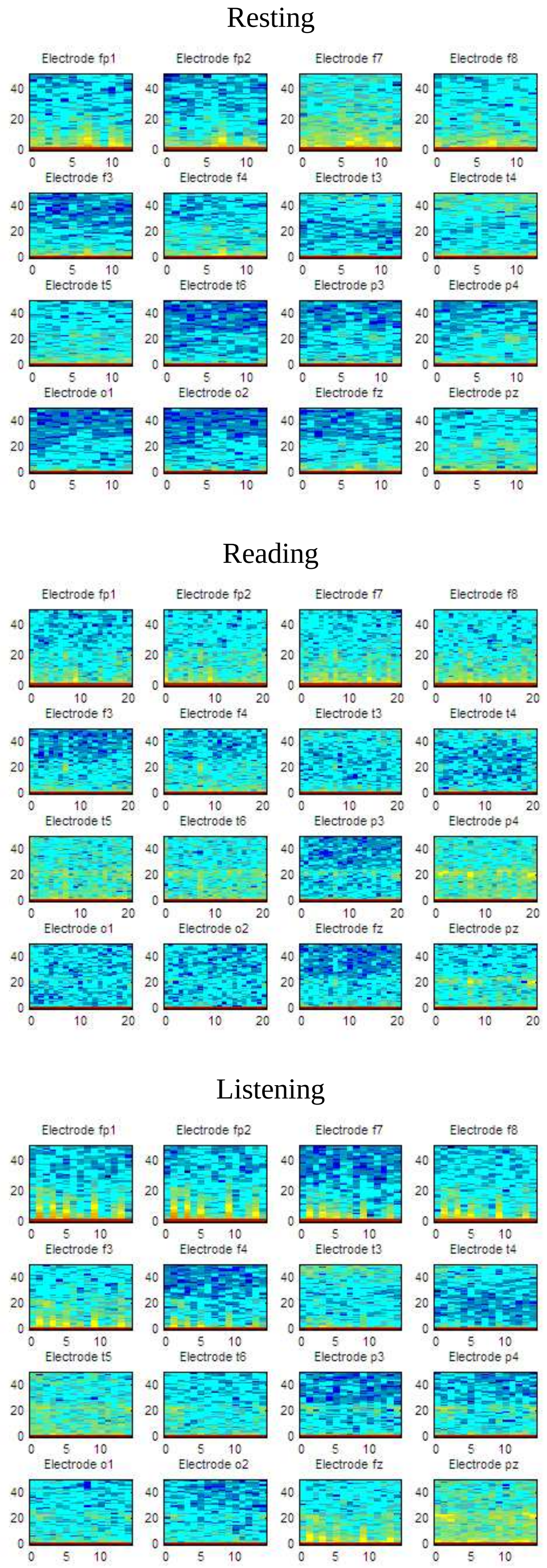


Figure 2.25: Spectrograms for 16 electrodes for the user states resting (top), reading (middle) and listening (bottom). The energy for the different frequency bands is represented on a ”temperature” scale where blue means low and red means high. The x-axis shows time in seconds, the y-axis gives the frequency in Hz.

levels, relying on the assumption that both parameters are correlated. (Often authors even use the terms workload or task load to characterize the use of mental resources, i.e. task demand according to our definition.)

In [Jung et al., 1997] and [Duta et al., 2004] (see section 3.2 for more details on these works) natural fluctuations of the availability of mental resources during tasks requiring sustained attention was examined, while workload was kept constantly. These fluctuations are commonly referred to changes in alertness or vigilance (see below). In [Duta et al., 2004] EEG data segments for different vigilance levels were obtained by manual annotation of the data, while in [Jung et al., 1997] the rate of failures to respond correctly to given stimuli was used as an indicator of vigilance. In both works also regression functions are trained which output an index reflecting the use of mental resources.

The performance of the different regression functions representing either task demand or alertness/vigilance was evaluated by correlating their outputs with different measures: [Berka et al., 2004, Izzetoglu et al., 2004] used the difficulty levels (i.e. the workload levels) of the performed task as references, [Pleydell-Pearce et al., 2003, Berka et al., 2004, Izzetoglu et al., 2004, Smith et al., 2001, Jung et al., 1997] an error measure or a performance measure which are both functions of failures to give correct responses to certain stimuli, [Duta et al., 2004] an expert rating of vigilance obtained by visual inspection of the EEG and [Smith et al., 2001] self-estimations of task demand.

We conclude that it seems difficult (or even impossible) to obtain objective measures for the use of mental resources which can be used to develop an EEG-based task demand index. Either one has to rely on subjective estimations of task demand, or task demand must be estimated indirectly relying on the hypothesis that it is well correlated with other parameters. (For some physiological parameters such as ECG or EEG this correlation is proved to be rather strong.)

While the terms alertness (in its common sense meaning) and vigilance can be treated as synonyms for this work, their relation to task demand needs some explanation.

Vigilance is defined as a physiological continuum which ranges from fully responsive (also referred to as active alertness) to sleepiness [Zschocke, 1995]. In the state of high vigilance sufficient mental resources are available and these resources are *used* either to process a huge number of stimuli or to expect stimuli attentively. Vigilance decreases either when less mental resources become available (because the individual is not relaxed but sleepy), while the amount of stimuli may remain constant, or when the number of stimuli decreases and/or they are expected with less attention. Note that stimuli can be either ”external” sensory stimuli, or ”internal” stimuli. The latter denote pure mental processes such as complex reasoning.

Since mental task demand is defined, according to our definition, as the amount of mental resources which are required for the execution of a specific task, we hypothesize the following relation between task demand and vigilance: In situations of high task demand a large amount of mental resources is required since there are many ”external” or ”internal” stimuli to be processed. That means that for a relaxed individual vigilance is high as well. When task demand is reduced, the number of stimuli decreases which means according to the above explanation a decrease of vigilance. Only if an individual is not relaxed, vigilance level and task demand level can be different, since it may happen in such cases that many mental resources are required but not available. Note that sometimes the term vigilance is even used to characterize task demand. In [Berka et al., 2004] a developed task demand index is called a vigilance index for example.

In the next section neural and EEG correlates of vigilance are described according to [Zschocke, 1995]. These should be correlates of task demand as well following the above hypothesis. Furthermore spectrograms from EEG data recorded for this work are shown, which confirm this hypothesis, since for different task demand levels they exhibit EEG characteristics corresponding to different vigilance levels. Finally some non-EEG correlates of task demand are enumerated.

2.3.2.1 Physiological Correlates of Vigilance and Task Demand

The formatio reticularis plays a crucial role for the regulation of vigilance [Zschocke, 1995]. This anatomical structure contains the ascending reticular activating system (ARAS) which controls (via the thalamus) the brain's α-activity. In a state of high vigilance (fully responsiveness or active alertness) α-activity decreases and desynchronization can be observed which is mostly connected with a cortex activity at higher frequencies which can be seen in the EEG.

This effect has already been explained in section 2.3.1 with the change of the operation mode of the thalamus (from oscillation mode to transfer mode). Reasons for this operation mode change are not exclusively the presence of sensory stimuli but also changes in vigilance level have a certain influence. In states of high vigilance the ARAS sends afferences to the thalamus so that its nuclei switch much easier into transfer mode (even when little stimuli are present). This causes that sensory inputs are transmitted more strongly (with higher impulse rate) to the cortex and that less nuclei remain in oscillation mode, so that less α-activity and more desynchronization can be observed all over the cortex. Additionally during states of high vigilance the ARAS makes the thalamus nuclei send afferences to cortex regions which are not primarily concerned with the processing of sensory inputs (diffuse projection) but which are responsible for higher mental processes and the high level processing of sensory inputs.

Thus the *degree* of vigilance is correlated correlated positively with the *amplitude* of non-α-activity and negatively with the *amplitude* of α-activity. If the reason for high vigilance is the execution of a particular mental task, such as complex reasoning to understand a talk or a presentation, especially an increase of 40Hz activity is to be expected, which is most pronounced in those regions which are mainly concerned with the required processing (see also section 2.3.1). In analogy to that, the amplitude of the α-frequency band is expected to decrease mostly in these regions, but it should be attenuated in other regions as well. For a relaxed individuals we hypothesize to observe the same phenomena when the level of task demand changes for the reasons explained above.

Based on the above explanations, the relation between the neural correlates of task demand and different user states shall finally be summarized briefly: In section 2.3.1 is has been explained, that during each user state different cortex regions are activated, i.e. a desynchronization (increase of non-α-activity and decrease of α-activity) can be observed. The *degree* of desynchronization in these regions is directly correlated with the degree of task demand.

Figure 2.26 shows the spectrograms for EEG signals of 16 electrodes recorded during periods low task demand and high task demand of subjects who perceived a slides presentation. For many electrodes it can be seen that there is more high frequent activity for high task demand compared to low task demand. For other electrodes the contrary is the case however, for example for electrodes Pz or Fp2. A possible explanation for this could be that one or a few subtasks required more mental resources when the overall task demand was rated to be low, compared to the case where the subject rated his overall task demand to be high. Nevertheless the hypothesis seems to hold that in general an increase of high frequent activity corresponds to an increase of task demand. The decrease of lower frequent activity for high task demand is very small, i.e. the amplitude of the α-activity seems to be attenuated only little. This is however not too surprising, since even in situations of low task demand already enough stimuli are present so that only little α-activity remains. The high amplitude of the lower frequency bands is not only explained with α-activity, but also with artifacts (see in particular the red columns in the spectrograms of the prefrontal and frontal electrodes in case of low task demand) and with other EEG activity (e.g. θ-activity and δ-activity, see table 2.2). Note that during the perception of a presentation virtually all cortex regions are involved: the temporal cortex for the understanding of speech, the temporal and parietal cortex for the understanding of slides,

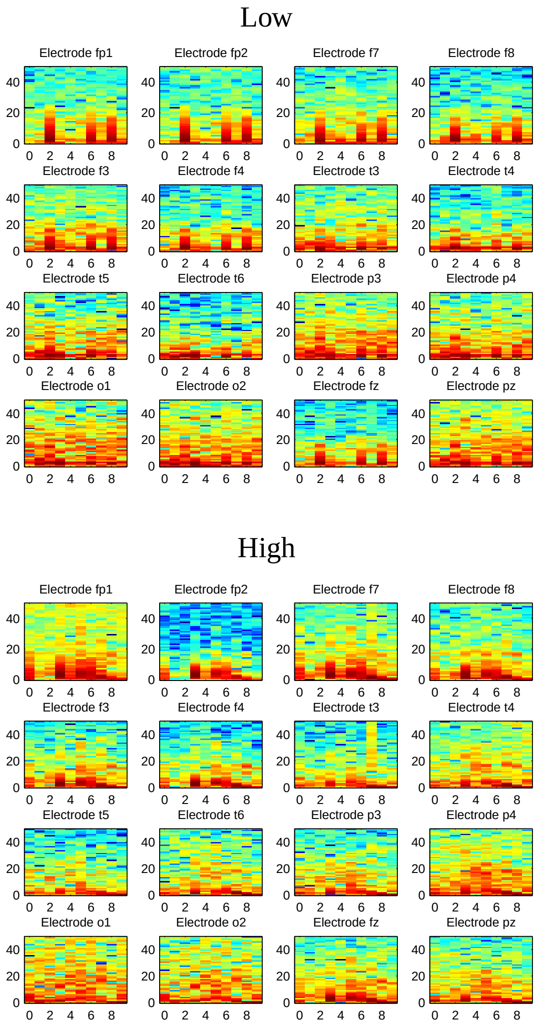


Figure 2.26: Spectrograms for 16 electrodes for low (top) and high (bottom) task demand during listening to a talk. The energy for the different frequency bands is represented on a ”temperature” scale where blue means low and red means high. The x-axis shows time in seconds, the y-axis gives the frequency in Hz.

the occipital cortex for visual perception and the frontal cortex for the control of attention. Therefore differences in the frequency content of the EEG signals for the different task demand levels can be seen for electrodes at positions all over the cortex. We conclude that the observations from the spectrograms confirm the hypothesis that EEG correlates of task demand and vigilance are closely related.

An increase of vigilance and task demand is also characterized by an increasing amount of oxygenated blood forehead, i.e. more oxygen is transported in the blood. For overload a decrease of blood oxygenation compared to the highest oxygenation level can be observed [Izzetoglu et al., 2004]. Furthermore the magnitude of pupil dilation has been shown to be a function of mental effort which is required to perform a cognitive task [Beatty, 1982], [Hoecks and Levelt, 1993].

Chapter 3

Related Work

A lot of research is concerned with the computational processing of human brain activity with non-clinical purpose. In most cases EEG is used for data recording but also functional magnetic resonance tomography (fMRI) and functional near infrared spectroscopy (fNIRS) are applicable techniques for physiological data acquisition. Three main research goals can be distinguished:

1. Identification of different mental states, i.e. different mental activities. These activities can be very different such as resting vs. performing mental arithmetic or very similar such as the perception of different word categories (e.g. fish vs. for-legged-animals vs. trees vs. flowers).

   Different motivations exist for this research goal: Human cognition can be studied by analyzing the properties of classifiers which discriminate mental states [Mitchell et al., 2004], i.e. inferences about the effect of different mental states on neural activity can be made. Furthermore the knowledge about states which can be discriminated particularly well and about methods which are suitable for that is an important *preparatory* work for the development of Brain-Computer-Interfaces (BCIs), which attempt to control electronic devices using only brain activity. Note that we distinguish the research which is concerned with the shear discrimination of mental states (mostly but not exclusively for BCI purposes, [Anderson et al., 1995, Anderson and Sijercic, 1996, Ford, 1996, Culpepper, 1999]) from the actual construction of BCIs (see item 3), where also the closed-loop feedback is an essential component [Lethonen, 2003] (see also section 3.3). Finally the development of intelligent user interfaces [Chen and Vertegaal, 2004] is to be mentioned as a motivation for the identification of different mental (and physical) states.

2. Assessment of mental task demand, alertness or vigilance (see section 2.3.2 for a detailed description of these terms). In most experimental setups of the works pursuing this research goal one single task was performed for which the difficulty was varied in order to provoke different levels of task demand of the recorded subject. Other researchers made experiments where the task difficulty stayed constantly but a task requiring sustained attention had to be executed for a longer period of time. From the physiological data recorded during task execution it was attempted to predict natural fluctuations of vigilance. Because we hypothesize a close connection between task demand and vigilance which has been explained in section 2.3.2, we summarize the research which is concerned with either of these parameters in this category.

   Two primary reasons for the assessment of task demand, alertness of vigilance can be distinguished: During critical tasks it is important to be able to predict and to avoid mental overload and too low alertness in order to prevent dangerous situations [Smith et al., 2001, Pleydell-Pearce et al., 2003, Duta et al., 2004]. Furthermore operator performance can be increased, when it is possible to maintain the operator's cognitive load on an optimal level by adapting the way data is presented appropriately [Jung et al., 1997, Berka et al., 2004, Izzetoglu et al., 2004]. A lot of the research which follows this motivation for task demand estimation has been done in a military context, namely for the DARPA Augmented Cognition Program [Schmorrow and Kruse, 2002].

3. Brain-Computer-Interfaces (BCIs). As mentioned above, the goal of BCIs is to give commands to electronic devices using only brain activity. Prominent examples for BCIs are systems to control prostheses or to allow locked-in patients to communicate although they are no longer capable of voluntary movements.

3.1 Identification of Mental States

In the past decade a lot of work aiming at the identification of mental states has been done which used the data set recorded by Keirn and Aunon [Keirn and Aunon, 1990]. This dataset consists of EEG data recorded at the electrode positions O1, O2, P3, P4, C3, C4 (see figure 2.12 for the precise positions) and an additional EOG channel for artifact detection. The data was recorded with a sampling frequency of 250 Hz and the following tasks were performed during EEG recording:

Baseline – Alpha Wave Production: Subjects were asked to relax and to close and open their eyes in five seconds intervals. Doing this, α-activity can be observed, at least when eyes are closed.

Mental Arithmetic: Subjects had to solve non-trivial multiplications without vocalizing or moving.

Geometric Figure Rotation: Subjects were shown a drawing of a complex geometric figure. Then the figure was moved out of sight and subjects were instructed to imagine the rotation of this figure.

Mental Letter Composing: Subjects were instructed to mentally compose a letter to a friend or relative, without moving or vocalizing.

Visual Counting: Subjects had to imagine a black board and mentally to visualize numbers being sequentially written on the board.

Note that these kinds of tasks are not very likely to occur in every day situations in contrast to the activities which are considered in this work (see section 1.1). However the focus of the research in [Keirn and Aunon, 1990] was rather to reliably discriminate different mental states to be able to give commands to a computer then to discriminate user states.

In [Anderson et al., 1995], [Anderson and Sijercic, 1996], [Ford, 1996] the data set described above is used to train and test systems for the discrimination of mental states. In [Culpepper, 1999] data for the same tasks was recorded for one subject, using the electrodes FPz, F3, Fz, F4, FCz, C3, Cz, C4, Pz, P3 and P4 and an additional EOG channel.

To allow an easy comparison of the mentioned research works, the used data portions, the applied methods and obtained results are summarized below using a common scheme.

Author: [Anderson et al., 1995]

Considered mental states: Baseline task, arithmetic task

Used Data: Two recording sessions of one subject, 10 trials per task from both sessions. From 790 quarter second long patterns per task 80% are used for training, 10% for validation and testing respectively. The data for each set is randomly chosen from both recording sessions. 10-fold cross-validation is performed.

Methods:

Artifact Removal: none

Feature Extraction: Three representations of quarter second long time segments are investigated: unprocessed time signals, a reduced dimensional representation obtained using the Karhunen-Loève transform, a frequency spectrum based representation estimated with a sixth-order auto regressive (AR) model.

Classification: Feed forward two-layer neural networks with up to 40 neurons in the hidden layer

Results: An accuracy of 74% for the discrimination of both states is obtained using the frequency spectrum based data representation.

Authors: [Anderson and Sijercic, 1996]

Considered mental states: All five states described above

Used Data: Data from four subjects, two recording sessions per subject, 10 trials per task and subject over both sessions. 277 half second long patterns per task and subject are available, after discarding eye blinks. Training set size: 80% of the whole data, validation and test set size respectively: 10% of the whole data. The data for each set is randomly chosen from both recording sessions. 10-fold cross-validation is performed.

Methods:

Artifact Removal: Time segments containing eye blinks are discarded using information from the EOG channel.

Feature Extraction: All six coefficients of a sixth-order autoregressive model for each electrode are used, resulting in a total feature vector dimensionality of 36.

Classification: Feed forward two-layer neural networks with 20 neurons in the hidden layer

Results: An average accuracy of 54% in subject dependent experiments is obtained when averaging over the network outputs for 20 consecutive half second long time segments.

Author: [Ford, 1996]

Considered mental states: Baseline task, arithmetic task

Used Data: Data from four subjects, two recording sessions per subject, 10 trials per task and subject over both sessions. 241 half second long patterns per task and subject are available, after discarding eye blinks and balancing the data set. Training set size: 80% of the whole data, validation and test set size respectively: 10% of the whole data. 10-fold cross-validation is performed.

Methods:

Artifact Removal: Time segments containing eye blinks are discarded using information from the EOG channel.

Feature Extraction: All six coefficients of a sixth-order autoregressive model for each electrode are used, resulting in a total feature vector dimensionality of 36.

Classification: Learning Vector Quantization

Results: In session and subject dependent experiments and average accuracy of 73% is obtained. Mixing of data from both sessions for each subject yields in an average accuracy of 85.5%.

Author: [Culpepper, 1999]

Considered mental states: All five states described above; only different triplets and pairs are considered for discrimination.

Used Data: Two recording sessions from one subject, 10 trials per task over both sessions. From 200 half second long time segments per task 50% are used for training and 50% for testing and validation.

Methods:

Artifact Removal: Independent component analysis is applied to the data to isolate eye blinks to one component. This component is identified by visual inspection and then rejected.

Feature Extraction: A frequency representation of half second long to $\frac{1}{20}$ second long segments of the independent components is obtained using a discrete Fourier transform.

Classification: Three-layer feed forward neural networks with 40 to 1000 neurons in the first and 5 to 100 neurons in the second hidden layer

Results: Classification accuracies for best discriminated task pairs (results for other pairs are not reported): 94% figure rotation vs. arithmetics, 90% for baseline vs. letter composition, 82% for baseline vs. counting
Classification accuracies for best discriminated task triplets (results for other triplets are not reported): 86% for baseline vs. letter composition vs. arithmetics, 77% for counting vs. figure rotation vs. letter composition, 74% for letter composition vs. figure rotation vs. artihmetics

One important conclusion to be drawn from the above presented research works is that the discrimination of mental states using EEG data is possible in general. Furthermore statistical learning methods such as neural networks and a representation of the EEG signals in frequency domain seem to be suitable for this task.

More recently other work concerning the discrimination of mental states has been done, which does not rely exclusively or not at all on EEG data.

[Chen and Vertegaal, 2004] proposed a physiologically attentive user interface (PAUI) which is able to distinguish four user states which are defined by the current level of mental and motor activity: ”resting” (low mental activity, no movement), ”moving” (low mental activity, sustained movement), ”thinking” (high mental activity, no movement) and ”busy” (high mental activity, sustained movement). The mental activity level is estimated using heart rate variability, the degree of motor activity is determined using EEG data from one electrode over the central cortex where motor activity is controlled[1]. Spectral analysis of the heart rate variance is applied to detect increases of mental load which is characterized by increasing heart rate variance. The onset of motor activity and thus a user state change is detected by the decrease in power for the μ-frequency band (see section 2.3.1) in the EEG .

For each state the cost for different kinds of interruptions by a mobile device is defined: While this cost is low for all kinds of interruptions during resting, it is low during moving for speech-related events only (e.g. for answering a phone call, since no additional complex movements must be made for that), but high for events which require complex motor activity such as responding to a text message or a chat request. For the user state thinking only non-auditory interruptions (i.e. vibrations, visual notifications) are defined to be acceptable, i.e. they have a low cost, since audible alerts would potentially interfere with mental engagement according to the authors. For the state busy the cost for all kinds of interruptions is set to be high. As a sample application a cell phone has been augmented with user state detection capabilities and in a six person trial the correct user state could be identified in 83% of all cases.

[Mitchell et al., 2004] use fMRI data to discriminate mental states like looking at a picture vs. reading a sentence describing that picture, reading a non-ambiguous sentence vs. reading an ambiguous sentence or hearing words from different categories (food, people, buildings etc.). Note that the considered states seem to be little different from each other compared to those considered in [Keirn and Aunon, 1990] or to states like reading, perceiving a presentation, resting etc. with which we are concerned in this work. Since fMRI data provides a much better spatial resolution and thus a much better insight into brain activity than EEG data, the discrimination of very similar states becomes more realistic.

For each set of mental states data from five to fifteen subjects is used. Motion artifacts are removed from the data and several seconds long time segments of data consisting of activity values for each voxel (i.e. for each cubic volume element of the recorded three dimensional data) are used as input features for different classifiers. Activity values are given as deviation from the mean activity measured during a resting condition. The performance of Gaussian Naive Bayes classifiers, Support Vector Machines and k-nearest neighbor classifiers is investigated. Since very high dimensional feature vectors are obtained from fMRI data, two different methods for feature reduction are explored. The first method attempts to select those features which discriminate the states to be distinguished best. This is done by training one classifier separately for each feature which uses this features only to make predictions. Then those features are selected for which the classifiers yield the best classification accuracies. In contrast to that the second method selects those voxels as features having the best signal-to-noise ratio. Such features have the property that they discriminate best between

[1] Note that this approach does not agree with the intuition that *mental* activity would be estimated best with EEG data and *motor* activity via the heart rate. It has been shown however that there is a correlation between mental load and the heart rate variability and between the degree of motor activity and the EEG of the μ-frequency band (see [Chen and Vertegaal, 2004] for appropriate references).

a resting condition and an activity condition under which all states to be actually discriminated are summarized. The second method is shown to outperform the first one in many cases.

Results are reported in terms of normalized rank error which considers the rank the classifiers assign to the correct hypothesis. It ranges from 0 when the correct class is ranked most likely, to 1 when it is ranked least likely. Using Gaussian Naive Bayes classifiers and Support Vector Machines, average normalized rank errors between 0.11 and 0.24 are reported for subject specific experiments. For the picture vs. sentence study classifiers are trained across subjects as well. Using support vector machines a normalized rank error of 0.25 is reported for these experiments.

Some findings from this research are also interesting for the computational processing of EEG data, since in both cases the problem of data sparseness paired with a high feature vector dimensionality has to be addressed:

- With *linear* Support Vector Machines at least comparable or even better classification results can be obtained compared to non-linear classifiers (such as *k*-Nearest Neighbors or Gaussian Naive Bayes classifiers), when the dimensionality of the feature space is high.
- For noisy data the selection of features which discriminate a baseline condition from all other conditions can improve classification results, since the signal-to-noise ratio is improved.
- There are similarities in the neural correlates of different mental states across subjects, so that training of subject independent classifiers for the discrimination of these states becomes possible.

3.2 Assessment of Task Demand, Vigilance or Alertness

A lot of research in the domain of task demand, vigilance or alertness assessment during the execution of tasks requiring sustained attention has been done in recent years. To provide a better overview over the research works reviewed in this section and to allow their comparison, the most important aspects of each work are summarized below using a common scheme.

Authors: [Jung et al., 1997]

Goal: Estimation alertness during a sustained attention task.

Task: Subjects have to respond to auditory stimuli. The task should simulate sonar target detection.

Electrodes: 5 electrodes from central, parietal and occipital cortex are used.

Amount of data: In three recording sessions lasting 28 minutes 10 subjects were recorded. One session is used for training, validation and testing respectively. 967 time segments of 1.6s length are extracted from each session.

Methods:

Artifact removal: Signals exceeding a certain threshold are discarded as artifactual data. Median filtering is used to further minimize the presence of artifacts.

Features: The log spectral power for 81 frequency bands between 0.5Hz and 50Hz is computed. Then principal component analysis is applied and the four principal components covering most variance of the data are retained.

Prediction method: Feed forward neural networks with 3 neurons in one hidden layer and multivariate linear regression functions are used to predict an alertness index which is a function of failures to respond to the given stimuli residing in $[0, 1]$.

Results: An RMS-Error between classifier prediction and the alertness index of 0.163 for linear regression and 0.156 for neural networks is reported. All results are obtained in session independent but subject dependent experiments.

Authors: [Smith et al., 2001]

Goal: Monitoring of subject specific task loading

Task: The MATB task is used which simulates activities of a pilot: systems monitoring, resource management, communication and tracking. The subjects must react with keyboard inputs on changes of gauges, warning lights and auditory stimuli. A joystick is used for the tracking task which simulates manual control of the aircraft position. Task difficulty is varied from passive watching to high load. Note that the variation of task difficulty showed a significant correlation with a subjective rating (NASA Task Load Index) performed by the recorded subjects.

Electrodes: Afz, Fz, F3, F4, Fz, P3, P4, Pz

Amount of data: For 16 subjects three session were recorded, each load level (passive watching, low, medium, high) was simulated for five minutes per session. One session is used for training, validation and test respectively.

Methods:

Artifact removal: Adaptive filtering techniques are applied for artifact removal.

Features: Spectral power is estimated for several sub-bands of alpha (8-13Hz) and theta (4-7Hz) frequency ranges for four seconds long time segments.

Prediction method: Subject-specific multivariate distance functions are estimated using data from load levels passive watching and high load. These functions compute a subject-specific task load index between 0 (passive watching) and 1 (high task load).

Results: For subject specific but session independent experiments, significant correlations between subjective task difficulty ratings and results of the estimated task load predictors or between reaction times on stimuli and results of the task load predictors can be found. Furthermore significantly higher task loads are predicted for load level high compared to load level low for 15 out of 16 subjects. Significant differences in predictions for low vs. moderate task load or high vs. moderate task load are observed only for half of the recorded subjects.

Authors: [Pleydell-Pearce et al., 2003]

Goal: Prediction of task load, investigation whether different predictors are suited for different subjects.

Task: Gauge monitoring task: Pushbuttons and a joystick are used to maintain changing gauges at a certain level. Task difficulty is varied by varying the tendency of the actual gauge value to move away from the optimal value. Performance is measured as deviation of the gauges' current values from their optimal values. All performance measurements for each subject are sorted and then split by the median so that finally only low and high task performance is distinguished.

Electrodes: Fp1, Fp2, Fpz, F7, F8, Fz, Cz, T5, T6, P3, P4, Pz, O1, O2 and vertical and horizontal EOG

Amount of data: Six subjects participated in two recording sessions. During each session each of five difficulty levels was simulated 10 times. Each time 40 seconds of data were recorded. This results in 33 minutes of data per subject.

Methods:

Artifact removal: Artifacts are removed using information from the EOG channel.

Features: 5166 dependent variables are computed for 10 second long time segments. Features include string length, mean area under the curve of the time signals, inter-electrode coherence, cross phase and cross power correlation between electrodes and spectral power for selected frequency bands between 0Hz and 100Hz. The features which are correlated best with task performance are then selected to be included in the prediction functions.

Prediction method: A multivariate linear regression model is used for prediction of task performance (not of the task difficulty). The underlying hypothesis here is that worse task performance indicates higher task load which is more challenging and stressful for a subject and therefore should have correlates in the EEG.

Results: Accuracies of 71.9% for the discrimination of low and high performance averaged over all six subjects in session dependent and 70.8% in session independent experiments are reported. For subject independent experiments an accuracy of 71.0% is obtained.

Remarks: Different features are found to be the best predictors for different subjects. Therefore the authors conclude that idiosyncratic aspects of EEG patterns reflect genuine and reproducible differences between subjects. Some predictors could be identified however which are valid for all subjects and it can be seen from the results that they work well for predictions across subjects.

Authors: [Duta et al., 2004]

Goal: Estimation of alertness during a sustained attention task.

Tasks: Tracking a position indicator on a computer screen, reacting to a visual stimulus, keeping the value of a continuously changing gauge at zero. The subjects' vigilance during the tasks was rated by experts who manually labeled 15 seconds long segments of the signals using a seven-class alertness classification system.

Electrodes: A1, A2 (mastoid EEG)

Amount of data: Approximately two hours of data for each of eight subjects was collected. In a round robin manner data from one subject is used for test, while training and validation is performed on the data from the remaining subjects.

Methods:

Artifact removal: Artifacts are detected by visual inspection of the raw data and discarded.

Features: The coefficients of fifth order AR models which estimate the power spectrum in a range between 0Hz and 30Hz for 1 second long time segments for each electrode channel are used as features.

Prediction method: A feed forward two-layer neural network with 10 neurons in the hidden layer is used to estimate a regression function which predicts alertness. Outputs are then grouped in three classes (alertness, intermediate, drowsiness) using thresholding

Results: Accuracies for the comparison with expert labels range between 39% and 62% for subject independent experiments.

Remarks: An interesting method for data analysis is applied here. This is necessary since the manually labeled segments have a length of 15 seconds but only one second long time segments are used for computational processing. Thus it is possible that some of the one second long segments have wrong labels. Self-organizing maps are trained here to cluster all data points of the training data. Then those data points are removed which are mapped in the same areas as other data points belonging a different alertness category.
Another interesting aspect of this work is that only electrodes at the mastoids are used which are easy to attach and comfortable to wear. This is particularly of advantage for applications where EEG is recorded constantly for longer periods of time.

Authors: [Berka et al., 2004]

Goal: Monitoring task loading during a command and control task

Tasks: (1) Warship Commander Task [John et al., 2002]: Incoming aircrafts must be identified as hostile or friendly and appropriate actions must be taken. Task difficulty is varied by varying the frequency with which events occur.
(2) Three level cognitive task: Digits are displayed on a computer screen. A response must be given only for certain combinations of consecutive digits. Workload is increased in three levels by increasing the complexity of digit combinations to be recognized.
(3) Recognition of previously memorized images interspersed randomly in a collection of new images. For this task the rate of failures to give correct responses is measured.

Electrodes: Bipolar recordings of Fz to POz and Cz to POz, unipolar recordings of Fz, Cz and POz

Amount of data: Task (1): 13 subjects were recorded. Each difficulty level was simulated three times for 75 seconds.
Task (2): 16 subjects were recorded. Each difficulty level comprised 250 trials.
Task (3): Two sessions for 19 subjects were recorded.

Methods:

Artifact removal: Eye blinks are detected using cross-correlation analysis with a sine shaped artifact model. Eye blink contaminated data is then replaced by the high pass filtered signal. Other artifacts are discarded using out of bounds detection.

Features: The log power spectrum for frequencies between 3Hz and 40Hz with 1 Hz resolution is computed for 1 second long time segments. Furthermore for each frequency band the relative power compared to the total power between 3Hz and 40Hz, and *z* scores are calculated.

Prediction method: A linear discriminant function is used to predict four states of alertness: high vigilance, low vigilance, relaxed wakefulness and sleepy. Note that data points from task variations with low difficulty are used as examples for low vigilance and data points from task variations with high difficulty are used as examples for high vigilance.

Results: For tasks (1) and (2) a significant correlation between task difficulty and percentage of 1 second epochs classified as high vigilance is reported. For task (3) there is a correlation between response errors and percentage of epochs classified as high vigilance per session. This suggests a practice effect between the first and the second session where the number of errors and the percentage of high vigilance epochs are lower. All results are obtained using subject-specific classification models.

Remarks: The EEG recording hardware used in this work is interesting for practical applications. The EEG headset is battery powered and it transmits signals wirelessly to a computer for signal processing. Furthermore the authors claim that it is easy to attach and comfortable to wear even for several hours.

Authors: [Izzetoglu et al., 2004]

Goal: Estimation of task load by monitoring blood hemodynamics using fNIR

Task: Warship Commander Task [John et al., 2002]: Incoming aircrafts must be identified as hostile or friendly and appropriate actions must be taken. The task difficulty ranges over four difficulty levels where for increasing difficulty the frequency with which events occur increases. Task difficulty can be increased even more using auditory messages interspersed from time to time which require responses.

Sensors: Data from 16 pixels of the prefrontal cortex with about 1cm length at each side is obtained.

Amount of data: For eight subjects one session was recorded. Each of the four difficulty levels was repeated three times for 75 seconds.

Methods:

Artifact removal: Adaptive filtering techniques are used for elimination of motion artifacts.

Features/Prediction method: Sensor measurements for the left and right hemisphere are averaged. Then a single index for blood oxygenation is derived.

Results: A strong positive correlation between blood oxygenation and task difficulty can be found. Furthermore blood oxygenation decreases significantly with decreasing task performance at the highest difficulty level. This supports the hypothesis that subjects who became overwhelmed, i.e. who experienced too high task demand became disengaged.

Authors: [Iqbal et al., 2004]

Goal: Estimation of task demand from the pupil diameter during an interactive task.

Tasks: Object manipulation (sorting e-mails in appropriate folders), searching a product in a list, adding numbers, reading comprehension. Each task contained two difficulty levels.

Sensors: A head mounted eye tracker.

Amount of data: 12 subjects were recorded.

Methods:

Artifact removal: none

Features/Prediction method: A regression function which predicts task difficulty from the average percentage in pupil diameter change (relative to a baseline condition where the computer screen had to be fixated) is estimated.

Results: Significant changes in relative pupil diameter can only be observed for a cognitive subtask of the object manipulation task.

Remarks: The applied technique has the advantage to be less intrusive EEG or even fNIR since ideally no measurement device has to be attached to the subject. However measurements are sensitive to environmental factors (e.g. illumination) and emotional states.

From the works reviewed above we conclude that the assessment of task demand, alertness or vigilance using EEG data, fNIR data or even pupil diameter (which is no direct correlate of neural brain activity) seems possible. Both statistical learning methods like neural networks or simple linear parametric models like linear regression or discriminant analysis seem to be applicable for the posed problem. Note however that the results in most works reported above were obtained in highly controlled experiments where it was easily possible to simulate different task difficulty levels which evoke different levels of task demand. Furthermore objective correlates with task demand such as task difficulty or task performance for a particular time segment could be obtained easily. For the problem of task demand assessment during a meeting or a talk this is not necessarily the case.

3.3 Brain-Computer-Interfaces

Wolpaw et al. define a brain computer interface (BCI) to be ”a communication system that does not depend on the brain’s normal output pathways of peripheral nerves and muscles” [Wolpaw et al., 2000]. In other words, a BCI can be used to control computer applications or other electronic devices using only brain activity but no speech or voluntary movements. According to this definition an EEG-device which estimates user state and task demand and uses these estimates to configure the user’s cell phone appropriately can be called a BCI.

However BCIs are commonly viewed as systems which immediately provide feedback to the user about the hypothesis of the user’s command [Lethonen, 2003] and some types of BCIs even require this closed loop feedback to work properly as will be explained shortly. Immediate feedback about the detected user state or workload level to the user is not necessary and perhaps not even desirable for the system developed in this work, since the goal is here to *minimize* disturbances of the user. For this reason we decided not to use the term BCI for the system developed here.

In this section only the basic principles of BCIs are explained briefly to complete the overview of applications for computational processing of EEG with non-diagnostic purpose. A more detailed review about BCIs can be found in [Vaughan et al., 2003]. Many classification and signal processing methods which are commonly used in BCI research, are interesting as well for the discrimination of mental states and the assessment of task demand. These methods are presented in chapter 4.

The following two approaches for the construction of BCIs can be distinguished [Lethonen, 2003]:

Pattern recognition approach: In this approach the user of the BCI has to perform different mental tasks such as those proposed by Keirn and Aunon [Keirn and Aunon, 1990] which are described in section 3.1. If these tasks can be distinguished from each other, a particular activity can be executed when the system recognizes that the user performs a specific task. For this approach the BCI must be pre-trained, its user however does not need any training at all in the best case.

Operant conditioning approach: In this approach the user has to learn to control his EEG response, for example the μ-rhythm amplitude of his brain activity. While this requires extensive training for the user, the BCI does not need to be trained at all since it simply looks for changes in specific characteristics of the EEG signal. Real-time feedback and positive reinforcement for correct behavior are essential to make the operant conditioning approach work.

Note that the discrimination of mental states is a central aspect of both the pattern recognition approach for BCIs and for user state identification. While in the BCI context however mental states can be chosen which are easy to discriminate, the mental states are already given in in case of user state identification and the system must be designed to discriminate them as good as possible.

Chapter 4

Methods

Several processing steps have to be accomplished before a hypothesis for the current user state or the current task demand level can be obtained from raw EEG data. These steps are explained in greater detail in this chapter. To provide a better overview over the following sections, figure 4.1 shows the integration of all applied methods into the whole system which is proposed here. For each method the corresponding section number of this chapter is displayed in the figure.

4.1 Artifact Removal

Out of the large number of artifacts which can contaminate EEG data (see section 2.2.1.5) eye blinking and eye movement artifacts seem to have the largest impact concerning the computational analysis of the EEG: They occur very frequently [Jung et al., 2000b] and their frequency range overlaps with the frequencies of the actual EEG signal, which makes simple band pass filtering for artifact removal impossible [Duta et al., 2004].

Filtering is only suitable to eliminate high frequency artifacts from muscular activity and noise introduced by the AC lines, since a low pass filter can be applied here with a cutoff frequency above the highest relevant EEG frequency. For computational analysis of the EEG in frequency domain however, it is sufficient to simply select those frequency bands which are relevant for EEG analysis while all others need not to be considered. Thus artifacts occurring at high frequencies are eliminated automatically without the need for low pass filtering. Note furthermore that low pass filtering of the signals prior to further computational processing in frequency domain has even been shown to be detrimental to classification accuracies for the discrimination of mental states (see [Anderson and Sijercic, 1996] and [Ford, 1996]). A reason for this might be the problem that low pass filters generally also attenuate frequencies below the cutoff frequency, since a filter can not be designed such that there is a sharp cutoff between those frequencies which pass the filter and those which are blocked.

For the reasons explained above only eye activity related artifacts are considered in this work. (Note that in virtually all other research concerned with the computational processing of EEG, artifact removal focuses on eye activity only as well, see also section 3.1.)

4.1.1 Removal of Eye Activity Artifacts

Several methods for eye activity artifact removal have been proposed in the past:

- Simple rejection of artifactual data. Short time segments of the signal (between 0.25 seconds and 2 seconds length) containing eye activity are simply discarded. Eye activity is identified

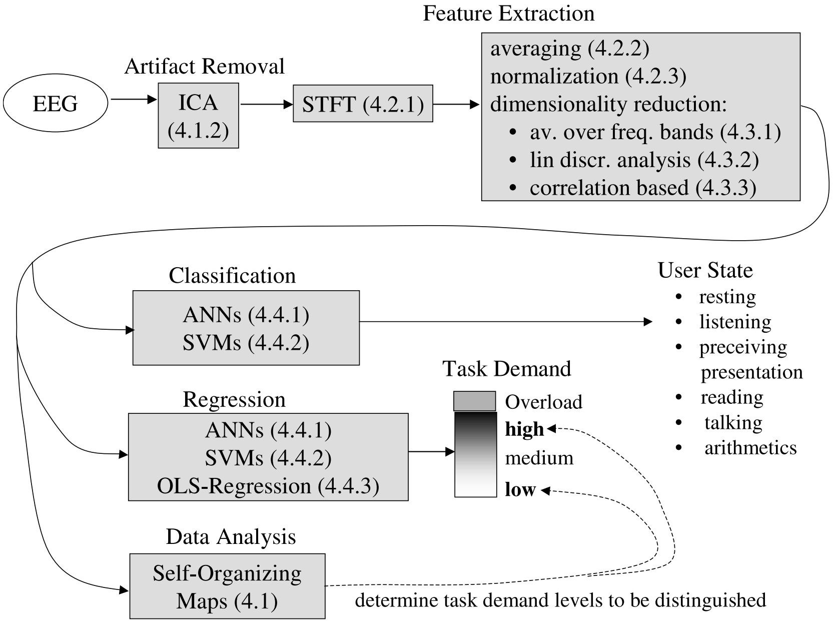


Figure 4.1: Overview over the system for user state identification and task demand estimation. The numbers in the boxes indicate the section numbers of this chapter, where the particular methods are explained.

by visual inspection [Duta et al., 2004] or automatically by rejecting all segments where the signal exceeds a certain threshold [Jung et al., 1997]. Another method for automatic eye activity detection and rejection uses a separate channel which records the eye activity only (i.e. the EOG which is short for electrooculogram) [Anderson et al., 1995, Anderson and Sijercic, 1996, Ford, 1996]. Since very little EEG is contained in the EOG data, artifacts are identified more reliably using EOG when the exceedance of a threshold is used as criterion.

- For many experimental setups eye activity artifacts occur very frequently in the EEG (e.g. when visual stimuli are involved in the experiments), and almost no data would be left, if all such artifacts were discarded [Jung et al., 2000b]. Therefore techniques which use the EOG channel information for artifact removal without discarding the data have been proposed. These methods are commonly known as regression techniques, since linear regression models are applied to describe the superposition of the EEG and the eye activity in the observed signals:

$$\mathrm{OBS}(t) = \alpha_1 \cdot \mathrm{VEOG}(t) + \alpha_2 \cdot \mathrm{HEOG}(t) + \mathrm{EEG}(t) \tag{4.1}$$

where OBS represents the actual observation, VEOG and HEOG the (appropriately filtered) EOGs for horizontal and vertical eye movements and EEG the actual EEG signal. After the estimation of the regression coefficients α_1 and α_2 the pure EEG signal can be easily determined from equation 4.1. Regression does not necessarily have to be performed for time domain signals but it is also possible for the signals in frequency domain. An overview over regression based techniques for artifact removal is provided in [Croft and Barry, 2000]. However two fundamental problems are related to these techniques: One or two (in case of vertical and horizontal EOG) extra recording channels are required, and EEG activity from the frontal cortex which is still slightly present in the EOG channels is mapped to all other channels by the artifact removal procedure.

- [Berka et al., 2004] do not need extra electrode channels for artifact removal. They use the positive half of a 40 μV 1.33Hz sine wave as artifact model and apply cross-correlation with the EEG signals to detect artifacts, which are characterized by maxima in the correlation function. Then artifactual data is replaced by the 4Hz high pass filtered signal of the artifactual data segment. The two main problems of this approach are that high pass filtering might eliminate important EEG information and that the shape of the artifact model has to be determined using prior knowledge.

- Finally, component based methods are an interesting group of artifact removal techniques which also do not require an extra EOG channel. Principal Component Analysis (PCA) or Independent Component Analysis (ICA) can be used to decompose the EEG signals in uncorrelated components (in case of PCA) or statistically independent components (in case of ICA). The underlying hypothesis here is that artifactual signals and signals related to neural processes are uncorrelated or emerge from independent processes so that they can be separated using methods like PCA or ICA.

An artifact removal method which can be used in the system developed for this work should not require extra EOG electrodes since this would decrease the user comfort. Furthermore the use of prior knowledge to determine the shape of artifacts is problematic, since it may differ across subjects and it may depend on the exact electrode positions. Therefore we decided to use a component based artifact removal method in this work, namely ICA, which has been shown to be superior to PCA [Jung et al., 1998]. Component based methods have the further advantage that other (artifactual) processes than eye activity can be detected and eliminated as well, if this is required.

The principle of ICA and its use for artifact removal is summarized briefly in the following section. A more detailed introduction on ICA can be found in [Hyvärinen and Oja, 2000].

4.1.2 Independent Component Analysis

Due to volume conduction in the body and external sources of noise (e.g. from AC lines), the signal measured at one electrode does not only reflect the local neural activity underneath this electrode, but it represents a superposition of several signals originating from neural and artifactual processes such as muscular activity or noise from external electrical devices. Assuming a weighted linear superposition of these processes, this can be expressed as follows:

$$x_j(t) = \sum_{i=1}^{e} a_{ji} \cdot s_i(t) \tag{4.2}$$

Here $x_j(t)$ is the observed signal over time at electrode j, $s_i(t)$ the signal corresponding to the i^{th} underlying process and a_{ji} the weight which determines how much $s_i(t)$ contributes to $x_j(t)$. We assume that the number of electrode channels e is equal to the number of underlying processes or independent components (ICs) here. This is an important assumption for ICA, which allows that the ICs can be recovered from the given signals.

In practice this assumption is only approximately true. On the one hand it is possible that the number of independent processes is larger than the number of electrodes, when the neural processes under each electrode are different and the signal contains additionally biological and technical artifacts. On the other hand it may happen that in cortex regions underneath many electrodes the same process takes place, namely synchronized α-activity, so that the number of independent processes is smaller than the number of electrodes. Nevertheless the approximate equality of both quantities seems sufficient for the estimation of the ICs [Makeig et al., 1996].

In matrix vector notation equation 4.2 can be reformulated as

$$\mathbf{x}(t) = A \cdot \mathbf{s}(t) \tag{4.3}$$

where $\mathbf{x}(t)$ is a vector containing the observed signals, $\mathbf{s}(t)$ a vector containing the signals corresponding to the ICs and A the ”mixing matrix” which defines how much each IC from $\mathbf{s}(t)$ contributes to the signals $\mathbf{x}(t)$ observed at the different electrodes.

For detection and isolation of artifacts we are interested in the ICs which can be obtained using the unmixing matrix $W = A^{-1}$:

$$\mathbf{s}(t) = W \cdot \mathbf{x}(t) \tag{4.4}$$

Those ICs containing artifactual data (eye activity but also AC noise and high frequent muscular activity are well detected using ICA [Jung et al., 1998]) can be identified by visual inspection or by heuristic criteria from the training data. Using the matrix W which is estimated on the training data, the ICs can also be obtained for the validation data and for the test data. Thus it is also possible to use cross validation to find one or more components to be rejected, so that the overall system performance improves most. Finally, the components which have been previously selected for removal are eliminated in the test data.

Let $\mathbf{s}'(t)$ be the vector where the artifact contaminated components are removed and A' the mixing matrix where the columns with the indices of the artifact contaminated components of $\mathbf{s}(t)$ are removed. Then an estimate for the artifact free data $\mathbf{x}'(t)$ is obtained by:

$$\mathbf{x}'(t) = A' \cdot \mathbf{s}'(t) \tag{4.5}$$

The above described artifact removal process is illustrated graphically in figure 4.2.

The remaining problem is the estimation of the matrices W or A. Usually only W is estimated and A is computed as the (pseudo) inverse of W. There are several methods to obtain an estimate for W:

- Let us assume that the number of electrode channels e equals exactly the number of underlying (statistically independent) processes contributing to the EEG. Then the components in $\mathbf{s}(t)$ are independent from each other, iff only one process contributes to each component $s_i(t)$. At that point the central limit theorem can be used which states that the more processes are superimposed in one random variable, the more its probability density function (pdf) resembles a gaussian pdf. Therefore our goal must be to estimate the signals $s_i(t)$ for $i = 1, \ldots, e$ in a way that their pdfs are as little gaussian as possible[1]. Our actual goal is however the estimation of the unmixing matrix W. Therefore we use the identity $s_i(t) = \mathbf{w_i}^T \cdot \mathbf{x}(t)$, where $\mathbf{w_i}^T$ is the i^{th} line of W, so that our new goal becomes the gaussianity minimization of the pdfs of $\mathbf{w_i}^T \cdot \mathbf{x}(t)$ for $i = 1, \ldots, e$.

  The gaussianity of a random variable can be measured using kurtosis which is zero for a gaussian random variable and becomes more negative or positive for more sub- or supergaussian random variables. Thus the lines $\mathbf{w_i}^T$ of W have to be estimated such that that $|\mathrm{kurt}(\mathbf{w_i}^T \cdot \mathbf{x})|$ becomes maximal for $i = 1, \ldots, e$. (We denote $s_i = \mathbf{w_i}^T \cdot \mathbf{x}$ as a random variable here whose distribution can be estimated from the time series $s_i(t) = \mathbf{w_i}^T \cdot \mathbf{x}(t)$.) In other words the e highest local maxima of $|\mathrm{kurt}(\mathbf{w_i}^T \cdot \mathbf{x})|$ have to be found in order to determine all e lines of W. Note that the processes contributing to the EEG must not have a gaussian pdf themselves in order

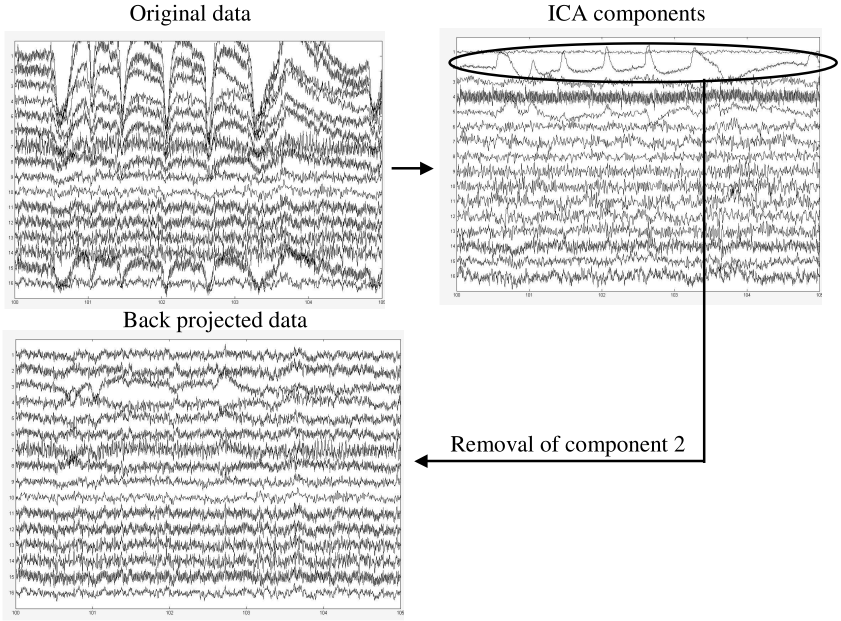


Figure 4.2: Application of ICA for artifact removal in EEG data. The eye activity related artifacts are identified by large peaks in the original data. They are isolated to the independent component 2. After removal of that component and back projection of the remaining components almost artifact free data is obtained.

to make this approach work. This assumption is however usually fulfilled for EEG signals and EEG artifacts [Hyvärinen and Oja, 2000].

- Since kurtosis is very sensitive towards outliers, negentropy can be used for estimating the gaussianity of a random variable. Let g be a gaussian random variable with $\mathrm{Cov}(\mathbf{w_i}^T \cdot \mathbf{x}) = \mathrm{Cov}(g)$ and let H denote the operator which computes the entropy of a random variable.

  When
  $$J(\mathbf{w}_i^T \cdot \mathbf{x}) = H(g) - H(\mathbf{w}_i^T \cdot \mathbf{x}) \tag{4.6}$$
  is maximal, the density of $\mathbf{w_i}^T \cdot \mathbf{x}$ is as little gaussian as possible, since a gaussian density has the highest entropy among all random variables of equal variance. A drawback of this approach is that for the computation of the entropy (non-parametric) estimates for pdfs of $\mathbf{w_i}^T \cdot \mathbf{x}$ for $i = 1, \ldots e$ have to be computed (histograms are not sufficient) which can be difficult in practice [Hyvärinen and Oja, 2000]. Therefore negentropy is often estimated using kurtosis, where the outlier problem arises again.

- The minimization of mutual information is a criterion for the estimation of independent components, which is inspired by information theory. Although the principle of gaussianity minimization to find independent components is not used here explicitly, it can be shown, that this approach is equivalent to the maximization of negentropy [Hyvärinen and Oja, 2000]. Therefore it can serve as a rigid mathematical justification for the more intuitive reasoning above,

[1] If the number of processes is larger then the number of electrode channels, the goal remains the same: When as little processes as possible are superimposed in one signal, the gaussianity of the corresponding pdf is minimized. Only in the rare case when there are less processes contributing to the EEG than electrode channels this approach becomes problematic.

which finally leads to kurtosis maximization or negentropy minimization. The mutual information of the random variables $s_1, \dots s_e$ which belong to the time series $s_1(t), \dots s_e(t)$ is defined as follows:

$$\mathrm{MI}(s_1, \dots, s_e) = \sum_{i=1}^{e} H(s_i) - H(s_1, \dots, s_e) \tag{4.7}$$

It is zero, when the sum of entropies of the single random variables equals the joint entropy of all random variables. If the joint entropy is smaller than the sum of single entropies, that means that one or more random variables can be used to predict partly the values of one or more others. Random variables which can be used to predict the values of each other are not anymore independent from each other however. Therefore minimization of the mutual information maximizes the independence between the random variables which belong to the components to be estimated.

- The infomax algorithm [Bell and Sejnowski, 1995] attempts to maximize the following criterion:

  $$L = H(g_1(\mathbf{w}_1^T \mathbf{x}), \dots, g_e(\mathbf{w}_e^T \mathbf{x})) \tag{4.8}$$

  where the arguments of the joint entropy operator can be interpreted as outputs of artificial neurons with non-linear activation functions $g_i(\cdot)$ and weight vectors $\mathbf{w}_i$.

  If we ignore the activation functions $g_i(\cdot)$ for a moment (or set them to the identity), we obtain the joined entropy $H(s_1, \dots, s_n)$ from equation 4.7. Maximization of the joined entropy would lead to minimization of the mutual information, if the sum of the single entropies was almost constant. For signals with a kurtosis larger then three, which is usually the case for EEG signals, this has been found to be a valid assumption [Bell and Sejnowski, 1995]. If however only the linear functions $\mathbf{w}_i^T \mathbf{x}$ were used as arguments of the joint entropy, it could be increased simply by increasing the variances of the arguments, i.e. the norm of $\mathbf{w}_i$. This is no more possible when $\mathbf{w}_i^T \mathbf{x}$ is passed to a non-linear squashing function such as $\tanh(\cdot)$ and the output of the squashing function is used as argument for operator computing the joint entropy [Bell and Sejnowski, 1995].

  Other conditions for the applicability of the infomax algorithm are that the signal sources must really be independent, that the signals are virtually not delayed by the ”mixing medium” between source and sensors (liquor, bone and skin in case of EEG signals) and that there is a sufficient number of independent processes (which is at least equal to the number of sensors). All these conditions seem to be fulfilled for EEG data [Makeig et al., 1996], so that the infomax algorithm is now widely applied for artifact removal in EEG data (see for instance [Jung et al., 2000a, Jung et al., 2000b, Delorme and Makeig, 2004]).

In this work ICA weights (i.e. the lines of the unmixing matrix W) are estimated using the implementation of the infomax algorithm from the MATLAB$^{\mathrm{TM}}$ Open-Source toolbox EEGLAB [Delorme and Makeig, 2004]. The actual algorithm which maximizes the criterion in equation 4.8 is basically a gradient ascent on L with respect to the weights in W. Its presentation is omitted, since it would go beyond the scope of this work. It can be found in [Bell and Sejnowski, 1995].

4.2 Feature Extraction

4.2.1 Obtaining Feature Vectors

The features extracted from the time signal of each electrode channel represent the signal's frequency content for a short time segment. For each electrode channel two seconds long segments which overlap one second are extracted using a hanning window. Then a short time Fourier transform (STFT) is computed for each segment according to

$$\mathrm{STFT}(r,\omega) = \sum_{m=-\infty}^{\infty} x[r+m] \cdot w[m] \cdot e^{-j\omega m} \tag{4.9}$$

where r is the index of the current time segment, ω the frequency for which the STFT is computed and $w[\cdot]$ denotes the windowing function which is used to extract the segment from the continuous signal $x[\cdot]$. (In practice the bounds of the sum in equation 4.9 must be adapted in a way, that the maximal and minimal values of $x[\cdot]$ and $w[\cdot]$ are not exceeded.)

For a given signal the result of $\mathrm{STFT}(r,\omega)$ is complex valued and contains information about amplitude and phase shift for the time segment r and the frequency ω. Since the amplitude value only is the information we are interested in, the phase is eliminated by taking the absolute value of the STFT result. Thus a spectrogram is obtained:

$$\mathrm{spectrogram}[r,\omega] = |STFT[r,\omega]|^2 \tag{4.10}$$

Since two seconds long time segments are considered, we end up with an amplitude value for each frequency band of $\frac{1}{2}Hz$ width for each time segment and electrode channel, which we call a feature. Remember that time segments overlap one second, so that new features for the different frequency bands are obtained every second (figure 4.3).

Features from the different electrode channels are concatenated to form one large feature vector. Let $E_1, \ldots, E_e$ be the set of electrodes which are used for recording and let furthermore N_E be the number of features which are obtained per electrode. (Since we have a $\frac{1}{2}$ Hz resolution N_E equals to the width of the considered frequency range in Hz times two.) The value corresponding to a particular feature for an arbitrary time segment is denoted by x_i. Then for this time segment and electrode E_i, we obtain the feature vector

$$\mathbf{x}^{E_i} = \begin{pmatrix} x_1 \\ \vdots \\ x_{N_E} \end{pmatrix} \text{ for } i \in \{1, \ldots, e\} \tag{4.11}$$

The final feature vector for this time segment is obtained by concatenating the feature vectors of all electrodes:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}^{E_1} \\ \vdots \\ \mathbf{x}^{E_e} \end{pmatrix} = \begin{pmatrix} x_1 \\ \vdots \\ x_{N_E} \\ x_{N_E+1} \\ \vdots \\ x_N \end{pmatrix} \tag{4.12}$$

where $N = e \cdot N_E$ is the total number of features. (Note that we assume to have the same number of features per electrode.) In a typical setting of the data collection conducted for this work (see chapter

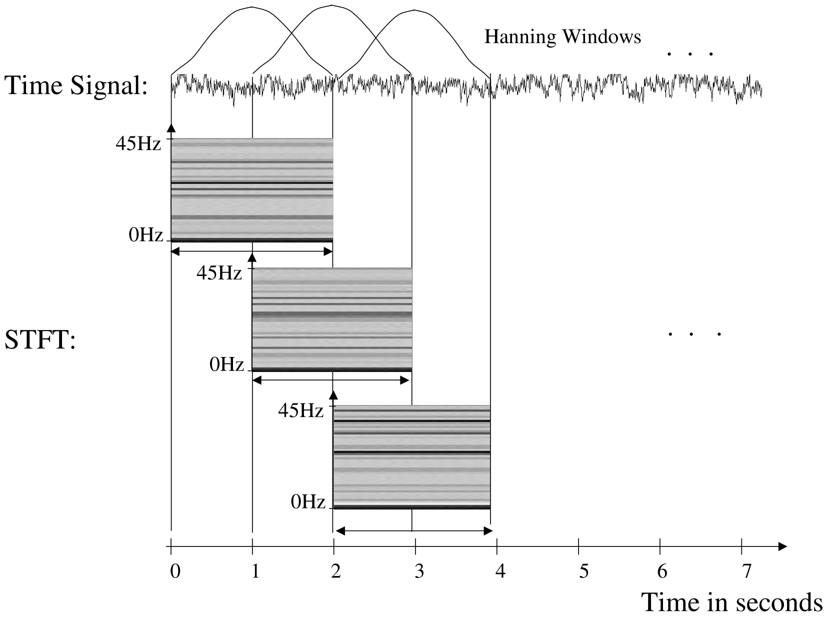


Figure 4.3: Feature extraction using spectrograms for one electrode channel. The gray boxes represent the spectrograms, where the lines correspond to different frequency bands (i.e. features) and the blackness of each line indicates the amplitude value for this frequency band (i.e. the value for this feature).

5) data from 16 electrodes was recorded and frequencies between 0Hz and 45Hz were considered per electrode which results in a feature vector dimensionality of $N = 16 \cdot 90 = 1440$ (remember that we obtain a feature for each frequency band of $\frac{1}{2}Hz$ width). Such a large feature vector dimensionality suggests that either methods for feature reduction are required or a huge amount of data is needed to be able to estimate models reliably. The feature reduction methods applied in this work are described in the next section and in chapter 6 it is examined whether feature reduction is applicable to improve overall system results.

This problem will be further investigated in chapter .

4.2.2 Averaging

EEG signals often exhibit fluctuations over time which are not relevant for user state identification or task demand estimation. Reasons for this can either be artifacts, but also processes caused by neural activity contributing to the EEG. For example it is almost impossible to maintain concentration on a given task for a longer period of time without short lapses which cause outliers in the signal. Therefore averaging over the k previous feature vectors is applied to reduce the influence of such outliers and of artifacts which haven't been eliminated before, using other artifact removal techniques (see section 4.1). For time t_0 the averaged feature vector $\tilde{\mathbf{x}}(t_0)$ is obtained by

$$\tilde{\mathbf{x}}(t_0) = \sum_{i=0}^{k} \gamma(i) \cdot \mathbf{x}(t_0 - i) \tag{4.13}$$

The factor γ could be used to decrease the influence of feature vectors with increasing i. For the experiments in this work it was set to one.

4.2.3 Normalization

Feature values for different frequency bands might have different ranges and different offsets. However a large range of a certain feature does not necessarily imply its large importance for classification. Furthermore, the transformation of all features to a known interval is of advantage to achieve quick convergence of the training procedures for some classifiers such as neural networks (see section 4.5.1.5). Therefore, three simple normalization techniques are investigated:

GlobalNorm: On the training data mean and variance is calculated for each electrode and each frequency band. The obtained values are then used for mean subtraction and variance normalization on the training, validation and test data:

Let $\overline{x}_i$ be the mean for the i^{th} feature over the whole training set and σ_i^2 the estimate for the variance of this feature. For a given feature value x_i of an arbitrary data set the normalized feature value x_i^{norm} is then obtained by

$$x_i^{norm} = \frac{x_i - \overline{x}_i}{\sigma_i} \tag{4.14}$$

UserNorm: Mean and variance normalization for each frequency band is performed separately on training, validation and test data. If one of these data sets contains data from more than one subject, mean and variance is also computed separately for each subject. Then the same normalization procedure as in equation 4.14 is applied to obtain the normalized feature values.

RelPower: To preserve relations of feature values across frequency bands, the feature value for each frequency band is divided by the sum of feature values over all frequency bands, which assures that each feature value falls within [0,1]. For a given feature vector **x** of dimension N the normalized value x_i^{norm} for the i^{th} feature x_i is computed by

$$x_i^{norm} = \frac{x_i}{\sum_{j=1}^{N} x_i} \tag{4.15}$$

Note that a possible drawback of this method is, that information about the relation of the (total) power between adjacent feature vectors gets lost, since the sum all components of the normalized feature vector equals to 1.

4.3 Feature Reduction

As mentioned above the dimensionality of the obtained feature vectors can be very high, when the number of used electrode channels and the considered frequency range per electrode channel are sufficiently large. Feature reduction might help here to improve the reliability of the trained models, especially in situations where training data is very sparse. Furthermore one can expect, that feature reduction methods remove those features which mainly contain noise and which are therefore detrimental to the overall system performance. Several methods for feature reduction are investigated in this work.

4.3.1 Averaging over Frequency Bands

A very straight forward method for feature reduction is to combine features by averaging over adjacent frequency bands (i.e. adjacent features) of one feature vector. It is either possible to take always a fixed number of features over which averaging is performed or to put features in physiologically meaningful groups, which could be for example the frequency ranges of the commonly distinguished EEG frequency bands (see table 2.2) or of their sub-bands.

4.3.2 Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) is a popular method for feature reduction in classification problems with a large number of features. In domains like face recognition [Li et al., 1999] or speech recognition [Haeb-Umbach and Ney, 1992] it is commonly applied. In this work it is used to perform feature reduction on the user state data. The underlying idea of LDA is that the feature vector dimensionality is reduced while the loss of discrimination information by eliminating features is kept minimal.

Let $c_1, \dots c_K$ be the classes of a classification problem and let the feature vectors to be classified have the form $\mathbf{x} = (x_1, \dots x_N)^T$. LDA attempts to find a linear function f of all features so that differences between feature vectors can be expressed in a single variable, namely in the value of f:

$$f(\mathbf{x}) = Y = \mathbf{x}^T \cdot \mathbf{b} \tag{4.16}$$

where $\mathbf{b} = (b_1, \dots b_N)^T$ is the vector of discriminant coefficients which weight each feature according to its importance for the discrimination between the given classes. Feature reduction can be performed by ranking all features according to the absolute value of their discriminant coefficient and selecting the k highest ranked features to be included in a new feature vector. Each of these features is then multiplied with its discriminant coefficient and renormalized using one of the methods described in section 4.2.3. (Renormalization is required for some classification methods as mentioned above.)

The remaining problem is to find the discriminant vector $\mathbf{b}$ which must have the property that is maximizes the ratio of the between class variance and the within class variance *after* all feature vectors have been projected in the one dimensional ”discriminant space” using equation 4.16. Formally the value of γ in the following expression has to be maximized:

$$\gamma = \frac{\sum_{i=1}^{K} R_{c_i}(Y_{c_i} - \overline{Y})^2}{\sum_{i=1}^{K} \sum_{\mathbf{x} \in c_i}(f(\mathbf{x}) - Y_{c_i})^2} \tag{4.17}$$

$\overline{Y}$ denotes the mean over the discriminant values of all feature vectors. The centroid Y_{c_i} for class c_i which is computed according to

$$Y_{c_i} = \frac{1}{R_{c_i}} \sum_{\mathbf{x} \in c_i} f(\mathbf{x}) \tag{4.18}$$

where R_{c_i} denotes the number of training examples for class c_i.

By substituting equation 4.16 into equation 4.17 we obtain the final criterion to be maximized which explicitly depends on $\mathbf{b}$:

$$\gamma = \frac{\mathbf{b}^T B \mathbf{b}}{\mathbf{b}^T W \mathbf{b}} \tag{4.19}$$

The between class scatter matrix B and the within class scatter matrix W of equation 4.19 are defined as follows:

$$B = \sum_{i=1}^{K} (\overline{\mathbf{x}}_{c_i} - \overline{\mathbf{x}})^T \cdot (\overline{\mathbf{x}}_{c_i} - \overline{\mathbf{x}}) \tag{4.20}$$

$$W = \sum_{i=1}^{K} \sum_{\mathbf{x} \in c_i} (\mathbf{x} - \overline{\mathbf{x}}_{c_i})^T \cdot (\mathbf{x} - \overline{\mathbf{x}}_{c_i}) \tag{4.21}$$

where $\overline{\mathbf{x}}_{c_i}$ denotes the mean over all feature vectors of class c_i and $\overline{\mathbf{x}}$ denotes the mean over all feature vectors.

The criterion 4.19 can be maximized by setting its derivative with respect to **b** to zero:

$$\frac{\partial \gamma}{\partial \mathbf{b}} \overset{!}{=} 0$$

This results in the following eigenvalue problem:

$$(B - \gamma W)\mathbf{b} = 0 \tag{4.22}$$

The eigenvector **b** for the largest eigenvalue γ corresponds to the desired discriminant vector since it maximizes the criterion 4.19. Note that if the feature vector dimensionality is very high and training data is sparse, the matrices W and B are likely to be almost singular which can be easily seen from their definitions in equations 4.20 and 4.21. In this case LDA is not applicable for feature reduction, since the eigenvalue problem in equation 4.22 does not have a stable solution. In [Li et al., 1999] the application of the QZ-algorithm for the numerical solution of the eigenvalue problem is proposed which succeeds to find a stable solution in cases where other algorithms (e.g. simple matrix inversion) fail. The QZ-algorithm is applied in the LDA algorithm which has been implemented for this work. Nevertheless we will see in section 6.1.7 that in some cases too little data is available, so that LDA can not be applied.

4.3.3 Feature Reduction of Regression Tasks

LDA is a suitable feature reduction method for *classification* problems (e.g. for user state identification) which has been illustrated in the previous section. The problem of task demand estimation has however much more the properties of a regression task, since task demand is originally a continuous scaled variable. Therefore some problems would arise, if LDA-based feature reduction was applied here:

- For a continuous scaled dependent variable, no finite set of ”labels” (in analogy to class labels) for the different feature vectors exists.

- If the dependent variable of a regression task was ordinally scaled and if it had only a finite number of possible values, each value of this variable could be interpreted as a class label, so that LDA could be applied. This is actually the case for task demand estimation, since we are considering only different *levels* of task demand: low, medium, high, overload. However LDA does not respect the *relation* between such labels, i.e. an important part of information contained in the data would not be used for LDA-based estimation of the features' discriminative power.

A feature reduction method which is suitable for regression tasks is to consider the correlation coefficient between each feature and the dependent variable. Similarly to LDA based feature reduction, features can be ranked according to the absolute value of their corresponding correlation coefficient, which is computed for the feature x_i as follows:

$$r_{x_i,y} = \frac{\sum_{j=1}^{R}(x_i^{(j)} - \overline{x}_i)(y^{(j)} - \overline{y})}{\sum_{j=1}^{R}(x_i^{(j)} - \overline{x}_i)^2 \cdot \sum_{j=1}^{R}(y^{(j)} - \overline{y})^2} \tag{4.23}$$

where $x_i^{(j)}$ denotes the i^{th} feature of the j^{th} feature vector, $\overline{x}_i$ the mean of the i^{th} feature value over all feature vectors, $y^{(j)}$ the value of the dependent variable belonging to the j^{th} feature vector and $\overline{y}$ the mean over all values of the dependent variable.

When using this method the occurrence of multi-collinearities is very likely, especially when the number of features is high. In other words, it is very likely that features are selected which are strongly correlated with each other. This problem could be overcome by the computation of the uncorrelated projections of the features using principal component analysis (PCA) prior to feature reduction. Then those components could be selected as features which are best correlated with the dependent variable without having the problem of multi-collinearities. PCA has however the same limitations as LDA, since an eigenvalue problem has to be solved to obtain the principal components which does not have a stable solution in case of high feature vector dimensionality and data sparseness.

Finally methods like forward selection or backward elimination which describe the data using a linear regression model could be used for feature reduction [Chatterjee and Price, 1991]. With such a linear regression model one tries to find a minimal subset of features which predict the dependent variable best.

- Forward selection starts with a regression function which includes only that feature which is correlated best with the dependent variable. This regression function is then extended by successively adding that feature which is best correlated with the *residuals* obtained from the previous regression function estimation. This procedure is continued until a certain number of features has been included or until the sum of the squared residuals falls below a predefined threshold. Thus features which explain the dependent variable are selected in a more sophisticated way than in case of the correlation based method proposed above. However forward selection is much more time consuming, since in each step one regression function has to be derived and the correlation between the residuals and all the remaining features has to be computed. Finally forward selection does not guarantee that multi-collinearities are avoided, but it makes only sure that in each step that feature is selected which minimizes the sum of the squared residuals in the new regression function more than any other feature. Note that this is not necessarily the case for the correlation based method.

- The starting point for backward elimination is the linear regression function which includes all features. Now those features are eliminated successively which reduce least the sum of the squared residuals. It can be shown that the feature to be eliminated next must have the lowest t-value (for the test statistic that the corresponding regression coefficient is zero) of all candidates [Chatterjee and Price, 1991]. Backward elimination is better suited to cope with the problem of multi-collinearities, since those features with a low t-value either don't explain the dependent variable well or they are highly correlated with other features which are already included in the regression function. This elimination procedure can be continued until the lowest t-value is above a certain threshold. Similar to forward selection, the computational effort required for this feature reduction method is much higher compared to the correlation based approach, since in each step a complete regression function must be estimated. Furthermore its application becomes impossible for too large feature vectors and sparse training data since then the matrix inversions required for the regression estimation can not be computed anymore (see also section 4.5.3).

Due to the problems of forward and backward selection concerning computational resources and high feature vector dimensionalities, only the correlation-based method which has been described above is used in this work. For the same reasons no PCA to uncorrelate features with each other is performed before computing correlations between features and the dependent variable. Results obtained using this feature reduction methods are reported in section 6.2.7.

4.3.4 Problems of Feature Reduction Methods

One common drawback of almost all feature reduction methods described above (LDA for classification tasks and correlation or linear regression based methods for regression tasks) is that they are linear models. In case of LDA linear separability between the classes is assumed, the methods for feature reduction in regression tasks assume a linear relationship between the features and the dependent variable. Furthermore LDA, PCA-based decorrelation techniques and backward selection are not applicable in situations of data sparseness paired with high feature dimensionality.

Therefore, other methods for feature reduction have to be investigated in the future which overcome these limitations. For classification problems a promising approach which can deal with high feature dimensionality in cases of sparse training data has been proposed for example in [Mitchell et al., 2004]. Nevertheless the application of the methods proposed above yields acceptable results as well (see section 6).

4.4 Self-organizing Maps (SOMs) for Data Analysis

4.4.1 Motivation

Self-organizing maps (SOMs) represent a sophisticated technique for clustering data which has been originally proposed by Tevu Kohonen [Kohonen, 1995]. Compared to traditional clustering techniques such as k-means, the main advantage of SOMs is that proximity relations between clusters can be better visualized: While in traditional methods proximity is given by some distance measure between cluster prototypes, SOMs represent the proximity between cluster prototypes in a two or three dimensional grid. Consider the example in figure 4.4. From the euclidean distances between the clusters (determined by the k-means algorithm for example) it is not straight forward and in any case difficult to infer the proximity relation which is found by a SOM. While this is also not required when results from cluster analysis are simply passed to the next step of computational processing, a proximity relation which can be visualized is extremely helpful for manual data analysis. Note that for many applications (and also in this work) not the real distances between prototypes in feature space are displayed on the grid. Instead the relation holds, that points which are close together on the grid correspond to prototypes which are close together in feature space. For many data analysis applications this is sufficient or even desirable.

In this work SOM-based clustering is used to gain insight into the structure of the task demand data. Features are extracted as described in the previous sections and then SOMs are trained with feature vectors from different task demand levels. For each feature vector the best matching unit

Traditional clustering (e.g. k-means):

Distance matrix:

	A	B	C	D
A	0	1.414	1	1.414
B	1.414	0	1	2
C	1	1	0	1
D	1.414	2	1	0

SOM:

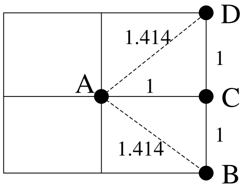


Figure 4.4: Comparison of the output of a traditional clustering algorithm (e.g. k-means) and of a SOM. While the traditional method outputs only distances between clusters, SOMs can be used to visualize the proximity relations between clusters on a grid.

(BMU) on the grid is computed and visualized which allows easy analysis of the spatial relations between feature vectors corresponding to different task demand levels. The BMU for a given feature vector is the unit on the grid to which the prototype with the smallest distance (in feature space) to that feature vector belongs (see also figure 4.7).

Feature vectors with a large (euclidean) distance in feature space have BMUs which are also far away from each other on the grid, while for vectors with smaller distances the corresponding BMUs are also closer together or they are even identical. Therefore the spatial relations between BMUs which belong to feature vectors for different task demand levels can be used to make inferences about how well these task demand levels can be discriminated.

Figure 4.5 shows two SOMs together with their BMUs, one trained on data from all four task demand levels, the other one trained on data from low and high task demand only. When all four task demand levels are considered, it can be seen that the BMUs belonging to feature vectors for medium task demand overlap a lot with those belonging to feature vectors for low and high task demand. This indicates that medium task demand can not be distinguished well from the other two task demand levels by some prediction method (i.e. a classifier or a regression method). In the other SOM one finds that BMUs belonging to feature vectors for low and high task demand are well separated suggesting that a prediction method can discriminate well between these two task demand levels. In section 6.2.1 the SOMs trained on task demand data from other recording sessions are shown and inferences which can be made from their analysis are discussed. Note however that one can not necessarily infer the performance of a prediction method only from the analysis of SOMs. It can only provide valuable hints which have to be verified experimentally using the prediction method in question. A good example for this are feature vectors for task demand level overload which are well separated from the feature vectors for other task demand levels in the SOM in figure 4.5, but which are strongly confused with these feature vectors by all prediction methods examined in this work. (For more details about that problem, please refer to section 6.2.1.)

4.4.2 Principle

Two different spaces must be distinguished for SOMs (figure 4.6):

- The map: This low dimensional (mostly two dimensional) space contains several units called neurons which are organized in a regular grid. Neurons are connected to adjacent neurons by

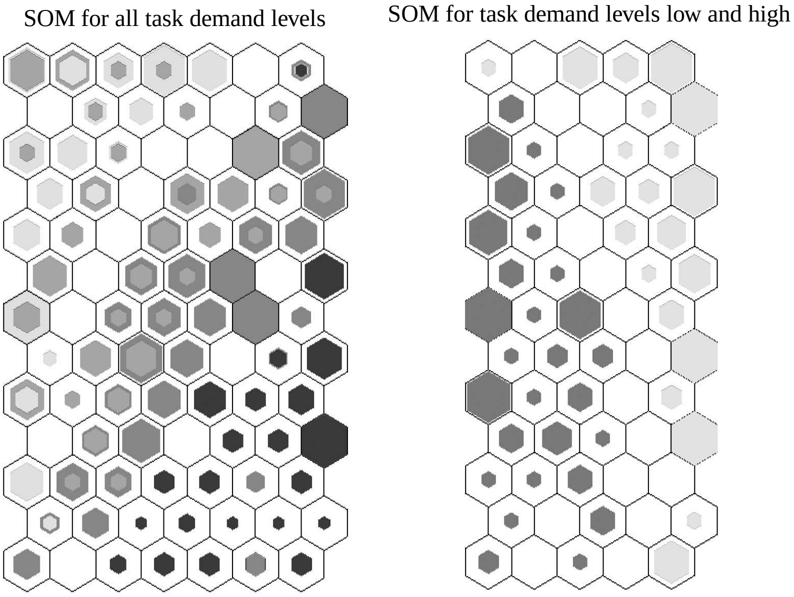


Figure 4.5: SOMs trained on data from all four task demand levels (left-hand side) and on data from low and high task demand only (right-hand side). The intensity of gray indicates the task demand level, ranging from low task demand coded in light gray to overload coded in black.

a neighborhood relation, which dictates the topology or structure of the map. Each neuron is associated with a prototype vector (or weight vector[2]) in feature space.

- The feature space: This is a vector space of arbitrary dimension containing the example vectors and the prototype vectors for each cluster.

Before starting the actual training procedure, a SOM has to be initialized, i.e. the number of units of the grid M and the sidelength of the grid has to be determined. These parameters can either be user defined based on prior knowledge, or they can be determined in a data driven manner which is done in this work following [Vesanto et al., 2000]. The value of M is calculated from the number of training examples R using the heuristic formula $M = 5\sqrt{R}$. The ratio of grid sidelengths is based on the ratio of the largest eigenvalues of the data covariance matrix, and the actual sidelengths are set so that their product is as close to M as possible.

SOMs are trained iteratively. After a sufficient number of training steps, it is guaranteed that units (or neurons), which are neighbors in the map, also have prototypes which are close together, no matter how the prototypes have been initialized. Thus a projection of high dimensional data in a lower dimensional space is obtained.

Sequential training proceeds with the following steps:

1. Select a data point $\mathbf{x}$ randomly.

2. Find the Best Matching Unit (BMU) c for $\mathbf{x}$ in the map, i.e. the neuron c with the closest prototype vector at time t $\mathbf{m}_c(t)$ to $\mathbf{x}$:

$$c = \arg\min_i \|\mathbf{x} - \mathbf{m}_i(t)\| \tag{4.24}$$

[2] The interpretation as weight vector justifies the expression neuron since then a unit can be seen as artificial neuron with the identity as activation function.

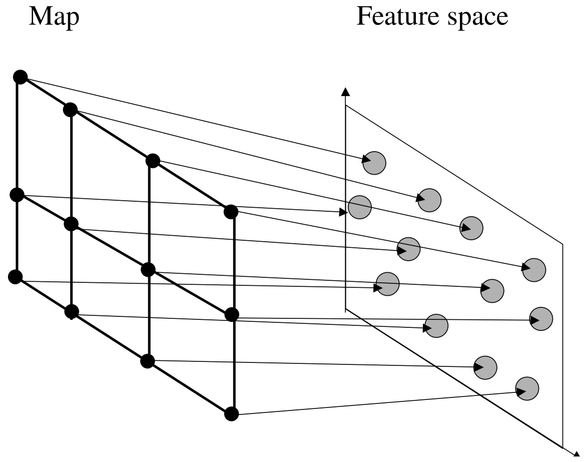


Figure 4.6: Map and feature space of a SOM, both of dimensionality two. For each unit on the grid the corresponding prototype is shown. In the state depicted here, the SOM is already trained, i.e. units which are close together in the map correspond to prototype vectors which are close together in feature space.

where i is an index over all neurons in the map.

3. Now the prototype vector of each unit is altered according to the following update rule:

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)h_{ci}(t)[\mathbf{x} - \mathbf{m}_i(t)] \tag{4.25}$$

Here $\alpha(t)$ is the learning rate, and $h_{ci}(t)$ is the neighborhood kernel around the BMU c. The neighborhood kernel usually decreases with increasing distance between c and the unit i to which the vector $\mathbf{m}_i(t)$ belongs. Furthermore it is non-increasing with respect to the time t. A gauss kernel

$$h_{ci}(t) = e^{\frac{d(c,i)^2}{\sigma_t^2}}$$

with an arbitrary distance function $d(\cdot,\cdot)$ on the grid (e.g. a block distance) and a value of σ_t which depends somehow on t would be suitable for example.

Different stopping criteria are applicable for this algorithm. One common choice is to abort it, when the update quantities for the prototype vectors drop below a threshold.

Equation 4.25 can be interpreted as follows: All prototypes $\mathbf{m}_i$ are moved in the direction of the example $\mathbf{x}$. Prototypes are altered more, the *larger* the distance between $\mathbf{m}_i$ and $\mathbf{x}$, and the *smaller* the distance between the BMU c and the unit i corresponding to the prototype $\mathbf{m}_i$. Thus even prototypes which are far away from $\mathbf{x}$ are altered little, when their corresponding neuron in the map is far away from the BMU c, while other prototypes which are far away as well from $\mathbf{x}$ are altered a lot, when their corresponding unit is close to the BMU c or when it is c itself (see figure 4.7).

Training is often divided in two phases. First a large neighborhood radius is chosen (i.e. h decreases slowly with increasing distance from the BMU) and the learning rate is large as well, so that all prototypes migrate quickly close to their final positions. In the second phase fine tuning is performed with a small neighborhood radius and a small learning rate.

SOMs can also be trained in a batch manner where all data points are considered at one time. That means that all data points play a role for the update of an arbitrary prototype $\mathbf{m}_i$ which belongs to unit

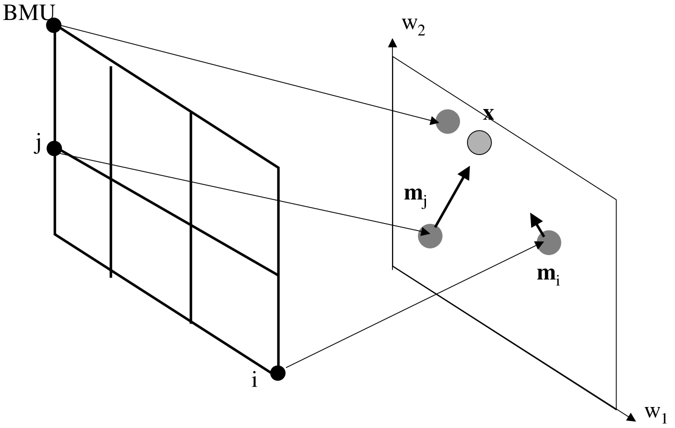


Figure 4.7: Update of prototypes during SOM training. While a prototype corresponding to a neuron which is far away from the BMU is altered little ($\mathbf{m}_i$), another prototype which has a neuron close to the BMU is altered much more ($\mathbf{m}j$).

i:

$$\mathbf{m}_i(t+1) = \alpha(t)\frac{\sum_{j=1}^{R} h_{c_j i}(t)\mathbf{x}_i(t)}{\sum_{j=1}^{R} h_{c_j i}(t)} \tag{4.26}$$

where $j = 1, \ldots R$ is the index over all training examples and $c_j = \arg\max_k \|\mathbf{x}_j - \mathbf{m}_k(t)\|$, i.e. c_j is the BMU for the training example $\mathbf{x}_j$.

Similarly to the sequential training algorithm one can see here that the new prototype vector $\mathbf{m}_i(t+1)$ is influenced most by those data points $\mathbf{x}_j$ for which the BMU c_j is close to the neuron i on the map. For this work the SOM implementation of the MATLAB$^{\text{TM}}$ SOM Toolbox [Vesanto et al., 2000] was used, where batch training is performed in two phases (approximate learning and fine tuning phase) as explained above. Note that batch training and sequential training generally converge to the same results, so that the explanations given above for the sequential training algorithm are valid for the batch training algorithm as well.

4.5 Classification and Regression

To determine the user state for a given feature vector, classification techniques must be applied which compute a mapping

$$f_c : \mathbf{X} \rightarrow C \tag{4.27}$$

where $\mathbf{X}$ denotes the feature space and $C = \{c_1, \ldots c_K\}$ the set of possible class labels.

In contrast to that, task demand is measured on a continuous or at least on an ordinal scale. Therefore regression functions are suitable to estimate the task demand level for given a feature vector. A regression function computes a mapping:

$$f_r : \mathbf{X} \rightarrow \mathbb{R} \tag{4.28}$$

Note that if the dependent variable in a regression task can only have a finite number of values (which is for example the case when this variable is ordinally scaled), also classification techniques can be applied. Then the feature vectors which belong to a particular value of the dependent variable are put into one class, whose class label is that particular value. In this case however the information about the relation between values of the dependent variable gets lost, since class labels are usually unrelated to each other. The task demand levels considered in this work are ordinally scaled, so that both classification and regression techniques are investigated for task demand assessment (see section 6.2). Remember that for exactly the same reasons LDA is applicable to the problem of task demand estimation, but the correlation-based method is expected to perform better, since it respects the ordinal scaling of the considered task demand levels (see section 4.3.3).

Two statistical machine learning methods are applied for classification and regression in this work: Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). Both concepts are explained in the remainder of this section.

4.5.1 Artificial Neural Networks (ANNs)

4.5.1.1 Basic Principle

The structure of a single artificial neuron is depicted in figure 4.8. It receives several inputs $x_1^{(in)}, \dots x_I^{(in)}$, weights each input $x_i^{(in)}$ with a weight w_i, computes the sum of the weighted inputs and finally passes the weighted sum to an "activation" function g. (The term "activation" function is used because of an analogy to biological neurons which generate an axonal response only if the sum of the incoming impulses is large enough: if the weighted sum is above a threshold (often zero) the value of the activation function is positive (or close to one), otherwise it is be negative (or positive but close to zero).) Common activation functions are "squashing functions" which map an arbitrary input to a certain interval (e.g. $g(h) = \tanh(h)$ or $g(h) = \frac{1}{1+e^x}$), linear functions or gaussians. (For the latter two the analogy to biological neurons does not hold.) Mathematically the relation between neuron input and output can be formulated as follows: Let $\mathbf{x}^{(in)} = (x_1^{(in)}, \dots, x_I^{(in)})^T$ be a vector of neuron inputs, and $\mathbf{w} = (w_1, \dots, w_I)^T$ the weight vector of the neuron. Then the neuron computes the output function

$$x^{(out)}(\mathbf{x}^{(in)}) = g([\mathbf{x}^{(in)}]^T\mathbf{w}) \tag{4.29}$$

Often a bias b (which plays the role of the threshold mentioned above) is subtracted from the sum of the weighted inputs before the non-linear activation function is evaluated. Then equation 4.29 becomes

$$x^{(out)}(\mathbf{x}^{(in)}) = g([\mathbf{x}^{(in)}]^T\mathbf{w} - b) \tag{4.30}$$

When setting however $\tilde{\mathbf{x}}^{(in)} = (x_1^{(in)}, \dots, x_I^{(in)}, -1)^T$ and $\tilde{\mathbf{w}} = (w_1, \dots w_I, b)^T$ the formulation of equation 4.29 can be used, where the bias is included implicitly when $\mathbf{x}^{(in)}$ is replaced by $\tilde{\mathbf{x}}^{(in)}$ and $\mathbf{w}$ by $\tilde{\mathbf{w}}$. For simplicity we assume in the following the implicit inclusion of the bias, however we omit the tilde.

The analogy between artificial and biological neurons which are described in section 2.1.2 does not only refer to the activation function but some more similarities can be found as well. The inputs of the artificial neurons correspond to the electrical impulses biological neurons receive via the dendrites. The weight for each input could be interpreted as the distance of the dendrite to the soma since this determines the contribution of dendritic impulses to the overall potential in the soma. Finally the subtraction of a bias and the computation of a (non-linear) activation function correspond to the way impulses are processed in the soma to produce an output impulse which is transmitted via the axon as explained above. The processes in biological neurons however are much more complex than those modeled by artificial neurons. Therefore the analogy which can be established between both concepts is rather weak.

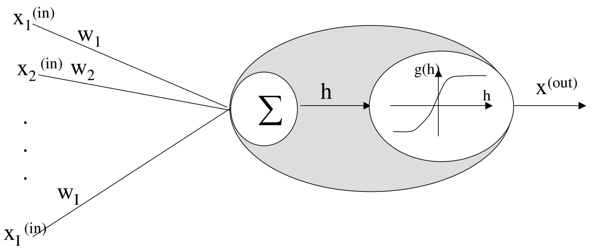


Figure 4.8: A single artificial neuron with a squashing function as activation function (e.g. $g(h) = \tanh(h)$). See text for explanation.

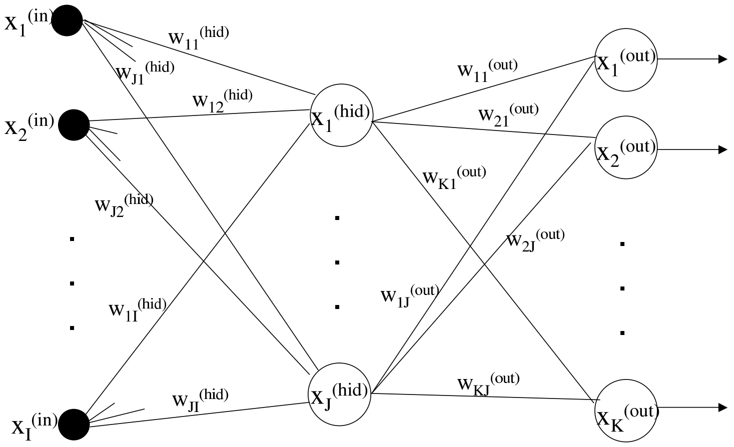


Figure 4.9: A multi-layer feed-forward neural network. See text for explanation.

An artificial neural network combines several artificial neurons, so that the output of one neuron serves as one input for another one. In this work we consider only feed-forward neural networks, where the artificial neurons are grouped in layers and each layer receives inputs only from the previous layer (see figure 4.9). The inputs of the first layer are contained in the input vector $\mathbf{x}^{(in)}$, which can be a feature vector of a classification problem for example. The outputs of the last layer (which are the outputs of the whole network at the same time) can be for instance class labels which are coded according to some convention. All layers except the output layer are also referred to as hidden layers, since their outputs can not be observed directly.

The function which is computed by a multi-layer feed-forward network can also be formulated mathematically. We define $\mathbf{x}^{(out)} = (x_1^{(out)}, \ldots, x_K^{(out)})^T$ to be the vector of network outputs and $\mathbf{w}_k^{(out)} = (w_{k1}^{(out)}, \ldots, w_{kJ}^{(out)})$ to be the weight vector of the k^{th} output unit. Furthermore we define $\mathbf{x}^{(hid)} = (x_1^{(hid)}, \ldots, x_J^{(hid)})^T$ to be the output vector of the hidden layer and $\mathbf{w}_j^{(out)} = (w_{j1}^{(out)}, \ldots, w_{jI}^{(out)})$ to be the weight vector of the j^{th} hidden unit. Then we can combine all output layer weights in a matrix $W^{(out)} = (\mathbf{w}_1^{(out)}, \ldots \mathbf{w}_K^{(out)})$ and the hidden layer weights in a matrix $W^{(hid)} = (\mathbf{w}_1^{(hid)}, \ldots \mathbf{w}_J^{(hid)})$. Finally we define the functions $g^{(out)}(.)$ and $g^{(hid)}(.)$ to be the activation functions for the output layer and the hidden layer respectively which take a vector of weighted sums as inputs and return an output vector whose components represent the outputs of the units of the corresponding layer. The network inputs are given as for the single neuron by $\mathbf{x}^{(in)} = (x_1^{(in)}, \ldots x_I^{(in)})^T$. Now we can write:

$$\mathbf{x}^{(out)}(\mathbf{x}^{(in)}) = g^{(out)}[W^{(out)} \cdot \underbrace{g^{(hid)}(W^{hid} \cdot \mathbf{x}^{(in)})}_{=\mathbf{x}^{(hid)}}] \tag{4.31}$$

In the remainder of this section the application of neural networks for classification and regression tasks is briefly described. A more detailed description can be found in [Bishop, 1995].

4.5.1.2 ANNs for Classification Problems

Let us return to the single neuron whose behavior is described by equation 4.30. Its use for classification is best illustrated geometrically. The weight vector **w** together with the bias b define an I-dimensional hyperplane whose orientation is given by **w** and whose a distance from the origin is $\frac{-b}{||w||}$ (figure 4.10). If we set the neuron's activation function $g(h) = \tanh(h)$, its output is larger then zero if an input vector is located on that side of the hyperplane where the vector **w** points to (which is the case for the example vectors depicted with "+" in the figure), and it is smaller than zero, if the input vector is located on the opposite side (which is the case for the example vectors depicted with "-" in the figure). Finally the neuron's output has the exact value of zero, if the input vector is located *exactly* on the hyperplane.

Now we assign each input vector a target value which is either $+1$ or -1. (For a binary classification problem the target values can be interpreted as class labels.) In the following we define that the input vectors which are depicted with "+" have the target value $+1$ and they are referred to as positive examples, while the input vectors which are depicted with "-" have the target value -1 and they are referred to as negative examples. Thus the example vectors $\mathbf{x}^{(in)}$ in figure 4.10 are all classified correctly if we compute the network predictions by $\mathrm{sgn}(x^{(out)}(\mathbf{x}^{(in)}))$. We conclude that a single artificial neuron is able to discriminate between positive and negative examples as long as they can be separated by a hyperplane. In other words, a single artificial neuron can be used as binary classifier for linearly separable problems.

Naturally a training procedure is required before an artificial neuron can be actually used for classification, since one can not expect that for a random initialization of the vector **w** positive and negative examples are classified correctly. For the simple case where only a single neuron is used, there is an intuitive explanation of the training procedure. Starting with a randomly initialized normal vector, all training examples are considered and whenever a training example is misclassified (i.e. on the wrong side of the hyperplane), the normal vector is moved a bit in the direction of that example if the example is positive, or in the opposite direction if the example is negative (figure 4.11). Mathematically this can be described as a gradient descent on an error function which is described briefly below for the more general case of multi-layer feed-forward neural networks.

Note that if we have I-dimensional input vectors, we obtain an $I+1$ dimensional hyperplane which passes through the origin, if the encode the bias implicitly in the weight vector as explained above. When using an explicit bias, an I dimensional hyperplane is obtained which does not necessarily pass though the origin as shown in figure 4.10. Both formulations are completely equivalent. While the second one is more suitable for visualization (the behavior of a single neuron with two non-bias-related inputs can be easily visualized in a two dimensional coordinate system here), we use the first one in the following because of its mathematical simplicity.

The classes of real-world classification problems can not always be linearly separated. Thus a single neuron is not suitable to address such problems. Using however a multi-layer neural network

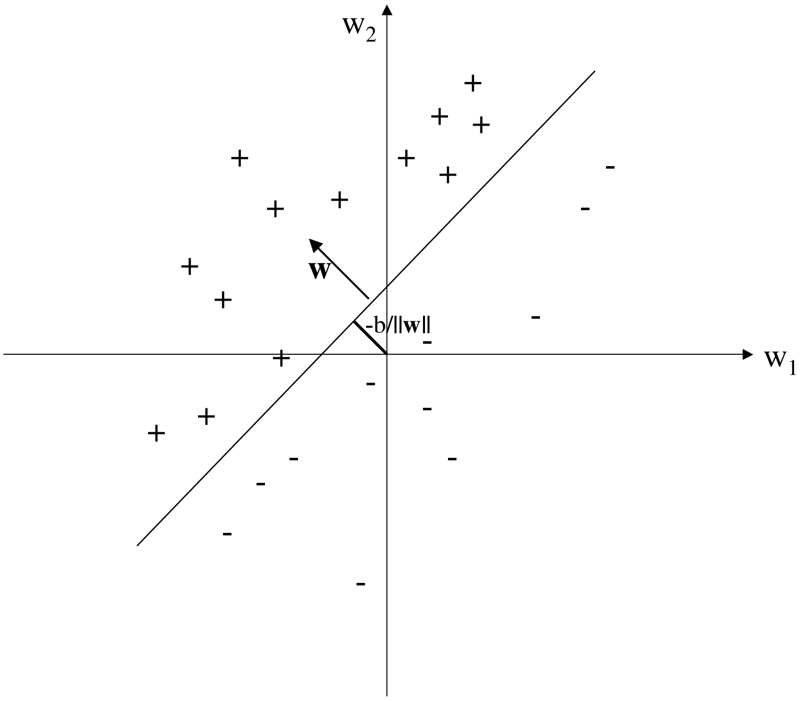


Figure 4.10: Geometric interpretation of the behavior of a single neuron with two inputs, i.e. in a two dimensional feature space. The hyperplane defined by the weight vector **w** has a distance of $\frac{-b}{||w||}$ from the origin. For the example vectors denoted with "+" the neuron outputs a value larger than 0, for the examples vectors denoted with "-" it outputs a value smaller than 0, when $g(h) = \tanh(h)$ is used as activation function.

with at least one hidden layer and a sufficiently large number of neurons in this hidden layer, two classes can be separated in virtually every way. Kolmogorov's theorem even states that a network with two hidden layers and a certain number of neurons in each layer is able to learn every mapping from an arbitrary input space to a continuous output variable. This implies that such a neural network is *theoretically* able to represent an arbitrary continuous function (which is interesting for regression problems) or an arbitrary partitioning of data into two classes (in case of classification problems). The practical relevance of this theorem is limited however since special requirements are imposed on the activation functions which can't always be fulfilled when using standard training techniques. In practice often networks with one hidden layer are used and the number of hidden-layer neurons is adapted to the complexity of the problem.

Up to now only binary classification problems have been considered, where predictions for a vector **x** can be computed via $\text{sgn}(x^{(out)}(\mathbf{x}))$, when the class labels are -1 and $+1$ as explained above. For a multi-class classification problem networks with more than one output unit have to be used and the class labels have to be translated to an appropriate output coding. For the experiments in this work the following output coding was chosen (which is suitable for $g(h) = \tanh(h)$ as activation function):

Let $\{1, \ldots, K\}$ be the labels for K classes. Then K neurons in the output layer are used and class i is assigned a target vector $\mathbf{t}_i$ where all components are -1 except the i^{th} component which is $+1$. Thus if we distinguish 6 classes for example, class 4 would be assigned the target vector $\mathbf{t}_4 = (-1, -1, -1, 1, -1, -1)^T$.

The predicted class i for a given network output $\mathbf{x}^{(out)}$ can now be found by determining that target

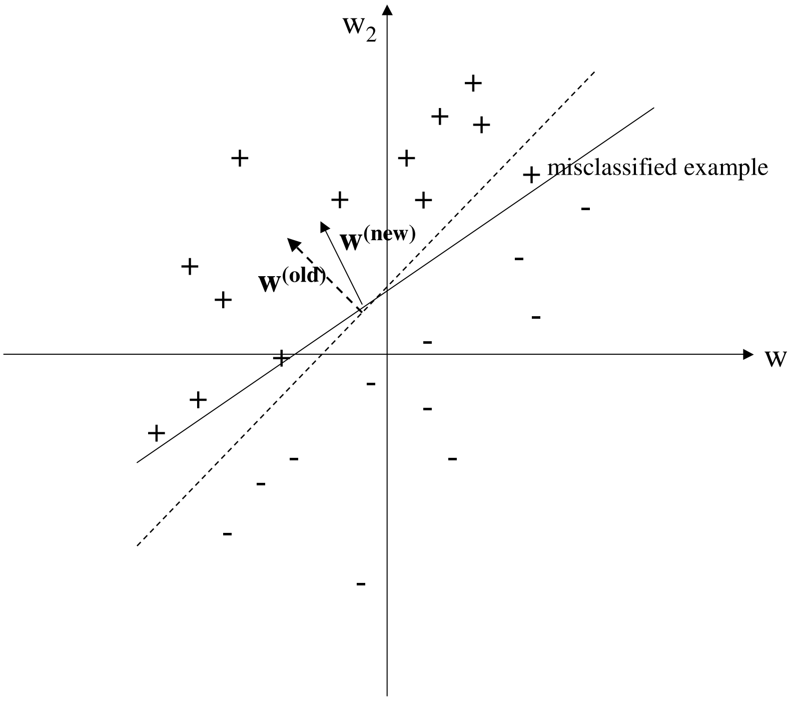


Figure 4.11: Training of a single neuron for its application as binary classifier. The example on the upper right is misclassified by the original hyperplane (dashed line). In the next training step the normal vector of the hyperplane is moved into the direction of the misclassified example so that the new hyperplane (solid line) does not produce this misclassification anymore.

vector for which the euclidean distance to the network output is minimal, i.e.

$$i = \arg \min_{j=1,\ldots,K} ||\mathbf{x}^{(out)} - \mathbf{t}_j||$$

It can easily be shown that this is equivalent to determining the index the largest component of the output vector $\mathbf{x}^{(out)}$, i.e.

$$i = \arg \max_{j=1,\ldots,K} x_j^{(out)}$$

The latter method is computational more efficient, since there is no need to compare the output vector with the target vector for each class.

4.5.1.3 ANNs for Regression Problems

For the estimation of a scalar regression function using a multi-layer feed-forward neural network only one output unit with a linear activation function is required (figure 4.12). If this function is set to the identity, only the weighted sum of the outputs from the hidden layer neurons is computed. No output coding is required here. The output of the network can be interpreted directly as prediction of the dependent variable of the regression function. For the training procedure the given values of the dependent variable from the training examples can be used directly as targets. Note however that it is advisable to normalize target values before training so that they reside in a known interval (and to

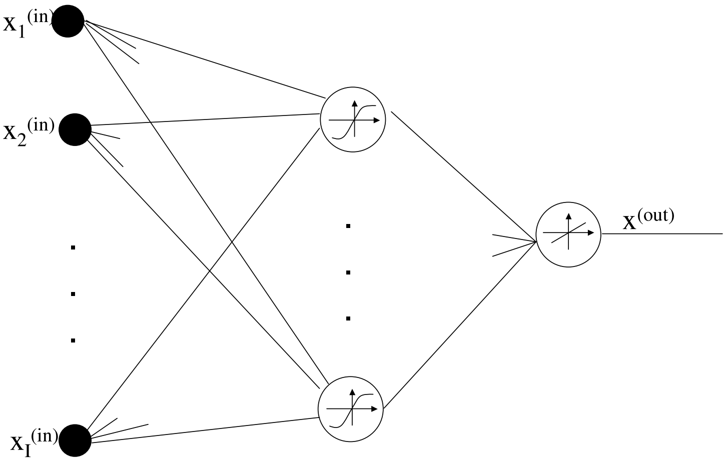


Figure 4.12: Topology of a multi-layer neural network for regression estimation. The hidden layer units use tanh activation functions while the output unit uses the identity as activation function as depicted by the symbols.

unnormalize predictions later) in order to be able to select the initial weights in a way that the training procedure converges quickly (see also section 4.5.1.5).

4.5.1.4 ANN Training

The remaining issue is how a neural network learns the appropriate mapping for a classification or regression task. While this can be explained easily with a geometric analogy for a binary classifier which consists of a single neuron (see above), the training procedure for a neural network is slightly more complex. In this work we use a simple algorithm called error back-propagation for ANN training. Only the principle of this algorithm is illustrated here. For further details and the rather technical derivation of the algorithm refer to [Bishop, 1995].

The goal of error back-propagation is to minimize some error function on the training data. A very common error function is the sum of the squared distances between the targets and the network outputs for all training examples in the training set $TR = \{\mathbf{x}, \mathbf{t}\}$:

$$E(W^{(hid)}, W^{(out)}) = \sum_{(\mathbf{x},\mathbf{t})\in TR} ||\mathbf{t} - \mathbf{x}^{(out)}(\mathbf{x})||^2 \tag{4.32}$$

The relation between the weight matrices $W^{(hid)}$ and $W^{(out)}$ and the computed network output $\mathbf{x}^{(out)}(\mathbf{x})$ is given by equation 4.31.

Minimization of the error function 4.32 is performed using a gradient descent with respect to the network weights. This results in different update equations for the weights in $W^{(out)}$ and $W^{(hid)}$ of the output and the hidden layer. In the update equations for the weights in $W^{(hid)}$ local errors are used which are observed at the units of the output layers. This explains the name error back-propagation for this training procedure.

4.5.1.5 Practical Problems

Some practical problems must be considered for the application of neural networks in classification or regression tasks. They are summarized briefly in this section. A more detailed discussion of these issues is provided in [Bishop, 1995].

- When using the error function of equation 4.32 it is important to adjust the amount of training data for all classes to be approximately equal, i.e. to create balanced training data. Otherwise the network might become biased towards those classes which contain more training examples, since the minimization of the error for the examples of such classes often minimizes the overall error function best.

- If the value of the dot product between input vectors and the weight vector $h = \mathbf{x}^T\mathbf{w}$ is too large, changes of the network weights cause virtually no change in the network output since the slope of sigmoid- and tanh- activation functions for large arguments is extremely small. This may lead to a very slow convergence of the training procedure. A good heuristic to avoid this problem is to initialize the network weights randomly to small values, e.g. from the interval $[-1, 1]$. The effect of small, randomly initialized weights gets lost however when the network inputs are too large. Therefore the network inputs should be either transformed so that their values reside also in a small interval around zero (again $[-1, 1]$ is suitable for instance), or each network weight must be adapted to the range its input, so that at the end the value of h is kept in a reasonable range. In this work features are normalized before they are used as network inputs so that they have values which are at least close to $[-1, 1]$ (see section 4.2.3). Therefore weights are simply initialized randomly in $[-1, 1]$ to assure reasonable fast convergence of the training algorithm.

- Random weight initializations are also the reason that different local minima of the error function are found, when a network is trained several times with the same training data. When now the process of training and evaluation of the network performance on test data is repeated several times, results exhibit certain fluctuations. They can be reduced, although not avoided, when for a given training and test set more than one networks are trained and evaluated. Then majority decisions (in case of classification) or averaging over all network predictions (in case of regression) can be used to obtain more stable estimates for correct class labels or regression function values.

- The problem of overfitting the training data can be treated in several ways for neural networks [Bishop, 1995]. In this work a technique commonly known as early stopping is used. The error on the validation set is computed after each training epoch, using the error function from equation 4.32. The goal is to find that epoch where the validation error has its *global* minimum. Using the network with the weights from this epoch, one assumes that generalization performance and thus the error on unseen data is best (figure 4.13). To find the global minimum of the validation error a heuristic is applied: Each time after a potential minimum (which is not necessarily global) has been found, training is continued for a few more epochs. Only if during these epochs the validation error does not decrease again (or alternatively if it does not fall below the previously found minimum), training is aborted and the weights from the epoch where the validation error was minimal are used. Otherwise training is continued. To make sure that the algorithm stops after a finite number of epochs, training is aborted as well when the training error drops below a certain threshold.

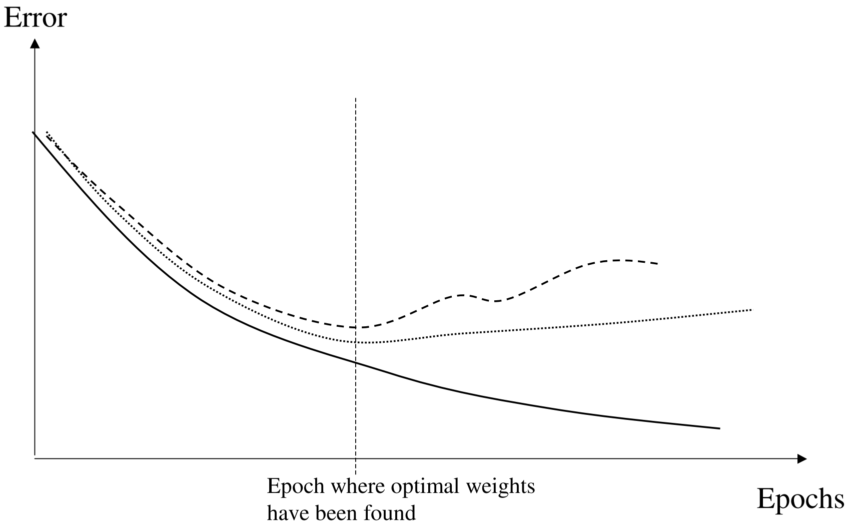


Figure 4.13: Training error (black line) validation error (gray line) and error for unseen data (dashed line) over the number of training epochs. Ideally the error for unseen data and the validation error have their minimum at approximately the same epoch.

4.5.2 Support-Vector-Machines (SVMs)

SVMs are an interesting alternative to ANNs for the construction of classifiers or the estimation of regression functions. There are two essential advantages of SVMs compared to ANNs:

1. The topology of a neural network, i.e. the number of hidden layers and the number of neurons per hidden layer, has to be determined before network training. Therefore either prior knowledge is required to determine the network topology, or time consuming cross validation must be applied to select the best topology out of a huge number of possibilities. For SVMs no such topology selection is required.

2. SVM training algorithms converge to a *unique* solution, i.e. when several SVMs are trained with the same data, they produce exactly the same results on test data, in contrast to neural networks where results fluctuate as explained above.

Furthermore SVMs have usually good generalization properties and no validation set is required to estimate the generalization performance of the learned models, in contrast to ANN training procedures where methods like early stopping require validation data. Instead a penalty term is used which keeps the model complexity low and thus avoids overfitting. Note that similar techniques known as weight decay are also applicable to neural networks [Bishop, 1995]. However the weight of the penalty term plays an important role here and it is often determined using again cross validation (see for instance [Duta et al., 2004]). This problem is less severe for SVMs. More details on how generalization is achieved for SVMs are given in section 4.5.2.5.

In the remainder of this section the basic principle of SVMs is summarized briefly (see for instance [Burges, 1998] for a more detailed introduction) and extensions of the original SVM formulation, i.e. of the use of SVMs as binary classifiers, are described. These include the application of SVMs for multi-class classification problems according to [Crammer and Singer, 2001] and for the SVM-based estimation of regression functions which is described in detail in [Smola and Schölkopf, 1998].

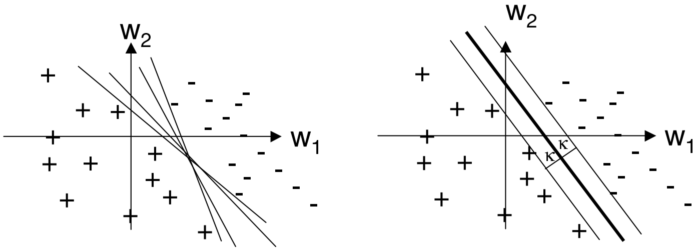


Figure 4.14: Separation between two (linearly separable) classes in a two dimensional feature space found by an ANN (left) and an SVM (right). While the ANN training procedure can converge to several possible solutions, the SVM determines a unique hyperplane which maximizes the margin κ to the positive examples (denoted with ”+”) and negative examples (denoted with ”-”).

4.5.2.1 Basic principle

In their original version SVMs are binary classifiers. The most simple variant, the linear SVM, can separate two classes in a feature space of arbitrary dimension by a hyperplane. The same problem can also be solved by a single artificial neuron (see section 4.5.1.2). The difference between both methods can be understood intuitively when comparing the separations which are found for linearly separable classification problems in a two dimensional feature space by each of the classifiers (figure 4.14).

The single neuron may find different separating hyperplanes (lines in the two dimensional case), dependent on the (local) minimum of the error function to which the training procedure converges. The SVM instead determines a separation between the classes which is in a certain sense optimal. It maximizes the ”security margin” κ to both, the positive (denoted with ”+”) and the negative (denoted with ”-”) training examples.

Mathematically the hyperplane which is found by an SVM is given by

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0 \tag{4.33}$$

where $\mathbf{w}$ is the normal vector of the hyperplane and b an offset which is proportional to the hyperplane’s distance from the origin (the distance is exactly $\frac{b}{||w||}$). An example is classified to be positive if $\mathbf{w}^T\mathbf{x} + b > 0$, otherwise it is classified to be negative. Let now the class label y_i for a training example $\mathbf{x}_i$ be $+1$ if $\mathbf{x}_i$ is a positive example and -1 otherwise. Then the following conditions hold for the training data $TR = \{(\mathbf{x}_i, y_i)\}$ with respect to the hyperplane defined by equation 4.33 which has been determined during training:

$$\forall(\mathbf{x_i}, y_i) \in TR : y_i(\mathbf{w} \cdot \mathbf{x}_i + b \geq \kappa) \tag{4.34}$$

In other words, all training examples have a distance of at least κ from the hyperplane. (Note that the above conditions can only be fulfilled, if the training data is linearly separable.)

To obtain maximal margins, the value of κ has to be maximized. Division by κ and setting $\tilde{\mathbf{w}} = \frac{\mathbf{w}}{\kappa}$ and $\tilde{b} = \frac{b}{\kappa}$ results in the following condition which must be fulfilled for the training data:

$$\forall(\mathbf{x_i}, y_i) \in TR : y_i(\tilde{\mathbf{w}} \cdot \mathbf{x}_i + \tilde{b}) \geq 1 \tag{4.35}$$

Now the value of κ can be maximized via the minimization of $||\tilde{\mathbf{w}}||$. Thus we end up with the following constrained optimization problem which has to be solved to find the desired separation with

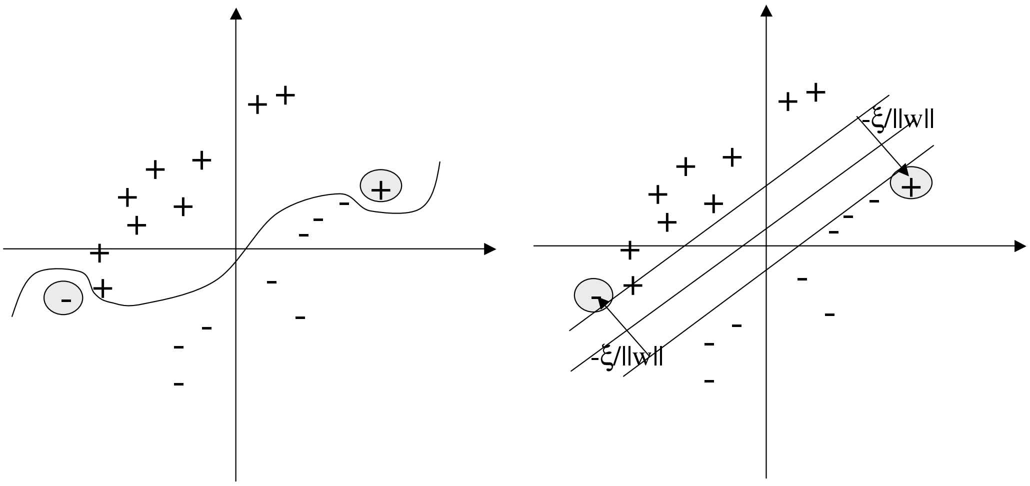


Figure 4.15: Two outlier examples (highlighted in gray) which make a linear separation of the classes impossible (left) and the solution of this problem using soft margins (right).

a maximal margin. (From now on we omit the tilde in equation 4.35 for better readability):

$$\text{Minimize } ||w||$$
$$\text{subject to}$$
$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \; \forall(\mathbf{x}_i, y_i) \in TR \tag{4.36}$$

Note that the final separation is determined by only a few training vectors which are referred to as support vectors. The support vectors have the property that they are located exactly on the security margin.

Figure 4.15 illustrates a situation where only because of two training examples no linear separation between the two classes is possible anymore, although both examples are likely to be outliers. To cope with that problem soft margins are introduced, which allow that training examples violate the security margin at a certain cost. Equation 4.35 becomes now

$$\forall(\mathbf{x}_i, y_i) \in TR: \; y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \tag{4.37}$$

where ξ_i determines how much the margin is violated.

Using soft margins, the optimization problem of equation 4.36 must be extended to:

$$\text{Minimize } ||w|| + C \textstyle\sum_i \xi_i$$
$$\text{subject to}$$
$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i \; \forall(\mathbf{x}_i, y_i) \in TR \tag{4.38}$$

C is a user-specified factor here which determines the cost for the violation of the security margin. High values of C favor little violations which results in a small margin in consequence, lower values of C produce more margin violations thus allowing a larger margin.

The optimization problem 4.38 can theoretically be solved using standard techniques for constrained optimization. However these techniques involve the computation of matrices whose size grows quadratically with the feature vector dimensionality. Therefore the application of standard techniques becomes already unfeasible for feature vectors of medium size because of speed and memory reasons. To overcome that problem several approximations and modifications of the standard optimization algorithms have been developed. For more details concerning the solution of the SVM optimization problem, please refer to [Burges, 1998].

4.5.2.2 Non-linear SVMs

SVMs are also applicable to classification tasks where classes are not linearly separable. In such a case the feature vectors **x** which we assume now to be elements of $\mathbb{R}^N$ are mapped into an euclidean space with a higher (possibly infinite) dimensionality which we call $\mathcal{H}$ using a mapping Φ:

$$\Phi : \mathbb{R}^N \to \mathcal{H}$$

Because of several findings of statistical learning theory (see [Burges, 1998]) it is a plausible assumption that if only the dimensionality of $\mathcal{H}$ is high enough, the classes can again be separated linearly in $\mathcal{H}$. Note however that in this case the security margin is maximized only for the linear separation in $\mathcal{H}$ but not for the (non-linear) separation of the classes in the original feature space. Nevertheless the generalization performance of non-linear SVMs is usually still very good. The optimization problem from equation 4.38 looks as follows for the non-linear case:

$$\begin{aligned} &\text{Minimize } ||\mathbf{w}|| + C \textstyle\sum_i \xi_i \\ \text{subject to} & \\ &y_i\Big(\Phi(\mathbf{w}) \cdot \Phi(\mathbf{x}_i) + b\Big) \geq 1 - \xi_i \ \forall(\mathbf{x}_i, y_i) \in TR \end{aligned} \tag{4.39}$$

The mapping Φ is never computed explicitly, since this would be computationally very expensive for a sufficiently high dimensionality of $\mathcal{H}$ and even infeasible if the dimensionality of $\mathcal{H}$ was infinite. The explicit evaluation of Φ is avoided using a method commonly referred to as ”kernel trick”:

During optimization it turns out, that the vector **w** can be written as linear combination of some training vectors, the ”support vectors”, which we define to belong to a set SV. As explained above, these vectors have the property that they lie directly on the margin and thus uniquely define the separation between the classes. Thus **w** becomes

$$\mathbf{w} = \sum_{i \in SV} \alpha_i \mathbf{x}_i \tag{4.40}$$

where α_i represent truly positive weights for the support vectors $\mathbf{x} \in SV$.

Furthermore during classification, i.e. the evaluation of $sgn(\mathbf{w} \cdot \mathbf{x} + b)$ for some training example **x** and also during optimization only dot products between feature vectors are computed. Therefore we can define

$$k(x_i, x_j) = \Phi(x_i)^T \cdot \Phi(x_j) \tag{4.41}$$

to be the dot product between two feature vectors which have been mapped before to $\mathcal{H}$ using Φ.

Substitution of equation 4.40 in equation 4.39 and application of the identity 4.41 yields in the following optimization problem where no explicit mapping into $\mathcal{H}$ is required anymore:

$$\text{Minimize } ||\mathbf{w}|| + C \textstyle\sum_i \xi_i$$

subject to

$$y_i\Big(k(\mathbf{w} \cdot \mathbf{x}_i) + b\Big) \geq 1 - \xi_i \ \forall (\mathbf{x}_i, y_i) \in TR \tag{4.42}$$

The function $k(\cdot,\cdot)$ is called kernel function and it can be any function which fulfills Mercer's condition (see [Burges, 1998] for an explanation of this condition). If this is the case it can be proved that the kernel function computes a dot product in some high dimensional euclidean space $\mathcal{H}$ which needs not necessarily to be known.

We conclude that the kernel trick for computing the dot product between feature vectors is useful to apply SVMs for non-linearly separable problems without increasing the amount of computational cost tremendously. Only the evaluation of the kernel function needs usually more computation than the evaluation of the dot product in feature space. Therefore the computational effort for non-linear SVMs increases only by a multiplicative constant compared to linear SVMs.

Popular kernel functions are

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \cdot \mathbf{y} + c)^d \tag{4.43}$$

$$k(\mathbf{x}, \mathbf{y}) = e^{||\mathbf{x}-\mathbf{y}||/(2\sigma^2)} \tag{4.44}$$

$$k(\mathbf{x}, \mathbf{y}) = \tanh(\kappa(\mathbf{x}^T \cdot \mathbf{y}) - \delta) \tag{4.45}$$

where d, σ, κ and δ are free parameters which are usually determined using cross-validation. Note that because of the need for cross-validation to tune the kernel parameters, often linear SVMs are preferred to non-linear SVMs when it is not known a priori that the data follows a highly non-linear relationship.

4.5.2.3 Multiclass SVMs

Up to this point only binary classification problems have been considered in the context of SVMs. The problem of user state identification which is addressed in this work involves however the discrimination of more than two classes. One possibility to apply SVMs for multi-class classification is to train an SVM for each class which discriminates the examples of this class from all others. Then the distance from the margin computed by the different SVMs can be used as a criterion to make a final decision for the predicted class, if more than one SVMs decide that an example belongs to ”their” class. In this work a more elegant solution of that issue, proposed in [Crammer and Singer, 2001], is used which addresses the multi-class classification problem for SVMs directly. The function f_c which maps a feature vector to a class label is defined here as follows:

$$f_c(\mathbf{x}; M) = \arg\max_r \{\mathbf{M}_r \cdot \mathbf{x}\} \tag{4.46}$$

f_c simply decides that an example $\mathbf{x}$ belongs to that class r for which the similarity between $\mathbf{x}$ and the prototype vector $\mathbf{M}_r$ for class r (expressed via the dot product of both vectors) is maximal. The matrix M contains the prototype vectors $\mathbf{M}_r$ as lines.

The empirical error of the classifier in equation 4.46 on the training set $TR = \{(\mathbf{x}_i, y_i)\}$ can thus be formulated as:

$$E_{emp} = \frac{1}{|TR|} \sum_{(\mathbf{x}_i, y_i) \in TR} (1 - \delta_{f_c(\mathbf{x}_i; M), y_i}) \tag{4.47}$$

The concept of margin maximization is introduced now by replacing $(1 - \delta_{f_c(\mathbf{x}_i;M),y_i})$ in equation 4.47 by

$$\max_r \{\mathbf{M}_r \cdot \mathbf{x}_i + 1 - \delta_{r,y_i}\} - \mathbf{M}_{y_i} \cdot \mathbf{x}_i \tag{4.48}$$

Thus the error for an example $(\mathbf{x}_i, y_i)$ is zero only if the similarity between $\mathbf{x}_i$ and the prototype of its correct class $\mathbf{M}_{y_i}$ is at least larger by one than the similarity between $\mathbf{x}_i$ and the second similar prototype. Otherwise the error the error increases continuously.

The optimization problem to be solved for multi-class SVMs is similar to the problem for binary SVMs:

$$\text{Minimize } ||M||^2$$

subject to

$$\mathbf{M}_{y_i} \cdot x_i + \delta_{y_i,r} - \mathbf{M}_r \cdot \mathbf{x}_i \geq 1 \ \forall r \ \forall (\mathbf{x}_i, y_i) \in TR \tag{4.49}$$

The incorporation of slack variables and kernels is analogous to the procedure for binary SVMs which has been explained in sections 4.5.2.1 and 4.5.2.2. For more details please refer to [Crammer and Singer, 2001]. In this work an extension of the publicly available SVM^{light} software [Joachims, 1999] is used for SVM based multi-class classification, which embeds the above formulation in a more general framework (see [Tsochantaridis et al., 2004] for details).

4.5.2.4 Regression SVMs

For the estimation of regression functions the training examples $(\mathbf{x}_i, y_i) \in TR$ contain a feature vector $\mathbf{x}_i$ as in case of classification and a continuous value y_i which is the target value of the regression function f to be predicted.

The simplest case is to learn only a linear function:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{4.50}$$

Usually it is unfeasible in the presence of noise to fit data exactly to a linear function (even if the data follows a linear relationship). For ordinary least square regression this problem is addressed by minimizing the squared error between the values of the estimated function and the targets of the training examples. In case of SVM-based regression, noise is considered simply by requiring the training examples to lie within an ϵ-tube around the estimated function in order to cause no cost (see also figure 4.16). Thus the following constraints are imposed for the estimation of f with respect to the training data:

$$\forall (\mathbf{x}_i, y_i) \in TR$$

$$y_i - \mathbf{w} \cdot \mathbf{x}_i - b \leq \epsilon$$

and

$$\mathbf{w} \cdot \mathbf{x}_i + b - y_i \leq \epsilon \tag{4.51}$$

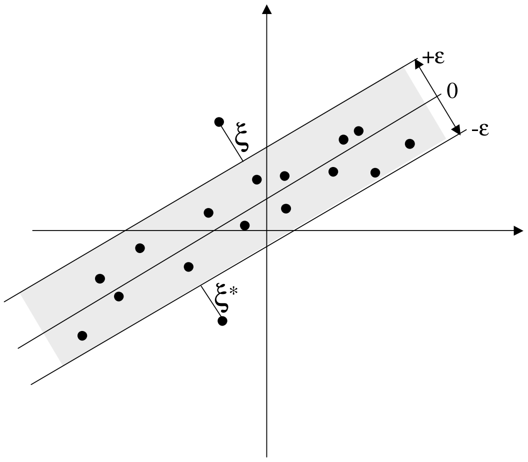


Figure 4.16: Use of soft margins for SVM regression. The ϵ-tube where targets of the training data may lie without causing a cost is shaded in gray. Only outliers which do not lie within the ϵ-tube cause a cost which is proportional to ξ or ξ^* respectively.

Additionally one requires that the flattest possible function is chosen out of all functions which fulfill the above constraints. In analogy to classification SVMs, flatness is defined here in terms of $||\mathbf{w}||$, i.e. $||\mathbf{w}||$ is to be minimized to obtain the flattest function. For a more rigid justification of that, please refer to [Smola, 1998]. Intuitively the flattest function can be thought as that function which adapts least to the noise of the training data, which is particularly evident for non-linear regression functions.

Thus we have the following optimization problem for SVM regression:

$$\text{Minimize } ||\mathbf{w}|| \text{ subject to}$$

$$y_i - \mathbf{w}\cdot\mathbf{x}_i - b \leq \epsilon$$

$$\text{and}$$

$$\mathbf{w}\cdot\mathbf{x}_i + b - y_i \leq \epsilon \quad \forall(\mathbf{x}_i, y_i) \in TR \tag{4.52}$$

Similarly to the classification case the introduction of soft margins makes sense here too, in order to reduce the influence of single outliers on the estimated function (figure 4.16). Then the optimization problem becomes

$$\begin{aligned} \text{Minimize } ||\mathbf{w}|| + C\sum_i \xi_i + \xi_i^* \\ \text{subject to} \end{aligned} \tag{4.53}$$

$$\begin{aligned} y_i - \mathbf{w}\cdot\mathbf{x}_i - b \leq \epsilon + \xi_i \\ \mathbf{w}\cdot\mathbf{x}_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \quad \forall(\mathbf{x}_i, y_i) \end{aligned} \tag{4.54}$$

Exactly in analogy to classification SVMs, kernels can be used as well to estimate non-linear regression functions. Note however that here again the use of kernels requires cross-validation to determine appropriate kernel parameters.

For the solution of the optimization problems related to SVM regression the same techniques as for SVM classification are applied. The software SVM^{light} [Joachims, 1999] is used in this work for SVM-based regression function estimation.

4.5.2.5 Generalization Properties of SVMs

Now that the basic concepts of SVMs are introduced their generalization properties shall be analyzed briefly. The *actual* goal of each classification or regression task for which a mapping $f(\mathbf{x}) = y$ must be learned, is to minimize some error function for unknown data (i.e. the actual error E_{actual}). The data is assumed to be generated by a probability distribution $P(\mathbf{x}, y)$.

$$E_{actual} = \int c\Big(\mathbf{x}, y, f(\mathbf{x})\Big) dP(\mathbf{x}, y) \tag{4.55}$$

where c is an arbitrary loss function such as the squared error $(y - f(\mathbf{x}))^2$

Since $P(\mathbf{x}, y)$ is usually not known, the actual error can not be computed exactly. Neural networks (without regularization techniques such as early stopping or weight decay, see section 4.5.1.5) approximate the actual error using the empirical error on the training set TR:

$$E_{emp} = \frac{1}{|TR|} \sum_{(\mathbf{x},y) \in TR} c(\mathbf{x}, y, f(\mathbf{x})) \tag{4.56}$$

which is clearly suboptimal since it makes overfitting very likely. Therefore in case of SVMs a penalty term is added to E_{emp} so that the following error function is minimized:

$$E_{SVM} = E_{emp} + \lambda ||\mathbf{w}|| \tag{4.57}$$

It has been shown above that the minimization of $||\mathbf{w}||$ corresponds to finding the hyperplane (in feature space for linear SVMs, in $\mathcal{H}$ for non-linear SVMs) with the maximal margin to the classes to be separated for a classification problem, or to finding the flattest function which approximates the data for a regression problem. The above formulation shows now that there is a relation between these criteria and the approximation of the actual error in case of SVMs. The minimization of $||\mathbf{w}||$ is used here to avoid overfitting since the overall error can not be minimized simply by minimizing the empirical error only. Therefore SVMs have usually good generalization properties although no extra cross-validation is required.

4.5.3 Multiple Linear Regression

Multiple linear regression is a very simple method for the estimation of regression functions. In contrast to SVMs (with kernels) and multi-layer neural networks which can estimate non-linear functions, only linear relationships can be obtained using this method. Given the training data $TR = \{(\mathbf{x_1}, y_1), \ldots (\mathbf{x}_R, y_R)\}$, the goal is to learn a function

$$f(x) = \mathbf{w}^T \cdot \mathbf{x} + b \tag{4.58}$$

such that the empirical error for the training set TR is minimized. In ordinary least square (OLS) regression this is the mean square error

$$E_{ols} = \frac{1}{|TR|} \sum_{(\mathbf{x}_i, y_i) \in TR} (y_i - f(\mathbf{x}_i))^2 \tag{4.59}$$

Now we explain how a regression function is found using OLS regression. Let the training data be

For the sake of clarity we eliminate first the parameter b from equation 4.58 by setting

$$\begin{aligned} \mathbf{w} &:= (w^{(1)}, \ldots w^{(N)}, b)^T \\ \mathbf{x}_i &:= (x_i^{(1)}, \ldots, x_i^{(N)}, 1)^T \text{ for } i = 1, \ldots R \end{aligned}$$

where $x_i^{(j)}$ denotes the j^{th} component of the i^{th} feature vector and $w^{(j)}$ the j^{th} regression coefficient. Then equation 4.58 becomes

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

Now we define $\mathbf{y} = (y_1, \ldots, y_R)^T$ and

$$X = \left(\mathbf{x}_1^T \vdots \mathbf{x}_R^T \right)$$

Using the above definitions the regression coefficients $\hat{\mathbf{w}}$ which minimize the error from equation 4.59 are the best approximative solution of the system of linear equations:

$$\mathbf{y} = X \cdot \mathbf{w}$$

Obviously an exact solution for equation 4.5.3 can only be found if the matrix X has a rank of N which is usually not the case. Note that an exact solution of the equation would yield in an empirical error of zero, since if $\hat{\mathbf{w}} = \mathbf{w}$ we have

$$0 = ||\mathbf{y} - X \cdot \hat{\mathbf{w}}|| = \sum_{(\mathbf{x_i}, y_i) \in TR} (y_i - \hat{\mathbf{w}}^T \mathbf{x}_i)^2 = E_{ols}$$

Normally the problem 4.5.3 is however either under determined or over determined. In that case an approximate solution can be computed which minimizes E_{ols}, using the pseudo inverse X^+ of X. This matrix has the important property that the estimate

$$\hat{\mathbf{w}} = X^+ \cdot \mathbf{y}$$

for $\mathbf{w}$ minimizes $||\mathbf{y} - X \cdot \mathbf{w}||^2$ which also minimizes E_{ols}. The pseudo inverse of a matrix is usually computed using a singular value decomposition and its computation is possible even if the matrix is not square and of full rank. For a description of more properties of the pseudo inverse and an explanation of the algorithm for singular value decomposition please refer to [Golub and Loan, 1996].

More information about multiple linear regression methods and the statistical analysis of the obtained regression function can be found in [Chatterjee and Price, 1991]. Note however some conditions must be fulfilled so that statistical methods for the interpretation of the regression function can be applied. In particular the features are required to be uncorrelated and the distribution of the variable to be predicted is assumed to be gaussian. Both conditions are not fulfilled when we apply OLS regression to the problem of task demand assessment from high dimensional EEG data.

Note finally that ordinary least square regression minimizes only the empirical error without performing any regularization in order to attempt to minimize the actual error in contrast to SVMs.

Therefore one should expect that SVMs have better generalization properties than OLS regression. Furthermore SVMs can estimate regression functions even in cases of high feature dimensionality and sparse training data. In case of OLS regression this leads to strongly under-determined systems of linear equations for which even the computation of a pseudo inverse is not possible anymore.

4.6 Performance Measurements

In this section some measures to evaluate the performance of classification or regression functions are introduced. This overview should help the reader to interpret the results presented in chapter 6 correctly.

4.6.1 Performance Measurements for Classification Problems

As explained in section 4.5 a classification function assigns a class label c_k drawn from a finite set of labels $C = \{c_1, \ldots, c_K\}$ to a given feature vector $\mathbf{x}$: $f(\mathbf{x}) = c_k$. Let the class labels be now positive integers for simplicity, i.e. $C = \{1, \ldots, K\}$. Then for a given test set $TE = \{(\mathbf{x}, y)\}$, where y denotes the correct class label for the feature vector $\mathbf{x}$, the performance of a classification function on this test set can be represented in a confusion matrix CM:

$$\begin{aligned} CM &= \{c_{ij}\}_{i,j=1}^{K} \text{ with} \\ c_{ij} &= \sum_{(\mathbf{x},y)\in TE} \delta_{y,i} \cdot \delta_{f(\mathbf{x}),j} \end{aligned} \tag{4.60}$$

The entries of the confusion matrix can be interpreted as follows: The element c_{ij}, i.e. the element in the i^{th} line and the j^{th} column contains the number of test examples which belong to class i and which are classified by the function f to belong to class j. Thus in the diagonal of CM the number of those examples is found which are classified correctly.

The accuracy A_i for class i is defined as the ratio of the number of examples belonging to class i which were classified correctly to the number of all examples belonging to class i in the test test. It can be calculated from the confusion matrix:

$$A_i = \frac{c_{ii}}{\sum_{j=1}^{K} c_{ij}} \tag{4.61}$$

The average accuracy A over all classes is then defined as:

$$A = \frac{1}{K} \sum_{i=1}^{K} A_i \tag{4.62}$$

Note that if test sets are not balanced, i.e. if the different classes do not contain an equal amount of data, it is important to distinguish between the average accuracy and the total accuracy $A^{(t)}$ which we define here to be the ratio of all correctly classified examples to all examples in the test set:

$$A^{(t)} = \frac{\sum_{i=1}^{K} c_{ii}}{\sum_{i,j=1}^{K} c_{ij}} \tag{4.63}$$

It is evident that classes containing more examples contribute more to the total accuracy, if the test set is not balanced. This introduces unwanted biases in the results, since one can not assume that user states, for which much data is available from our data collection, also occur most frequently in real life situations. In this work unbalanced test sets are used in order to exploit all available data for testing. This has the advantage that accuracies for those classes containing many test examples are more reliable. The above problem of biased results in case of unbalanced test sets is avoided by reporting always average accuracies instead of total accuracies.

In the following accuracies for different classes or average accuracies for complete test or validation sets are reported in most cases to evaluate the system performance, since this measure is more concise than a confusion matrix. Only for a few analyses confusion matrices are considered as well. This is especially interesting, when the accuracy for one or more classes is not good, since then the classes which are confused most with each other by the classifier can be identified in the confusion matrix.

Note that only the term accuracy is used in the next sections, since it becomes clear from the context whether the average accuracy for a whole data set or the accuracy for a particular class is meant. Usually the mean value over the accuracies for all recording sessions from a particular data collection is reported, however in some cases also accuracies for particular recording sessions are given. Also also here it can be inferred from the context which kind of accuracy is meant.

Average accuracies for classification problems can only be compared, when they were obtained for the same number of classes, since the ”chance accuracy” $A^{(c)}$ obtained as average accuracy by a trivial classifier which always predicts the same class, varies with the number of classes:

$$A^{(c)} = \frac{1}{|C|} \tag{4.64}$$

Therefore an average accuracy of 60% would be still an acceptable result for the discrimination of 10 classes ($A^{(c)} = 10\%$ here) but only little above the chance accuracy of 50% for a binary classification problem. A figure which evaluates the performance of a classifier *independently* from the number of classes is the normalized expected loss, which relates the average *error* $E = 1 - A$ to the error $E^{(c)}$ produced by the trivial classifier which always selects the same class :

$$NEL = \frac{E}{E^{(c)}} = \frac{1 - A}{1 - A^{(c)}} \tag{4.65}$$

The smaller the value of NEL the better the performance of a classifier, no matter how many classes are discriminated. Therefore the normalized expected loss can also be used to compare results from classification problems with a different number of classes. Note that for ”reasonable” values of E, i.e. for $0 \leq E \leq E^{(c)}$, the normalized expected loss resides in $[0, 1]$ and it has the following properties:

$$E = 0 \Leftrightarrow NEL = 0$$
$$E = E^{(c)} \Leftrightarrow NEL = 1$$

This is not the case when the average *accuracy* A is related to the chance accuracy $A^{(c)}$. The value of $\frac{A}{A^{(c)}}$ is bounded by $\frac{1}{A^{(c)}}$ for $A = 1$, i.e. it still depends on the number of classes while the normalized expected loss is 0 when $A = 1$, regardless of the number of classes. For the above reasons the normalized expected loss is used in the following to compare results when different numbers of user states are considered by a classifier.

When accuracies for different parameter configurations of the system are compared, it is important to determine whether improvements are significant. Let A_{C_1} and A_{C_2} be vectors containing the accuracies for the test sets of different recording sessions obtained with two different system configurations

C_1 and C_2. If the mean value over the components from the difference vector $A_{C_2} - A_{C_1}$ is now significantly larger than zero, this means that configuration C_2 is significantly better than C_1. Significances are reported here always on the 5% level if nothing else is mentioned explicitely, i.e. the probability is less than 5% that C_2 is not better than C_1 for *arbitrary* data, although one decides the contrary because of the result of the significance test which is based on the accuracies for the *given* test data. Significance was tested throughout this work using right-tailed t-tests, i.e. the alternative hypothesis was that the mean over the components of $A_{C2} - A_{C1}$ was larger than zero (which corresponds to a null-hypothesis that the mean is smaller than zero).

4.6.2 Performance Measurements for Regression Problems

As already explained in section 4.5 the different task demand levels considered in this work are ordinally scaled and they can be predicted from EEG data as well with regression and classification methods. Also the performance of a system predicting an ordinally scaled variable such as task demand levels can be evaluated with methods which are commonly used for the evaluation of regression or classification functions. Since such methods for classification functions have already been described in the previous section, this section concentrates on methods for the evaluation of regression functions and their modifications to make them suitable for the prediction problems given here. Furthermore it is described how class labels from the predictions of a regression function can be derived.

A very simple method to analyze the performance of a regression function is the squared error SE between the targets y and the predictions of the function $f(\mathbf{x})$ for a test set $TE = \{(y, \mathbf{x})\}$. The squared error is usually defined as:

$$SE = \sqrt{\frac{1}{|TE|} \sum_{(\mathbf{x},y)\in TE} (f(\mathbf{x}) - y)^2} \tag{4.66}$$

Since test sets are unbalanced for the reasons explained above also in case of the task demand data, i.e. the amount of test data for each task demand level is not equal, equation 4.66 has to be modified such that the contribution of each task demand level to the overall error is considered equally. Let the different task demand levels be now positive integers, i.e. they belong the a set $C = \{1, \ldots, K\}$. Furthermore let N_i be the number of examples available for task demand level i. Then the squared error for unbalanced data sets SE_{ub} can be defined as follows:

$$SE_{ub} = \sqrt{\frac{1}{K} \sum_{i=1}^{K} \frac{1}{N_i} \sum_{(\mathbf{x},y)\in TE} \delta_{y,i}(f(\mathbf{x}) - y)^2} \tag{4.67}$$

This formulation has the advantage that it accounts for unbalanced data sets, but it equals to the squared error from equation 4.66 for a balanced data set.

Also the correlation coefficient r between predictions and targets is useful for the evaluation of regression functions. While a low squared error indicates that targets match predictions very well, this is not necessarily the case for a correlation coefficient close to one. A high correlation can already be obtained if the *difference* between two targets is always similar to the *difference* between the corresponding predictions, regardless of the offset between targets and predictions. (Note that nevertheless a low value of SE usually implies a high value of r but not vice versa.) A common definition of r is:

$$r = \frac{\sum_{(\mathbf{x},y)\in TE}(f(\mathbf{x}) - \overline{f(\mathbf{x})})(y - \bar{y})}{\sum_{(\mathbf{x},y)\in TE}(f(\mathbf{x}) - \overline{f(\mathbf{x})})^2 \cdot \sum_{(\mathbf{x},y)\in TE}(y - \bar{y})} \tag{4.68}$$

where $\overline{f(\mathbf{x})}$ denotes the mean over all predicted values and $\bar{y}$ the mean over all targets. As in case of the squared error, this basic formulation of the correlation coefficient does not account for unbalanced data sets. This problem can be solved by computing correlation coefficients $r^{(i)}$ separately for each task demand level i (by considering only those examples $(\mathbf{x}, y) \in TE$ for which $y = i$) and then taking the average over all $r^{(i)}$ for $i = 1, \dots K$. Thus the correlation coefficient r_{ub} for unbalanced data sets is defined as:

$$r_{ub} = \sum_{i=1}^{K} r^{(i)} \tag{4.69}$$

Note that r_{ub} does not equal to r for a balanced data set. However it can be shown that the differences between both are not very large for balanced data sets, when only a small number of task demand levels is considered and the variances of the examples are not too different for the different task demand levels. Both assumptions are often fulfilled in practice.

Even more insight into the relation between targets y and predictions $f(\mathbf{x})$ can be gained by computing a linear regression function which uses the targets as dependent variable and the predictions as independent variable:

$$y = b_1 \cdot f(\mathbf{x}) + b_0 \tag{4.70}$$

Similarly to the correlation coefficient r, the the slope of the regression function b_1, is close to one when the difference between two targets is always similar to the difference between the two corresponding predictions. The coefficient b_0 reflects the offset between targets and predictions, i.e. it indicates whether the predicted values are systematically smaller or larger than the targets.

Also classification accuracies can be obtained for the predictions of task demand using a regression function. For that purpose it is necessary that each task demand prediction is assigned a label, namely the target value for some task demand level. As above we assume that the different task demand levels are taken from a finite set of integer numbers $C = \{1, \dots, K\}$. Then a function $l(\mathbf{x})$ can be defined which assigns each prediction $f(\mathbf{x})$ its closest target value as label:

$$l(\mathbf{x}) = \arg\min_{c \in C} |f(\mathbf{x}) - c| \tag{4.71}$$

The obtained labels can now be interpreted as predicted class labels, so that all methods for the evaluation of classification functions from section 4.6.1 can be used to evaluate the system performance, even when a regression method has been used for the prediction of task demand. In the following results for task demand estimation are reported mostly in terms of classification accuracies because of the conciseness of this figure. Only for more detailed analyses the modified performance measures for regression functions from above are applied.

Chapter 5

Data Collection

To be able to conduct the experiments which are reported in chapter 6, EEG data had to be collected first. In this chapter the technical recording setup and the scenarios for the collection of user state and task demand data are described, and some figures illustrating the amount of collected data are given.

5.1 Recording Setup

Most EEG data has been collected using a standard EEG-cap (an ElectroCap$^{\mathrm{TM}}$, [Electro-Cap International, Inc.,], figure 5.1) which is usually used for clinical EEG recordings. This recording device is equipped with 19 $Ag/AgCl$ electrodes at the positions defined by the international 10-20 system (see figure 2.12). For our data collection 16 electrodes at the positions Fp1, Fp2, F3, F4, Fz,F7, F8, T3, T4, T5, T6, P3, P4, Pz, O1 and O2 were selected for recording. These electrodes cover all cortex regions but the motor cortex (see figure 2.23), since we assume that motor activity is of little importance for the identification of user states or the assessment of mental task demand. We are aware of the possible relation between different user states or levels of task demand and changes in facial expression which should have correlates in the motor cortex. However, the corresponding muscular activity is partly captured by the frontal EEG electrodes and it remains to be investigated whether this activity can be identified in the EEG of the motor cortex as well.

Fewer data were recorded with a headband, in which we sewed-in four $Ag/AgCl$ electrodes with a tip size of a few millimeters over the forehead (positions Fp1, Fp2, F7, F8) (see figure 5.2). A

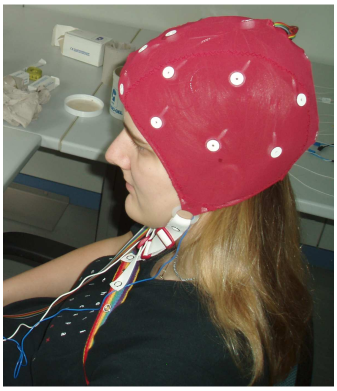


Figure 5.1: The ElectroCap$^{\mathrm{TM}}$ EEG-cap used for the data collection conducted for this work.

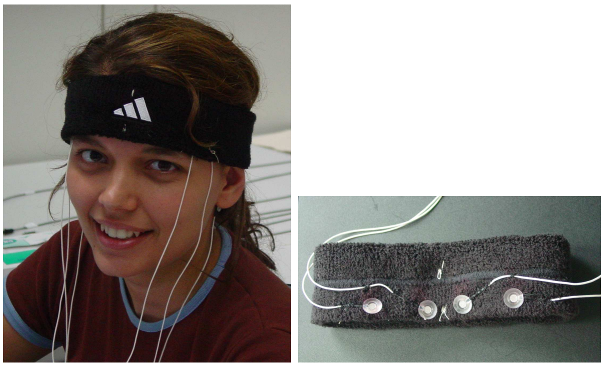


Figure 5.2: Headband for EEG recording (left), electrodes attached on the headband at positions Fp1, Fp2, F7 and F8 (right).

disposable electrode at the back of the neck was used as ground electrode. The advantages of the headband over the ElectroCap$^{\text{TM}}$ are that it is more comfortable to wear and easier to attach, two major factors for everyday life applications. Also, not electrode gel gets in contact with the hair. Last but not least, the position and number of electrodes in the headband compared to the ElectroCap$^{\text{TM}}$ proves to be sufficient for the identification of user states and the estimation of task demand in many cases (see sections 6.1.8 and 6.2.9). Note however that eye-activity and artifacts introduced by facial muscles are most pronounced in the forehead EEG which is recorded using the headband. It remains to be investigated whether this information is helpful or detrimental for the given tasks.

Electrodes at the positions A1 and A2 were chosen as references for both recording devices. For the headband recordings references were placed directly at the mastoids, for the ElectroCap$^{\text{TM}}$ recordings at the ear lobes. The signals from both electrodes were averaged before amplification (unipolar recording with technical reference, see section 2.2.1.2). This electrode montage has the advantages that the signals from all electrodes are directly comparable to each other and that, in contrast to bipolar recordings, they represent approximations of the actual cortical activity. Furthermore electrodes at these positions are nearly indifferent towards physiological processes, i.e. they capture only little muscular artifacts, ECG and EEG from the temporal cortex. All reference electrodes were reusable $Ag/AgCl$ electrodes.

For signal amplification and digitization a VarioPort$^{\text{TM}}$ amplifier [Becker, 2003] which 16 EEG channels was used (figure 5.3). Table 5.1 summarizes the technical characteristics of the amplifier.

Data can be transferred instantaneously from the amplifier to a computer via an RS232 port, so that online processing can be performed, if the amount of data per unit of time does not exceed the serial port's maximal capacity of 115200 Bits per second. This corresponds to 28 electrode channels which can be recorded simultaneously, when data is recorded with a sampling frequency of 256Hz. This sampling frequency was used for all data collections made for this work, although sampling with a lower frequency should be sufficient to avoid aliasing when considering the amplifier's upper cutoff frequency of 60Hz (see table 5.1). For technical reasons the slope of the band pass filter implemented in the amplifier is very small however, so that we decided to choose this rather high sampling frequency.

5.2 User State Data

Data for the following user states was recorded:

Amplification factor	2775
Input Range	±450μV
A/D conversion	12 Bit (4096 steps)
Resolution	0,22 V / Bit
Frequency Range	0,9 ... 60 Hz

Table 5.1: Technical characteristics of the Varioport^{TM} EEG amplifier used for data recording.

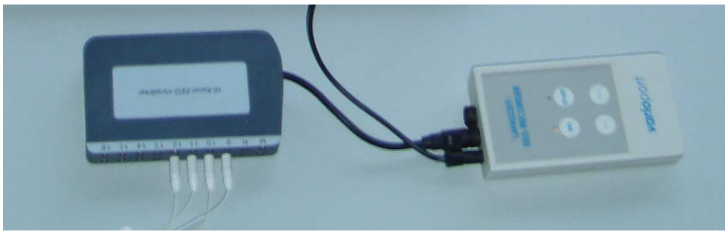


Figure 5.3: The Varioport^{TM} EEG amplifier. Left: the actual amplifier, right: the recorder which controls the amplifier, stores recorded data and established the connection to a computer.

Resting (R): Subjects were asked to relax and not to concentrate on a particular issue. Eyes had to stay open and most subjects preferred looking out of the window. Little to no background noise was present during the resting periods, each lasting about 70 seconds.

The requirement to keep eyes open during the resting period should make sure that resting is not simply identified by very pronounced α-activity in the EEG which can usually be observed only in a relaxed state when eyes are closed. Instead a typical resting period during a meeting or a lecture should be simulated where eyes are not necessarily closed.

Listening to a talk (L): Subjects were given a talk, mostly about a medical topic, which contained a lot of facts to memorize and descriptions of complex processes, so that extensive use of the working memory and visual imagination was required to be able to follow the talk. This is very common for talks where no additional visual information supporting the speech is available. To make sure that subjects listened attentively to the talk, they were asked to summarize its content afterwards.

Perceiving an audio-visual presentation (P): A short presentation slide presentation was given about a scientific topic with which the recorded subject was not familiar. Thus at least medium attention was required to be able to understand the presentation. Images and graphical explanations of facts were largely used to make sure that there were auditory *and* visual stimuli in contrast to user state listening. After the presentation subjects were asked to summarize it with the help of the slides which should guarantee their attention during the presentation.

Reading an article (RE): An article from a popular news magazine of medium length (one to three pages) had to be read. Subjects were allowed to select an article which appeared interesting to them out of a large collection. This should make sure that they remembered the article well enough to be able to summarize it afterwards.

Summarizing the read article (RS): Directly after reading, subjects had to summarize the read article. To facilitate this task and to enable subjects to talk long enough so that enough data could be collected, they were allowed to look from time to time at the text they had read before.

Performing arithmetic operations (A): Arithmetic operations had to be performed on a sheet of paper. Depending on the mathematical background of the subjects, different tasks were selected: multiplications, divisions, summations and subtractions of four digit numbers, matrix multiplications or the computation of simple derivatives and integrals.

Not only the arithmetic task, but also the talk and the slide presentation were adapted to the background of the recorded subject, so that in all cases a reasonable amount of mental resources should be required for task execution. This is important, since in cases of very low task demand, the neural correlates of different user states might be too weak (see section 2.3.2). Mental overload on the other hand might overwhelm the recorded subject and then lead to disengagement and thus again to weak EEG correlates of the currently recorded user state.

All user states were recorded consecutively. After each state or block of states (reading and summarization of the read article are considered as block) there was a resting period. The order of the other user states was chosen randomly in order to avoid biases in the data due to a fixed order.

When a classification function is trained, it is important to avoid biases due to non-equal amount of training data per class (see also section 4.5.1.5). This can either be done by balancing the training data so that for each class approximately the same number of samples is available or by introducing weight factors for samples of different classes (similar to prior probabilities). Since the latter was difficult to integrate in the classifier implementations used for this work, we decided to balance the data, thus accepting that some data had to be discarded.

As explained above, the resting periods lasted about 70 seconds only, while for the other user states much more data per recording session was acquired. Therefore the data for user state resting would be the limiting factor for amount of data per class after balancing. This problem was overcome at least partly by concatenating for each session the data from two resting periods so that at least 140 seconds of data from user state resting were available. Note that the duration of the resting periods could have been increased as well, or more than two resting periods could have been concatenated. Subjects reported however that is was difficult to remain relaxed for a longer period then about 70 seconds during the experiments, which excludes the possibility of recording longer resting periods. Concatenation of more than one resting period on the other hand would introduce more variability in the data for user state resting only, thus making the system behave systematically different for this user state compared to the others[1]. For these reasons we chose the compromise to concatenate only two resting periods of 70 seconds length.

Data was collected from different subjects at different locations using either the ElectroCap^{TM} or the headband. All subjects were university students which volunteered for the experiments. Subjects were not paid for their participation. In the remainder of this thesis the following data sets are distinguished for the user state data:

CMUSubjects: Six computer science students (four males, two females) between 23 and 33 years of age were recorded at Carnegie Mellon University in Pittsburgh (USA). None of them was a native English speaker, but they all had very good knowledge of the English language. 8767 seconds of data in total were collected. In the following identifiers (U1), . . ., (U6) are used to refer to the recording sessions from the subjects of this data set.

UKASubjects: Three university students (two males, one female) with varying academic backgrounds between 23 and 26 years of age were recorded at University of Karlsruhe (Germany). One of

[1] Preliminary experiments have shown that classification accuracy increases for user state resting, when data from more than one resting period (from the same recording session!) is contained in the test and training set.

them was not a native German speaker. 9232 seconds of data are available from this data collection. The **UKASubjects** recording sessions are referred to using the identifiers (U7a), (U7b), (U8a), (U8b) and (U9), where the letters ‘a’ and ‘b’ are used to distinguish the particular recording sessions when one subject was recorded twice.

HeadBandSubjects: 3267 seconds of EEG data with the headband were collected at the University of Karlsruhe from two female university students with 21 and 23 years of age. The identifiers (U10) and (U11) refer to the two **HeadBandSubjects** recording sessions in the following.

Note that the amount of data in seconds given for the different data sets includes only EEG data recorded for one of the six user states described above, i.e. only data which can actually be used for subsequent processing. During the explanation of the recording procedure and during the time which subjects needed to prepare for the next state to be recorded no data was acquired.

In total data from eleven recordings with the ElectroCap$^{\text{TM}}$ and two recordings with the headband are available. Table 5.2 shows for each user state the mean, the minimum and the maximum amount of data over all these recording sessions. In figure 5.4 this information is displayed separately for each for the three data sets **CMUSubjects**, **UKASubjects** and **HeadBandSubjects**. Finally, in figure 5.5 each recording session is considered separately and mean, minimum and maximum duration in seconds over all recorded user states is shown.

Two interesting observations can be made from table 5.2 and figures 5.4 and 5.5:

- The mean value for the state resting is lowest of all user states (see table 5.2 and figure 5.4), which suggests that the amount of data for this user state limits the size of the usable training data per subject in many cases, since training sets are balanced. Indeed it can be seen from figure 5.5 that the duration of the shortest user state is often around 140 seconds (which corresponds to the concatenation of two resting periods lasting about 70 seconds). Closer inspection of the data from the different recording sessions reveals that the user state with this minimum duration is mostly resting. Remember that the amount of available data for this user state was already enlarged by concatenating the data from two resting periods. For the reasons explained above no more data are available for that user state however.

- Furthermore it is interesting to compare the average durations of a recording session of the three data sets: For the **CMUSubjects** data the average duration is 1461 seconds which is significatnly less than the average duration for the **UKASubjects** data (1846 seconds) and the **HeadBandSubjects** data (1861 seconds). Furthermore it can be seen in figure 5.5 that the mean amount of data for one user state is smaller for the **CMUSubjects** recording sessions (sessions (U1), . . ., (U6)) compared to the other recording sessions (except sessions (U9) and (U10)). When looking at the single states, a significant difference in duration can be found for the states perceiving a presentation (P) and summarizing the read article (RS).

  As explanation for the difference for user state (P) might be, that for both data sets presentations were held by a native German experimenter who talked German to the subjects which were recorded at Karlsruhe (**UKASubjects** and **HeadBandSubjects** data sets) and English to the subjects which were recorded at Pittsburgh (**CMUSubjects** data set). Thus it is well possible, that the experimenter was more brief in English than in his native language (where he naturally knows better to express himself), although for the English presentations exactly translated versions of the German slides were used. A reason for the difference in duration for user state (RS) could either be that the articles read by the subjects in Pittsburgh (from the ”TIME” magazine) were simply shorter than those read by the subjects in Karlsruhe (from the magazine

User State	Mean	Minimum	Maximum
(R)	145	143	149
(L)	264	169	388
(P)	360	238	495
(RE)	408	223	646
(RS)	206	124	345
(A)	252	139	496
Complete session	1636	1267	2132

Table 5.2: Mean, minimum and maximum amount of data in seconds over all recording sessions for each user state. In the last line mean, minimum and maximum length of a complete recording session are shown.

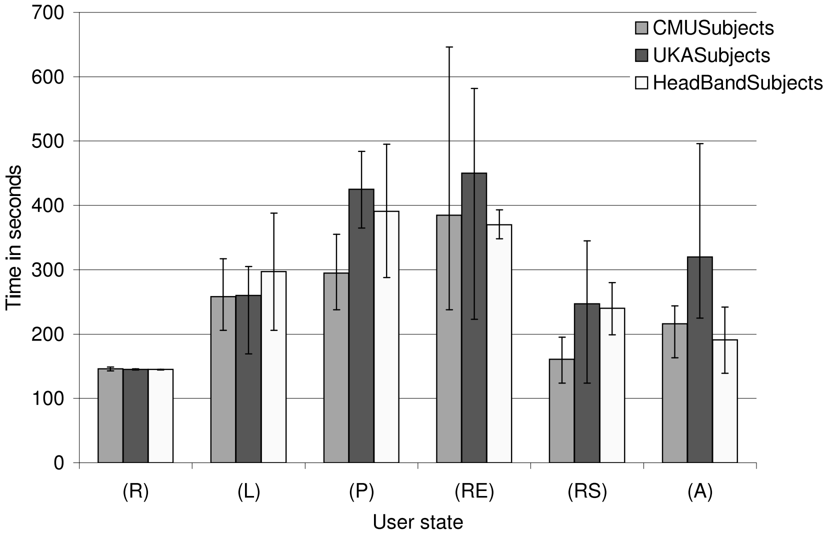


Figure 5.4: Mean amount for data in seconds for each user state over all recording sessions of the three data sets **CMUSubjects**, **UKASubjects** and **HeadBandSubjects**. The shortest and longest duration of each state is depicted by the whiskers.

”Der Spiegel”). On the other hand it is possible that because of differences in style in both magazines, articles in the ”TIME” magazine contained less facts to report about than the articles of ”Der Spiegel”.

5.3 Task Demand Data

Mental task demand data was collected while a 15 to 20 minutes long slide presentation with varying difficulty levels was given to the recorded subjects. During the presentation the laptop screen (slides and mouse pointer) and the voice of the experimenter giving the presentation were recorded on a video tape. Directly after the presentation the recoded subjects were shown the video and asked to evaluate

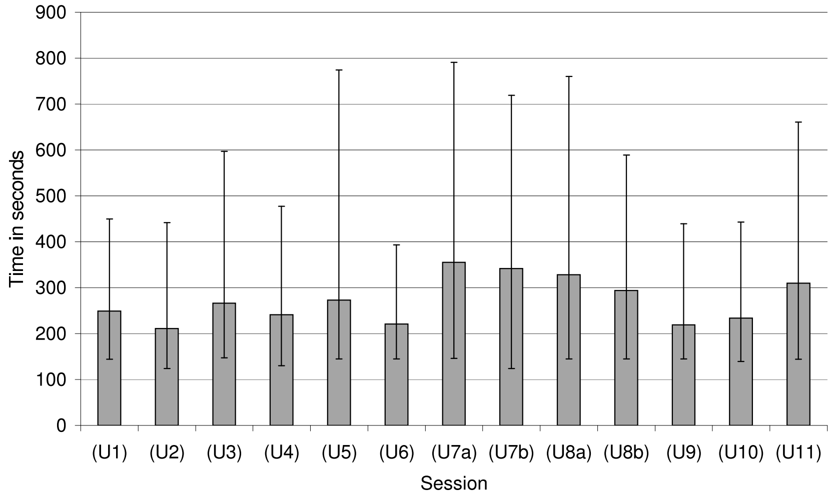


Figure 5.5: Mean duration of one user state for each recording session. The duration of the shortest and longest state for each session is depicted by the whiskers.

their task demand over time which they experienced during the *initial* perception of the presentation. The following task demand levels could be chosen:

Low (L): It is possible to follow the presentation without any problems and without larger mental effort. Everything is understood, also details.

Medium (M): Some mental effort is required to be able to follow the presentation. Still everything is understood, possibly except a few details.

High (H): Almost all available mental resources are required to be able to keep up with the presentation. Details are mostly not understood anymore, however the essence of the presented content is still clear. Task demand level high should also be chosen, if the talk is *not* understood anymore, but still all mental resources are mobilized to try to understand it at least.

Overlad (O): The presentation is not understood anymore, the subject became overwhelmed, then disengaged and makes no effort to gain the plot again.

The goal for the design of the presentations was that all task demand levels are experienced equally by the recorded subjects during their perception. Since background knowledge of the subjects varied a lot, it was attempted to adapt topics and the way of their presentation individually for each subject to reach that goal.

Many subjects reported that it was very difficult to estimate the own task demand which they experienced during the initial perception of the presentation. Often they had difficulties to distinguish between their task demand during the presentation and during its video replay, and they were temped to simply evaluate their current task demand while watching the video. Furthermore almost all subjects reported problems to find the exact transition between task demand level high and overload, i.e. the state where they still tried to understand the presentation but they could not and the state where they became overwhelmed and therefore did not pay attention anymore. The question how reliable the subjects' self-estimates of task demand are, is addressed at the end of this section.

Two different task demand data sets were recorded at the University of Karlsruhe:

ElectroCapSubjects: In total 7690 seconds of task demand data were recorded using the ElectroCap^{TM}. Six university students (three males, three females) with varying academic backgrounds between 23 and 26 years of age volunteered for that data collection. Two subjects were none-native German speakers. The recording sessions from the different subjects of this data collection are referred to in the following using the identifiers (T1), . . ., (T6). One subject was recorded twice; the corresponding recording sessions are therefore identified with (T3a) and (T3b).

HeadBandSubjects: 1918 seconds of task demand data in total were collected from one male university student (28 years of age) and one female university student (21 years of age) using the headband. Both volunteered for the data collection, one subject was a non-native German speaker. These recording sessions are assigned the identifiers (T7) and (T8).

Table 5.3 shows mean, minimum and maximum duration for the different task demand levels over all recording sessions. In figure 5.6 the duration of each task demand level, the mean duration of one level and the total duration of the whole presentation is shown for each recording session. Note that one task demand level could occur several times during the presentation. The durations of the particular task demand levels displayed in table 5.3 and figure 5.6 result from the concatenation of all data segments corresponding to the same level.

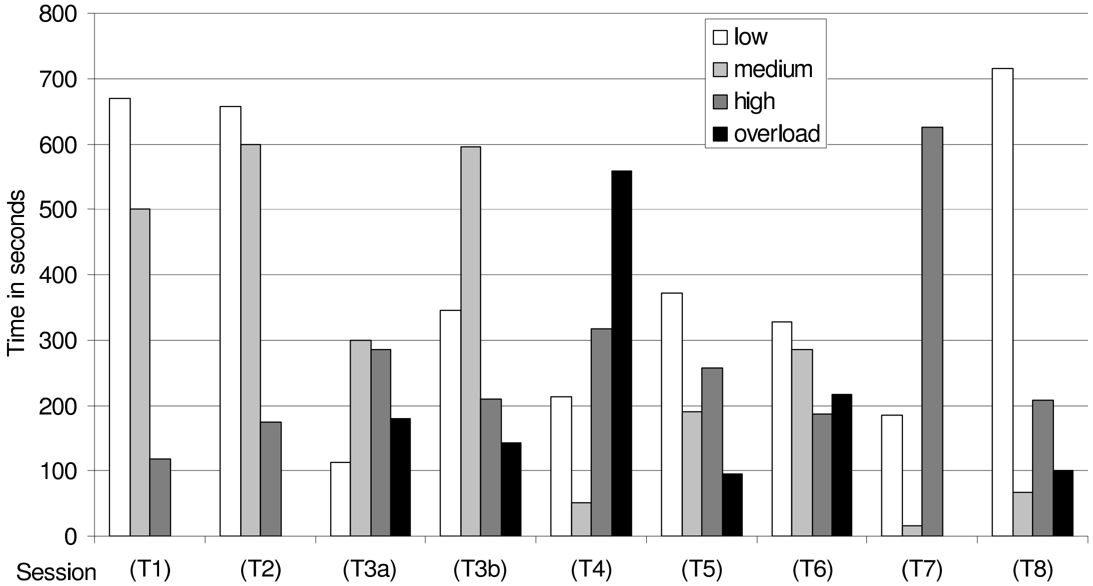
From the first column of table 5.3 one can see that the mean duration of 144 seconds for task demand level overload is the shortest of all task demand levels. It is in the same range as the mean duration of the user state for which in average least data is available, namely resting with 145 seconds (see table 5.2). The *minimum* durations of the different task demand levels (second column of table 5.3) are however much shorter than those of the different user states (see table 5.2). This indicates that in general a balanced training set for task demand prediction contains less data than a balanced training set for user state identification. In figure 5.6 one can see that indeed for many recording sessions much less than 145 seconds of data are available for at least one task demand level. This is not the case for the user state data, where for one state at least 124 seconds and in most cases even more were recorded, see figure 5.5.

Note also the large fluctuations of the available data per task demand level for the different recording sessions which can be seen in figure 5.6. The goal for the design of the presentations to make subjects experience each task demand level equally could not be reached. This shows, that it is extremely difficult to anticipate the effects of a presentation on a listener, even if his or her background knowledge is well known. Precisely this finding can serve as a hint, that EEG based task demand assessment can be very useful during a presentation in order to stay informed about changes in task demand of the audience or to be able to analyze the effect of the own talk later.

It was mentioned above that most subjects found it very difficult to estimate their task demand during the initial perception of the presentation given the video replay. Therefore we wanted to find out how reliable their estimates are. This was done by asking the four subjects recorded in the sessions (T2), (T3b), (T4) and (T6) to repeat the evaluation of their task demand using the video a second time. These second evaluations took place two to eight weeks after the initial recording session. The reason why the period between both evaluations was chosen to be sufficiently long was to avoid that subjects simply remembered and reproduced their decisions made during the first evaluation. Instead subjects were asked to estimate the task demand they experienced while watching the video, i.e.

Task demand level	Mean	Minimum	Maximum
low	400	112	716
medium	289	15	600
high	265	118	626
overload	144	0	558
Total	1098	826	1431

Table 5.3: Mean, minimum and maximum amount of data in seconds over all recording sessions for the different task demand levels.



Mean	322	358	219	323	285	229	254	206	273
Total	1287	1431	876	1293	1140	916	1017	826	1092

Figure 5.6: Amount of data in seconds for each task demand level and each recording session. Mean and total amount of data for the particular recording sessions are shown below the diagram.

while perceiving the original presentation a second time. Assuming that the subjects' background knowledge did not change much between both evaluations and that the learning effect after the first perception of the presentation was small enough to be neglected, the estimates from both evaluations should be comparable. (According to the subjects, both assumptions were fulfilled in all cases.)

Figure 5.7 shows the results of both evaluations for all four subjects. One can see that for session (T3b) the overlap of time segments belonging to the same task demand level in both evaluations is relatively high, while there is much less overlap for sessions (T2), (T4) and (T6).

From each of both evaluations another partitioning of the data, i.e. another assignment of task demand levels to data segments, can be derived. Furthermore a third partitioning can be obtained when considering only those data segments, which have been assigned the same task demand levels in both evaluations. In the following the identifiers **Eval1**, **Eval2** are used to refer to the partitionings of the data according to the first or second task demand evaluation and **EvalCombined** is used to refer to the combination of both.

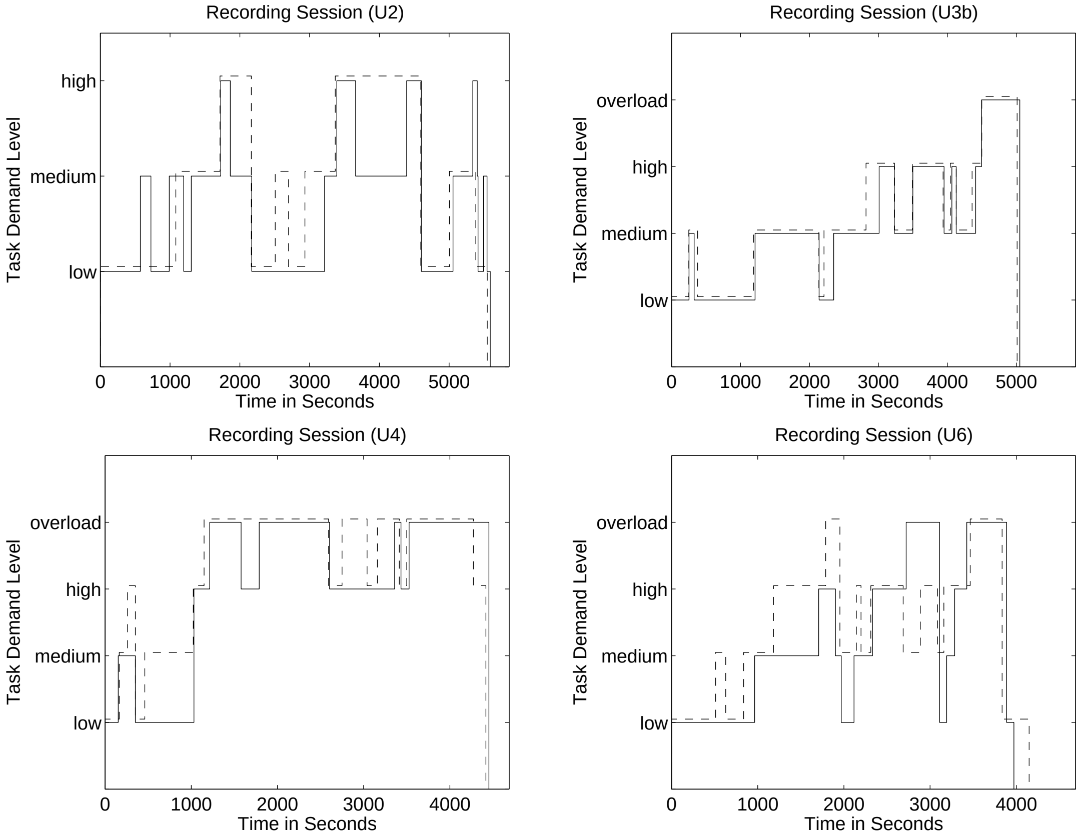


Figure 5.7: Comparison of the first evaluation (solid line) and the second evaluation (dashed line) of the task demand experienced during the presentation for recording sessions (T2), (T3b), (T4) and (T6)

Table 5.4 compares the amount of data per task demand level which is available for the different partitionings. The percentages given for each task demand level of partitioning **EvalCombined** represent the overlap of time segments belonging to this task demand level in both evaluations relative to the maximum amount of data available for this task demand level in one of both. They confirm the findings from figure 5.7 that the overlap between both evaluations is largest for recording session (3b) compared to sessions (T2), (T4) and (T6). Closer inspection of the table shows that at least for task demand level low the overlap is larger than 62% for sessions (T2), (T3b) and (T6) and for task demand level overload an overlap of at least 71% for sessions (T3b) and (T4) was reached.

Note furthermore that the reason for little overlap is in some cases that almost all data segments belonging to one task demand level in one evaluation belong to the same task demand level in the other evaluation but not vice versa. An example for that is task demand level high of recording session (T2) where almost all data segments belonging to that task demand level in the first evaluation belong to task demand level high as well in the second evaluation. One the other hand much more data segments were classified to belong to high task demand in the second evaluation than in the first evaluation (see figure 5.7). (In such a situation it can be seen in table 5.4 that the amount of data for partitioning **EvalCombined** is not much smaller the the minimum amount of data available for partitioning **Eval1** or **Eval2**.) This finding suggests that ”core segments” belonging to one task demand level were reliably identified at least in one evaluation, i.e. in that evaluation where only the same data segments were chosen to belong to a particular task demand level as in the other evaluation.

	Eval1				**Eval2**				**EvalCombined**			
Session	(L)	(M)	(H)	(O)	(L)	(M)	(H)	(O)	(L)	(M)	(H)	(O)
(T2)	657	600	174	0	572	419	429	0	484 (74%)	246 (41%)	156 (36%)	0
(T3b)	345	596	210	142	291	583	278	131	290 (84%)	524 (88%)	206 (74%)	131 (92%)
(T4)	213	51	318	558	70	169	185	707	67 (31%)	25 (15%)	115 (36%)	504 (71%)
(T6)	328	286	186	217	257	267	394	137	206 (62%)	96 (34%)	148 (38%)	95 (42%)

Table 5.4: Comparison of the amount of data in seconds available for each task demand level for data partitionings **Eval1**, **Eval2** and **EvalCombined** for recording sessions (T2), (T3b), (T4) and (T6). The percentages given for the last columns relate the amount of data per task demand level for partitioning **EvalCombined** to the maximum amount of data obtained from one of the partitionings **Eval1** and **Eval2**.

(In the above example this is the first evaluation.)

An interesting question related to the overlapping segments of the same task demand level which are contained in the partitioning **EvalCombined** in general, is whether predictions can be improved when they are used for training and testing only. The underlying hypothesis here is that task demand estimates are more reliable for these segments, since they were assigned the same task demand level in both evaluations. This hypothesis is further investigated in section 6.2.8.

The generally poor overlap indicates, how difficult it is to subjectively evaluate the own task demand during the perception of a presentation. This agrees as well with the comments of the recorded subjects. Therefore it would not be too surprising, if training and test data were very noisy and feature vectors belonging to different task demand levels were very similar, so that the prediction of the correct task demand level becomes extremely difficult.

Chapter 6

Experiments and Results

A lot of experiments have been conducted on the collected user state and task demand data, in order to analyze the impact of the different processing steps on the whole system for user state identification and task demand estimation and to optimize the overall system performance by configuring each processing step appropriately. The results of these experiments are reported and discussed in this chapter using the performance measures introduced in chapter 4.6.

6.1 User State Identification

6.1.1 Experimental Setups

Several experimental setups are distinguished for the user state data in the following:

UD **U**ser and session **d**ependent experiments: Data portions of the same recording session were used for training (80% of the whole session), testing and validation (each 10% of the whole session). All sessions from the **CMUSubjects**data collection (recording sessions (U1)-(U6)) were used for this type of experiments. Training sets comprise here 668 seconds of data in average, validation sets 84 seconds and the test sets 146 seconds. If not indicated differently, results reported for this setup are always averages over the accuracies obtained for the test sets (or validation sets) of all recording sessions of this setup.

HB The same type of experiments as for setup **UD** were conducted on the **H**ead**B**andSubjects data (recording sessions (U10) and (U11)). The average training set length is here 681 seconds, the validation set length 84 seconds and the test set length 163 seconds. Also here usually averages over the test set (or validation set) results from both recording sessions are reported in the following.

UI **U**ser **i**ndependent experiments: For the **CMUSubjects** data collection the system was trained on data from five recording sessions (in average 3338 seconds of training data) and tested on data from the remaining session in a round-robin manner. For comparability of the results the same test sets as for setup **UD** were used and results are reported as averages over all recording sessions as above. For validation 1082 seconds of data from the sessions (U7a), (U8a) and (U9) from the **UKASubjects** data collection were available. If results for a particular recording session are given for this setup in the following, the session number refers always to that session from which the test data is taken.

SI User dependent but **s**ession **i**ndependent experiments: For the subjects from the **UKASubjects** data set which were recorded twice (recording sessions (U7a), (U7b), (U8a) and (U8b)) the system was trained using 80% of the the data of one session (in average 672 seconds) and tested on the 10% of the data recorded from the same subject in the other session (in average 198 seconds). Thus there are always results from 4 test sets available for averaging in the following. Note that for this setup no good validation data exists. For the early stopping regularization method for ANNs 10% of data from the training sessions were used which bears the risk of overfitting however and does not reflect the properties of a real session *independent* experimental setup. For these reasons and since the data from the test session must not be touched for validation (this comes too close to parameter tuning on the test set), no cross validation was applied to determine system parameters for setup **SI**. Instead parameter values were chosen which are somewhere in-between the parameters estimated for setups **UD** and **UI** following the intuition that properties of setup **SI** are in-between those of the other two setups.

Note that the length of the different data sets in seconds corresponds to the number of feature vectors per data set, since features are extracted from two seconds long time windows which overlap one second as explained in section 4.2. Note furthermore that the fraction of data for training has been balanced after its extraction. That applies to the validation set, too, however not for the test set which does not have to be balanced for the reasons explained in section 4.6. This also explains why the available amount of data is different for validation and testing, even when originally the same fraction of data was extracted for both.

As described above for setups **UD** and **UI** *exactly* the same test sets were used, which allows direct comparison of the results. In contrast to that the test sets for setups **SI** and **HB** are both based on data from other recording sessions. This has to be taken into account when comparing the results obtained for these setups with others. For this reason accuracies obtained for setups **UD** and **UI** are clearly separated from those obtained for setup **SI** or setup **HB** in the following sections.

6.1.2 ANN topology selection

Before ANN-based classifiers can be trained and used to make predictions, their topology, i.e. the number of hidden layers and the number of neurons per hidden layer, must be determined. If the topology can not be inferred from a priori knowledge, this is usually done using cross-validation. The above described validation sets however are already required for early stopping regularization (see section 4.5.1.5). Therefore training, validation and test data from recording sessions (U7a), (U8a) and (U9) (the ”validation sessions” of setup **UI**) was taken to determine the network topology for setup **UD**. For setup **UI** additional data from the same sessions which is however not contained in the original validation sets was used for topology selection. (The original training sets were kept here.) No topology selection was performed for setup **SI**, because no appropriate validation data was available for this setup as explained above. Since ANNs with one hidden layer are suitable for most classification problems in practice, cross-validation was used here only to determine the number of hidden units for a network with a single hidden layer.

Figure 6.1 shows the accuracies obtained for the topology selection data sets of setup **UD** and setup **UI** for different numbers of hidden units. One can see that for both setups the fluctuations of accuracies are very small. In particular they are smaller than the standard deviations (depicted by the whiskers) over five repetitions of the same experiments. Note that these standard deviations are still high (in particular for setup **UI**), although for each of the five experiments five networks were trained with the same data and a majority decision over the predictions of all networks was made to obtain

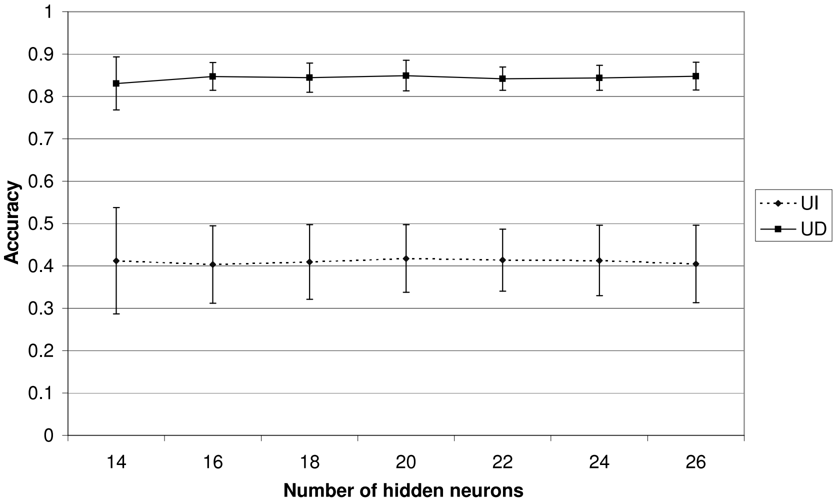


Figure 6.1: Accuracies for different numbers of hidden neurons obtained on the topology selection data of setups **UD** and **UI**. The whiskers show the standard deviations over five repetitions of the same experiment.

a final prediction. We conclude that the problem of user state identification is insensitive towards the number of neurons in the hidden layer, at least for the range between 14 and 26 neurons examined here. Therefore 20 neurons in the hidden layer were used for the remaining experiments with ANNs on the user state data.

The insensitivity towards the number of neurons in the hidden layer might be an indicator that there is a comparatively simple separation between the given classes in feature space, perhaps even a linear separation. Much more experiments concerning the network topology would have to be done to gain deeper knowledge about the structure of the feature space, which is however too time consuming to be within the scope of this thesis. The hypothesis that there exists a linear separation between classes in feature space is revisited in the next section however where the ANN results are compared with results obtained using linear SVMs.

6.1.3 Classifier comparison

In this section the performance of ANNs and SVMs is compared for the baseline configuration of the proposed system. This configuration has the following characteristics:

- no artifact removal
- no averaging over previous features
- normalization with method **GlobalNorm**
- no feature reduction
- Linear SVMs or ANNs with 20 neurons in one hidden layer and with tanh as activation function for all units are used for classification. To reduce the fluctuations of the ANN results, predictions

are always majority decisions over five networks. Furthermore ANN experiments are always repeated five times. Therefore mean values and standard deviations over all repetitions are reported in the following.

Table 6.1 shows the results achieved by both classifiers for the baseline system for setups **UD UI** and **SI**. One can see that ANNs perform better than SVMs for setups **UD** and **SI**. A significant difference in results can only be found for setup **UD** however. (For ANN experiments only the mean values over the five repeated experiments are considered for the significance test.) For setup **UI** SVMs perform slightly but not significantly better than ANNs. Two conclusions can be drawn from this:

1. It seems that the given classes can almost (but not perfectly) be linearly separated, since results achieved with the linear SVM classifier are not too much, but significantly, below those achieved with non-linear ANNs for setup **UD**. One can assume that particularly for this setup the obtained results reflect the properties of the real separation between the different classes best. For the other setups results are largely influenced by the variability in the training data and between the training and test data.

2. On the other hand SVMs seem to be able to cope better with exactly this variability which is strongest for setups **SI** and **UI**. Here ANNs are not (significantly) better anymore than SVMs, possibly because SVMs have better generalization properties.

Finally the standard deviations over repeated experiments with ANNs must be considered which are larger for the setups **SI** and **UI** than for setup **UD**. A reason for this could be the larger variability between training, validation and test data for the first two setups. Because of this variability it is more likely, that the network configuration found during training corresponds to a position in the test set error function which is far away from a (local) minimum. Therefore small differences in the found weight configuration when training is repeated several times may lead to large differences in the test set error.

Fluctuations in results for repetitions of the same experiment make it difficult to compare the effect of different parameter configurations on the overall system performance. Furthermore it can become problematic for practical applications that ANNs with early stopping regularization approximate the actual error for unseen data using the training and validation set only. It may happen that by coincidence the classification performance for newly recorded data is bad, just because this data does not fit to the error approximation which is biased towards the validation and training set. The risk for that to happen is particularly high when only little (or little representative) validation data is available, which is unfortunately often the case for our experimental setups.

For these reasons we decided to use exclusively SVMs in the following experiments. It must be kept in mind however, that in some cases, e.g. in a situation similar to setup **UD** where ANN results fluctuate little, they achieve possibly better results than linear SVMs. In practice such cases will be rare however. Furthermore it must be noted that only linear SVMs were used in all experiments in this work and non-linear SVMs would possibly outperform non-linear ANNs in all cases, since they are also able to cope with non-linear separations between different classes *and* they have better generalization properties than ANNs as hypothesized above.

since they would address better the slightly non-linear separation of the different user states. The use of non-linear SVMs requires however extensive parameter tuning which is very time consuming and beyond the scope of this work.

Figure 6.2 shows the accuracies for the particular user states for the setups **UD**, **UI** and **SI**. As expected, results obtained for setup **UD** are generally better than those obtained for the other two

	ANN	SVM
UD	92.3% (±2.9%)	89.7%
UI	37.6% (±5.7%)	38.2%
SI	58.6% (±6.6%)	56.8%

Table 6.1: Accuracies for different classifiers and experimental setups. The percentages in braces in the first column show the standard deviations over five repetitions (training and testing) of the same ANN experiment. The displayed accuracies for ANN experiments represent the means over the results from these five repetitions.

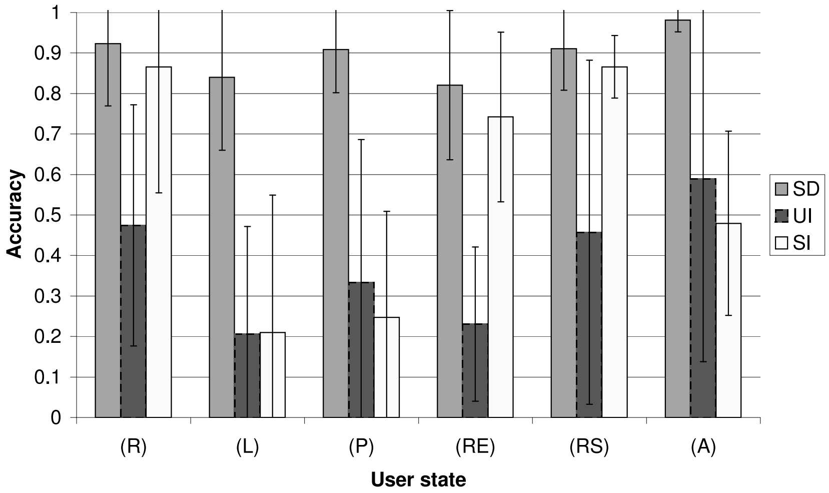
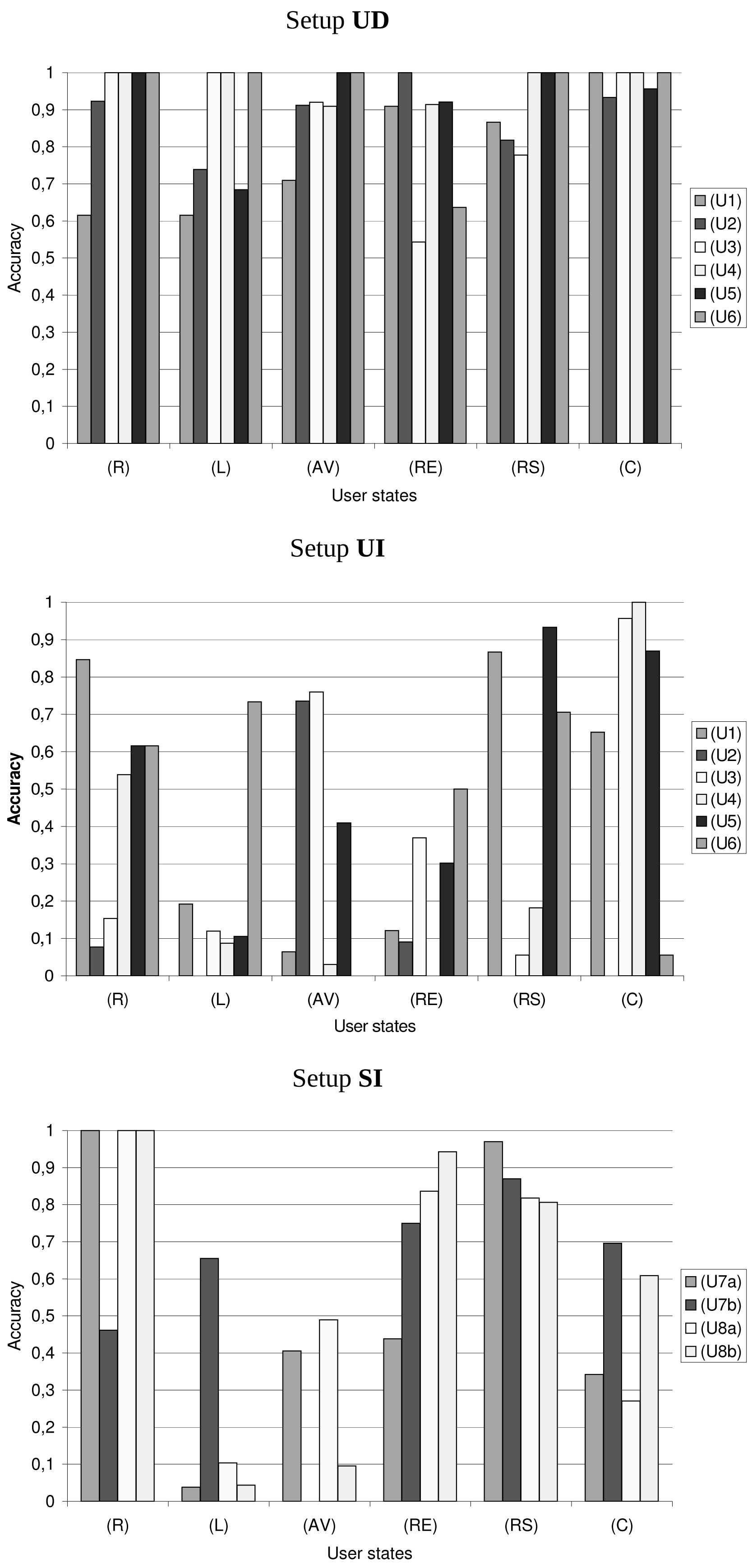


Figure 6.2: Results for different user states for the different experimental setups. The standard deviation over the recording sessions which belong to one setup is depicted by the whiskers.

setups. Note however that for user states resting (R), reading (RE) and the summarization of the read article (RS) accuracies for setups **SI** and **UD** are approximately in the same range. This shows that there seem to be some user states which are distinguished more robustly than others, when electrode positions and mental fitness of the recorded subject vary, which is inevitable across sessions. It is possible that not only EEG but also eye movements and the activity of the facial muscles play a role for the discrimination of these states. This hypothesis and other physiological reasons for this finding remain to be investigated.

Note furthermore the large standard deviations over all recording sessions for the single tasks, which are even larger when the mean accuracy over all recording sessions is low. Figure 6.3 gives a possible explanation for that: It can be seen (in particular for setups **UI** and **SI** where mean accuracies are low) that for each test set there are different states for which the system performance is good or bad. In other words, no common subsets of user states for all recording sessions can be identified, for which exclusively high or low accuracies are achieved.

Finally it is interesting to analyze the confusion matrices obtained in the experiments for the different recording setups (see table 6.2) to gain some more insight in the behavior of the classifier. Note that the matrices for *unbalanced* test sets are displayed here, which plays however a minor role for the following analysis. (The overall performance of the baseline system is analyzed already above in terms of average accuracies which address the problem of an unequal amount of data per class, see section 4.6.1.)


Setup **UD**

Setup **UI**

Setup **SI**

Figure 6.3: Results for each user state and each recording session for the different experimental setups. For each state, the bars represent from left to right the results for recording sessions (U1)-(U6) (for setups **UD** and **UI**) or recording sessions (U7a), (U8a), (U7b) and (U8b) (for setup **SI**) respectively.

While for setup **UD** the largest entries of the confusion matrix are found on the main diagonal, this is not anymore the case for setups **UI** and **SI** where moderately high values are spread all over the matrices. This is not surprising when the accuracies achieved for the different setups (see table 6.1) are taken into account. User states which are confused particularly often for setups **UI** and **SI** are analyzed now.

For both setups the states listening (L) and perceiving a presentation (P) are often confused with resting (R). A reason for that could be natural fluctuations in alertness during the states (L) and (P), i.e. subjects can not stay *constantly* alert for several minutes while listening to a talk or perceiving a presentation. Furthermore the basic alertness level during state (R) varies across subjects and sessions. Therefore it becomes likely that a short lapse in alertness during states (L) or (P) is classified as resting (R) for setups **UI** or **SI**, since for one or more training sessions recorded from subjects with a high basic alertness level, a similar activity pattern could have been observed during state (R).

Next, it can be seen that state (L) is often confused with state (P) for setups **UI** and **SI** and state (P) is often confused with state (L) for setup **SI** and with state (A) for setup **UI**. This can possibly be explained with the fact that similar cortex regions are activated for states (L), (P) and (A) (see also section 2.3.1): All three states require the understanding of symbols (letters, numbers) which involves the parietal cortex. Furthermore states (P) and (L) require the processing of speech and language which takes places in the temporal cortex. Therefore possibly only small differences in the EEG between the particular states exist, which are still sufficient for their discrimination in case of setup **UD**, but not anymore in case of the other two setups due to the variability across sessions and subjects.

Finally the arithmetics state (A) is often confused with reading (RE) for setup **SI** and the state (RE) is often confused with states (P) and (A) for setup **UI**. The same explanation as given before might hold here as well, since in all cases the parietal cortex for the understanding of symbols is strongly involved.

6.1.4 Averaging

The first attempt to improve the performance of the baseline system was to apply averaging over the previous k feature vectors (see section 4.2.2). The optimal value of k was obtained using cross validation. We consider a value of k to be optimal here, when improvements achieved by increasing it even more remain small compared to the improvements obtained before. The accuracies on the validation sets for different values of k are shown in figure 6.4 for setups **UD** and **UI**.

While for setup **UD** there are no considerable improvements for $k > 2$ anymore, the accuracy for setup **UI** improves up to $k = 4$. Therefore the following experiments were continued with values of $k_{opt} = 2$ for setup **UD**, with $k_{opt} = 4$ for setup **UI** and with $k_{opt} = 3$ for setup **SI**. The value of k_{opt} for setup **SI** was chosen, following the reasoning from section 6.1.1 that properties of this experimental setup lie somewhere in-between those for the other two setups. Table 6.3 compares the baseline results ($k = 1$) with those which are obtained when averaging over the k_{opt} previous feature vectors is performed for the test data of the different experimental setups. In all cases significant improvements for $k = k_{opt}$ are achieved compared to the baseline system.

It is interesting that different values of k_{opt} are found for the different setups. This can possibly again be explained with the different degree of noise and variability contained in the data of the different setups. When using slack variables, an SVM is able to cope with a certain amount of variability and noise in the data. While for setup **UD** the SVMs have to cope only with the natural temporal fluctuations of the EEG signal, for setup **UI** additional variability is introduced by the individual differences in the EEG data of the subjects in the training set. Therefore the influence of the temporal signal fluctuations has to be reduced more for this setup which is done by choosing a larger value of

Setup **UD**

		Prediction					
		(R)	(L)	(P)	(RE)	(RS)	(A)
Target	(R	72	4	0	1	1	0
	(L)	9	124	8	1	4	0
	(P)	5	5	151	1	2	4
	(RE)	12	1	21	181	3	3
	(RS)	0	1	0	5	79	2
	(A)	0	0	1	0	1	120

Setup **UI**

		Prediction					
		(R)	(L)	(P)	(RE)	(RS)	(A)
Target	(R)	37	9	14	10	4	4
	(L)	49	34	33	14	10	6
	(P)	45	9	56	17	3	38
	(RE)	23	10	43	53	16	76
	(RS)	3	6	16	5	42	15
	(A)	0	0	18	23	3	78

Setup **SI**

		Prediction					
		(R)	(L)	(P)	(RE)	(RS)	(A)
Target	(R)	45	7	0	0	0	0
	(L)	44	24	17	8	7	7
	(P)	53	55	42	14	4	1
	(RE)	1	22	0	141	2	31
	(RS)	0	3	1	8	86	0
	(A)	1	1	1	71	2	56

Table 6.2: Confusion matrices for the different recording setups. The displayed matrices represent the sum over the confusion matrices of all test sets of one recording setup.

k, i.e. by considering a larger context. Thus the classifier is able to address other types of variability. When the considered context is too large however, a change of user state might be detected with delay only, since at the time where the new user state begins too many samples of the old user state are still taken into account.

6.1.5 Normalization

Up two this point, all reported results were obtained using the baseline normalization method **GlobalNorm** which performs mean subtraction and variance normalization globally for the whole data, using mean and variance estimates from the complete training set. As we have seen in the previous sections, the variability between subjects or sessions for setups **UI** and **SI** seems to be detrimental to the classification results. One possibility to address that problem which is investigated in this section, is to use normalization method **UserNorm**, where mean subtraction and variance normalization is performed separately for each recording session contained in the training, validation or test set. Fur-

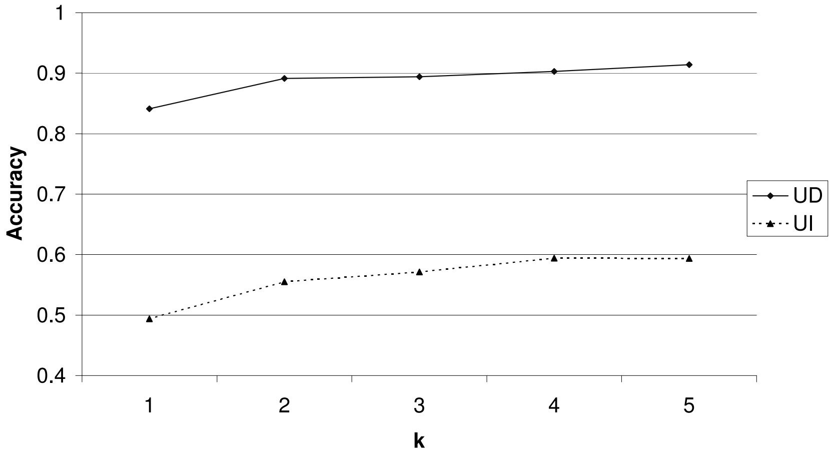


Figure 6.4: Validation set accuracies for averaging over k feature vectors for different numbers of k. The results for setup **UD** are depicted by the solid line, the results for setup **UI** by the dotted line.

	$k = 1$	$k = k_{opt}$
UD ($k_{opt} = 2$)	89.7%	93.4%
UI ($k_{opt} = 4$)	38.2%	46.7%
SI ($k_{opt} = 3$)	56.8%	61.8%

Table 6.3: Comparison of test set results for averaging over $k = 1$ (baseline system) and $k = k_{opt}$ previous features. The value of k_{opt} for each setup is given in braces.

thermore the hypothesis that relations between different frequency bands must be preserved for good classification performance, is examined here by using normalization method **RelPower**. (See section 4.2.3 for a more detailed description of the different normalization methods.)

Table 6.4 shows the results which were obtained using the different normalization methods. (Note that the first column of table 6.4 corresponds to the last column of table 6.3 since the optimal value of k was used for all experiments here.) Normalization method **UserNorm** seems to be able to reduce variability for setup **UI**, where significant improvements in results can be achieved compared to the baseline method **GlobalNorm**. Also for setup **SI** improvements are observed compared to the baseline when using method **UserNorm** which are however small and not significant. Closer inspection of the results for the single recording sessions of setup **SI** reveals, that only for session (U8a) accuracies increase strongly, while they decrease slightly for all others. The small decrease in results for the application of normalization method **UserNorm** compared to method **GlobalNorm** for setup **UD** is not significant as well.

Normalization method **RelPower** leads finally to a significant decrease in results for all experimental setups. That shows that possibly the information about the total power of each feature vector, which is eliminated using this normalization method, is more important for classification than the relations between the feature values of different frequency bands which are preserved correctly only by this method.

6.1.6 Artifact Removal

ICA for the removal of eye activity artifacts did not work well for our system which is unexpected, since it has been applied successfully for this purpose in other research (see section 4.1 for refer-

	GlobalNorm	UserNorm	RelPower
UD	93.4%	93.1%	42.2%
UI	46.7%	58.9%	34.1%
SI	61.8%	62.7%	25.9%

Table 6.4: Accuracies for different normalization methods for the different experimental setups.

ences). Only for experimental setups **UD** and **SI** eye activity could be isolated to one component in the training data (as shown in figure 4.2 in section 4.1.2), while for setup **UI** it is usually spread over multiple components. Therefore the proposed procedure for eye activity removal from the data (visual inspection of the ICs in the training data an rejection of *one* component containing eye activity) is not applicable for that setup. Also the rejection of more than one component is no alternative here, since it would remove to much actual EEG activity from the data.

The most likely reason that eye activity can not be isolated properly for setup **UI**, is that the training data for this setup does not meet one necessary condition for the application of ICA: Although only n electrode channels are available, the data is likely to be generated by much more than n processes. Since one can not assume a statistical dependence between the EEG of the five subjects in the training set, the number of processes contributing to the probability density functions (pdfs) related to the *concatenated* measurements of the different electrode channels is approximately $5n$ in the worst case. Thus it is impossible to estimate n ICs (i.e. their corresponding pdfs) such that only one process contributes to each IC (i.e. to each pdf), and it is not very likely (although not impossible) that only those processes are summarized in one ICs by a standard ICA algorithm which correspond to the eye activity of the different subjects. In other words, an unmixing matrix which is suitable for the data of one subject, would not fit at all for the data of another subject, so that the procedure for ICA weight estimation, which takes the data from all subjects into account, can not find suitable weights at all.

For the other two setups the identification of the component containing the eye activity in the training data and its rejection from the test and training data yields the test set results shown in table 6.5. While for setup **UD** the average accuracy increases slightly but not significantly, the accuracy for setup **SI** even decreases. A possible reason for these poor results can be seen in figure 6.5 where the unmixing matrix which had been estimated on the training data of session (U1) was applied to determine the ICs of the validation data from the same recording session: The eye activity is now not anymore isolated to one component, but spread over two components. Furthermore actual EEG activity seems to be present as well in both ICs containing eye activity (more strongly in the first than in the second one). Thus it becomes clear that the removal of one or both ICs containing eye activity does not necessarily improve the results and may even lead to a decrease of accuracies.

Note that the results in table 6.5 were obtained by rejecting only that component which contained all eye activity in the training data. The rejection of both components containing eye activity in the validation data was attempted as well, which led to strong decreases in results however.

We conclude that ICA weights which are estimated on the training data do not generalize well for unseen data, even in case of setup **UD** where the unseen data is taken from the same recording session as the training data.

Although the generalization performance of the used ICA algorithm is obviously not very good, we wanted to find out whether there are nevertheless some processes which are detrimental to the classification accuracy and which can be reliably identified in the ICs computed for unseen data. Therefore cross-validation was applied to determine those components whose rejection yields in the largest improvements of results for setup **UD**. (Remember that for setup **SI** no validation set was

	no ICA	ICA
UD	93.4%	94.1%
SI	62.7%	56.8%

Table 6.5: Comparison of results achieved without and with ICA-based eye activity removal from the data for setups **UD** and **SI**. All results are obtained with the optimal system configuration for averaging and normalization.

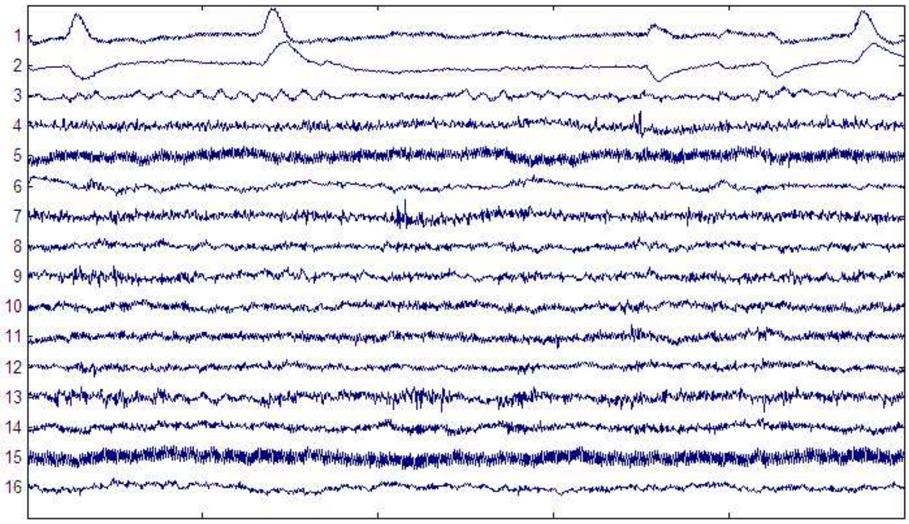


Figure 6.5: ICs of the validation data for recording session (U1) obtained with ICA weights estimated on the training data from the same session. It can be seen that the first and the second component contain eye-activity and at least the first component contains also EEG activity.

available, so that for this setup cross-validation was not applicable.)

Table 6.6 compares the results obtained without artifact removal, to those obtained using cross-validation or visual inspection of the ICs computed for the training data to find the components to be rejected. The cross-validation method yields slightly better results than the visual inspection method. Furthermore results obtained with the cross-validation method are significantly better than the baseline results where no ICs are removed. However it can be seen that the components selected for removal by the cross-validation method are completely different from those selected by the visual inspection method and they do not contain eye activity in any case.

We conclude that there are possibly processes which are detrimental to the classification accuracy and which can be isolated on unseen data using ICA weights estimated on training data. Cross-validation can be used to identify these processes, however future work has to find out more about their properties and their origin. Furthermore other variants of ICA or completely different algorithms for successful eye activity removal in our data have to be investigated in the future. Only if eye activity can be rejected reliably from unseen data, it will be possible to say whether it is detrimental to classification accuracies or not.

Session	no ICA	vis. inspection		cross validation	
(U1)	86.4%	1	92.1%	3	92.3%
(U2)	93.2%	-	93.2%	9	94.2%
(U3)	95.7%	3	93.5%	8	97.1%
(U4)	97.6%	2	97.8%	12	97.6%
(U5)	92.8%	3	92.4%	10	93.8%
(U6)	94.7%	5	95.5%	16	94.7%
Total	93.4%		94.1%		94.9%

Table 6.6: Comparison of results for the different recording sessions obtained without ICA to those obtained using ICA and visual inspection of the training data or cross-validation to find the components to be rejected. Left of the results for a particular method the number of the component which is rejected for each recording session is shown. A dash denotes that no component has been rejected.

6.1.7 Feature Reduction

Two potential feature reduction methods which are applicable for classification problems were presented in section 4.3: Averaging over adjacent frequency bands (referred to as FreqAvg method in this section) and linear discriminant analysis (LDA).

Figure 6.6 shows the validation set accuracies for setups **UD** and **UI** when adjacent features are put together in bins of size *b*, i.e. when averaging over *b* adjacent frequency bands is performed. In both cases no significant increase in results can be observed for any value of *b*, but results remain stable with increasing bin size for a while. For setup **UD** results start to decrease for with bin size of about 12 which corresponds to a total feature vector dimensionality of 128 features. (Note that the original feature vector dimensionality was 1440.) For setup **UI** results decrease already for $b = 7$ where in total 208 features are used. A very interesting observation is that results decrease only moderately up to $b = 45$ for both setups. In this case only two features per electrode are considered anymore: the average power for the lower frequencies (1-22 Hz) and the higher frequencies (23-45 Hz). Only for $b = 90$ (one feature per electrode) results drop strongly. This supports the hypothesis from section 2.3.1 that for each electrode information about the total power in the lower frequency range (around the α-band) and in the higher frequency range (around the 40Hz γ-activity) is sufficient to distinguish different user states.

As explained in section 4.3.2, LDA coefficients can only be estimated when enough data is available, since otherwise the eigenvalue problem to solve is ill-conditioned. Therefore LDA can only be applied for setup **UI** here. Figure 6.7 shows the validation set accuracies for LDA-based dimensionality reduction to different feature vector sizes.

For LDA-based feature reduction accuracies start to drop already for larger feature vector sizes compared to the FreqAvg method. Furthermore it can be seen that for the latter the accuracy for setup **UI** is still larger than 50% when only two features per electrode are considered (32 features in total), while the accuracy for LDA-based feature reduction to the same number of features is significantly lower. Closer inspection of the features selected by LDA for this feature vector size reveals that preferably very low frequency bands are selected, especially the bands 0 - 0.5Hz and 0.5 - 1Hz (see table 6.8) and that across training sets there is a strong overlap between the selected features. As explained in section 2.3.1 these low frequencies correspond to DC potentials which are related to the processing of sensory inputs. They are usually pertubed with sweating and electrode artifacts however and therefore attenuated by the amplifiers. Nonetheless LDA seems to be able to extract valuable information from these features. A reason that LDA performs worse than the FreqAvg method might

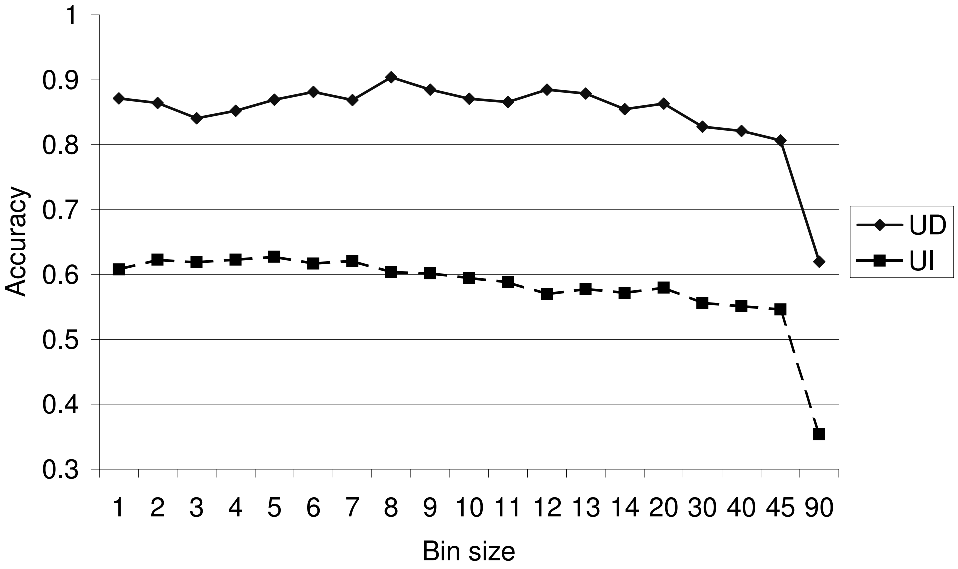


Figure 6.6: Validation set accuracies for setups **UD** (solid line) and **UI** (dashed line) for feature reduction with the FreqAvg method using different bin sizes. Note the non-equidistant scale of the x-axis.

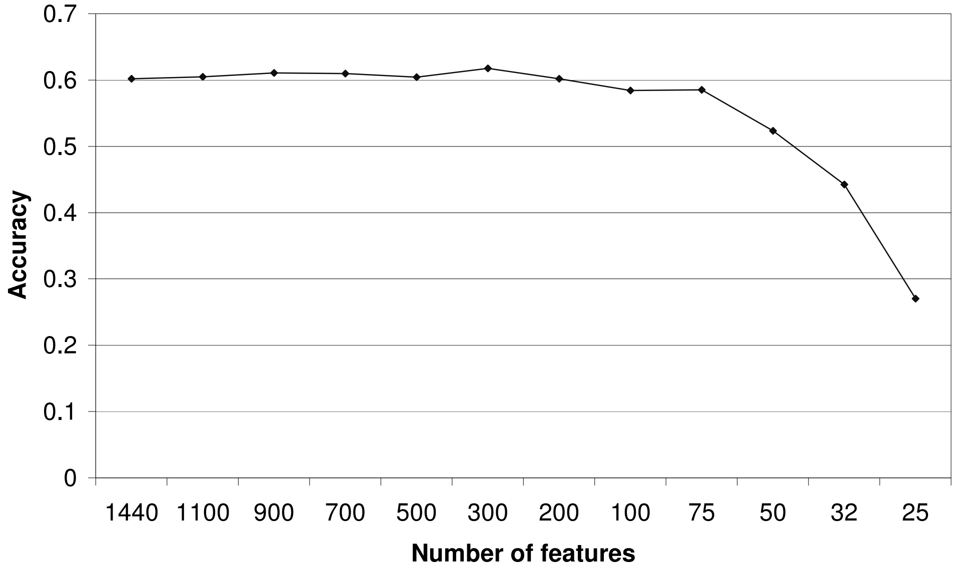


Figure 6.7: Validation set accuracies for setup **UI** for LDA-based feature reduction to different feature vector sizes. Note the non-equidistant scale of the x-axis.

Setup	No feat. reduction	LDA	FreqAvg
UD	93.4%	-	93.0% (144, b = 10)
UI	58.9%	57.5% (300)	53.1% (288, b = 5)
SI	62.7%	-	62.5% (205, b = 7)

Table 6.7: Results for feature reduction on the test data for the different experimental setups using LDA and the FreqAvg method compared to the baseline results obtained without feature reduction. A dash indicates that no results are available for a particular case. The number of features (and the value of b where applicable) are displayed in braces. For better comparability no ICA was used for setup **UD** although this might improve results slightly. All results are obtained with the optimal system configuration for averaging and normalization determined in the previous sections.

be that LDA imposes a linear separation between the classes which seems not to be true completely although linear SVMs achieve comparatively good classification results (see also section 6.1.3).

One common reason for the application of feature reduction is that the elimination of features introducing only noise in the data might lead to better overall results. From figures 6.6 and 6.7 it can be seen that feature reduction can not improve results here. A possible explanation is that the classifier itself is well able to cope with noise, because it selects a separation between the classes where noisy features play a minor role. Another advantage of feature vector dimensionality reduction is however that statistical models can be estimated more reliably, even when only little data is available. Furthermore less memory is required to store learned models and training and classification procedures may speed up a lot when the number of used features is reduced.

For these reasons it is interesting whether a feature vector dimensionality can be determined from the validation set results, so that results remain stable but the number of features is reduced significantly for unseen data. We decided here to make conservative choices for the reduced feature vector dimensionalities, i.e. they were chosen sufficiently larger then those dimensionalities where validation set results start to drop. Thus it should be more likely to obtain stable test set results.

Table 6.7 compares the test set results for both feature reduction methods (the feature vector dimensionality has been determined from the validation data as described above) with the baseline where no feature reduction is performed. All results drop slightly (but not significantly) compared to the baseline, however it can still be said that they remain stable. The only exception are the results obtained with feature reduction method FreqAvg for setup **UI**, where a significant decrease in results is observed. Possibly the choice of $b = 5$ from the validation data was not conservative enough, i.e. possibly the validation data does not reflect the properties of the test set well enough in this case.

Session	Fp1	Fp2	F7	F8	F3	F4	Fz	T3	T4	T5	T6	P3	P4	Pz	O1	O2
(U1)	0, 2.5, 4	0, 0.5, 5.5	0.5	0, 0.5	0.5, 2, 2.5	0, 0.5	0, 0.5, 4, 4.5	-	0.5	0, 0.5	0, 0.5	0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5
(U2)	0, 0.5, 2.5	0.5, 2	0	-	0, 0.5, 1, 1.5, 2.5	0, 0.5, 4	0, 0.5, 4	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0	-	0, 0.5	0, 0.5
(U3)	0, 4	0, 0.5	0, 0.5	0.5, 1	0, 0.5, 1.5, 2.5	0, 0.5	0, 0.5 3, 5	0, 0.5	0	0, 0.5	0, 0.5	0, 0.5	0	0.5	0, 0.5	0
(U4)	0, 0.5, 4	0, 0.5	0, 0.5	-	0, 0.5	0, 0.5, 4	0, 0.5, 4	0, 0.5	0	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5
(U5)	0, 3.5	0, 0.5, 2	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 3	0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5
(U6)	0, 0.5, 4	1.5, 2, 4	0	-	0, 0.5, 2	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5	0, 0.5

Table 6.8: Frequency bands selected for the different electrodes by LDA-based feature reduction to 32 features for the different training sets of setup **UI**. The session numbers identify the test sets to which the training sets belong on which ICA weights were calculated. The numbers in the rest of the table indicate the start frequency of the frequency bands with 0.5Hz width.

Setup	
UD	93.4%
UD^4	70.8%
HB	66.2%

Table 6.9: Accuracies for setup **UD** setup **UD**^{4} and setup **HB** for the discrimination of all six user states.

6.1.8 Electrode Reduction

As explained in section 5.1 the headband for EEG recording uses only four electrodes at the pre-frontal and frontal positions Fp1, Fp2, F7 and F8. In this section results are presented which were obtained with the headband or with the ElectroCap^{TM} when only data from these four electrodes is considered. Furthermore subsets of user states are identified which can be discriminated particularly well using the four pre-frontal and frontal electrodes only.

Table 6.9 compares the results for setup **UD** obtained with all 16 electrodes with those obtained using the four pre-frontal and frontal electrodes of the ElectroCap^{TM}only (This setup is henceforth referred to as **UD**^{4}.) and with the results obtained for setup **HB**. In all cases the optimal system configuration for averaging and normalization which has been found in the previous sections for setup **UD** was used. No ICA and component rejection was applied however, since the number of processes captured by the four frontal electrode channels seems to be much larger than four. In the ICs estimated for four electrode channels only, one can see that different processes such as eye activity are not isolated to single components at all. The same explanations which have been given for the failure of ICA application for setup **UI** (see section 6.1.6) seem to hold here, too.

It is not surprising that results drop significantly when only four pre-frontal and frontal electrodes instead of all 16 electrodes are used. Nevertheless these four electrodes seem to be sufficient to discriminate user states comparatively well, although during most user states cortex areas are particularly active which are not located at a pre-frontal or frontal cortex lobe at all: During user states listening, perceiving a presentation and summarization of a read article high activity should be observed especially in temporal cortex regions which are responsible for the processing of speech and language. Furthermore during user states reading, arithmetics and perceiving a presentation the parietal cortex where the understanding of symbols takes place is particularly active.

The most likely explanation for the comparatively large difference in results between the setups **UD**^{4} and **HB** where the same electrode positions are used, is that different subjects have been recorded in both cases and results usually fluctuate across subjects. The signals recorded with the headband seem to be at least of comparable or even of better quality than those recorded with the ElectroCap^{TM}, since the contact between the skin and the headband electrodes is much better compared to the ElectroCap^{TM} electrodes.

Next it must be analyzed whether there are subsets of user states which can be discriminated particularly well when using only the four pre-frontal and frontal electrodes. For this purpose the confusion matrices obtained for the validation sets of setups **HB** and **UD**^{4} are examined (table 6.10). Interestingly the pairs of states for which many confusions are produced differ often across the two setups, which indicates that the information about user states which can be captured from the frontal cortex is at least partly subject dependent. Nevertheless some subsets of user states can be identified for which sufficiently little confusions are made for both setups:

1. Resting (R), listening to a talk (L), reading and article (RE), summarization of the read article

Setup **HB**

		Prediction						
		(R)	(L)	(P)	(RE)	(RS)	(A)	Accuracy
Target	(R)	8	0	7	5	0	5	33.0%
	(L)	2	22	0	0	0	1	87.2%
	(P)	1	0	10	2	8	4	40.4%
	(RE)	0	0	0	22	0	3	88.1%
	(RS)	1	1	2	0	21	0	83.7%
	(A)	6	0	6	4	2	7	28.5%

Setup **UD**^{4}

		Prediction						
		(R)	(L)	(P)	(RE)	(RS)	(A)	Accuracy
Target	(R)	28	14	7	8	8	3	40.5%
	(L)	1	58	4	3	0	2	85.8%
	(P)	2	9	45	5	3	4	68.7%
	(RE)	3	8	0	41	8	8	56.1%
	(RS)	0	5	0	2	58	3	83.9%
	(A)	3	8	9	13	1	34	52.4%

Table 6.10: Confusion matrices for the validation sets of setups **HB** and setup **UD**^{4} for the discrimination of all six user states. The matrices represent the sum over the confusion matrices obtained for all validation sets of a particular setup. In the last columns the classification accuracy for each state is shown.

(RS)

2. Perceiving a presentation (P), listening to a talk (L), reading an article (RE)

Note that for the states in the first subset little confusions between all of them can only be observed for setup **HB**, while for setup **UD**^{4} examples for the state resting (R) are comparatively often classified to belong to another state. Nevertheless we included resting in at least one subset of reduced states, since we consider this state to be important for practical applications.

Table 6.11 compares the results for setups **HB**, **UD**^{4} and **UD** obtained for the discrimination of all six user states with those which are achieved when only the user states from the two subsets are taken into account. Since in each case a different number of states is considered, results are displayed in terms of the normalized expected loss (see section 4.6) to make them comparable.

For both user state subsets the normalized expected loss improves for the setups which use four electrode channels only. For the second subset improvements are even significant. This leads to the conclusion that there are subsets of user states which can be discriminated particularly well with only four pre-frontal and frontal electrodes. Nevertheless the best results are still achieved when using all 16 electrodes. This is not very surprising, since the largest part of the neural processes related to the different user states do not take place in the pre-frontal or frontal cortex as explained above.

6.1.9 Analysis of Best System Configuration

Table 6.12 shows parameters for the best system configurations for setups **UD**, **SI** and **UI** which were determined in the previous sections. The improvements for each setup achieved by using these optimal

Setup	all states	(R), (L), (RE), (RS)	(P), (L), (RE)
UD	0.07 (93.4%)	0.11 (92.4%)	0.08 (94.6%)
UD^4	0.35 (70.8%)	0.33 (75.2%)	0.24 (84.0%)
HB	0.41 (66.2%)	0.31 (77.0%)	0.17 (88.6%)

Table 6.11: Mean normalized expected losses and mean accuracies (in braces) for the different sets of user states over all recording sessions of setups **UD**, **UD**^{4} and **HB**.

Setup	Averaging (value of k_{opt})	Normalization method	Artifact Removal	Feature Reduction
UD	$k_{opt} = 2$	**GlobalNorm**	ICA using cross-validation for component rejection	none
UI	$k_{opt} = 4$	**UserNorm**	none	none
SI	$k_{opt} = 3$	**UserNorm**	none	none

Table 6.12: Parameters of the best configuration of each processing step for setups **UD**, **SI** and **UI**

parameters for each processing step are displayed in table 6.13.

For all setups results improve considerably when averaging over the k_{opt} previous feature vectors is performed. Only for setup **UI** large improvements are observed as well when normalization method **UserNorm** instead of the baseline method **GlobalNorm** is used. As a reason for that finding, it was suggested in section 6.1.5, that the variability in the data of this setup, which is larger compared to the other to setups, can be reduced by this normalization method. ICA based artifact removal could finally be applied with success only to the data of setup **UD** because of the bad generalization properties weight matrices estimated by the Infomax algorithm. Note however that in contrast to our original intention no eye activity but other potentially artifactual processes were rejected with the help of ICA.

For the best system configuration the results for setup **UI** and **SI** do not differ much, while for the baseline system much better results for setup **SI**could be achieved. Intuitively one would consider problem of user state classification across sessions from the same subject (setup **SI**) to be much easier than user state classification across subjects (setup **UI**). It must be noted however that about five times less training data was available for setup **SI** compared to setup **UI** which might explain that only a small difference in results between both setups remains, after unwanted variability between subjects has been removed from the data using normalization.

Closer analysis of the results for different recording sessions and different user states per recording session reveals, that they all improve without exception when comparing the baseline results to those obtained with the best system configuration for setups **UD** and **UI**. This is not the case for setup **SI**, where the classification accuracy increases strongly for recording session (U8a), but it decreases for the other three recording sessions when normalization method **UserNorm** is applied instead of

Setup	Baseline	Averaging	Normalization	Artifact Removal
UD	87.9%	93.4% Δ_{abs} = 5.5% Δ_{rel} = 45%	-	94.9% Δ_{abs} = 1.5% Δ_{rel} = 23%
UI	38.2%	46.7% Δ_{abs} = 8.5% Δ_{rel} = 14%	58.9% Δ_{abs} = 12.2% Δ_{rel} = 23%	-
SI	56.8%	61.8% Δ_{abs} = 5.0% Δ_{rel} = 12%	62.7% Δ_{abs} = 0.9% Δ_{rel} = 2%	-

Table 6.13: Accuracies, absolute (Δ_{abs}) and relative improvements (Δ_{rel}) after optimizing each parameter system parameter of the baseline system appropriately for setups **UD**, **UI** and **SI**. A dash indicates that no improvements could be achieved by changing the baseline configuration of the corresponding processing step.

the baseline method **GlobalNorm**. If the baseline normalization method is used for all recording sessions of this setup however, also here improvements for each session and almost each user state are observed. Only for user state resting the accuracy obtained with the best system decreases by 1% in average over all sessions compared to the baseline system.

The largest absolute improvements are observed for setup **UI**. Therefore it is interesting to analyze for this setup whether the system's weak points could be improved when the best system configuration is used. For this purpose we consider the confusion matrices of the baseline system, the best system and the difference between both, which are shown in table 6.14.

In particular the analysis of the difference matrix is interesting. Positive entries in the main diagonal denote accuracy improvements. Entries at other positions are positive when the number of confusions increased in the best system compared to the baseline system and negative otherwise. Since it was mentioned above that accuracies improved for all user states for the best system, it is not surprising that positive entries in the difference matrix are found mostly but not exclusively on the main diagonal. It is noteworthy however that examples of user state perceiving a presentation (P) are confused 26 times more with the summarization of a read article (RS) for the best system configuration compared to the baseline configuration. No simple explanation can be provided for that finding. Probably much deeper understanding of the neural processes related to these two user states and closer analyses of the effects of normalization and averaging on the data are required, to be able to give a plausible reason for the strong increase of confusions here.

In the baseline system the following user states were confused particularly often for setup **UI** (see also section 6.1.3): (L) and (P) with (R), (L) with (P), (P) with (A) and (RE) with (P) and (A). In the best system the number of almost all these kinds of confusion decreased clearly, only the number of confusions of state (L) with state (R) remained constantly.

We conclude that weaknesses of the baseline system could be eliminated at least partly in the best system. It must be noted however that also the best system confuses the above mentioned states still relatively often.

Setup **UI**, baseline configuration

		Prediction					
		(R)	(L)	(P)	(RE)	(RS)	(A)
Target	(R)	37	9	14	10	4	4
	(L)	49	34	33	14	10	6
	(P)	45	9	56	17	3	38
	(RE)	23	10	43	53	16	76
	(RS)	3	6	16	5	42	15
	(A)	0	0	18	23	3	78

Setup **UI**, best configuration

		Prediction					
		(R)	(L)	(P)	(RE)	(RS)	(A)
Target	(R)	44	16	10	5	3	0
	(L)	49	65	25	0	4	3
	(P)	31	14	77	10	29	7
	(RE)	34	15	29	100	11	32
	(RS)	4	0	4	3	73	3
	(A)	0	0	8	14	8	92

Difference between best and baseline configuration for setup **UI**

		Prediction					
		(R)	(L)	(P)	(RE)	(RS)	(A)
Target	(R)	7	7	-4	-5	-1	-4
	(L)	0	31	-8	-14	-6	-3
	(P)	-14	5	21	-7	26	-31
	(RE)	11	5	-14	47	-5	-44
	(RS)	1	-6	-12	-2	31	-12
	(A)	0	0	-10	-9	5	14

Table 6.14: Confusion matrices of the baseline system, the best system and the difference between both for setup **UI**.

6.1.10 Prototype System

To show the feasibility of user state classification in real time and to be able to get an immediate response of the system when data is recorded under different conditions, a prototype system has been developed (figure 6.8).

The workflow of the system can be described as follows: First raw EEG data is acquired with the headband and the amplifier which are described in section 5.1. Then a Pentium III 800MHz laptop is used for data processing. Data is received by the computer via the serial RS232 port using a C++-based data acquisition tool which has been developed at University of Karlsruhe. Then preprocessing and classification are performed with a MATLAB^{TM} implementation of the corresponding algorithms (which is partly based on standard tool boxes). Finally a hypothesis for the current user state is displayed by the data acquisition tool. Note that for technical reasons ANNs instead of SVMs are used as classifiers in the prototype system.

EEG is recorded continuously and every five seconds a new user state hypothesis is produced

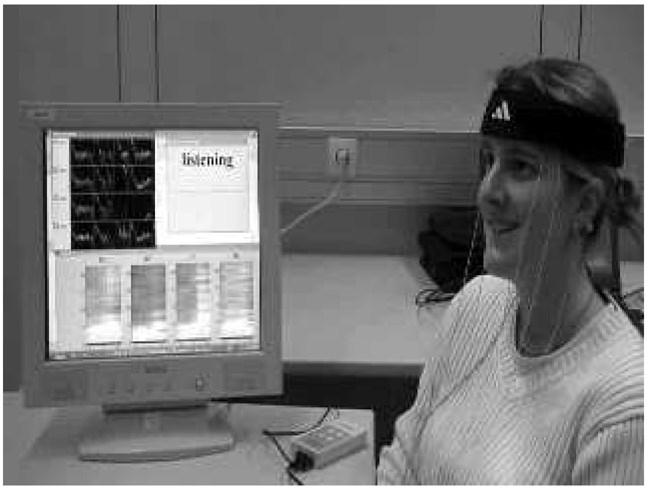


Figure 6.8: The prototype system for user state identification. In the upper left corner of the monitor the recorded EEG signals are displayed, the hypothesis for the current user state can be seen in the upper right corner. The spectrograms for the electrodes Fp1, Fp2 F7 and F8 are shown at the bottom.

using the data segment acquired during the last five seconds. From each data segment five feature vectors are extracted and classified using the methods described in section 4. Thus five user state hypotheses are produced per segment and the final hypothesis to be displayed is determined using a majority decision. Preprocessing and classification for five seconds of data takes much less than one second so that real time processing is easily possible. Currently the prototype system is able to distinguish the user states resting, reading, listening to a talk and talking.

The prototype system performs best when training and testing is performed in one session which is not surprising when comparing the results for setups **UD** and **SI** from the previous sections. Thus, to obtain an optimal system performance, training must be performed directly before the system is actually used and the headband must not be moved or even pulled off and put on again in-between. About 100 seconds of training material for each user state are sufficient to obtain already very good classification results. When training data from other sessions than the test session is used, system performance increases with the number of sessions from which training data is taken. Even when training and test session are the same, the use of additional training material from other sessions seems to have a positive effect in many cases.

The user states reading and resting are recognized very robustly across sessions. Not even a careful electrode placement is required to obtain a good classification performance for these states, but the user who is recorded can easily put on the headband himself. For the recognition of user state listening the prototype system is less robust across sessions. This confirms the findings from section 6.1.3 where for user states resting and reading much better classification accuracies across sessions (i.e. for setup **SI**) were reported than for user state listening.

6.2 Assessment of Mental Task Demand

6.2.1 Data Analysis using SOMs

Initial experiments on the task demand data showed that it is much more difficult to distinguish the four task demand levels low, medium, high and overload than different user states. In table 6.15 clas-

Session	Low	Medium	High	Overload	Average	Norm. expt. loss
(T1)	62%	27%	100%	-	63%	0.56
(T2)	44%	39%	100%	-	61%	0.59
(T3a)	90%	0%	40%	88%	53%	0.72
(T4)	75%	100%	10%	0%	46%	0.61

Table 6.15: Accuracies for each task demand level obtained in an initial subject and session dependent experiment for recording sessions (T1), (T2), (T3a) and (T4). Linear multiclass SVMs were used as classifiers here, for averaging a value of $k = 2$ was chosen and normalization was performed with method **GlobalNorm**. A dash at a certain position indicates that no data for that task demand level from the corresponding recording session was available.

sification accuracies for the discrimination of the task demand levels low, medium, high and overload are shown for recording sessions (T1), (T2), (T3a) and (T4). An experimental setup similar to setup **UD** for the user state data was used to obtain these results: Classification was performed with linear multiclass SVMs, for averaging a value of $k = 2$ was chosen and normalization method **GlobalNorm** was applied.

The obtained accuracies are all above the chance accuracy of 25%, however far below the results which were obtained for setup **UD**for the discrimination of six user states. Furthermore it can be seen from table 6.15 that for each recording session there are no more than two task demand levels for which the accuracies are clearly above chance. Closer inspection of the confusion matrices for the sinlge recording sessions reveals, that the examples for task demand levels with low accuracies are mostly classified to belong to *one* task demand level with a higher accuracy.

Also when using ANNs for classification or the regression versions of SVMs or ANNs results can not be improved. This leads to the hypotheses that either the neural correlates of task demand which can be measured in the EEG are too weak to be used for the prediction of task demand, or that the target values for the available data are too unreliable. Especially the latter hypothesis seems plausible, since the targets were obtained from subjective task demand evaluations by the recorded subjects , and subjects usually found it very difficult to identify the exact transition between two levels of task demand (see section 5.3). This hypothesis is supported even more when considering the poor overlap between time segments corresponding to the same task demand level for two subjective evaluations of one recording (see table 5.4).

To find out whether there is at least some structure in the data despite the potentially bad quality of the references, i.e. to find out whether there are at least two or three task demand levels which can be reliably discriminated, SOMs were trained to gain insight about the spatial relations of the available feature vectors. Only if all feature vectors belonging to different task demand levels are sufficiently far away from each other, the training of classification or regression functions for the discrimination of these task demand levels makes sense.

Figure 6.9 shows the SOMs for recording sessions (T1), (T2), (T3a) and (T4). After SOM training, the BMUs of the examples belonging to different task demand levels were indicated in the map as dots of different colors. The size of the dots is proportional to the number of examples of one task demand level sharing the same BMU. To make the generation of the SOMs computationally tractable, averaging over five adjacent frequency bands was performed which should not be too detrimental to the results according to the experience with the user state data (see section 6.1.7). (In section 6.2.7 it is confirmed that also results for the task demand data remain stable, even when averaging is performed over up to 45 adjacent features.)

The BMUs for task demand level medium overlap either with those for task demand level high (in case of session (T2)), with those for task demand level low (in case of session (T4)) or even with the BMUs for task demand level high and low (in case of sessions (T1) and (T3a)). This agrees with the above hypothesis that either the EEG activity related to medium task demand is not very different from the activity related to low or high task demand, or that subjects were simply not able to find the exact transitions between medium task demand and the other task demand levels. Since in table 6.15 the lowest accuracies are reported also for task demand level medium, we decided to exclude it from further experiments.

Only for two of the four considered recording sessions (sessions (T3a) and (T4)) examples for task demand level overload are available, and it can be seen from figure 6.9 that their corresponding BMUs almost do not overlap with the BMUs of other task demand levels. For session (T4) however no clear separation between the BMUs for the examples of task demand levels high and overload can be seen, which suggests bad generalization performance for unseen data. To analyze the relation between the three remaining task demand levels low, high and overload in greater detail, SOMs for sessions (T3a) and (T4) were trained with examples for these task demand levels only (see figure 6.10). Now some overlap between the BMUs for examples of task demand levels high and overload can be seen for session (T3a), while for session (T4) there is still no overlap but also no clear separation between the BMUs of examples of both task demand levels. The finding for session (T3a) that overload and high task demand are only separated well in the SOM trained on all data (figure 6.9) suggests, that the feature vectors of both two task demand levels belong to separate clusters which are however very close together. As a consequence of that for unseen data generalization might be bad. Furthermore it can be found for the initial experiments described above that either many confusions were made between high task demand and overload (in case of session (T3a)) or that examples of task demand level overload were exclusively classified to belong to other arbitrary task demand levels (in case for session (T4)). For these reasons also examples of task demand level overload were excluded from the following experiments.

Figure 6.11 shows finally the SOMs for the four recording sessions which are trained only on the examples for the remaining task demand levels low and high. Except for session (T1), one can see that examples for low and high task demand have mostly to non-overlapping BMUs which are separated clearly in the SOMs. This indicates that also for unseen data the prediction of these two task demand levels should work well. The overlapping BMUs of both task demand levels for session (T1) suggest that the variance *between* feature vectors for different task demand levels is smaller than the variance *within* the group for feature vectors belonging to one task demand level. Note however that this does necessarily imply bad classification results, since it is possible that also small changes of some feature values have a large predictive power. Thus this finding is only an indicator that generalization performance for unseen data might be bad[1].

For all SOMs presented here, the BMUs were determined for the examples which were used for training only, since the goal for using SOMs was to gain insight into the structure of the available data in general, and not to test the generalization performance of a SOM-based classification algorithm. To examine whether our hypothesis concerning the good separability of feature vectors for task demand levels low and high can also be generalized for unseen data, experiments with the same setup as above were conduced on test data, where these two task demand levels only were discriminated . The results are shown in table 6.16.

[1] Note that for the above decisions to reject the data from task demand levels medium and overload for the following experiments, not only overlapping or badly separated BMUs for the different examples in the SOMs were used as justification, but also actual classification accuracies from the initial experiments were considered. This is also done here (see table 6.16) to justify the decision that low and high task demand can be separated well.

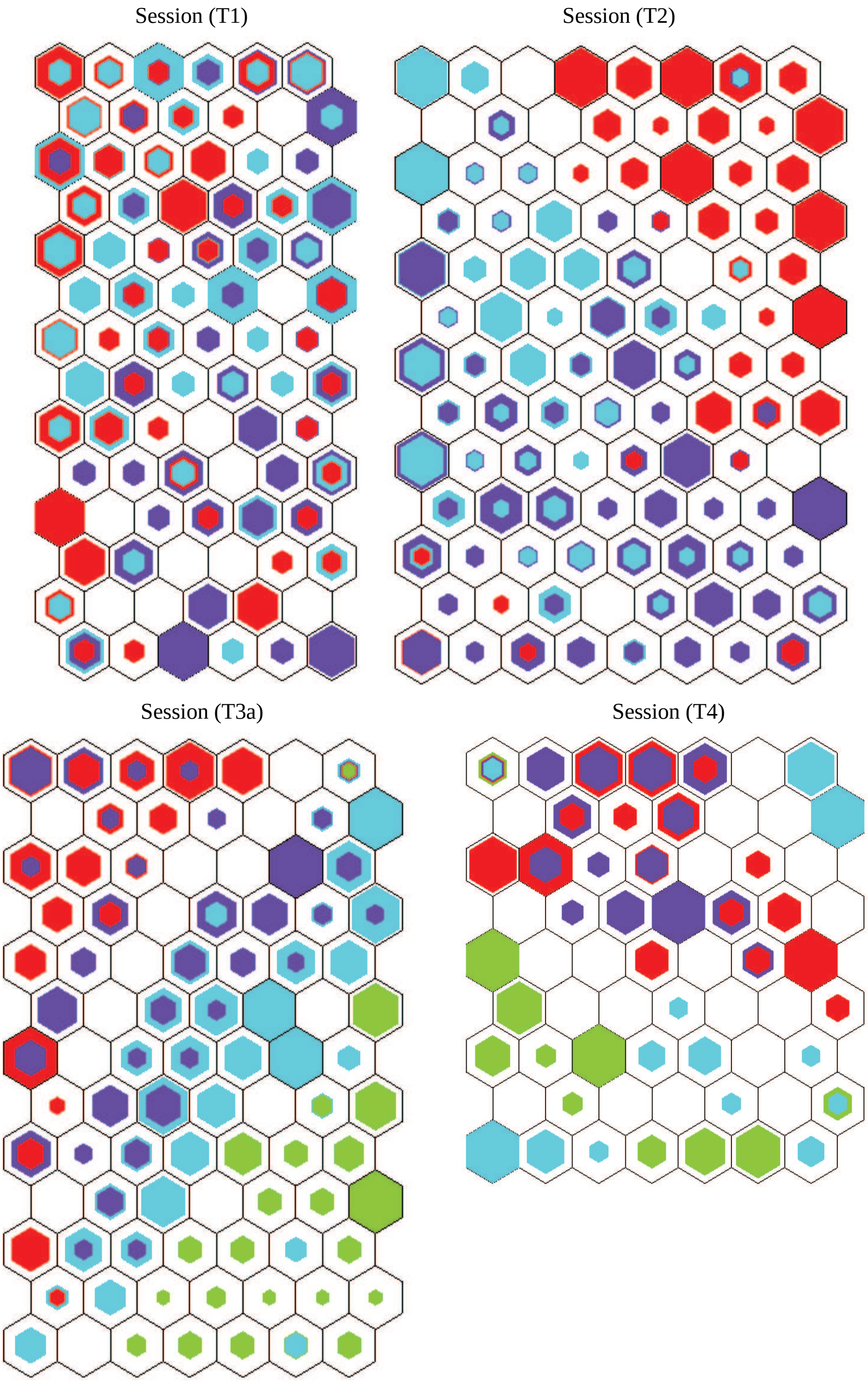


Figure 6.9: SOMs for recording sessions (T1), (T2), (T3a) and (T4). Red dots correspond to task demand level low, dark blue dots to medium, light blue dots to high and green dots to overload. The size of the dots is proportional to the number of examples having their BMU at the position of the corresponding dot.

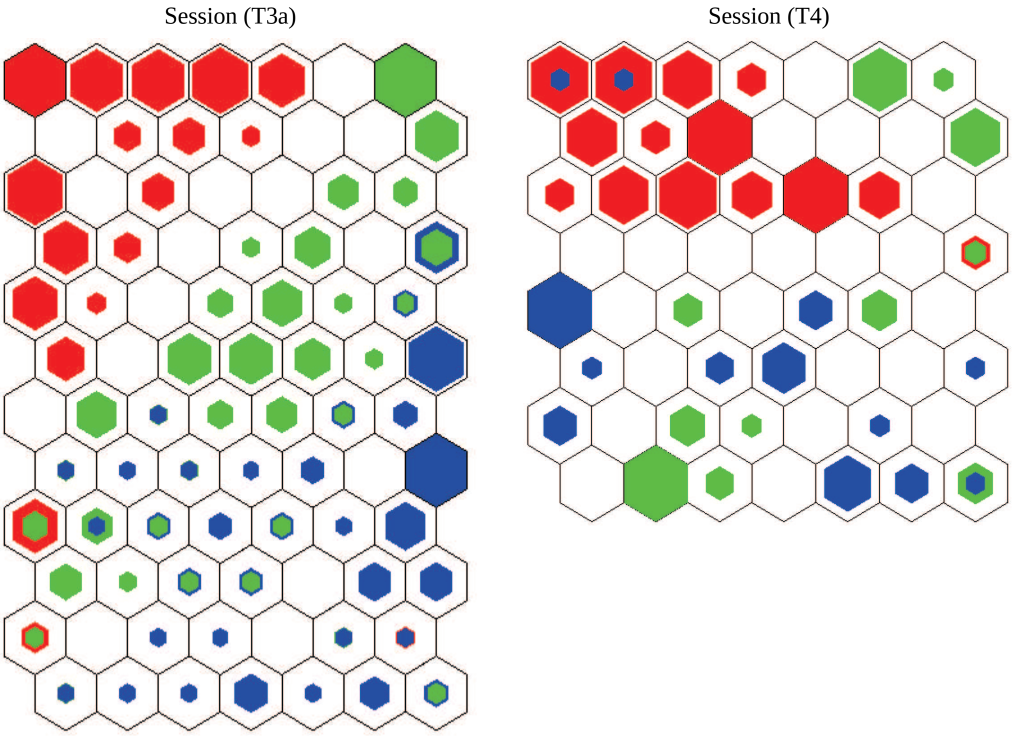


Figure 6.10: SOMs for recording sessions (T3a) and (T4) trained on data from task demand levels low (red dots), high (blue dots) and overload (green dots) only. The size of each dot is proportional to the number of examples having their BMU at the position of the corresponding dot.

Session	Low	High	Average	Norm. exp. loss
(T1)	88%	100%	94%	0.12
(T2)	72%	100%	86%	0.28
(T3a)	100%	100%	100%	0
(T4)	100%	133%	87%	0.58

Table 6.16: Accuracies for the discrimination of task demand levels low and high obtained in an initial experiment with a user and session dependent setup for recording sessions (T1), (T2), (T3a) and (T4).

Compared to the results obtained for the discrimination of all four task demand levels (table 6.15), the normalized expected loss improves for all recording sessions. (Improvements are clearly significant.) Two interesting observations can be made when comparing the SOMs from figure 6.11 and the results from table 6.16: For session (T1) very good classification results are achieved, although the SOM for the data for this session this session suggests the contrary. Thus is seems that there are really some features for which small changes of their values have high discriminative power. Second a surprisingly low accuracy is achieved for recording session (T4), in particular for task demand level high. One reason for this might be that no suitable *linear* separation could be found between the examples of both task demand levels, although some separation seems to exist which is suggested by the structure of the SOM. On the other hand it is possible, that the data of this recording session is too noisy, so that the trained classifier does not generalize well for the test data here.

From the findings in this section we conclude that the discrimination of task demand levels low

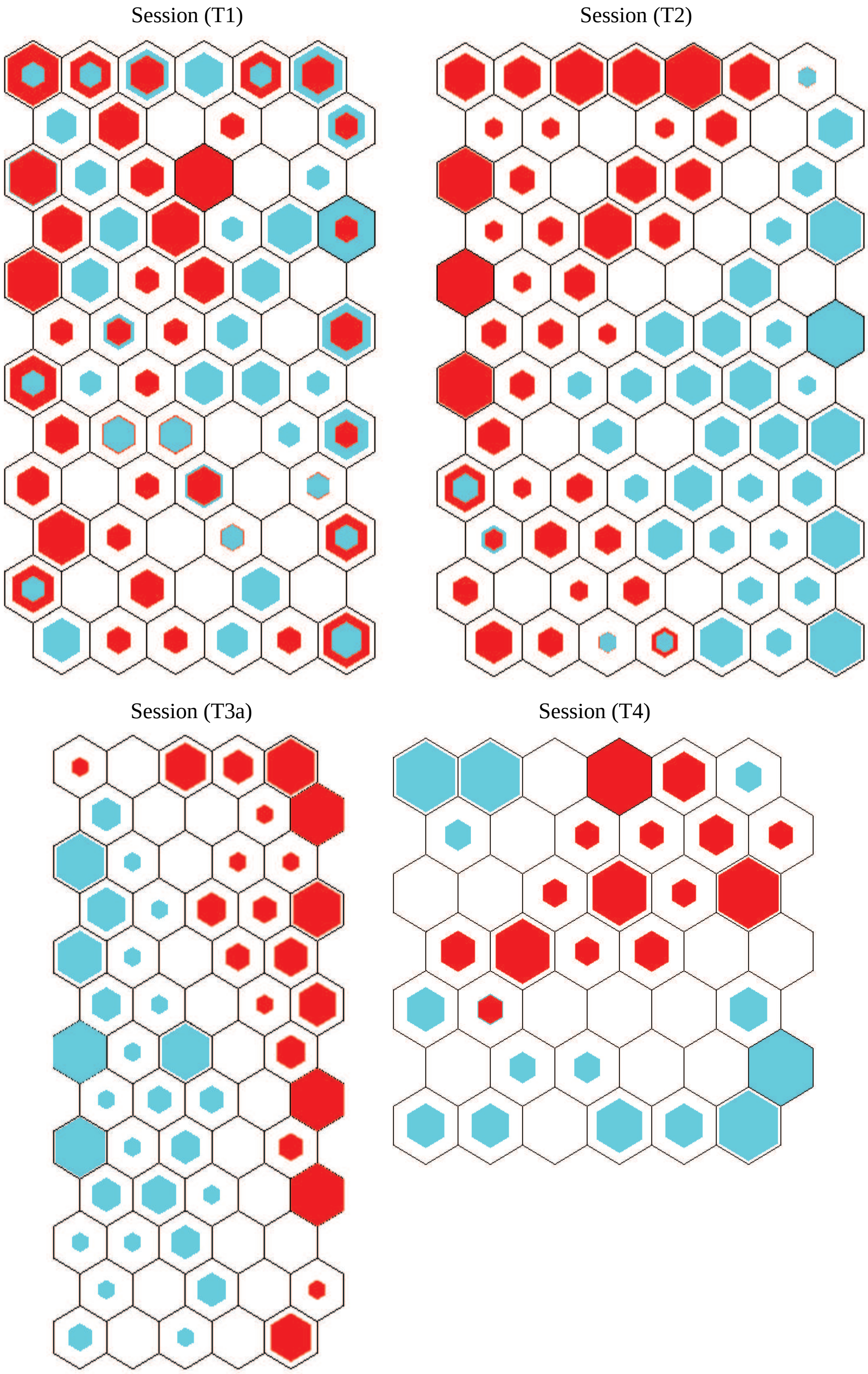


Figure 6.11: SOMs for recording sessions (T1), (T2), (T3a) and (T4) trained only on examples for task demand levels low (red dots) and high (blue dots). The size of each dot is proportional to the number of examples having their BMU at the position of the corresponding dot.

and high seems to work relatively well. Therefore only these two task demand levels were considered in the experiments presented in the following sections.

6.2.2 Experimental Setups

The experimental setups for the task demand data are very similar to those which were distinguished for the user state data. Therefore the same names for similar setups setups are used here to make it easier to remember their characteristics. Each setup is now described briefly below:

UD **U**ser and session **d**ependent experiments: Different data portions of the same recording session were extracted for training (80% of the whole session), testing and validation (each 10% of the whole session). Recording sessions (T1), (T2), (T3b) and (T4) from the **ElectroCapSubjects** data collection were used for this setup and results are always reported as averages over the test or validation set accuracies for all these sessions if not indicated differently. Training sets comprise in average 247 seconds of data, validation sets 31 seconds and test sets 64 seconds.

HB The same type of experiments as for setup **UD** was conducted on the **HeadBandSubjects** data (recording sessions (T7) and (T8)). The average size of the training sets is here 314 seconds, the average validation set size 39 seconds and the average test set size 86 seconds. Also here reported results are averages over the accuracies obtained for the validation or test sets of both recording sessions.

UI **U**ser **i**ndependent experiments: For recording sessions (T1), (T2), (T3b) and (T4) from the **ElectroCapSubjects** data the system was trained in a round-robin manner on a subset of three recording sessions (in average 740 seconds of data) and tested on the remaining session. For comparability of the results the same test sets as in setup **UD** were used and results were averaged in the same way. For validation 229 seconds of data from the sessions (T5) and (T6) from the **ElectroCapSubjects** data collection were available.

SI User dependent but **s**ession **i**ndependent experiments: Only for one subject task demand data was collected twice (recording sessions (T3a) and (T3b)). For this data, experiments across sessions were conducted, i.e. training was performed on 80% of the data of one session (in average 257 seconds of data), testing on 10% of the other session (in average 48 seconds of data) and vice versa. Results are reported as averages over the accuracies from both test sets. As in case of the user state data, no data was available for validation here, which reflects the properties of a session independent experimental setup. For early stopping regularization 10% of non-training data from the training sessions was used (in average 32 seconds). In all other cases plausible guesses for system parameters were made based on the estimates for setups **UD** and **UI**.

Note that also for the task demand data the length of the different data sets in seconds corresponds to the number of feature vectors per data set, since for each second a new feature vector is available, as explained in section 4.2. Note furthermore that data sets for training and validation were balanced after their extraction from the different recording sessions (as in case of the user state data), so that the percentages given above do not represent their fraction of all available data. For the test data this is not the case, which also explains the difference between the available test and validation data, even when the same amount of data had been extracted for both prior to balancing.

Similarly to the user state data (see section 6.1.1), the test sets for setups **UD** and **UI** are exactly the same, while the test sets for the other two setups are based on different recording sessions. For this reason accuracies obtained for setups **UD** and **UI** are clearly separated from the accuracies obtained for the other setups in all following tables in the same way as is has been done for the user state data.

6.2.3 ANN topology selection

As explained in section 6.1.2 for the user state data, ANN topology selection had to be performed as well for the task demand data, before ANNs can be trained and used for task demand prediction. Also here exclusively networks with one hidden layer are used so that only the number of neurons in the hidden layer remains to be determined.

For setup **UD** this is done using the data from recording sessions (T5) and (T6) (the validation data sessions for setup **UI**). For setup **UI** networks are trained on the original training data and early stopping regularization is performed using the original validation data. As topology section test set data portions from recording sessions (T5) and (T6) are used which are disjoint with the original validation data. No data is available to perform topology selection for setup **SI**. Therefore the network topology for this setup was guessed based on the results obtained for the other two setups.

Figure 6.12 shows the accuracies for different numbers of neurons in the hidden layer obtained with regression and classification ANNs for setups **UD** and **UI**. Accuracies represent always averages over five repetitions of the same experiment. The standard deviations over these repetitions are depicted by the whiskers in the figure. From the predictions of regression ANNs accuracies were obtained using equation 4.71.

Several observations can be made in figure 6.12. For both experimental setups regression ANNs perform slightly better than classification ANNs, however differences in results are in most cases very small (except when less than 8 hidden units are used in case of setup **UD**) and often within the standard deviations over the repeated experiments. Thus the hypothesis from section 4.5 that different task demand levels are predicted better when using a regression function instead of a classification function (since the first method exploits the information contained in the ordinal scaling of the different task demand levels) is supported only slightly by that finding. Furthermore closer inspection of the results for particular recording sessions reveals that fluctuations across subjects are larger when a regression ANN instead of a classification ANN is used.

For setup **UD** the classification results increase when increasing the number of hidden units up to 14, while regression results seem to be mostly insensitive towards the number of hidden units. Therefore in the following ANN experiments using setup **UD** 18 hidden units were used for classification ANNs and 14 hidden units for regression ANNs.

For setup **UI** it can be seen that results are only slightly above chance accuracy. Therefore it has to be investigated in the following, whether it is generally possible to predict task demand across subjects, and if this is the case, whether ANNs are applicable for this task. Besides that it can be seen that results for classification and regression ANNs are insensitive towards the number of hidden units. Therefore 14 hidden units were used for both methods in the following experiments using setup **UI**. Also for setup **SI** 14 hidden units were used for all ANN experiments.

6.2.4 Comparison of Prediction Methods

In this section the performance of the regression and classification versions of SVMs and ANNs are compared for the following baseline system configuration:

- no artifact removal
- no averaging over feature vectors

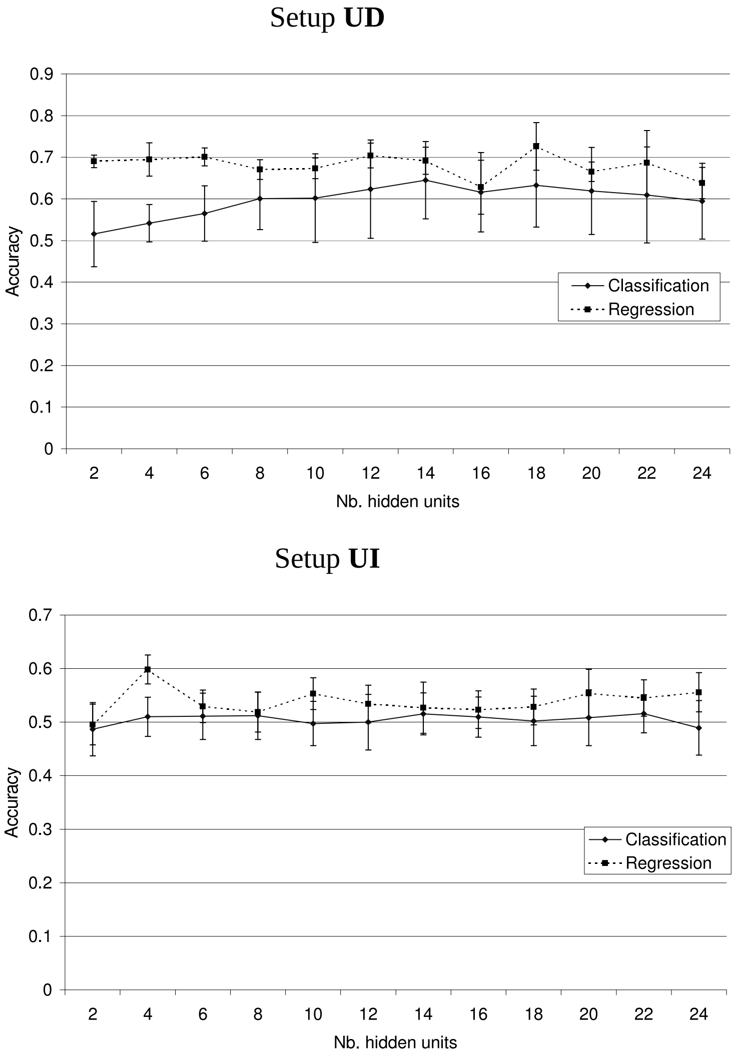


Figure 6.12: Results obtained on the topology selection test sets when using different numbers of hidden units for classification (solid lines) and regression (dotted lines) ANNs for setups **UD** and **UI**. The whiskers depict the standard deviations over five repetitions of the same experiment.

- normalization with method **GlobalNorm**
- no feature reduction
- Linear SVMs or ANNs with one hidden layer were used for classification and regression. For the number of hidden ANN units the values determined in the previous section were chosen. To reduce the fluctuations of the ANN results, predictions were computed using majority decisions (in case of classification) or averaging (in case of regression) over the outputs of five networks trained and tested on the same data.

Table 6.17 shows the results for the different classification and regression methods and the different experimental setups. From the predictions of the trained regression functions, accuracies were derived here (and for all other experiments with regression methods reported in the remainder of this work) using equation 4.71. For ANNs the same experiment (i.e. training and testing) was always

Setup	SVM^{class}	$SVM^{regress}$	ANN^{class}	$ANN^{regress}$
UD	81%	79%	78% (±7%)	71% (±3%)
UI	72%	74%	70% (±3%)	69% (±3%)
SI	66%	73%	53% (±5%)	66% (±5%)

Table 6.17: Results for the different classification and regression methods for all experimental setups. The figures in braces reflect the average standard deviations over five repetitions of the same experiments in case of ANNs.

repeated five times in order to be able to analyze the fluctuations in the results. Therefore accuracies obtained with ANNs are always reported as means over these five repetitions and standard deviations are given as well.

Note that OLS-regression for task demand prediction is not applicable for the baseline system, since training data is sparse and feature vector dimensionality high, so that the problem of estimating the regression coefficients becomes ill-conditioned. This prediction method can only be used when the feature vector dimensionality has been reduced significantly before. Therefore results obtained with OLS-regression are reported in section 6.2.7, where different feature reduction methods are examined and for the best performing method an optimal feature vector dimensionality is selected.

It can be seen from table 6.17 that (in contrast to the user state data, see section 6.1.3) SVMs perform better the corresponding ANNs for all experimental setups, as well for the estimation of regression functions and classification functions. In most cases, the differences in results are not significant however. For the same reasons which are already mentioned in section 6.1.3 for the user state data (risk of extraordinary low accuracies for unseen data just by coincidence, difficult comparability of results because of fluctuations), all following experiments were conducted using SVMs. It should be kept in mind however that ANNs seem to be an applicable method for task demand estimation as well.

Note that ANNs results for setup **UI** are much better than expected when comparing them with the accuracies obtained on the topology selection test sets which were only slightly above chance (see section 6.2.3). The reason for that might be that the validation data coincidentally contains recording sessions for which the estimation of task demand does not work well. Furthermore it is interesting to observe that classification ANNs perform better than regression ANNs for setups **UD** and **UI** here in contrast to the findings from section 6.2.3. The differences are however not significant which suggests that classification and regression ANNs are possibly equally suitable for the prediction of task demand, at least when only two task demand levels are considered.

Also no big differences in results between the classification and regression variants of SVMs can be observed. Only for setup **UI** the regression SVMs produce significantly better results at the 10% level than classification SVMs, while in no case SVM-based classification performs significantly better than SVM-based regression.

For all following experiments the regression version of SVMs was used. This decision is motivated by the intuition that for the prediction of an ordinally scaled variable such as task demand regression methods are more suitable. Although no benefits of their application compared to the application of classification methods can be seen when only two task demand levels are distinguished, regression methods might outperform classification methods when a larger number of task demand levels must be predicted. In that case it makes more sense to exploit information about the *order* of the different task demand levels, which is exactly what methods for the estimation of regression functions do.

In figure 6.13 the predictions of the learned regression functions are compared with the target values for low task demand (target values 1) and high task demand (target value 2). The classification accuracies which can be derived from these predictions are shown in figure 6.14 for each recording session and each experimental setup.

The relation between the regression function predictions (figure 6.13) and the classification accuracies derived from these predictions (figure 6.14) is now discussed separately for each experimental setup.

For setup **UD** it can be seen that the predictions for session (T3a) match the targets very well and that they show little variability. This results in a classification accuracy of 100% for this session. For sessions (T1) and (T2) much more variabilty in the predictions can be observed. However predictions which belong to examples for high task demand have in average higher values than those beloning to examples for low task demand. The values of the latter are generally much higher than their target values. The effect of both observations on classification accuracies is, that the performance for high task demand is much better than the performance for low task demand for both sessions. All predictions for the test data of session (T4) are approximately on the same level. It can easily be seen that those belonging to examples of high task demand are much too low in order to produce the correct classification. Note however there is a slight tendency of these predictions to be higher than the predicted values for examples for low task demand.

For experimental setup **UI** it is interesting to observe that for sessions (T2) and (T4) better classification accuracies are achieved compared to setup **UD**. That means that data recorded from other subjects is better suited to predict the task demand for these two sessions, than data which is taken from these sessions themselves. The variability of predictions is large for setup **UI**, too. In particular for session (T1) it surprising that classification accuracies clearly above chance are achieved, although predictions seem almost randomly spread between the target values for low and high task demand. In general results for setup **UI** are much better than expected according to the experience made for the user state data (see section 6.1.3).

When comparing the predictions for recording session (T3a) for setups **UD** and **SI**, it can be seen that the predicted values contain much more variability when another recording session from the same subject is used for training. Nevertheless predictions belonging to examples for low task demand are sufficiently low, and those for high task demand are sufficiently high also for setup **SI**, so that still good classification accuracies are obtained. For session (T3b) a similar observation as for session (T4) in case of setup **UD** can be made: The predictions for examples for high task demand stay too close to the target value for low task demand, so that only the chance classification accuracy of 50% is obtained, since all examples are classified as low task demand.

Table 6.18 finally shows for each recording session and each experimental setup some statistical measures which are suitable for the evaluation of regression functions (see section 4.6.2) together with the average classification accuracy for comparison.

It can be seen that high values for r_{ub} and low values of the squared error do not necessarily imply good classification accuracies. The main reason for this seems to be, that despite the large variability of the predictions in many cases, the predicted values are still closer to *their* target value than to the target value of the other task demand level, so that good classification accuracies are obtained. In our case that means that predictions for examples for low or high task demand which are only slightly below or above 1.5 may lead to good classification accuracies, while they produce high squared errors and low correlation coefficients. Furthermore it is interesting to observe the (in most cases) largely positive offsets of the linear regression functions. This shows that task demand predictions are generally higher than they should be, what can also be seen in figure 6.13, in particular for the examples for low task demand.

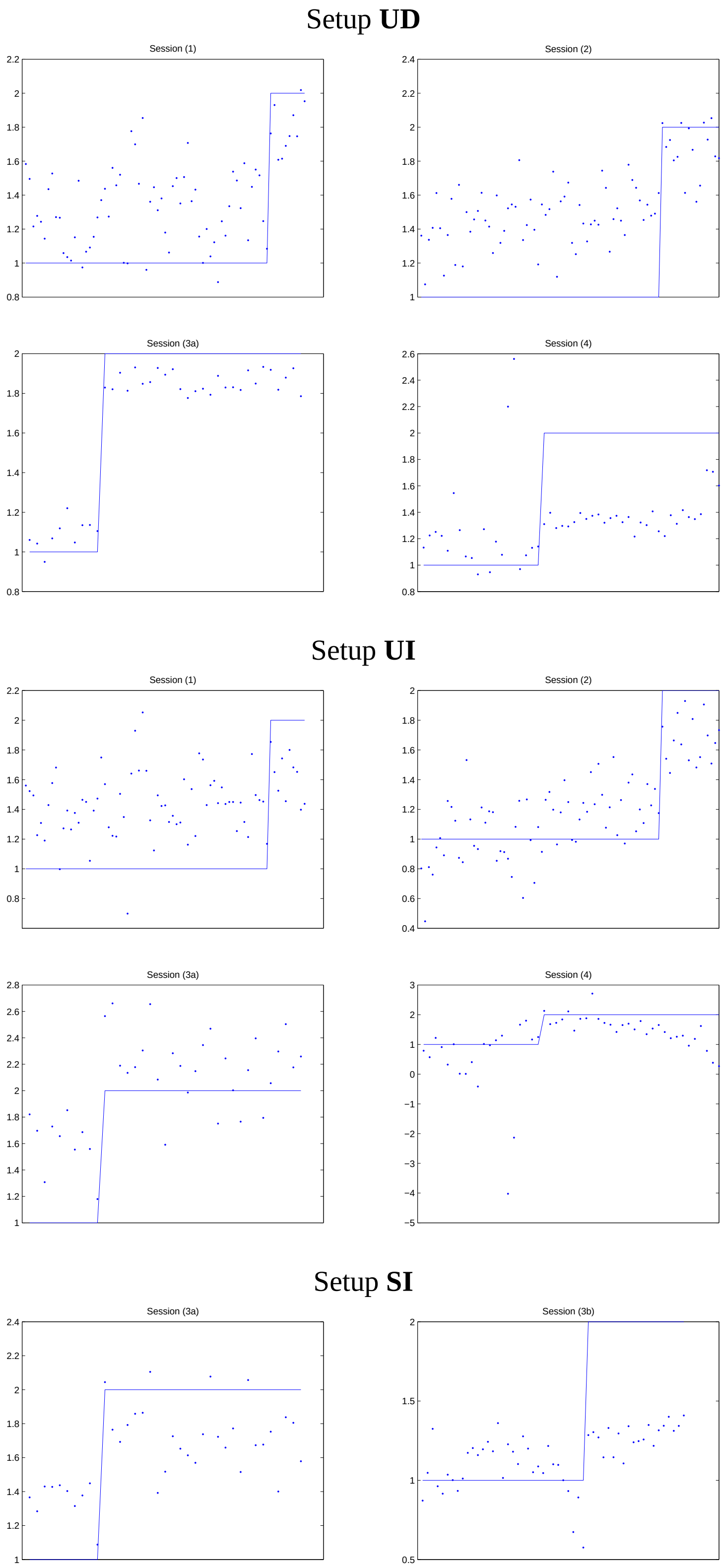
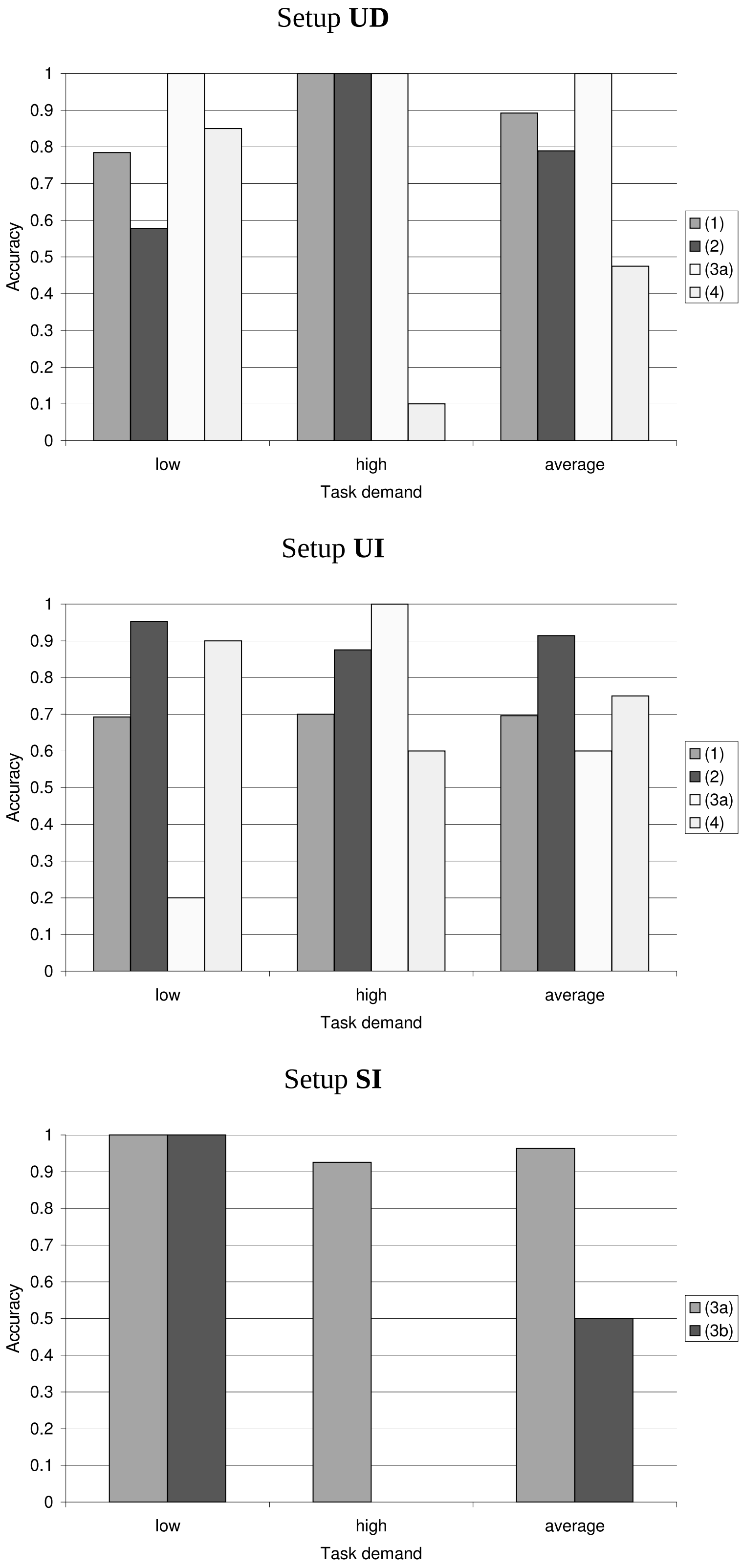


Figure 6.13: Predictions of the learned regression functions on test data (dots) compared their targets (solid line) for all recording sessions and all experimental setups. The target value for task demand level low is 1, for task demand level high it is 2.



Setup **UD**

Setup **UI**

Setup **SI**

Figure 6.14: Classification accuracies for task demand levels low and high for all recording sessions and all experimental setups.

Setup **UD**

Session	r_{ub}	SE_{ub}	reg. fctn	av. accuracy
(T1)	0.62	0.32	$t = 0.48p + 0.83$	89%
(T2)	0.68	0.37	$t = 0.40p + 1.06$	79%
(T3a)	0.98	0.13	$t = 0.77p + 0.32$	100%
(T4)	0.24	0.56	$t = 0.10p + 1.16$	48%

Setup **UI**

Session	r_{ub}	SE_{ub}	reg. fctn	av. accuracy
(T1)	0.43	0.44	$t = 0.20p + 1.23$	70%
(T2)	0.70	0.31	$t = 0.57p + 0.53$	91%
(T3a)	0.71	0.51	$t = 0.59p + 1.02$	60%
(T4)	0.55	1.12	$t = 1.07p - 0.62$	75%

Setup **SI**

Session	r_{ub}	SE_{ub}	reg. fctn	av. accuracy
(T3a)	0.71	0.35	$t = 0.38p + 0.98$	96%
(T3b)	0.65	0.53	$t = 0.21p + 0.86$	50%

Table 6.18: Correlation coefficient r_{ub} and squared error (SE_{ub}) for unbalanced data sets and linear regression functions between targets (t) and predictions (p)obtained for the different recording sessions and different experimental setups. In the last column the average classification accuracies are shown for comparison.

We conclude that for all experimental setups SVM-based regression function estimation for task demand prediction works comparatively well, although single recording sessions exist where no more than chance accuracy is achieved (session (T4) for setup **UD** and session (T3b) for setup **SI**). This indicates that there are data sets for which task demand can not be predicted well with our methods, at least for the baseline system configuration. Next it is important to remark that in many cases where good classification accuracies are achieved, the variability of the predicted values is nevertheless large and predictions often do not match their targets very well, what can be seen from figure 6.13. The statistical measures characterizing the estimated regression functions confirm that finding. This shows that in many cases predictions of class labels are not very stable and that the problem of task demand estimation, even when it is reduced to the problem of distinguishing two task demand levels, still remains difficult.

Although it is important to keep in mind that classification accuracies are always derived from predictions of regression functions here, experimental results are reported mostly in terms of accuracies in the following. We believe that this measure provides the best overview over changes of the system performance when different parameters are varied.

6.2.5 Averaging

Also for the task demand data the effect of averaging over the k previous feature vectors was examined. Figure 6.15 shows the validation set accuracies for setups **UD** and **UI** for different values of k.

Compared to the user state data (see figure 6.4), improvements achieved by averaging are relatively small. For setup **UI** the best results are observed for $k = 2$ in contrast to the expectation that results

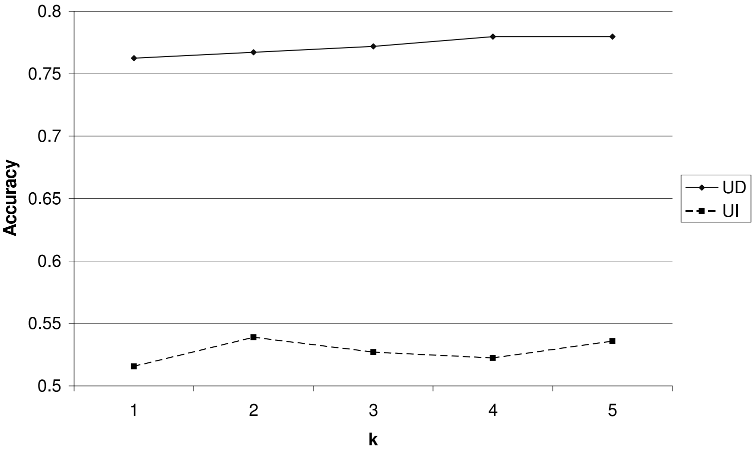


Figure 6.15: Validation set accuracies for setups **UD** and **UI** for different values of k.

	$k = 1$	$k = 2$
UD	78%	82%
UI	75%	79%
SI	73%	73%

Table 6.19: Results for the different experimental setups obtained for the baseline system configuration ($k = 1$) and a modified configuration where a value of $k = 2$ is chosen for averaging.

increase with increasing values of k. Note however that the validation set accuracies for this setup are only slightly above chance. Therefore one has to be extremely careful with their interpretation and it is possible that the accuracy for $k = 2$ represents an outlier.

Since accuracies do not increase much with increasing values of k, all following experiments were conducted with a value of $k = 2$. This represents a compromise between improvements in results and the flexibility of the system to react sufficiently fast to changes of task demand: For too large values of k it is likely that the previous task demand level is predicted for a few more seconds when the level of task demand changes, since data recorded some seconds ago has still large influence on the current feature vector.

In table 6.19 one can see that considerable improvements of the test set accuracies (which are also significant) can be achieved for setups **UD** and **UI** when for a value of $k = 2$ is chosen compared to the baseline configuration where $k = 1$. Interestingly for setup **SI** no improvements at all can be observed.

The latter observation and the fact that the potential of averaging was possibly underestimated on the validation set, suggest that perhaps data sets are too small (and therefore too little representative) to make reliable inferences about the potential of averaging for the task demand data. The significant improvements in results when averaging is applied for setups **UD** and **UI** however, allow at least the careful statement that averaging has a positive effect on the overall system performance.

6.2.6 Normalization

Since the results obtained with normalization method **RelPower** were far below those obtained with the other normalization methods for the user state data (see section 6.1.5), this method is not applied anymore for the task demand data[2]. In table 6.20 the impact of the other two normalization methods

	GlobalNorm	UserNorm
UD	82%	73%
UI	79%	80%
SI	73%	87%

Table 6.20: Results for different normalization methods for the different experimental setups. The accuracies in the first column correspond to those shown in table 6.19 for an optimal value of k for averaging.

on overall results for the task demand data is compared for all experimental setups.

Similar findings as for the user state data (see table 6.4) can be made here. However for setup **UD** results drop much stronger here, when normalization method **UserNorm** is used instead of the baseline method **GlobalNorm**. A reason for this might be, that the task demand data test sets are smaller than the user state data test sets: Only two task demand levels are considered in total, and often little data per task demand level is available (see also section 5.3). Therefore mean and variance can possibly not be estimated reliably enough on the test sets which has to be done however during the normalization procedure of method **UserNorm**.

The slight improvements observed for setup **UI** when using normalization method **UserNorm** are not significant. They are mainly explained with an accuracy improvement of 24% for session (T3a) and a decrease of 17% for the accuracy of session (T2). This indicates that separate normalization of the data for each recording session does not necessarily have a good effect on results for subject dependent experiments. A major reason for the decrease in results for session (T2) might be as well the data sparseness leading to unreliable mean and variance estimates.

For setup **SI** the strongest improvements in results are achieved. This is in particular due to improvements for session (T3b) from the chance accuracy of 50% to 85%. Since data from only two sessions for this experimental setup is available however, one has to be very careful with the conclusion that the application of normalization method **UserNorm** helps to improve the system performance for task demand prediction across sessions. The results obtained here support this hypothesis, but do not provide strong evidence for it.

Although no significant improvements could be achieved when using normalization method **UserNorm** instead of method **GlobalNorm** for setups **UI** and **SI** this method was used in the following experiments for both setups, since we believe that its application would improve results significantly, when more data per recording session was available so that more reliable mean and variance estimates could be obtained.

6.2.7 Feature Reduction

Two feature reduction methods were investigated for the task demand data: Averaging over adjacent frequency bands, henceforth referred to as FreqAvg method (see section 4.3.1), and selection of the features which are best correlated with the target values to be predicted (see section 4.3.3). Backward selection which is able to cope better with multi-collinearities than the simple correlation-based method, can not be applied here, since the dimensionality of the original feature vectors is too large while training data is sparse. Forward selection requires much more computation than the correlation-

[2] As explained in section 2.3.2.1 the information about the total power content of different feature vectors (which is eliminated when using method **RelPower**), is particularly important for task demand prediction, since it seems to be correlated directly with the degree to which mental resources in a certain cortex region are required (see also figure 2.26).

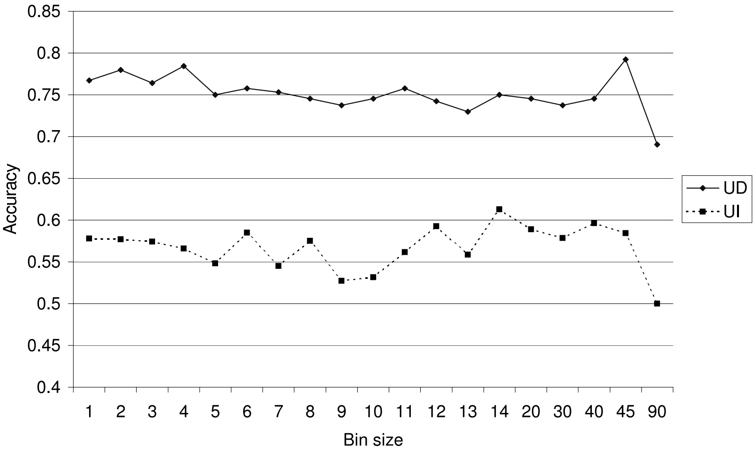


Figure 6.16: Validation set accuracies for setups **UD** and **UI** when different bin sizes for the FreqAvg method are used. The optimal system configuration for averaging and normalization (as determined in the previous sections) was used for both setups.

based approach, but it can not be shown that is better able to avoid the problem of multi-collinearities (see section 4.3.3). Therefore it is not applied here as well.

Figure 6.16 shows validation set accuracies when averaging over a different number of adjacent frequency bands is performed, i.e. when adjacent frequency bands are put into bins of different sizes *b*, for setups **UD** and **UI**. A very similar tendency as already observed for the user state data (see section 6.2.7) can be seen here: Accuracies fluctuate in a comparatively small range for all bin sizes between 1 and 45 and they decrease strongly only for a bin size of 90. We conclude that also for the task demand data the information about the power content of lower and upper frequencies per electrode as a whole seems to be sufficient to obtain acceptable predictions. Note that here again validation set results for setup **UI** are only slightly above chance. Therefore one must be careful with their interpretation.

Figure 6.17 shows validation set results for the correlation-based feature reduction method for setups **UD** and **UI**. For setup **UD** results for feature vector dimensionalities below 100 are much better than those obtained with higher feature vector dimensionalities or even without feature reduction, and they do not drop even when the number of features is reduced down to 16. Only 16 features, i.e. one feature per electrode, are also used for a bin size of 90 for the FreqAvg method. Here results drop strongly compared to larger feature vector sizes which can be seen from figure 6.16.

For setup **UI** no large changes in results can be observed when reducing the feature vector dimensionality with the correlation-based method. However, also for this setup results do not decrease when using only 16 features are used, compared to the case where the feature vector dimensionality is reduced to 16 with the FreqAvg method. A reason that the correlation-based method can not improve results for this setup could be (besides the problem of validation set accuracies close to chance), that for different subjects different features have the largest predictive power (see also table 6.22 below). Therefore the features selected by the correlation based method are not necessarily highly correlated with the targets for the test data, but with the targets of data from different subjects in the training set.

We conclude that the correlation-based feature reduction method seems more suitable for feature reduction than the FreqAvg method for the problem of task demand prediction. Note that on the task demand data for setup **UD** even *improvements* of results could be achieved when correlation-based feature reduction was performed, in contrast to the user state data, where for the application of both LDA (to setup **UI**) and the FreqAvg method (to both setups) results could only be kept constantly

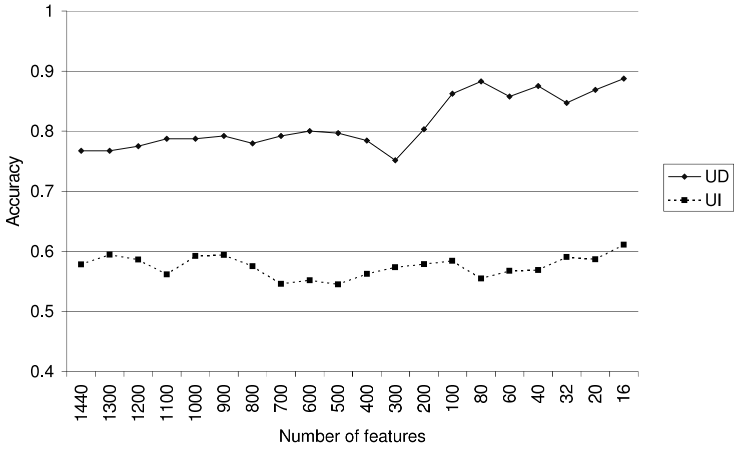


Figure 6.17: Validation set accuracies for correlation-based feature reduction to different feature vector dimensionalities for setups **UD** and setup **UI**. The optimal system configuration for averaging and normalization (as determined in the previous sections) was used for both setups.

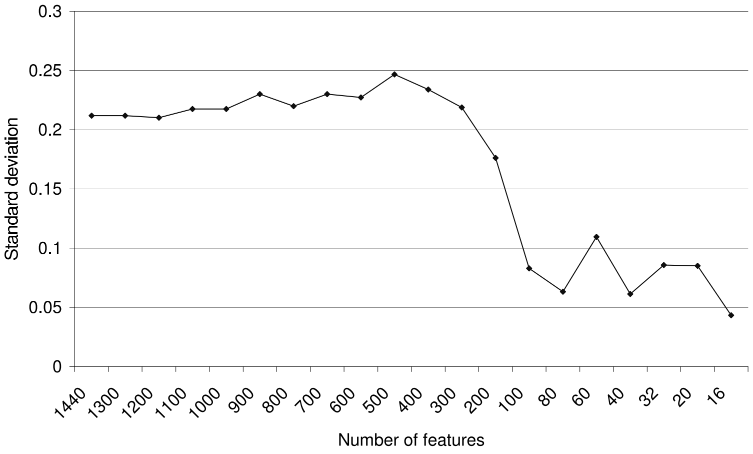


Figure 6.18: Development of the standard deviation over the validation set accuracies obtained for different recording sessions for setup **UD** when the number of features is reduced with the correlation-based method.

when reducing the feature vector dimensionality. That shows that for session dependent estimation of task demand linear feature reduction methods are suitable.

For setup **UD** it is interesting to observe the relation between the standard deviation over the results from different recording sessions and the feature vector dimensionality when correlation-based feature reduction is performed (see figure 6.18): For a smaller number of features the standard deviation becomes smaller as well. Accuracies which are already high for a larger feature vector dimensionality decrease a little (sessions (T1) and (T3a)), while previously low accuracies increase considerably (sessions (T2) and (T4)) for a reduced number of features. This indicates that feature reduction is in particular useful when for a high number of features accuracies are low. In that case, the correlation-based feature reduction method seems to be able to sort out features which introduce a lot of noise in the data (apparently more than SVMs could cope with) and whose presence in the feature vectors is therefore harmful to the predictions.

	No feature reduction	Corr.-based feat. reduction
UD	82%	92% (80 features)
UI	80%	77% (240 features)
SI	87%	61% (240 features)

Table 6.21: Results obtained for feature reduction with the correlation-based method for the different experimental setups compared to the results obtained without feature reduction. In braces the number of used features for each setup is given. Results for the particular setups are always achieved with their optimal system configuration for averaging and normalization.

Table 6.21 compares finally the test set results obtained without feature reduction to those obtained using correlation-based feature reduction method. Based on the validation set results from figure 6.17 a feature vector dimensionality of 80 was determined for setup **UD**. For setup **UI** we argued above that the features selected on the training data are not necessarily the best correlated features with the target values of the test data. Therefore we decided to allow a three times larger feature vector size of 240 for this setup to increase the chance that also features with high predictive power for the test data are included. This feature vector size was also chosen for setup **SI**.

While accuracies for setup **UD** increase by 10%, a decrease of results for the other setups can be observed, although a larger feature vector size was allowed here. The most likely reason for that is that features with the largest predictive power for the test sets are still not included in the feature vectors. This shows that features which are suitable for the prediction of task demand seem to differ a lot across subjects and even across sessions.

For setup **UD** where correlation-based feature reduction yields improvements in results, it is interesting to analyze which features are selected for the different recording sessions, i.e. for the different subjects. In table 6.22 for each session the number of features selected per electrode channel is shown. No commonalities for all recording sessions can be found, however some similarities between two pairs of sessions are noteworthy. For recording sessions (T1) and (T2) electrode T4 has a large importance, although the pre-frontal electrodes Fp1 and Fp2 play an even more important role in case of recording session (T1). For session (T2) only for electrode Fp2 some more features than for other electrodes are selected, but the largest number of features is still taken from electrode T4. Recording sessions (T3a) and (T4) share the importance of electrode O1. While for session (T3a) the data from O1 only is sufficient to make good predictions however, for session (T4) also the frontal and the fronto-temporal cortex plays an important role.

We conclude that features which are most correlated with task demand are generally subject-specific, although there are remarkable similarities at least between groups of subjects. On the other hand one finds for setup **SI** that there are little similarities between the features selected for the two sessions recorded from the same subject ((T3a) and (T3b)). While for (T3a) all 80 features are chosen from electrode O1 as shown above, the fronto-temporal cortex and the parietal cortex are preferred for session (T3b). Therefore it seems that the selected features are not (or not only) subject-specific, but possibly also task-specific. This becomes clear when considering that different types of mental resources are required when different presentations are given to one subject. While during one presentation mathematical reasoning might be more important, for another one the understanding of sketches and visual perception could be required. Naturally, in both cases the correlates of different task demand levels are observed most pronounced at different cortex regions as explained in section 2.3.

Session	Fp1	Fp2	F7	F8	F3	F4	Fz	T3	T4	T5	T6	P3	P4	Pz	O1	O2
(T1)	36	13	3	0	3	0	1	1	12	0	6	1	1	3	0	0
(T2)	2	9	0	2	1	0	2	2	59	3	0	0	0	0	0	0
(T3a)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	0
(T4)	0	6	4	8	5	4	5	5	6	1	3	1	5	3	21	3

Table 6.22: Number of features selected per electrode for the different recording sessions of setup **UD**, when correlation based reduction to 80 features is performed.

	$SVM^{regress}$	OLS-Regression
UD	92%	79%
UI	77%	77%
SI	61%	70%

Table 6.23: Comparison of results achieved by regression SVMs and OLS-regression for the different experimental setups. For all setups the feature vector dimensionality is reduced with the correlation-based method to the values shown in table 6.17 for this method.

The analysis of the selected frequency bands for feature reduction to 80 features for the recording sessions of setup **UD** reveals, that mostly frequency bands above 20Hz are preferred, except for session (T3a) where almost all frequency bands from electrode O1 are chosen. This agrees with the hypothesis from section 2.3.2.1 that in particular high frequent EEG activity is a good correlate of task demand.

For a comparatively small feature vector dimensionality it becomes finally possible to examine the performance of OLS-regression for the prediction of task demand. Table 6.23 compares the results obtained with regression SVMs to those obtained using OLS-regression for a feature vector dimensionality of 80. Interestingly only the results for setup **UD** seem to confirm the hypothesis from section 4.5.3 that regression SVMs perform better than OLS-regression, since the latter method does not perform any regularization to minimize the acual error instead of the training set error. For setup **SI** the application of OLS-regression improves results even by 10%. The reason for this is the average accuracy obtained with a regression SVM for session (T3b) of 26% (which is far below chance) is increased to 79% using OLS-regression. Much deeper analysis of the data from sessions (T3a) and (T3b) would probably be required to understand the differences in results obtained with both methods.

6.2.8 Combination of Two Subjective Task Demand Evaluations

In section 5.3 it was mentioned that some subjects evaluated the task demand they experienced during the perception of the presentation twice. These two evaluations were then used to create the three data partitionings **Eval1**, **Eval2** and **EvalCombined**. In this section the results from experiments with all three data partitionings are reported and discussed in order to analyze the following issues:

1. Is system performance better for partitioning **Eval1** or **Eval2**? With this question, we want to find out whether subjects are still able to estimate their task demand *while* they were given the presentation some days or even weeks *after* the actual presentation.

2. Do results improve when using partitioning **EvalCombined**? Here we want to examine whether in the data from partitioning **EvalCombined** (which contains the overlapping time segments from both evaluations) only those time segments are included for which differences in task

Session	**Eval1**	**Eval2**	**EvalCombined**
(T2)	88%	78%	72%
(T3b)	48%	57%	57%
(T4)	98%	91%	90%
(T6)	55%	92%	70%
Mean	72%	79%	72%

Table 6.24: Average accuracies for the different recording sessions and means over all sessions for data partitionings **Eval1**, **Eval2** and **EvalCombined**. Results were obtained using experimental setup **UD** for which the optimal parameters for averaging, normalization and feature reduction (as determined in the previous sections) were chosen.

demand are particularly pronounced, since time segments for which subjects were unsure how to evaluate them are probably removed in this partitioning.

Table 6.24 shows the average accuracies for the different recording sessions (for sessions (T2), (T3b), (T4) and (T6) two task demand evaluations are available) and the mean over all sessions for partitionings **Eval1**, **Eval2** and **EvalCombined**. Results are obtained using experimental setup **UD**. One can see that in average better results are achieved for partitioning **Eval1** compared to partitioning **Eval2**. In particular the results for sessions (T3b) and (T6) which are below or close to chance accuracy for partitioning **Eval1** improve when using partitioning **Eval2**, while results for sessions (T2) and (T4) drop. The difference of the mean accuracies over all recording sessions between both partitionings is not significant.

We conclude that even some weeks after the actual perception of the presentation subjects were still able to give estimates of their task demand during the initial perception of the presentation which are correlated with their EEG. This seems surprising when the poor overlap between both evaluations (see table 5.4) is considered. On the other hand it must be taken into account however that between both evaluations mostly high task demand and overload or low task demand and medium task demand were confused (see figure 5.7) and that medium task demand and overload are not considered here for prediction. Thus the time segments belonging to task demand levels low and high for which predictions are made, virtually do not overlap between both partitionings, so that it becomes plausible again that acceptable results are obtained for both partitionings.

For the partitioning **EvalCombined** better results were expected because of the above explained hypothesis. However it is important to note that the amount of available data decreases when combining both evaluations (see also 5.4) which might explain that results do not improve compared to partitioning **Eval1** or **Eval2**. Nevertheless accuracies for all sessions are better compared to those obtained when using partitioning **Eval1**.

6.2.9 Electrode Reduction

Also for the task demand data experiments with a reduced set of electrodes were conducted. Table 6.25 compares results obtained for setup **UD** with those obtained for setups **HB** and **UD**^{4}, where data is taken only from the four pre-frontal and frontal electrodes which are also used for the headband recordings. When the data from four electrodes only was used for prediction, the application of feature reduction could not improve results. Therefore also for setup **UD** the results obtained without feature reduction are displayed in table 6.25 for better comparability.

Setup **UD**

Session	
(T1)	93%
(T2)	79%
(T3a)	100%
(T4)	57%
Mean	82%

Setup **UD**^{4}

Session	
(T1)	91%
(T2)	48%
(T3a)	86%
(T4)	50%
Mean	69%

Setup **HB**

Session	
(T7)	89%
(T8)	49%
Mean	69%

Table 6.25: Accuracies for the different recording sessions and means over all sessions for setups **HB**, **UD**and **UD**^{4}. In all cases the system is configured optimally according to the findings from the previous sections. Only feature reduction is not performed.

Setup	Averaging (value of k_{opt})	Normalization method	Feature Reduction
UD	$k_{opt} = 2$	**GlobalNorm**	correlation-based feature reduction to 80 features
UI	$k_{opt} = 2$	**UserNorm**	none
SI	$k_{opt} = 2$	**UserNorm**	none

Table 6.26: Parameters of the best system configuration for setups **UD**, **SI** and **UI**.

As expected, accuracies drop when using only four pre-frontal and frontal electrodes instead of all 16, however they are still clearly above chance. Furthermore there are large fluctuations between the results for different recording sessions for setups **HB** and **UD**^{4}Ḟor both setups for half of the sessions results are in the same range as for setup **UD**, while for the rest accuracies are not better than chance. The reason for these fluctuations is possibly the same as above where it was attempted to explain that across subjects and sessions different features are selected (section 6.2.7): Only for a certain group of subjects or a certain type task to be executed, the mental resources required primarily during task execution produce an EEG which can be measured using no more than four pre-frontal and frontal electrodes.

6.2.10 Analysis of Best System Configuration

In this section the best system configuration for each experimental setup is summarized. In table 6.26 for all processing steps and all experimental setups the parameters are displayed which lead for each processing step to the largest performance gain. Table 6.27 shows the improvements in results for each processing step when the optimal parameters for this step are chosen. All results are achieved with the regression version of a linear SVM.

Note that artifact removal is not considered in tables 6.26 and 6.27, since neither visual inspection based nor cross-validation based rejection of independent components could improve results for the task demand data. Besides the above mentioned problem that ICA weight matrices generalize badly

Setup	Baseline	Averaging	Normalization	Feature Reduction
UD	78%	82% Δ_{abs} = 4% Δ_{rel} = 19%	-	92% Δ_{abs} = 10% Δ_{rel} = 56%
UI	74%	79% Δ_{abs} = 5% Δ_{rel} = 17%	80% Δ_{abs} = 1% Δ_{rel} = 12%	-
SI	73%	73% Δ_{abs} = 0% Δ_{rel} = 0%	87% Δ_{abs} = 24% Δ_{rel} = 89%	-

Table 6.27: Accuracies, absolute (Δ_{abs}) and relative improvements (Δ_{rel}) obtained for setups **UD**, **UI** and **SI** when for the different processing steps the optimal parameter configuration is chosen. A dash indicates that no improvements could be achieved by changing the baseline configuration for the corresponding processing step.

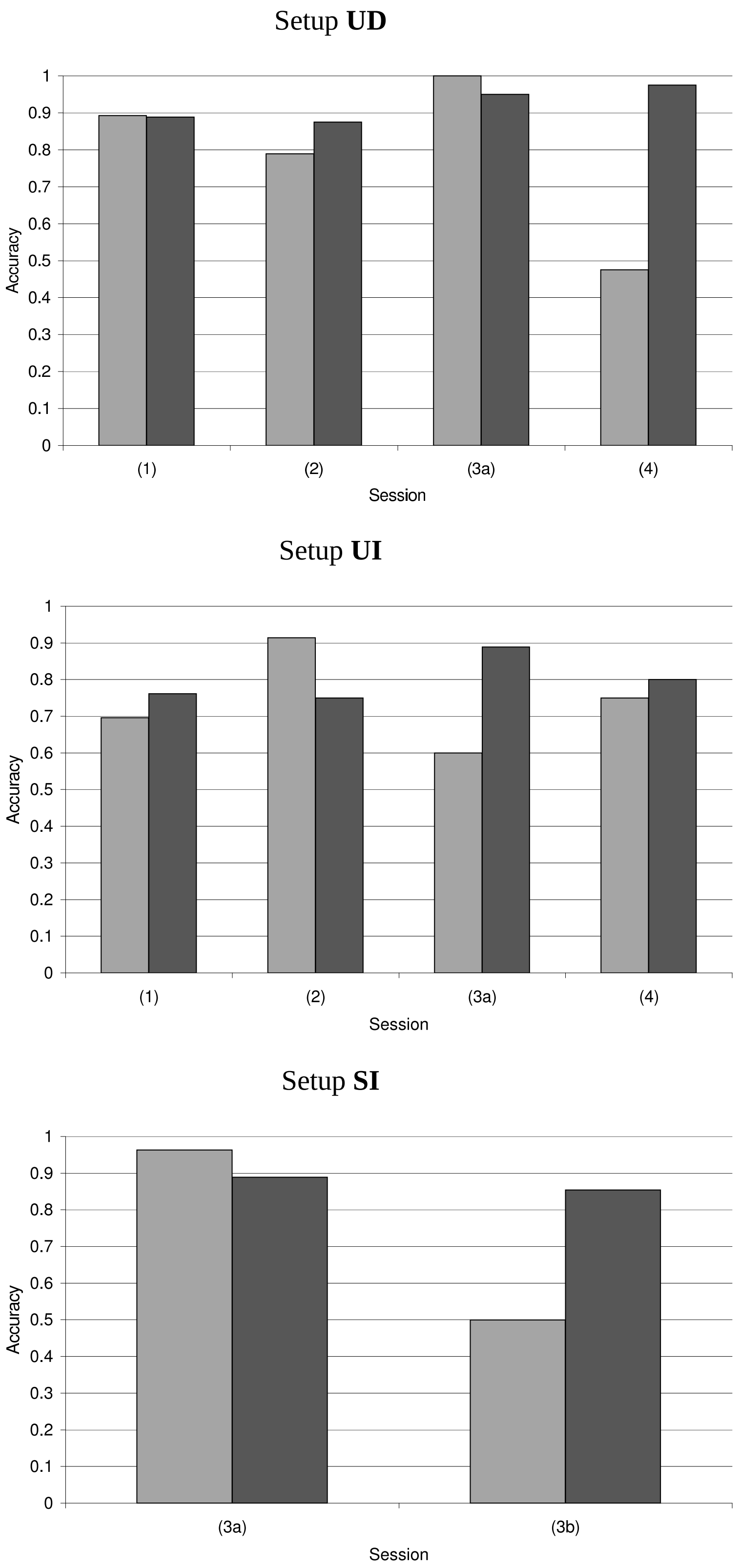
for unseen data (see section 6.1.6), one explanation for this finding could be that the eye-blink frequency is possibly correlated with task demand, so that eye activity information is potentially helpful for task demand estimation. Further research is required however to find evidence for or against this hypothesis. Since also cross-validation based component rejection was not able to improve results, it seems that other processes which are detrimental to the system performance (as in case of the user state data) do not exist for the task demand data, or they can not be isolated properly using ICA.

It can be seen from table 6.27 that similarly to the user state data, also for the task demand data results can be improved for setups **UI** and **SI** when using normalization method **UserNorm** instead of the baseline method **GlobalNorm**. Larger improvements for setup **UI** would be possibly achieved when more data per recording sessions was available so that means and variances per session could be estimated more reliably. The large gains for setup **SI** are due to very large improvements for one recording session. Since results for this setup are only reported as averages over two recording sessions, one has to be very careful with their interpretation.

Correlation-based feature reduction yields improvements in results only for setup **UD**. For the other two setups this feature reduction method would perhaps be applicable as well, if more (or better) validation data was available to estimate an appropriate feature vector dimensionality. This dimensionality is likely to be much larger than 240, which is the feature vector size for which experiments on the test sets of setups **UI** and **SI** were conducted. Averaging over adjacent frequency bands can probably be performed for all setups to keep results constant with a smaller number of features (see also figure 6.16), so that less memory and computational time is required for training and testing.

Figure 6.19 finally compares the accuracies for all recording sessions of all setups which were obtained with the baseline system and with the best system configuration.

For all three setups it can be seen, that often accuracies which are already high for the baseline system drop slightly when the best system configuration is used, while accuracies which are very low for the baseline system increase more or less strongly. Thus not only the average accuracy over all sessions is improved when using the best system configuration, but that also fluctuations across recording sessions are reduced. That suggests that the risk to obtain extremely bad predictions decreases, when using the best system configuration instead of the baseline system. Note however that much more data should be analyzed to be able to provide stronger evidence for or against this hypothesis.



Setup **UD**

Setup **UI**

Setup **SI**

Figure 6.19: Comparison of the accuracies for the different recording sessions between the baseline system and the best system for all experimental setups.

Chapter 7

Conclusions and Future Work

In this thesis a system for the identification of user states and the assessment of mental task demand using electroencephalographic data has been proposed. For the discrimination of the user states resting, listening, perceiving a presentation, reading an article in a magazine, summarizing the read article and performing arithmetic operations, classification accuracies of 94.9% in session and subject dependent experiments, 58.9% in subject independent experiments and 62.7% in subject dependent but session independent experiments could be obtained. For the prediction of low and high task demand during the perception of a presentation accuracies of 92.2% in session and subject dependent experiements, 80.0% in subject independent experiments and 87.1% in subject dependent but session independent experiments were achieved.

Potential fields of application for the proposed system are meeting, lecture or office scenarios, where interaction between users and mobile communication devices or face-to-face communication between persons can be improved, when information about user state and task demand is available (see section 1.2). In section 1.1 several goals were formulated which must be fulfilled to make EEG recording and EEG data processing practical for the mentioned fields of applications. Below it is summarized in how far each of these goals could be met by the research work described in this thesis:

- **Robustness**: For subject and session dependent experiments accuracies obtained for user state identification and task demand estimation are both above 90%. Since these results were achieved when no attempt was made to remove artifacts from the data, which usually occur very frequently when recording EEG under non-clinical conditions, it seems that artifacts have no strong negative influence on the overall system performance. ICA-based artifact removal was only partially successful. While the rejection of those components containing eye activity could not improve results, small but significant improvements were achieved when cross-validation was applied to identify those components which are most detrimental to classification accuracies. A general problem of the ICA algorithm used here seemed to be that weights learned on training data do not generalize well for unseen data. Besides that the robustness of the system towards the variability across sessions and subjects must be increased, since results achieved in session or subject independent experiments are far below those obtained for session and subject dependent experiments (at least for the user state data).

- **Acceptability**: In experiments with the headband for EEG recording (see section 5.1 for more details about this recording device) it was shown that user state identification and task demand assessment is possible with four electrodes over the pre-frontal and frontal cortex only. These are suitable electrode positions for EEG recording in meeting, lecture or office scenarios, since electrodes are easy to attach, no conductive paste gets in contact with the user's hair and the data recorded from these electrodes is sufficient to obtain acceptable user state and task demand predictions in many cases. In the future better positions for electrode attachment should be investigated. For instance they could be mounted on a glasses frame or attached behind the ear lobes. Thus more cortex regions would be covered compared to the headband which records signals from pre-frontal and frontal cortex regions only. Furthermore electrodes at these positions are less visible which might increase the user's readiness to wear them.

- **Real Time Behavior**: On a 800 MHz Pentium-3 laptop preprocessing of raw EEG data and prediction of user states takes about one second for more than 100 examples (which corresponds to 100 seconds of data) and much less than one second for five examples using a MATLAB^{TM}-based implementation of all algorithms[1]. The prediction of task demand requires even less time. With the prototype system (see section 6.1.10) it has been demonstrated that user state prediction is possible in real time.

- **Realistic Scenarios**: The scenarios used for the data collection and for the experiments with the prototype system are very different from the highly controlled recording conditions which must be fulfilled for clinical EEG recording. On the other hand the proposed system has not been tested yet in a real meeting or lecture or during realistic office work. This is however a very essential step which should be made soon. Furthermore the users' real needs during the scenarios we are aiming at should be investigated more, and it should be examined whether EEG recording during such scenarios is acceptable for users in general.

We conclude that all formulated goals haven been realized at least partly during the research described in this thesis, which we consider to be completely satisfactory, since no previous attempt towards EEG-based identification of user states and assessment of task demand in meeting, lecture or office scenarios has been made. Besides the above mentioned tasks to be accomplished, there remain also some other issues to be considered in future research:

- Much more data should be collected, in particular task demand data, to be able to make more reliable inferences about the system behavior when optimizing its parameters. Also other recording setups should be tested (i.e. other user states or other tasks which evoke different task demand levels) in order to learn more about the potential and the limits of the proposed methods.

- A deeper insight into the bio-medical background of the EEG would be of advantage to be able to incorporate more a priori knowledge into the system. It should be analyzed for instance which are the most meaningful features to be extracted. This does not necessarily have to be the frequency content of the signals which can be obtained using a Fourier transform. Frequency-based signal representations using wavelets or features which are extracted from the time signal (in the simplest case the time signal itself and/or its 'derivative', i.e. its slope) could be investigated for example.

- For many processing steps the application of more sophisticated methods should be examined. This includes the use of more robust methods to estimate the frequency spectrum of the signals,

[1] In the MATLAB^{TM} implementation ANNs are used for classification, since no suitable SVM implementation for MATLAB^{TM} was available. The SVM^{light} software [Joachims, 1999] which was used for SVM-based classification and regression usually requires about $1/100^{th}$ of a second to make predictions for a few hundred examples of user state or task demand data, when the learned model is already in memory.

e.g. autoregressive models, the investigation of feature selection techniques which do not imply a linear structure of the data and finally the application of non-linear SVMs.

At the end of this work it shall be pointed out that we believe in the large potential of EEG to obtain information about user state and task demand. Such information is extremely useful for the construction of attentive user interfaces which adapt themselves to the wishes and needs of their owners. Furthermore inferences about how people interact and communicate with each other can be made using information from the EEG. The largest challenge for practical applications of the proposed methodology in the future remains however to minimize the disturbance of users who wear mobile EEG recording devices, while maximizing the benefit of these devices.

Bibliography

[Anderson et al., 1995] Anderson, C., Devulapalli, S., and Stolz, E. (1995). Determining Mental State from EEG Signals Using Neural Networks. *Scientific Programming, Special Issue on Applications Analysis*, 4(3):171–183.

[Anderson and Sijercic, 1996] Anderson, C. and Sijercic, Z. (1996). Classification of EEG Signals from Four Subjects During Five Mental Tasks. In *Solving Engineering Problems with Neural Networks: Proceedings of the Conference on Engineering Applications in Neural Networks*, pages 407–414.

[Applied Anthropology Institute, 2001] Applied Anthropology Institute (2001). Lifeguard Vigilance. Technical report, Applied Anthropology Institute.

[Beatty, 1982] Beatty, J. (1982). Task-Evoked Pupillary Responses, Processing Load, and the Structure of Processing Resources. *Psychological Bulletin*, 91(2):276–292.

[Becker, 2003] Becker, K. (2003). VarioPort^{TM}. http://www.becker-meditec.com/.

[Beer et al., 2003] Beer, R., Lehmann, W., Noldus, L., Paternò, F., Schmidt, E., ten Hove, W., and Theuws, J. (2003). The usability lab of the future. In *Proceedings of Conference on Human-Computer Interaction – INTERACT03*.

[Bell and Sejnowski, 1995] Bell, A. J. and Sejnowski, T. J. (1995). An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7(6):1129–1159.

[Berka et al., 2004] Berka, C. et al. (2004). Real-Time Analysis of EEG Indexes of Alertness, Cognition, and Memory Acquired With a Wireless EEG Headset. *International Journal of Human Computer Interaction*, 17(2):151–170.

[Bishop, 1995] Bishop, C. M. (1995). *Neural networks for pattern recognition*. Clarendon Press.

[Bokranz and Landau, 1991] Bokranz, R. and Landau, K. (1991). *Einführung in die Arbeitswissenschaft*. Ulmer.

[Bolz and Urbaszek, 2002] Bolz, A. and Urbaszek, W. (2002). *Technik in der Kardiologie: eine interdisziplinäre Darstellung f¨r Ingenieure und Mediziner*. Springer, Berlin, Heidelberg.

[Burges, 1998] Burges, C. J. C. (1998). A Tutorial On Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2.

[Chatterjee and Price, 1991] Chatterjee, S. and Price, B. (1991). *Regression Analysis By Example*. John Wiley & Sons, Inc., New York.

[Chen and Vertegaal, 2004] Chen, D. and Vertegaal, R. (2004). Using Mental Load for Managing Interruptions in a Physiologically Attentive User Interface. In *Proceedings of the International Conference on Human Computer Interaction*.

[Crammer and Singer, 2001] Crammer, K. and Singer, Y. (2001). On the Algorithmic Implementation of Multiclass Kernel-Based Vector Machines. *Journal of Machine Learning Research*, 2(6):265–292.

[Croft and Barry, 2000] Croft, R. J. and Barry, R. J. (2000). EOG correction: Which regression should we use? *Psychophysiology*, 37:123–125.

[Culpepper, 1999] Culpepper, J. (1999). Discriminating Mental States Using EEG Represented By Spectral Power Density. Technical report, Harvey Mudd College.

[Defayolle et al., 1971] Defayolle, M., Dinaud, J., and Fourcade, J. (1971). Problèmes théoriques et pratiques de la vigilance. *Revue des Corps de santé*, 12(2):171–186.

[Delorme and Makeig, 2004] Delorme, A. and Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics. *Journal of Neuroscience Methods*, 134:9–21.

[Dössel, 2000] Dössel, O. (2000). *Bildgebende Verfahren in der Medizin: von der Technik zur medizinischen Anwendung*. Springer, Berlin, Heidelberg.

[Dudel and Backhaus, 1996] Dudel, J. and Backhaus, W. (1996). *Neurowissenschaft*. Springer.

[Duta et al., 2004] Duta, M., Alford, C., Wilon, S., and Tarassenko, L. (2004). Neural Network Analysis of the Mastoid EEG for the Assessment of Vigilance. *International Journal of Human Computer Interaction*, 17(2):171–195.

[Electro-Cap International, Inc.,] Electro-Cap International, Inc. Electro-Cap$^{\mathrm{TM}}$. http://www.electro-cap.com/.

[Faller, 1995] Faller, A. (1995). *Der Körper des Menschen*. Thieme.

[Ford, 1996] Ford, D. K. (1996). Analysis of LVQ in the Context of Spontaneous EEG Signal Classification. Master's thesis, Colorado State University.

[Golub and Loan, 1996] Golub, G. H. and Loan, C. F. V. (1996). *Matrix computations*. Johns Hopkins University Press, 3. ed. edition.

[Haeb-Umbach and Ney, 1992] Haeb-Umbach, R. and Ney, H. (1992). Linear Disriminant Analysis for Improved Large Vocabulary Continous Speech Recognition. In *Proceesings of the ICASSP*.

[Hoecks and Levelt, 1993] Hoecks, B. and Levelt, W. (1993). Pupillary Dilation as a Measure of Attention: A Quantitative System Analysis. *Behavior Research Methods, Instruments, & Computers*, 25:16–16.

[Honal and Schultz, 2005] Honal, M. and Schultz, T. (2005). Identifying User State Using Electroencephalographic Data. In *Proceedings of Workshop on Multimodal Multipary Meeting Processing of the ICMI*.

[Hyvärinen and Oja, 2000] Hyvärinen, A. and Oja, E. (2000). Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13(4-5):411–430.

[Iqbal et al., 2004] Iqbal, S. T., Zheng, X. S., and Bailey, B. P. (2004). Task evoked pupillary response to mental workload in human computer interaction. In *Proceedings of Conference of Human Factors in Computer Systems (CHI)*.

[Izzetoglu et al., 2004] Izzetoglu, K. et al. (2004). Functional Optical Brain Imaging Using Near-Infrared During Cognitive Tasks. *International Journal of Human Computer Interaction*, 17(2):211–227.

[Jasper, 1958] Jasper, H. H. (1958). The ten-twenty electrode system of the International Federation. *Electroencephalographic Clinical Neurophysiolgy*, 10:371–375.

[Joachims, 1999] Joachims, T. (1999). *Making Large-Scale SVM Learning Practical*, chapter 11. MIT-Press.

[John et al., 2002] John, M. S., Kobus, D. A., and Morrison, J. G. (2002). A Multi-Tasking Environment for Manipulating and Measuring Neural Correlates of Cognitive Workload. In *IEEE 7th Human Factors Meeting*.

[Jung et al., 2000a] Jung, T. et al. (2000a). Removing Electroencephalographic Artifacts by Blind Source Separation. *Psychophysiology*, 37(2):163–178.

[Jung et al., 1998] Jung, T., Humphries, C., Lee, M., Iragui, V., Makeig, S., and Sejnowski, T. (1998). Removing electroencephalographic artifacts: Comparison between ICA and PCA. In *Proceedings of IEEE International Workshop on Neural Networks for Signal Processing*.

[Jung et al., 1997] Jung, T., Makeig, S., Stensmo, M., and Sejnowski, T. (1997). Estimating Alertness from the EEG Power Spectrum. *IEEE Transactions on Biomedical Engineering*, 4(1):60–69.

[Jung et al., 2000b] Jung, T. P., Makeig, S., Westerfield, M., Townsend, J., Courchesne, E., and Sejnowsky, T. J. (2000b). Removal of Eye-Artifacts in Visual Event-Releated Potentials in Normal and Clinical Subjects. *Clinical Neurophysiology*, 111:1745–1758.

[Keirn and Aunon, 1990] Keirn, Z. and Aunon, J. (1990). A New Mode of Communication Between Man and his Surroundings. *IEEE Transactions on Biomedical Engineering*, 27:1209–1240.

[Kohonen, 1995] Kohonen, T. (1995). Self-organizing Maps. *Springer Series in Information Sciences*, 30.

[Laufs et al., 2003] Laufs, H., Kleinschmidt, A., Beyerle, A., Eger, E., Salek-Haddadi, A., Preibisch, C., , and Krakowa, K. (2003). EEG-correlated fMRI of human alpha activity. *NeuroImage*, 19:1463–1476.

[Lethonen, 2003] Lethonen, J. (2003). EEG-based Brain Computer Interfaces. Master's thesis, Helsinki University of Technology.

[Li et al., 1999] Li, Y., Kittler, J., and Matas, J. (1999). Effective Implementation of Linear Discriminant Analysis for Face Recognition and Verification. In *Proceedings of the 8th International Conference on Computer Analysis of Images and Patterns*.

[Makeig et al., 1996] Makeig, S., Bell, A., Jung, T., and Sejnowsky, T. (1996). Independent Component Analysis of Electroencephalographic Data. *Advances in Neural Information Processing Systems*, 8.

[Martinez-Montes et al., 2004] Martinez-Montes, E., Valdes-Sosa, P. A., Miwakeichi, F., Goldman, R. I., and Cohen, M. S. (2004). Concurrent EEG/fMRI analysis by multiway Partial Least Squares. *NeuroImage*, 22:1023 – 1034.

[Meyer-Waarden, 1985] Meyer-Waarden, K. (1985). *Bioelektrische Signale und ihre Ableitverfahren*. Schattauer.

[Mitchell et al., 2004] Mitchell, T. et al. (2004). Learning to Decode Cognitive States from Brain Images. *Machine Learning*, 57(1–2):145–175.

[Niedermeyer and da Silva, 1987] Niedermeyer, E. and da Silva, F. L. (1987). *Electroencephalography: Basic principle, clinical applications and related fields*. Urbarn and Schwarzenberg.

[Nielsen, 1993] Nielsen, J. (1993). *Usability Engineering*. Academic Press.

[Pineda et al., 2000] Pineda, J., Allison, B., and A.Vankov (2000). The effects of self-movement, observation, and imagination on μ rhythms and readiness potentials (RP's): toward a brain-computer interface (BCI). *IEEE Transactions on Rehabilitation Engineering*, 8(2):219–222.

[Pleydell-Pearce et al., 2003] Pleydell-Pearce, C. W., Whitecross, S. E., and Dickson, B. T. (2003). Multivariate Analysis of EEG: Predicting Cognition on the basis of Frequency Decomposition, Inter-electrode Correlation, Coherence, Cross Phase and Cross Power. In *Proceedings of 38th HICCS*.

[Schmidt and Thews, 1997] Schmidt, R. F. and Thews, G., editors (1997). *Physiologie des Menschen*. Springer.

[Schmorrow and Kruse, 2002] Schmorrow, D. D. and Kruse, A. A. (2002). DARPA's Augmented Cognition Program - Tomorrow's Human Computer Interaction from Vision to Reality: Building Cognitively Aware Computational Systems. In *Proceedings of IEEE 7th Human Factors Meeting*.

[Scientific Learning Cooperation, 1999] Scientific Learning Cooperation (1999). www.BrainConnection.com. http://www.brainconnection.com/.

[Smith et al., 2001] Smith, M., Gevins, A., Brown, H., Karnik, A., and Du, R. (2001). Monitoring Task Loading with Multivariate EEG Measures during Complex Forms of Human-Computer Interaction. *Human Factors*, 43(3):366–380.

[Smola, 1998] Smola, A. J. (1998). *Learning with Kernels*. PhD thesis, Technische Universität Berlin.

[Smola and Schölkopf, 1998] Smola, A. J. and Schölkopf, B. (1998). A Tutorial on Support Vector Regression. Technical report, Gesellschaft für Mathematik und Datenverarbeitung.

[Student Government, University of Maine, 2005] Student Government, University of Maine (2005). GSS Meeting 1. http://www2.umaine.edu/StudentGovernment/pictures.

[Tsochantaridis et al., 2004] Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. (2004). Support Vector Machine Learning for Interdependent and Structured Output Spaces. In *Proceedings of the ICML*.

[Vaughan et al., 2003] Vaughan, T. et al. (2003). Brain-computer interface technology: A review of the Second International Meeting. *IEEE Transactions on Neural Systems & Rehabilitation Engineering*, 11(2):94–109.

[Vesanto et al., 2000] Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J. (2000). SOM Toolbox for Matlab 5. Technical report, Helsinki University of Technology.

[Wolpaw et al., 2000] Wolpaw, J. R. et al. (2000). Brain-computer interface technology: A review of the first international meeting. *IEEE Transactions on Rehabilitation Engineering*, 8(2):164–173.

[Zschocke, 1995] Zschocke, S. (1995). *Klinische Elektroenzephalographie*. Springer.