

An investigation into time structure of EEG-based cursor control in a brain-computer interface using machine-classification methods

Pieter Laurens Baljon
stud. no. 1211692

September 2006

Supervisors:

prof. dr. Lambert Schomaker
dr. Tjeerd Andringa

Artificial Intelligence
University of Groningen



Contents

1	Introduction	7
1.1	Research question	8
2	Theoretical background	11
2.1	Measuring neuronal activity	11
2.2	Neurophysiology	13
2.3	Analysis of the EEG signal	15
2.3.1	Temporal analysis	15
2.3.2	Spectral analysis	15
2.3.3	Real-time spectral estimates and spatial filtering	17
2.4	Machine Learning	18
2.4.1	k Nearest Neighbors	19
2.4.2	Linear model	19
2.4.3	Time models	20
2.5	Brain-Computer Interface research	21
3	Methods	23
3.1	Experimental setup	24
3.1.1	Apparatus	24
3.1.2	Design of BCI2000	24
3.2	User tasks	25
3.2.1	Motor imagery tasks	25
3.2.2	Motor imagery sessions	26
3.2.3	Feedback sessions	26
3.3	Data analysis for feature selection	28
3.3.1	Data overview	28
3.3.2	Correlation	29
3.3.3	Mutual Information	31
3.3.4	Correlation over time	32
3.4	Data sets	33
3.5	Experiments with instantaneous classifiers	35
3.5.1	k Nearest Neighbors	36
3.5.2	Linear model	36
4	Models for trial-based operation of a BCI	39
4.1	Types of time structure	40
4.2	Hidden-Markov Model for classifying EEG	42
4.3	Practical issues in modeling	44

4.4	Variations on the re-estimation procedure	46
4.4.1	Flat HMM	46
4.4.2	Common-structure HMM	46
4.5	Experiments with HMM variants	48
4.6	Experiments on artificial data	48
5	Results	51
5.1	Performance of Temporal models on artificial data	51
5.2	HMMs compared to instantaneous classification	52
5.3	Performance on reduced training sets	56
5.4	Comparison of HMM variants	58
5.5	Qualitative results	59
6	Conclusion & Discussion	61
6.1	Instantaneous classifiers and temporal models	61
6.2	The type of time structure	62
6.3	Discussion of the results	63
6.4	Limitations	65
6.5	Future work	66
A	MatLab Manual	69
A.1	Continuous Observation HMM	69
A.2	Wrapper method for Machine Learning	72
A.3	Data Exploration Methods	73
B	Datasets	75

Notation

\vec{x}	Feature vector
d	Dimensionality of feature vector.
x_i	Feature (element of \vec{x})
y	Scalar value. Output of regression.
ω	Class label.
\hat{a}	Estimate of a . <i>Outcome of regression or classification can be regarded an estimate, e.g. in the latter $\hat{\omega}$.</i>
a^\top	Vector or matrix transpose of a .
s_x^2	Sample variance in variable x .
\bar{x}	Sample mean of x .
$\mathcal{N}(a, b)$	Normal distribution with mean a and standard deviation b .
$\chi^2(a)$	Chi-square distribution with a degrees of freedom.
ν	Number of degrees of freedom in χ^2 distribution.
$\langle x, y \rangle$	Standard inner product between vectors x and y .
β	Weight vector in linear model
r	Correlation.
ϕ_τ	Autocorrelation at lag τ .
n	Number of classes in classification, or number of observations used for computation of an aggregated measure such as correlation.
p	Order of AR-model.
k	Number of neighbors considered in k Nearest Neighbors.
T	Duration of a trial.
N	Number of training samples.
N_s	Number of states in Hidden-Markov Model.
$\mathbf{N}_k(\vec{x})$	Set of k neighbors around feature vector \vec{x} in k Nearest Neighbors.
s_i	State in Hiden-Markov Model ($i \in \{1 \cdots N_s\}$).
o_u	Observed feature.
O_t	Observed feature vector at time t . $O_t = \{o_1 \cdots o_d\}^\top$.
\mathbf{O}	Observed trial. $\mathbf{O} = \{O_1 \cdots O_T\}$
a_{ij}	Transition probability between s_i and s_j in Hidden-Markov Model.
A	Matrix of transition probabilities a_{ij} .
π_i	Prior probabilities of Hidden-Markov Model.
B	Set of parameters for observation distribution in Hidden-Markov Model.
λ	Instance of Hidden-Markov Model $\{A, \pi, B\}$.
$\gamma_t(i)$	The probability of being in state s_i at time t in a trial.
$H(X)$	Entropy in variable X .
$I(X, Y)$	Mutual Information of variable X with respect to variable Y .

Chapter 1

Introduction

To read minds has long been man's great desire. Opinions differ on whether this will ever be possible, be it by machines or humans. Though there exists abundant philosophical dispute about whether or not it is possible *in principle*, time must tell whether this ambitious goal will ever be attained.

In this thesis I pursue a simplified version of mind reading. Recent findings have led us to believe it is possible to discern several different cognitive states¹ from recordings of neural activity. During this research I try to obtain similar results. The ultimate goal of this project is to use these results to interact with a computer through the performance of mental activities.

In this research we built a so called Brain-Computer Interface (BCI). Such a BCI is opposite to the natural interface between brain and computer: the arms and hands operating a keyboard or mouse with the ensemble of muscles in the arms. In the natural case, the brain is seamlessly integrated with its interface to a computer as the control systems for arm movement are located in the brain itself.

A BCI may be required when one or more parts of this motor tract are deficient. This may be if a link in this tract lacks entirely as for example with an arm amputee or paralysis patients, or if the natural interface provides insufficient bandwidth, as for example with a fighter pilot.

Nowadays prostheses for missing limbs are controlled by neurons originally used for motor control but rendered useless by the amputation. The output of these neurons can still be controlled by the brain similar to conventional motor control. This branch of rehabilitation research is however not part of what is currently indicated by BCI research. During the the first international BCI conference in 1999 a BCI was defined stating that it

”must not depend on the brain's normal output pathways of peripheral nerves and muscles.” [1].

In practice this comes down to the requirement that whatever information we derive about the cognitive state of the participant must be recorded from the brain itself.

¹A *cognitive state* is the state of the brain during a specific type of mental activity. This may be compared to how posture is the state of the body during physical activity. Examples of mental activities include thinking about one's favorite tune, solving math problems or imagining the movement of a limb.

Recent years have seen an enormous growth of the field of BCI research. It has been shown in a multitude of institutes and experimental setups that we are able to provide subjects with reasonable control over a computer interface. The most widely used paradigms are continuous cursor control [2] and the operation of a spelling device [3].

However a lot of work remains to be done. Current accuracy leaves room for improvement, the training required by the subject is a strenuous enterprise and there exists uncertainty about the type and amount of control subjects have over measured neural activity. Overall the operation of a BCI by no means allows for practical use by the disabled outside an experimental setup.

This research is part of the project *Moving Thoughts* for BCI research in Groningen. Related research is performed in parallel aimed at developing a more natural form of control and adding a 'mouse click' to the paradigm of cursor control. Our combined efforts serve as a pilot research in the start-up phase of the Moving Thoughts project.

This thesis is organized as follows. The second chapter will provide an introduction to the sub-fields that constitute BCI research and present the theory used in the following chapters. Chapter 3 consists of three parts. The first part describes the methodology of the experiments with participants. The second part describes how we determined what part of the EEG was informative. The final part describes the machine learning approaches used for comparison to the more advanced methods. Chapter 4 presents these more advanced methods: we detect cognitive states by means of models of an entire trial. Chapter 5 will discuss the results of the comparison between the model-based approach and baseline methods. Finally, chapter 6 will attribute meaning to these results and sketch a framework for future work within this project. The appendices contain specific descriptions of software developed during this project and additional figures.

1.1 Research question

The original research question to be answered by this project was:

How can a machine learn a transformation of a subject's EEG signal into his intended movement, with 1) high accuracy of control by the subject, 2) robustness to suboptimal transformations and differences between subjects and finally 3) in little training time?

Over the course of this project I have continually focused this question. The aforementioned research question is broad in the sense that it captures the whole branch of research concerned with machine learning for BCI. In this project we answer a more limited research question, while we provide useful insights and practical directions for answering the remaining questions. The research question addressed in this research is the following:

Does accounting for time structure of EEG spectra within a trial improve a BCI over instantaneous classification of those spectra with respect to 1) accuracy and 2) training time?

The answer to this question contributes novel insights to current BCI research. It however also raises more fundamental questions about the nature of control participants have over their EEG recordings. Therefore I also try to address the question:

Does the time structure of spectra within a trial provide information about the cognitive state, subsidiary to the individual instantaneous spectra?

In order to answer these questions I will implement three types of trial models and draw conclusions from the comparison of the performance of these models to each other and to instantaneous classifiers.

Chapter 2

Theoretical background

This chapter will provide an overview of current BCI research, focusing on EEG based paradigms. BCI research is a practical form of research and it brings a range of other fields together such as neuro physiology and motor learning, EEG signal analysis and machine learning. We will first provide an overview of these relevant sub fields. In the last section we will use these fields as a natural framework to describe current themes in BCI research.

The chapter is structured as follows. Section 2.1 gives an overview of the state of the art of neuro imaging. This is relevant as all neuro imaging in principle might serve as the basis for a BCI. Section 2.2 describes the neuro-physiological processes leading us to believe a brain-computer interface is at all feasible. Section 2.3 provides an introduction into interpreting an EEG signal and gives an overview of technical issues of concern in our implementation. Section 2.4 describes the machine learning methods we use for comparison. Finally Section 2.5 gives an overview of current BCI research with respect to the aforementioned fields of research.

2.1 Measuring neuronal activity

Since long it has been man's desire to know what is going on in the brain. This research is only a very small piece of that puzzle. Current imaging techniques have brought us closer than ever to the answer. Figure 2.1 lists the five currently most prominent imaging techniques. The graph shows how these techniques relate with respect to their spatial resolution (what is the smallest difference in space a method can discern?) and temporal resolution (what is the smallest step in time a method can discern?).

Positron Emission Tomography (PET) uses a radioactive substance injected in the blood. This substance emits radiation that can be measured outside the body. The exact origin inside the body of the radiation can then be determined. The amount of radiation from a position is a measure of the concentration of the substance at that location. Now, if the radioactive substance is connected to oxygen, the radiation is taken as a measure of the amount of oxygen that is used, which is a measure of the cell activity. Though the spatial resolution is high, it takes considerable time for enough radiation to have been emitted, therefore temporal resolution is low. Another important drawback is the use of

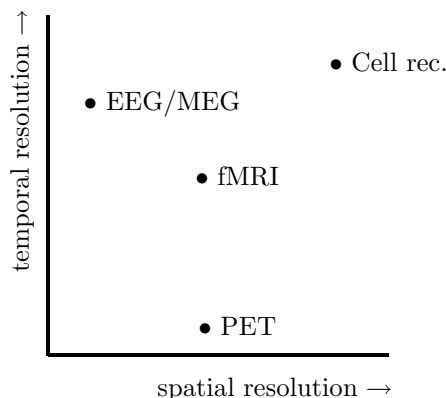


Figure 2.1: Qualitative comparison of current neuro-imaging techniques with respect to their spatial and temporal resolution. A high resolution implies small discernible differences which is desirable.

radioactive substances. PET is used primarily for the imaging of tumors and not for research related to brain-computer interfacing.

Functional Magnetic Resonance Imaging (fMRI) measures the oxygen levels in blood by using the difference in magnetic properties between blood with and without oxygen. The spatial resolution of this method is as high as PET resolution and the temporal resolution of fMRI is better; of the order of seconds. The method has no known negative effects on the health of a participant. This makes the method very useful for research on localization of functionality in cognitive tasks. fMRI has also been used to discern cognitive tasks e.g. in [4][5]. This would in principle allow a subject to interact with a computer. However the equipment required for fMRI measurements limits the practical use of the technique for actual BCIs. We can however use results obtained by high-resolution MRI to direct research in EEG with lower spatial resolution.

Recording the potentials at single cells is the most direct method of measuring brain activity. It has the highest possible spatial resolution, and temporal resolution is limited only by the sample rate of the recording. Single-cell recording introduces new difficulties. The method does not allow for a holistic view as it is impossible to record more than a fraction of the total number of neurons. Furthermore the invasive nature of this method makes it more problematic to use it on human subjects in experimental conditions. The work is primarily done on monkeys[6], though there is also a growing body of literature describing work on paralyzed humans, for example [7] or an overview in [8]).

Finally, Electroencephalography (EEG) measures the potential on the scalp. Neurons transmit information electrochemically, therefore neuronal activity creates an electric field¹. EEG measures the summed electric field of the neurons perpendicular to the scalp [9]. Though the influence of a neuron on the electrode decreases with the distance to that electrode, in principle the potential is summed over all neurons lying under the scalp. Therefore the spatial resolution of an EEG is low.

How neurons affect an electrode is different over the scalp. It is dependent

¹Neuronal activity also yields an accompanying magnetic field perpendicular to this field which is used in *Magnetoencephalography* (MEG).

on the neuro anatomy of networks of neurons. The orientation of a neuron determines where on the scalp its effect is maximally present. It is also unclear from how deep we measure effects. In clinical settings, brainstem potentials are measured by an electrode positioned central on the scalp.

Since the measured potential is the direct effect of neuronal activity, temporal resolution is high. EEG is currently the most widely used method for BCI research on humans [2][3][10][11][12].

These methods for measuring neural activity have led to an ever more precise knowledge of neural processes. Though the exact dynamics of the brain as a whole are still an open question, general principles of neural processing are now widely accepted in the neuro-scientific community. The fact that these principles are widely accepted in the neuro sciences, does not mean they are no longer the subject of lively debate in the philosophical community. See for an overview [13].

The relevant principles will be discussed in the next Section. We stress that these principles are not absolute truths but rather serve as principles allowing for effective practical research in a field where knowledge of the fundamental dynamics underlying the measurements is a distant goal.

2.2 Neurophysiology

One of the most important principles of current neural science is that it may prove beneficial to regard the brain not as a giant parallel processor, but as having some area's specialized for distinct functions [14]. The cognitive functions are then assumed to be located on the outer layer of the brain, the cerebral cortex. A modality such as visual processing is distributed over several even smaller regions of the cortex. These sub-functional regions itself consist of cortical columns [9], which are thought to be the elementary computational elements of the brain.

The sub-regions within modalities are assumed to be hierarchically structured with large numbers of interconnections between levels of the hierarchy. The first steps of sensory processing are basic steps which act as input for higher (i.e. more complex) processing area's. In this way, over 30 area's concerned with visual processing in humans have been identified. The order of the hierarchy of motor control is the inverse of sensory processing. The final cortical area involved in motor control is the primary motor cortex and it represents the most basic information, i.e. the information sent to the brainstem and spinal chord. The higher (i.e. earlier) motor areas are concerned with planning of longer series of actions.

Sensory and motor areas are topographically organized. This means that adjacent neurons in the cortex are innervated by adjacent sensory neurons or innervate adjacent muscles respectively. The organization of the primary motor cortex is similar for (almost) all humans. We may draw the limbs over the motor cortex, showing which part of the brain controls what limb, see Figure 2.2. In this context we say for example that the feet are *represented* within the central sulcus, and the face is *represented* to the side of the motor cortex.

The size of a representation on the motor cortex is not based on the size

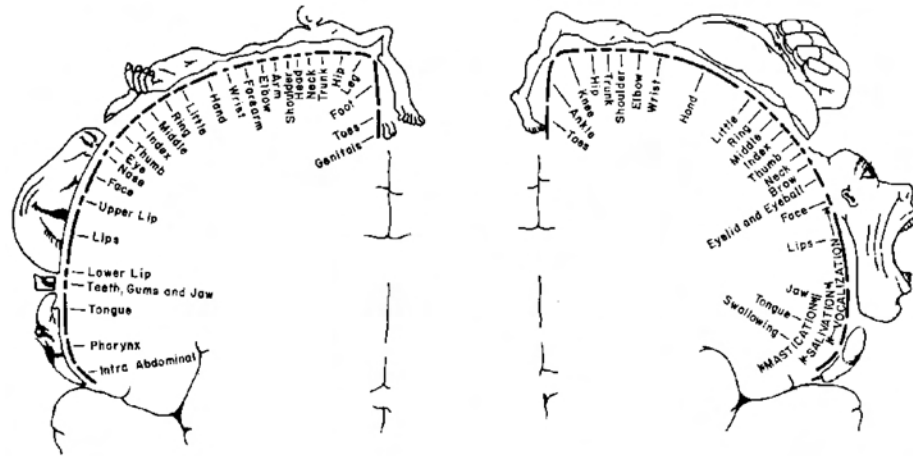


Figure 2.2: The 'Homunculus' illustrates which part of the motor cortex innervates which part of the body. We refer to area's in the motor cortex as *representations* of the limb they innervate. (From Love & Webb 1992, p. 19)

of the body part but on the measure of innervation of that body part. The hands allow for very fine control. This fine control is accomplished through very fine innervation from the motor cortex and therefore the hand representation occupies a large part of the motor cortex. The legs on the other hand contain large muscles, but as they do not require fine control, the innervation from the brain is limited, and therefore so is the representation on the cortex.

Both executing a movement and only *imagining* the execution of that movement (motor imagery), yield similar EEG patterns [15][9]. Since during voluntary movement we observe correlated changes in EEG recordings over the motor cortex, motor imagery seems a sensible starting point in the search for EEG components under voluntary control. Results from other groups indicate that motor imagery as an internal paradigm for the subject can indeed yield satisfactory control in a one-dimensional task [16][10][11].

Practically all functional systems are lateralized, e.g. muscles on the right side of the body are controlled by the motor cortex on the left hemisphere and vice versa. It is common to exploit this lateralization by using the difference between two points on the cortex opposite to one another. This can improve the signal to noise ratio because due to the lateralization, an effect for left- and right motor imagery should yield opposite signs which simplifies the problem of finding an appropriate bias. It does however consume one degree of freedom.

EEG measures the resultant activity of millions of neurons and therefore the spatial resolution is low as noted earlier. Smaller (motor-)representations are assumed to be more susceptible to disturbances from neighboring functional area's which might mask the effect due to this low spatial resolution. Therefore it seems desirable that the area on the cortex concerned with a certain motor function is large in order for imagery of that motor function to be suitable for this type of control.

We use this knowledge of the functional organization of the brain to instruct cognitive tasks to the subjects, of which we now know where to expect an effect. The exact relation between the mental state (during the cognitive task) and measured potential is not known. The challenge is now to extract features from the EEG that inform us best about the underlying mental state, i.e. constitute the inverse of the relation between mental state and potential. This is the subject of the next section.

2.3 Analysis of the EEG signal

The information in the EEG is hidden very well between large sources of noise and cognitive processes we are not interested in. At present we have a large array of analyzes at our disposal to uncover the information of which the most prominent and widely used will be discussed in the following section. The analysis of an EEG is performed in the time domain, frequency domain, or 'somewhere in between'. These alternatives span the following two subsections.

2.3.1 Temporal analysis

A common form of EEG research is in psychological experiments where the researcher is interested in the time course of the potential in two or more different conditions. To this purpose the subject is asked to perform an experiment repeatedly in each condition. The experiments trigger a behavioral response of the subject through a stimulus, with a measurable counterpart in the EEG. The change in EEG due to this stimulus is time locked to the trigger, whereas all other influences on the EEG are randomly shifted with respect to that trigger. By averaging over many repetitions of a trial the researcher can extract the Event Related Potential (ERP) since the other influences average to zero as a result of their random phase shift.

It is dependent on the effect size under consideration whether it is possible to detect such a time course in a single trial *without* averaging. If an effect is too small, unrelated fluctuations make the effect invisible. A well known temporal effect in the potential is the Bereitschafts Potential (BP, or Lateralized Readiness Potential: LRP). This effect is large enough to be detected in single trials and serves as a basis for BCI research [11]. In the present research we will not use any features from the time domain.

2.3.2 Spectral analysis

Spectral analysis breaks a signal down into its constituting frequencies [17]. Stated informally: it shows for every frequency how well a waveform of exclusively that frequency fits the original signal. Spectra are useful when we are interested in the frequency of a particular phenomenon or, as in this case, when we can attribute meaning to certain frequencies and study the phenomenon through its presence in the spectrum. The presence of a certain frequency is referred to as the *power* of that frequency.

The effect of a single neuron cannot be detected by an EEG electrode (the potential is too low). Furthermore, we cannot discern between two neurons alternating their spikes, both at 5Hz and one neuron firing at 10Hz. Therefore,

tradeoff between the reliability of a spectral estimate and the agility of the control system [17]. Though the tradeoff imposes a strict boundary, means of analysis exist that unite a time- and frequency representation such as Wavelets and other representations (specifically in BCI research [20]). Another method of this sort employed especially at our faculty is continuity preserving signal processing [21] which may also be used for this type of analysis in the future. These methods are however not yet widespread in BCI research. Most research either exclusively uses time or frequency features, or combines the two in the machine learning phase [22].

2.3.3 Real-time spectral estimates and spatial filtering

Most BCI research uses Autoregressive (AR) models to obtain estimates of spectral power. This method is best described under the name 'Maximum Entropy Method' or 'All Pole Method' in [23, p. 547] and [24, p. 430]. Another common method for spectral estimation is the Fast Fourier Transform (also described in [24, p. 381]). However, AR models have the advantage of allowing the user to reduce computational load by reducing the spectral resolution. This is a major advantage when the system has to work in real time.

Suppose we observe a signal y consisting of consecutive samples y_i . An AR model expresses the point y_t as a weighed combination of previous points $y_{t-p}..y_{t-1}$ with p the order of the model. An AR model for a zero-mean signal is of the form

$$\hat{y}_t = \sum_{i=1}^p a_i y_{t-i}$$

The modeling exercise lies in finding coefficients a_i . To estimate these coefficients one uses the autocorrelation ϕ . The autocorrelation ϕ_τ is defined as the correlation between points y_t and $y_{t+\tau}$ over t . This definition is intuitively close to the formulation of the coefficients a_τ . The exact computation is described in [24, p. 430].

AR models use the autocorrelation function over p lags on a segment of the signal to compute the spectral estimates. A longer segment increases the reliability of the estimate of the autocorrelation. Increased reliability of the autocorrelation also increases reliability of the spectral estimates. However, a longer segment of course also increases the delay of the feedback loop.

By using more lags τ of the autocorrelation ϕ_τ the *order* of the AR model increases. A higher model order increases the number of peaks in the spectrum we can discern. If the order is too low, separate peaks may be 'smeared' together. If the order is too high, a single large peak may be untruthfully separated. A proper setting for segment length and model order can be determined from other research groups and by comparing different settings in offline analysis of the raw signal. It is also common to use the coefficients a_i directly or derivatives other than spectral power as features. We prefer the fact that spectral powers bear a functional meaning as described earlier.

A way of improving spatial resolution is by using spatial filters [25]. Spatial filters operate as focused high-pass filters by subtracting the activity of neighboring electrodes. Different filters differ in the exact set of electrodes they subtract from the electrode of interest. Common average reference (CAR) subtracts the

mean signal of all the electrodes. Laplacian methods subtract the mean of only the surrounding electrodes. The Small and Large Laplacian subtract the mean of the electrodes at a distance of 3cm and 6cm respectively, see Figure 2.4.

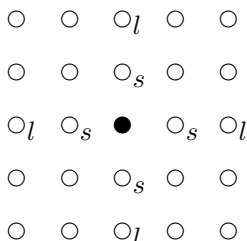


Figure 2.4: The operation of a Laplacian spatial filter with the black electrode that of interest. For a Small and Large Laplacian, the average of the electrodes labeled s and l respectively is subtracted. In the 10-20 system only around the electrodes over the central gyrus (Cz and C1 to C8) a perfect grid as here exist. For other locations the electrodes must be placed at the appropriate distances by hand.

2.4 Machine Learning

Currently there are two main approaches to machine learning in BCI research. We may discern groups performing *classification* and those performing *regression*. Classification can be characterized as a mapping of feature vectors \vec{x} to an element of a finite set of n classes ω_i .

$$\vec{x} \mapsto \hat{\omega} \in \{\omega_1, \omega_2, \dots, \omega_n\}$$

The set of classes in a simple BCI setting might for example be *up* and *down*.

Regression is a mapping of feature vector \vec{x} to scalar value y .

$$\vec{x} \mapsto \hat{y} \in \mathbb{R}$$

We can use the scalar y as a control signal for the cursor, for example as the horizontal velocity or displacement of the cursor.

Many forms of classification have been employed as reflected in the contributions to BCI competitions [26] and [27]. We will use k Nearest Neighbors as a method of classification as described in Section 2.4.1. For regression we will use a Linear Model as described in Section 2.4.2. This method is used for example in the BCI in Albany [2] after sophisticated feature selection.

A different approach to classifying EEG is through modeling the trial as a whole rather than to classify each point in time individually. This approach is central to our research question. We use Hidden-Markov Models to this purpose. Hidden-Markov Models are introduced in Section 2.4.3 and our particular approach is described in detail in Chapter 4.

2.4.1 k Nearest Neighbors

The method of k Nearest Neighbors [28, p. 174] is one of the most basic and perhaps most intuitive methods of supervised learning. The method makes no assumptions about distributions underlying the data. k NN needs a training set $T = \{(\vec{x}_i, \omega^i)\}$ of labeled examples prior to classification. When classifying a new instance \vec{x} , we define the subset $\mathbf{N}_k(\vec{x}) \subset T$ as the k points in T closest to \vec{x} . We define Ω as the labels in $\mathbf{N}_k(\vec{x})$. We classify the new instance \vec{x} as a function of the labels in Ω :

$$\hat{\omega} = f(\Omega)$$

This function can express plurality voting or majority voting. In the former case $f(\cdot)$ returns the mode² of the set. In the latter case a classification is returned only if the number of elements of one class exceeds $k/2$. In the case of a dichotomy these methods are equivalent. When more knowledge about the underlying type of data is inserted into the form of $f(\cdot)$ the method is strictly no longer k NN, but it is very well conceivable.

The only assumptions on the underlying structure are made in the choice of distance metric such as Euclidean, Hamming etc., and the number k of neighbors. The number k is usually chosen an odd number especially in the case of two-class problems, for obvious reasons. Furthermore this method has the nice property that with N the number of training samples, for $N \rightarrow \infty$ and $k \rightarrow \infty$ but $k/N \rightarrow 0$ and odd, k NN approaches the Bayes optimum [28].

Of course this method is limited in many ways. The most notorious deficits of the method are slow classification (since for every new instance, the distance to all training samples must be computed and they must be sorted) and its lack of generalization if the number of training samples is small. Still, a no-knowledge approach is attractive as a baseline for pattern recognition.

2.4.2 Linear model

A linear model is a function $f(\vec{x}) = \hat{y}$. One generally chooses an $f(\cdot)$ minimizing the difference $\hat{y} - y$ with y the label corresponding to \vec{x} . In a linear model f is a linear combination of the components x_i of \vec{x} and therefore $f(\vec{x})$ can be expressed as the inner product of \vec{x} with a weight vector β :

$$y = f(\vec{x}) = \langle \vec{x}, \beta \rangle = \sum_{i=1}^d x_i \beta_i$$

with d the dimensionality of the feature vectors. This weight vector is to be estimated from a training set. It is common to use *augmented* feature vectors $\vec{x}' = \{1, x_1, \dots, x_n\}^T$. The element β_0 of the corresponding augmented weight vector now represents the bias or offset in the model.

There are standard techniques to find a solution in the minimum squared error sense for the weight vector β . We use the method of the *matrix pseudo inverse* which is described in [28, p. 246]. The performance of a linear model

²The mode of a set is the element with the highest frequency. This statistic is also defined for nominal data types. For example, the mode of the set $\{left, left, right, up\}$ is *left*, as opposed to the mean of a set.

can be tested by the mean squared error on a test set of N samples.

$$MSE = \frac{1}{N} \sum_{i=1}^N (f({}_i\vec{x}) - {}_iy)^2$$

In this case the index i in ${}_i\vec{x}$ and ${}_iy$ is over feature vectors and corresponding labels in the test set rather than component features of a vector as in \vec{x}_i .

Another measure of performance that is common in BCI research is the correlation between $f(\vec{x})$ and label y . A problem may arise if the label y assumes binary values, or otherwise occupies only a subset of \mathbb{R} . This is problematic since the Pearson product-moment correlation produces a biased underestimate if it is computed between continuous variable X and binary variable Y . In that case we must use point-biserial correlation [29] with X_a indicating the values of X for which $Y = a$, s_x the standard deviation in X and p the proportion of $Y = 1$:

$$r = \frac{(\bar{X}_1 - \bar{X}_2)\sqrt{p(1-p)}}{s_x}$$

Whether classification or regression is more appropriate is an open question. Regression introduces the concept of a (linear) relation between the amount of power in a frequency band and the speed or location of the cursor. Classification coerces the system to be in either one of several states. Literature [19] allows for speculation about the ability of subjects to influence the effect size of spectral phenomena corroborating the hypothesis there exists an analog relation between power and intended motion of the cursor. There is also report of non-linear regression, using interactions between features [30].

2.4.3 Time models

The linear model and k NN disregard the fact that there is an underlying time structure in the data as it is obtained from a trial. In order to better understand EEG processes during brain-computer interaction, we will need to look at models able to capture these temporal processes. Hidden-Markov Models (HMM) are a well known method to model time series [28, p. 128]. HMMs have gained fame over their application in speech recognition [31] and handwriting recognition but are also a frequent instrument in EEG analysis and classification e.g. in [32] and [33].

A Markov model consists of a set of states $S = \{s_1, \dots, s_{N_s}\}$. Time is discrete and at every time step the system is assumed to be in one of these states. Over time the system goes from one state to the other; the system is said to make *state transitions*. An individual sequence of states over time $1..T$ may be denoted as $\{q_1 \dots q_T\}$ with every $q_t \in S$. Constructing a Markov Model consists of modeling 1. the individual states and 2. the dynamics of state transitions. We can attribute a meaning to these states post-hoc, but this is by no means necessary.

Markov models furthermore obey the Markov property. This property requires that the state the system is in at time t , depends *exclusively* on the state the system was in at the previous moment in time $t - 1$. Otherwise stated: all the information about the past is encoded in the present state. This property

constitutes a wide range of models. Markov models may differ in structural properties of the underlying states (is the state observed directly, or a derived feature vector) and in the dynamics of state transitions (the set of states that can be reached from a state may be bound).

In Hidden-Markov Models [31] what is hidden is the state s the system is in. We make observations O_t , the distribution of which is dependent on the state s_i , expressed as $p(O|s_i)$ with $i \in \{1 \cdots N_s\}$. The observations provide us with inconclusive evidence of the state the system is in. One may consider the observation distributions the coupling between the abstract notion of states and the 'real' feature space. The structural aspect of the model is entirely described by the probability distributions $p(O|s_i)$ for all s_i , and the a-priori probabilities $\pi(s_i)$ of being in a state at time t_1 .

The dynamics of state transitions are encoded in the square *transition matrix* denoted as A with elements a_{ij} and $i, j \in \{1 \cdots N_s\}$. The value of transition probability a_{ij} represents the probability of being in state s_j at time $t+1$, given the fact that the system was in state s_i at previous time point t .

$$a_{ij} \equiv p(q_t = s_j | q_{t-1} = s_i)$$

The probabilities a_{ii} are called recurrent probabilities: the probability of remaining in the same state over a time step. It is required that $\sum_i a_{ij} = 1$, but a model may impose further restrictions. For example a model may require a chronological ordering of the states $s_1 \dots s_N$. Such an ordering prohibits transitions $s_i \rightarrow s_j$ for which $j < i$. This is represented by transition probabilities $a_{ij} = 0$ for $j < i$, i.e. an upper-triangular matrix. Such a model is denoted a Bakis model. A transition matrix with all $a_{ij} > 0$ is called an ergodic model.

The type of observations we make defines the type of the distributions $p(O|s_i)$. In the traditional case we have observations from a finite alphabet with M tokens $\{\theta_1 \dots \theta_M\}$. Now b_{ij} is defined as the probability of observing token θ_j when in state s_i with $j \in \{1 \cdots M\}$. This constitutes a $N \times M$ matrix we call B . In this case $p(O|s_i)$ is of the discrete type with $p(O|s_i) = b_{i..}$.

In many applications we make multivariate continuous observations. We may approximate the observation distribution by a Normal distribution described by mean and variance vectors μ_i and σ_i^2 for every state. In this case the collection of observation parameters B is a set of vectors instead of a matrix.

We define the entire HMM as the tuple $\{A, \pi, B\}$ of transition probabilities, prior probabilities and parameters for the observation distributions. Our specific implementation as well as advantages and disadvantages of this approach are described in detail in Chapter 4.

2.5 Brain-Computer Interface research

Researchers at the Wadsworth center for Neurological disorders in Albany may be considered pioneers of this field. They developed the first one-dimensional BCI in 1991 [16] and their technology has advanced since. Recently they reported on a BCI providing two-dimensional control with high accuracies [2].

This group uses specific frequency components over the hand representations. This paradigm requires extensive training by the subject. An adaptive algorithm tracks the learning of the subject and constantly searches for the most informative components in the EEG. This algorithm continually constructs a model (a least mean squared error fit) based on the most recent trials that best describes previous data.

A group in Tübingen was also one of the first groups worldwide to investigate BCI [3]. Whereas the Albany group focuses on spectral features, this group uses Slow Cortical Potentials as features. The subject has to learn positive and negative influence on the potential on the scalp. The focus of this group is shifting from fundamental research to applied research of BCI [34].

The Fraunhofer institute in Berlin [11][35] reports on three main topics of research. Their emphasis is on methods of machine learning and feature selection. Through these methods they can provide a subject with reasonable control over a cursor in less than an hour. Secondly, this group uses Common Spatial Patterns, similar to independent component analysis, in order to extract more informative features. Finally, they explicitly combine the spectral features used in Albany (Event Related Desynchronization) and temporal features used in Tübingen (Lateralized Readiness Potential).

The Lausanne group [20] also use highly advanced machine learning and feature extraction methods. This group constructed a single time-frequency representation for classification by a support-vector machine. They do not provide continuous control to the user but stepwise movement of a cursor. They provide fundamental analysis on the use of different types of features, as well as practical solutions to eye-blink rejection.

The group in Oxford [33] aims primarily at statistically founded machine learning schemes. They pioneered the application of HMMs to BCI research, thereby acknowledging the existence of valuable time structure in trials. Another important part of this work is the investigation into adaptive learning schemes able to follow the learning user [36].

Chapter 3

Methods

During this project subjects participated in trials with a very basic brain-computer interface. Though the subjects performed the task in well described and controlled conditions the experiments did not leave the exploratory phase. In the upcoming phase we will be able to test larger numbers of subjects in varying conditions.

The experiments we ran served multiple objectives. In the first place we were interested in assessing the technical and procedural difficulties. The former were addressed by the technical staff, the latter together with my colleague junior researchers in psychology. They have gained valuable hands-on experience in training subjects to operate the BCI. The experimental protocol and the equipment used is described in Section 3.1.

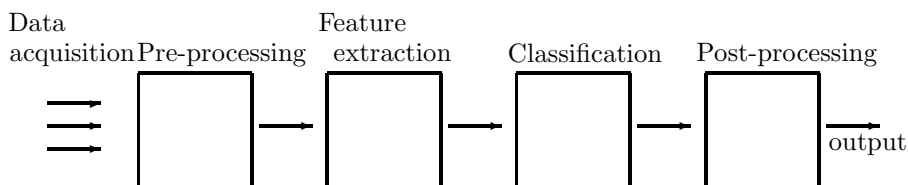


Figure 3.1: A Pattern Classification pipeline. Data acquiry typically represents an analog-to-digital converter or otherwise the link between sensor and computer. In pre-processing the signal is for example normalized through a form of detrending. Feature extraction requires knowledge about the input phenomena. Features represent the essential information from the input objects. Classification is then a rather straightforward decision function of the vector of features copmuted previously. In post-processing we may incorporate the classification result into a larger decision for example by combining multiple classifications into a single classification. In our case we use this to combine the instantaneous decisions during a trial into a single classification of a trial as a whole.

The emphasis in this project was on offline analysis of the data resultant from these experiments as my focus is on machine learning. Figure 3.1 shows the typical serial construction of a machine learning system. In this project I have implemented methods aiding manual feature extraction as described in Section 3.3 and for classification in Section 3.5. For this project I was particularly

interested in the time structure within trials using Hidden-Markov Models as described in Chapter 4. I compare this to the typical instantaneous classification schemes kNN and a linear model as described in Section 3.5.

3.1 Experimental setup

We used the BCI2000 software [37] to record and process the EEG of the subject. BCI2000 is a modular software package in C++ to acquire data from an EEG amplifier, apply a wide range of filters and provide feedback to the subject. The system allows the researcher to use included standard filters such as normalization and spectral analysis, and complement this with self made filters e.g. for classification. This section will describe the technical environment in which we obtained our data.

3.1.1 Apparatus

All subjects participated at the EEG cabinets of the Heymans institute after giving informed consent. The experiments were approved by the Ethical Committee for Psychology (ECP). The cabinets were equipped with Porti amplifiers (TMS International BV, Enschede, The Netherlands). A single amplifier can process up to eight electrodes but multiple amplifiers can be connected to separate USB ports of the computer to increase the number of electrodes. The driver for these amplifiers in BCI2000 was written specifically for this project and altered only slightly during the project.

The ground electrode for the subject was connected to the sternum and all electrodes are referenced to the common mean in the amplifier. The amplifiers have internal clocks to produce time-labeled data at predefined sampling rates. The amplifiers are galvanically separated from the computers by an optical link from the amplifier to a USB converter (TMS International BV, Enschede, The Netherlands). The USB signal is then transported through the wall of the shielded cabinet to the operator computer. There BCI2000 software processes the resultant EEG data. The participant looks at a 17-inch computer screen which is mirrored in the operator room. The operator has a second screen which shows trial-relevant data for visual inspection such as running time and spectral powers.

3.1.2 Design of BCI2000

The system consists of three modules. One module interfaces with the EEG amplifier to obtain the raw EEG signal. The settings for this module determine the initial selection of channels, scaling of the data and sample rate.

The second module performs the signal processing and consists of a series of filters. The researcher determines the elements in the series and the order. A filter is an object which the user can easily implement and extend as desired. The design of filters is governed by rules that are described in [38]. Every filter for example, has a predetermined number of inputs and outputs. The initial normalization filter has a number of inputs equal to the number of channels obtained from the amplifier. A classification filter in a 2D cursor control task

has two outputs, representing the control signals for both dimensions which, in turn, is the number of inputs to the feedback module.

The third module provides feedback to the subject. In our case the medium of feedback is the computer screen. Future applications might just as well provide feedback through a robotic prosthesis. The feedback module communicates the state of the system (cursor coordinates, target position etc.) to the first module for storage.

In our initial recordings we used the feedback module only to present a stimulus indicating which of two cognitive tasks the subject had to perform. Since the stimulus location and raw EEG were stored, this setup provided us with labeled recordings of cognitive tasks for offline analysis. In later sessions we used a very basic linear combination of features to control one- and two-dimensional movement of a cursor through a feedback loop. In the next sections we will elaborate on these tasks and the analysis of the resulting data.

3.2 User tasks

As the EEG signal exhibits very large variability, the change in EEG resultant from the cognitive task must be well discernible from unrelated variability on single trial basis. This is different from common practice in ERP computation where the researcher can average over trials. Therefore not all motor imageries are equally well suited. In our initial pilot experiments the subjects performed different motor imageries. In offline analysis we compared the discriminability of the corresponding EEG signals. The tasks that are well separable can serve as a basis for control in the BCI.

3.2.1 Motor imagery tasks

Our goal is to provide the user with two-dimensional control. This requires two -more or less- independent control signals to be extracted from the EEG signal. Hand/arm imagery is well established in the available literature as a basis for control in a BCI. The hand is densely innervated, the (healthy) subject has independent control over both hands and the representations are well separated on the cortex.

One possibility for an independent second component is using another spatial location[39]. That allows the subject to longer use the motor imagery paradigm. The legs and feet are (in general) also under independent voluntary control and might prove appropriate candidates. The corresponding areas on the primary motor cortex are located inside the central gyrus perpendicular to the scalp (see Figure 2.2).

There is considerably less literature on observed EEG patterns during imagined foot movements [40][18]. Due to the very limited spatial separation of the left and right representations and the limited access during measurement, this paradigm presumably conveys less information about the user's intentions than does the paradigm of imagined hand movements. We will however look at this type of motor imagery as well.

A more recent publication on two-dimensional control [2] uses motor imagery only in the initial training phase and departed from that internal paradigm when the subject gained control. The subjects were then able to independently control

another frequency component at the same spatial location without resorting to imagined movement.

Other considerations are related to the exact motor task the subject imagines. Automated motor programs are executed in the lower layers of the cortex [14] and are therefore presumed to be less visible in the EEG. The tasks the subjects performed were:

- Clench a soft ball with either the right or left hand.
- Operate a screwdriver with either right or left hand.
- Lift the big toe of either the right or left foot.
- Pick up a pen with the toes of either the right or left foot.

These tasks all require a consortium of muscles to cooperate in a way that is not too common. This is expected to require a large group of neurons to cease idling.

3.2.2 Motor imagery sessions

In the first round of exploratory experiments two male subjects participated in the experiment where they were asked to perform motor imageries. EEG was recorded at eight electrodes over the motor cortex: electrodes C1-C6, Cz and Fz (see Figure 2.3 for electrode locations). Activity was computed through common average reference.

We asked the subjects to imagine the sensation of actually performing the act, as opposed to imagining *how it looks* if someone else performs the act. The former is called kinesthetic-motor imagery, as opposed to the latter visual-motor imagery. This is known to impact the quality of the signal [41].

During the first three motor imageries we asked the subjects to perform the act at a pace of roughly twice per second. The order in which subjects performed these motor imageries was varied over subjects. During the experiment the subject was seated facing a computer screen in a comfortable chair. We asked to refrain from blinking during trials and keep hands and feet in a relaxed position without movement.

The session comprised four blocks with the different motor imageries. The imageries were verbally explained to the subject through standardized instructions. In a single trial a red bar appeared on either the left or right side of the computer screen for 4 sec. The location of the red bar corresponded to the desired side of imagery, see Figure 3.2

Between trials the subject had 3 sec. rest periods. A single run lasted 4 min. containing 35 trials. A block of one motor imagery consisted of 5 runs with 1 min. breaks in between. Between blocks the subject received new instructions. In total the experiment took about 2 hours.

3.2.3 Feedback sessions

After an initial assessment of the control participants had over different frequency bands of their EEG, we linked the power in the best-controlled frequency band to the horizontal movement of a cursor on the computer screen. We used

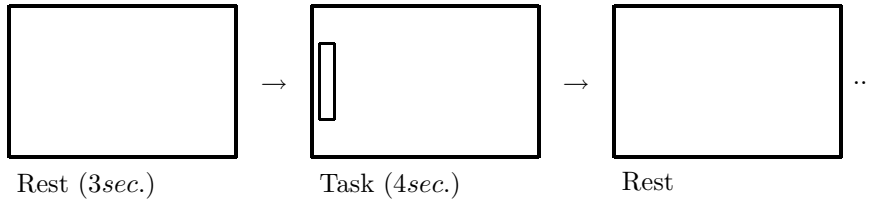


Figure 3.2: **Task layout for motor imagery.** The thin bar on the (in this case) left side of the middle screen indicates the desired side of motor imagery. The bar appears subsequently on either one of the two sides in a random order. The appearance indicates the onset of the task, the disappearance the end of the task, no further information is provided to the subject through the screen.

the difference in power in the 10-12Hz frequency band (mu-band) at the electrode over the hand representation (C3 and C4) as the basis for control. This band exhibited the highest predictive power with respect to the stimulus location in the motor-imagery session. How the predictive power is determined is described in Section 3.3.

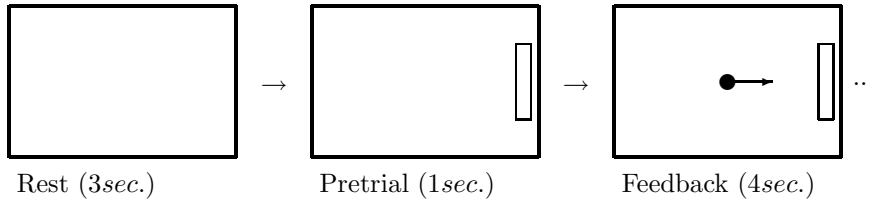


Figure 3.3: **Task layout for feedback tasks.** The thin bar on the (in this case) right side is the target for the cursor to hit. This target appears subsequently on either one of the two sides in a random order. The target appearance is the cue for the subject to start the cognitive task. One second later the cursor appears and immediately starts moving.

The subject was asked to perform the cognitive task that yielded the highest correlation in the previous session, this was the task of clenching a soft ball. Apart from the moving cursor, the experiments were set up identically to the initial measurements, see Figure 3.3. The stimuli were equal to those indicating the desired cognitive task, but were this time explained as being the target location for the moving cursor. One second after the target had appeared, a cursor appeared in the center of the screen and started moving. The horizontal speed was determined by the control signal: the difference in power in the frequency band over the hand representation.

The BCI2000 system automatically sets a bias and gain for the control signal to ensure that both targets are equally well attainable and that the cursor moves on average at a predetermined speed. We set this speed to a value such that the cursor only seldom hits the target. This prevents the trials to be of different length because the trial ends immediately when the target is attained. We explained this to the subject as well to prevent him getting frustrated.

We instructed the subjects to initiate the cognitive task as soon as the target appeared. The spectra are computed based on a 500ms window. That

introduces a delay in the feedback. By initiating the cognitive task prior to the appearance of the cursor, the effect of this delay is diminished.



Figure 3.4: **Target locations in 2D feedback task.** This figure illustrates the four target locations in the 2D feedback task. All targets are positioned along the left and right side of the screen.

As the control of one of the subjects improved, we added a dimension under the subject's control. While the horizontal movement remained based on the difference in power between C3 and C4 in the 10-12Hz band, the movement in the vertical direction is based on the total beta power namely the sum of power in bin 12 and 13 (23-26Hz) of both electrodes. The location of the four targets is shown in Figure 3.4.

3.3 Data analysis for feature selection

For spectral analysis of the data we used an Autoregressive model of order 10 (see Section 2.3.3) over windows of 500ms to find the spectral power in 2Hz-wide bands. The spectra are computed every 64ms. (after 16 new samples were observed with a sample rate of 250Hz). At this point we do not reject trials containing eye blinks or electromyography (EMG) on line. I performed the offline analysis in MatLab (The Mathworks Inc., Natick, MA, USA). The most important code is documented in Appendix A.

3.3.1 Data overview

The data we obtain can be expressed in four main dimensions: space (different electrode locations), frequency, time *within* a trial (typically three or four seconds) and the time axis *over* trials (typically 35 per run). We are able to express the relation between at most two of these four dimensions at a time, either keeping the other dimensions a constant, or averaging over it.

The visualization of spectral data over time is the spectrogram. In this case we keep the spatial dimension constant and we either average over all trials, or choose an exemplary individual trial. This plot is a rectangular grid with a horizontal time axis and a vertical frequency axis. The cells in this grid have a color representing the amount of power present in that frequency band at that given time. We can plot a spectrogram for every channel of interest of the EEG.

In many cases the difference between two spectra is more informative than the two separate spectra. This is also the derived feature we used as the control signal in our early feedback sessions. Therefore we might expect that a difference spectrogram is a good way to visualize the EEG data. We use difference

computations between electrodes on opposite sides of the scalp. Examples of difference spectrograms for a typical subject are given in Figure 3.5.

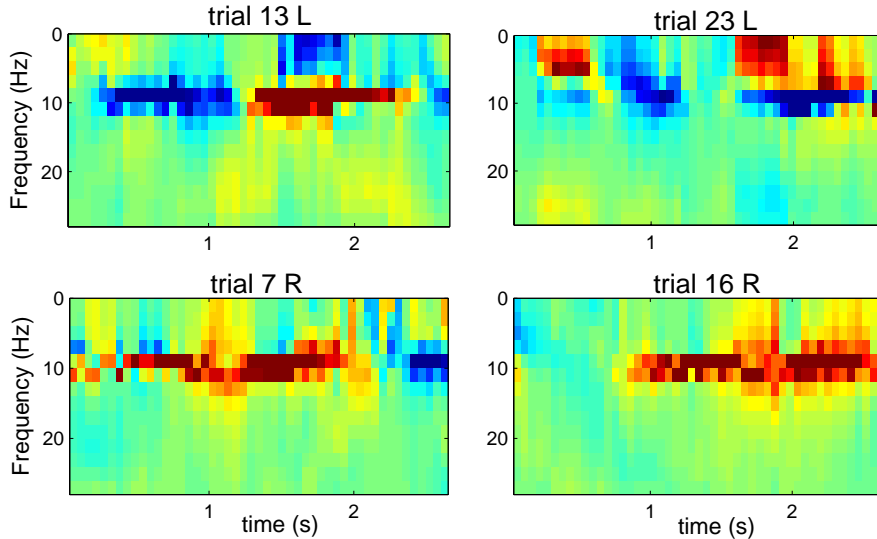


Figure 3.5: Example difference spectrograms. The panels show the difference in spectral power between electrodes C3 and C4 during four typical trials from conditions L (left target) and R (right target). The differences lie between +10 ($C4 > C3$; red cells), and -10 ($C4 < C3$; blue cells). The graphs give a clear indication for time structure in the data. Also note the frequency specificity of the structure. (*Data is Subject JC, session 2.2, run 1, trials 7, 13, 16 and 23.*)

In the analysis environment we can easily extend this so as to display the difference spectrograms for all the trials of a subject at once. These graphs give a good impression of the variability and consistency over trials.

To get an impression of phenomena that are common to all (or a subset of) trials we can compute the average of these difference spectrograms. A meaningful subset would for example be that of all trials with a common side of motor imagery. We can then use these averages to look for structural differences between these sides. An example of two such average difference spectrograms is given in Figure 3.6.

These graphs give us detailed information about the relation between spectral power and the time course in a trial. However this method does not provide information over different spatial locations. We are also interested in measures that provide information on different spatial locations while averaging over time. Such methods are discussed in the following sections.

3.3.2 Correlation

One measure of the amount of control a subject has over an EEG component, is the correlation r of the power in a frequency band to the handedness (left or right) of motor imagery (as used for example in [30]). We use dummy rep-

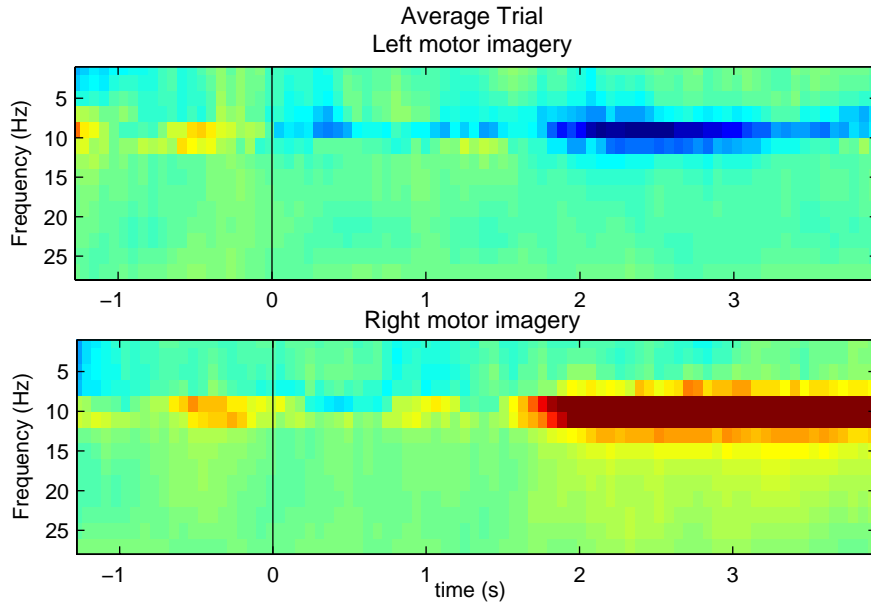


Figure 3.6: Example average difference spectrogram. The graph shows the average difference in power between electrodes C3 and C4. Again red cells indicate $C4 > C3$ and blue cells vice versa. The upper graph is the average over all left trials, whereas the lower over the right trials. The trial starts at $t = 0$, corresponding to the vertical black line, negative time corresponds to the resting prior to the trial for contrast. (Data is Subject JC, session 1: Ball-clench imagery runs 1 to 5. The average is over the two disjoint subsets corresponding to target code (both $n = 82$).)

representations left = 1 and right = 2 to compute correlations. When relating continuous variables, i.e. spectral power, to the dichotomous target code we must use point-biserial correlation as described in Section 2.4.

We use r as opposed to r^2 to retain the sign of the relation. As the labels of the cognitive tasks are arbitrary, so is the sign of the correlations. However we expect electrodes on opposite hemispheres to correlate with opposite signs [15]. The absolute difference in correlation between two electrodes is lost when using r^2 .

When using r as the measure of control we assume that there is no direct effect of the stimulus on the measured EEG, i.e. what we measure is assumed not to be stimulus related. It is conceivable that the higher visual cortices radiate activity up to the locations over the motor cortex. For now we leave this question open as this is a common experimental setup.

We compute the correlation of each frequency band to the target location for an entire session of a specific motor imagery or feedback task. These data points used for correlation are not independent. Due to the 500ms window spectral powers in neighboring time points have 112 out of 128 measurements in common. For the correlation estimate we remove the spectra computed between trials and spectra containing extreme values in any band. In practice the extreme values

are found only in the first 8 spectra, when there is not enough data to compute the spectrum.

This procedure yields correlation estimates for every frequency band at every spatial location. We can visualize this with a graph for several spatial locations spanning the x axis on which the frequency bands are set out. Figure 3.7 shows an example of this type of graph for the second feedback session of a typical subject, for four spatial locations.

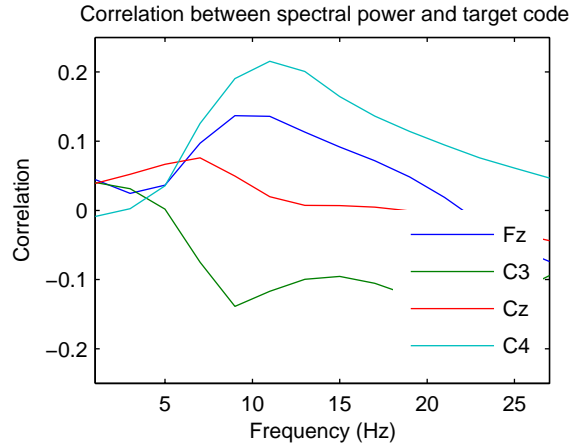


Figure 3.7: Example graph of the correlation between power in frequency bands and the side (left or right) of motor imagery. The different lines correspond to distinct spatial locations. In this example best performance is found around 10Hz for spatial location C4 with $r \approx 0.2$. Note that C3 and C4 on opposite sides of the scalp correlate inversely. For details the reader is referred to the text. (Data is subject *JC*, session 2.2, runs 1 to 5. Correlation is computed over $n = 6760$ (142 trials \times 47.6 points per trial) points)

In this specific graph the highest r is found around 10Hz. As expected we also see opposing signs for r at different spatial locations. This strokes with our observations in Figure 3.6 where the most prominent component in the average difference spectrogram for the left- and right motor imagery lies around 10Hz. Also there the difference in power has opposite signs for both left and right.

3.3.3 Mutual Information

An additional measure of the predictive power of a frequency band for the underlying intention is the Mutual Information between the distribution of powers (in a specific band) and the cognitive tasks [42][43]. This measure is similar to correlation in the sense that qualitatively, it expresses the same relation. However when using Mutual Information we are able to express non-linear relations whereas correlation is linear.

Mutual information is based on the amount of disorder that can be explained by the two conditions (sides of motor imagery). The disorder of a system can be expressed by the entropy H , so the mutual information is the difference between the overall entropy and the sum of entropies of the partial systems of one side

of motor imagery. The entropy for a continuous random variable X is defined as:

$$H(X) = - \int_X p(x) \log p(x) dx$$

In the practical case we make a histogram of the data to approximate the distribution with the sum over x the bins in the histogram, and $p(x)$ the relative frequency of that value:

$$H(X) = - \sum_x p(x) \log p(x)$$

In these equations X is the distribution of powers, and x the values we may observe. We use the marginal distribution of x with respect to y , $p(x|Y = y)$ to estimate the mutual information between x and y . The marginal entropy of x , expressed as $H(X|Y = y)$ is computed by substituting the marginal distribution $p(x|Y = y)$ for $p(x)$ in the formula for $H(X)$. The marginal entropy is interpreted as the disorder in variable X for a given y . The mutual information between X and Y is the difference in disorder in the variable X itself, and the constituent disorders for all the possible values for Y . Thus the mutual information $I(X, Y)$ between continuous X and nominal Y with levels y_i is computed as:

$$I(X, Y) = H(X) - \sum_{y_i} H(X|Y = y_i).$$

The mutual information is expressed in bits. We express the mutual information in a graph similar to that of Figure 3.7 where the running variable is the frequency but the y-axis is now in bits. Different lines display separate electrode locations and we again average over both time dimensions. A typical graph of this type is displayed in Figure 3.8.

3.3.4 Correlation over time

A more specific visualization of the data would be to use the time axis in the trial as the running variable and keep the frequency band a constant (i.e. only look at frequency bands of interest). We again use the correlation r as our measure of control, but this setting increases the required amount of data. The data requirement increases by the number of steps we wish to discern in the running variable. This could be overcome by letting the subject perform a great number of experiments but that raises another issue.

For all the averaging procedures we implicitly assume stationarity in the dimensions over which we average. As can be seen from Figure 3.6 the assumption of stationarity is violated when averaging over time within a trial. But what can we say about the stationarity of our signals over trials?

While the computer is learning where the information lies in the EEG, the subject tries to learn based on the feedback of the computer screen and the experience during motor imagery. We may assume that the subject intentionally changes structural parts of his EEG over a longer time, i.e. learning. Therefore for time intervals long enough (i.e. the time constant of learning) stationarity is lost.

That rephrases our question to what interval lengths allow us on the one hand to compute *reliable* correlations, and on the other hand provide satisfactory stationarity for *accurate* correlations. The Figures in Appendix B show

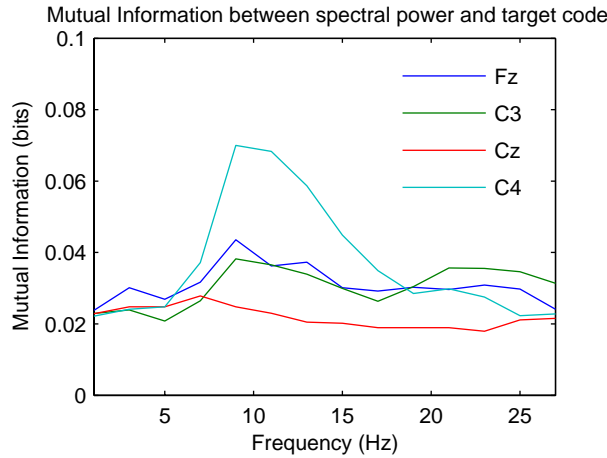


Figure 3.8: The mutual information between the observed spectral powers in different frequency bands, and the side of motor imagery. One bit of information would imply perfect information about the intended side of motor imagery. In this figure C4 bears the most information. (*Data is Subject JC, Session 2.2, runs 1 to 5. $n = 6760$ (142 trials \times 47.6 points per trial) for every frequency band, 250 bins in histogram to estimate the pdf.*)

average trials for training and test set separately. These sets are chronologically separated and thus give an impression of the non-stationarity over trials. At present we use one session of five runs for this analysis. This corresponds to half an hour in the experiments.

We can now compute the progress over time of the control a subject has, measured in the correlation of the power in the 11-13 Hz band to the side of motor imagery. Figure 3.9 shows the evolution of the correlation for the two spatial locations over the hand representations.

This graph indicates that averaging over an entire trial for correlations may obscure valuable structure. If the absolute correlation increases over time, the averaging procedure yields an underestimation. This indicates that accounting for the structure in the trial may prove beneficial. We will further introduce this method in Chapter 4. The remainder of this Chapter is concerned with describing the experimental procedures in assessing performance of the instantaneous classifiers.

3.4 Data sets

In our experiments we will make use of five main data sets compiled from the store of data. The following provides an overview of these data sets. More structural information on the data sets is provided in Appendix B.

The data sets should reflect the typical data encountered in BCI research. We supplemented our own data with one data set from the BCI Competition [26] containing data obtained in conditions similar to ours. The other four data sets originate from two subjects that were trained extensively in the early phases of the project.

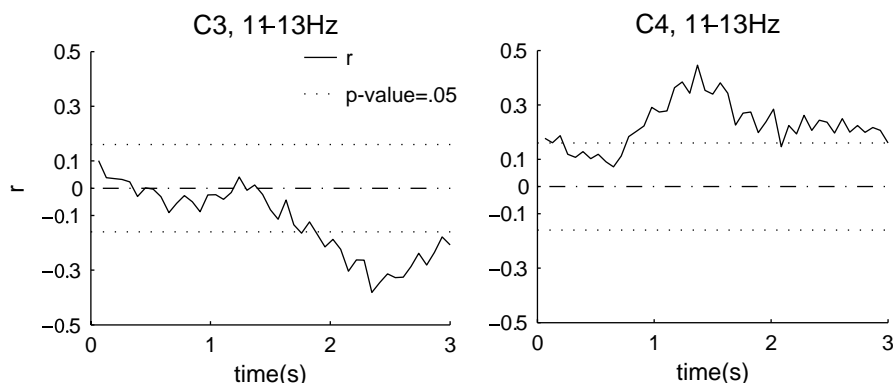


Figure 3.9: This graph shows the evolution of the correlation with the side of motor imagery over time *within a trial* for the two electrodes over the hand representations. Note the variability over time and the qualitative difference between the graphs. The dotted line denotes the level of significance at $\alpha = .05$ for these correlations. (Data is Subject JC, Session 2.2, runs 1 to 5. $n = 142$ trials for every point.)

Data set 1 was obtained in the first feedback session of subject JC. In a prior session he had performed motor imagery to assess the frequency bands under voluntary control. The subject participated in three sessions of half an hour, each consisting of five runs. We used the session exhibiting the highest average absolute correlation between power and target code over frequency bands. From that session we used the four best runs and divided them into an equally sized training- and test set. Data set 1a contains 14 features for the 2Hz-wide frequency bands between 1 and 28Hz representing the difference in power between electrode C3 and C4. Data set 1b contains 28 features with the raw powers of the two electrodes concatenated.

Data set 2 from subject RH. The subject also has feedback in the form of a moving cursor but the subject can control the horizontal as well as the vertical movement as described in the last part of Section 3.2. The data set is of similar size as Data set 1, but as the number of targets is twice the number in Data set 1, we have only half the number of trials *per condition*. For this data set we also computed features from other spatial locations, this is described in detail in the Appendix.

Data set 3 is data set III from the BCI Competition. It consists of trials from three different subjects: O3VR, X11 and S4. We performed the exact same pre-processing steps as in our on line system by feeding the data to an offline instance of BCI2000. These data sets are considerably larger than our own sets containing 1400 trials in total from two (left and right) conditions. However only half the trials were labeled for competition purposes, we used only these labeled trials. From these data sets we computed features representing the difference in power as this had proven in the previous data sets to yield good performance.

Data set 4 and 5 were collected in conditions similar to that in sets 1 and 2

from subjects JC and RH respectively. Data set 4 is relatively large containing 200 trials from both conditions in relatively stationary conditions. These data sets are preceded by a large amount of training by both subjects, therefore we may expect to find more information in the data. In these data sets the frontal electrode F4 provided far more information than C4 concerning the target code. Therefore we used the difference between C3 and F4 in power as features for classification.

3.5 Experiments with instantaneous classifiers

The Linear Model and kNN are referred to as instantaneous classification, meaning that a single feature vector is considered and classified into a class, disregarding earlier observations in a trial. In our setup the EEG is sampled at 250Hz. After every 16 samples (64ms) a spectrum is computed based on the previous 500ms. A spectrum is passed to the classifier as a vector with 14 features per channel, namely the spectral power in 2Hz wide bands from 1 to 28 Hz. Note that two subsequent spectra are based on data that is for roughly 85% the same. We train classifiers on spectral powers and on difference in spectral power between two channels.

A trial of four seconds consists of 63 feature vectors. The average accuracy of a classifier on individual feature vectors however is an underestimate of the expected accuracy *per trial* which we use to compare with time models. Therefore we need to combine the instantaneous classifications into a single classification for the trial as a whole.

Averaging a classification over all 63 separate classifications may drastically improve performance. For example, if the accuracy on individual feature vectors is 0.6, the probability that half or more of the spectra from a trial are classified correctly is larger than 0.6. This probability of classifying more than half of the spectra correctly is the probability of classifying the trial as a whole correctly. If feature vectors were independent, we could estimate the increased accuracy with $X \sim \text{Bin}(63, 0.6)$. The probability of classifying a trial correctly is now the probability of observing > 32 successes. This is larger than 0.6 for such a distribution.

To combine the 63 classifications into a classification of the trial we use the mode of the set of individual classifications. We use the mode rather than the rounded mean because the latter is not defined for non-metric values such as 'left' and 'right', and introducing a dummy representation would favor (in the case of more than two classes) the central values.

We will not perform cross validation in our tests on real data. For some data sets we see rather large non-stationarities between early and late trials (See Appendix B). These effects are presumed not to be random variations, but resultant from all sorts of influences such as learning, fatigue etc. By performing cross validation we would equalize the training- and test set. This would result in overestimates of actual performance. For results on artificial data (See Chapter 4) and other more specific results, we *do* use cross validation.

These non-stationarities are characteristic for this type of data, and by not removing them, we hope to obtain a more realistic estimate of the true perfor-

mance when a model is first trained with a subject, and then tested. We will check whether there are large differences when performing cross validation.

3.5.1 k Nearest Neighbors

We used kNN with different numbers of neighbors k . When making test runs we used the values 4, 9, 16, 25 and 36. These numbers are the squares of the numbers of states used in the HMM. kNN is a deterministic learning scheme just as the linear model, so we needed to perform a learning run only once per level of k per data set. This is different from the HMMs described in the next Chapter which rely heavily on initial conditions.

We implemented the classifier as an object in MatLab with a training and testing method. This way it can be easily integrated with other learning methods. The Linear model is implemented similarly. A run of the learning method can be made by invoking a wrapper method performing the bookkeeping such as consecutively calling training- and testing methods and saving the results to a standardized results file format. For the linear model, the wrapper method also augments the feature vectors to introduce a bias in the model. This is essential for the proper operation of the thresholding in that case. This wrapper method has a similar interface for all learning methods.

3.5.2 Linear model

When performing regression we obtain real values rather than class labels (See Section 2.4). This is useful when we want to extract a control signal, but poses problems when assessing the performance and/or compare it to other classifiers. However, we can transform the estimate $\hat{y} \in \mathbb{R}$ to a classification. In the case of two possible outcomes (e.g. left/right) we may code left and right as labels $[-1, +1]$. We can now classify a new instance \vec{x} as $sgn[f(\vec{x})]$. This creates a dichotomizer from the linear model.

In this case, correct/incorrect is an appropriate measure of performance because the labels accompanying the data only express a direction (up, down, left and right) and not a magnitude. A subject can manipulate the magnitude of ERD in various ways [19] but at this point we did not instruct the participants to do so. Therefore we have only labels indicating the direction and not magnitude of the cognitive tasks. The information we may extract from this type of analogous control in bits is larger, if the subject is able to exercise this type of control over features of its EEG. This is an open question. At this point classification is appropriate for the data we collected.

We fit a model from continuous predictor variables to binary response values. Binary responses normally require fitting a Generalized Linear Model (GLM) [44]. GLM introduces a link function between a linear predictor $\eta = \langle x, \beta \rangle$ and the outcome. The logit link function $\log[\eta/(1-\eta)]$ transforms the domain of binary outcomes $[0, 1]$ to $[-\infty, \infty]$ to ensure the model is well defined. However in our case we are only interested in the sign of the outcome.

Since the logit function is a monotonic function, imposing a threshold on $\log[\eta/(1-\eta)]$ for classification, is equivalent to imposing a threshold on η . Furthermore, if we use the difference in power over two channels we will argue for normality. Therefore I fit a simple linear model (see Section 2.4.2) and threshold

the linear predictor. Fitting on labels $[-1, +1]$ places the threshold at zero for equal priors. For $\eta = 0$, classification is always incorrect: the classifier does not guess.

In the case of more than two directions we can combine multiple dichotomizing linear models of the type described above. For example, one linear model may discern left from right and the other up from down. Together these models can make a four-way decision. In practice the number n of directions is always a power of 2, therefore $\log_2(n)$ linear models suffice. In this thesis the maximum number of directions is four.

Chapter 4

Models for trial-based operation of a BCI

The previous Chapter described several methods describing the data. Direct visualization of the features vectors (e.g. Figures 3.5 and 3.6) indicated that a time structure exists within a trial. The average over a number of consecutive trials, Figure 3.6, suggests that the time structure is at least partly constant over trials.

In Chapter 2 we also described literature on the EEG process under consideration. Neuper et al. describe a constant time structure of ERD in [19]. The typical time span of this phenomenon is of the order of 4 sec. This work also investigated the difference in the phenomenon for separate frequency bands. They found a non-trivial relationship between observed power in the upper- and lower mu band respectively.

When computing the correlation between spectral power in the mu band over time within a trial (Figure 3.9) we also find a structure over time. Also in this case the structure does not appear trivial, as for that example the time print of C3 and C4 do not overlap.

The aforementioned suggest there to be a relatively constant time structure *on average*. For a trial based approach it is however necessary to be able to recognize the structure in single trials. Figure 3.5 illustrated the variability in structure between trials which seems relatively large but not random. We see certain components of a spectrogram recurring over several trials, be it at different points in the trial.

The most natural components of this sort are the traces of red and blue in the 10Hz vicinity. Such components generally have a duration longer than 500ms thereby illustrating the fact that they are not merely resultant from smearing due to the windowing for spectral estimates. More complex components constitute frequency modulation, present in the spectrogram as diagonal traces e.g. in trial 23 of Figure 3.5.

Thus we have seen there is a rich time structure in these trials. It is however unclear what to gain from this time structure. In this chapter we will introduce several more advanced methods of machine learning that are able to incorporate

the time structure in a model of the underlying process. By comparing these models with models that do *not* account for time structure we determine whether these time models are a promising new lead for BCI research.

We gave an intuitive argument for using time structure, based on a qualitative analysis of early results. Another aim is to further investigate the type of time structure of this process. To achieve this latter goal we also construct alternative models exhibiting different performance for different types of time structure in artificial data sets. We use the performance on artificial data sets with a known time structure as a measuring stick for the performance on real data to describe the time structure underlying trials with our BCI.

The next Section will describe different types of time structure. In Section 4.2 we describe our implementation of a continuous-observation Hidden-Markov Model and argue for the design choices in that implementation. Section 4.3 describes issues encountered in modeling EEG trials, while Section 4.4 describes our alternative models: the flat HMM and the Common-Structure HMM. Sections 4.5 and 4.6 describe the procedures for offline analysis of real- and artificial data respectively.

4.1 Types of time structure

In speech recognition time information is essential to discern between words that are built up from the same phonemes in different orderings such as for example *law* and *all*. In other cases there may exist a time structure, but it does not necessarily bear information of the specific condition of that process. Figure 4.1 shows three fictive processes that exhibit a different but clear time structure as well as discrete states.

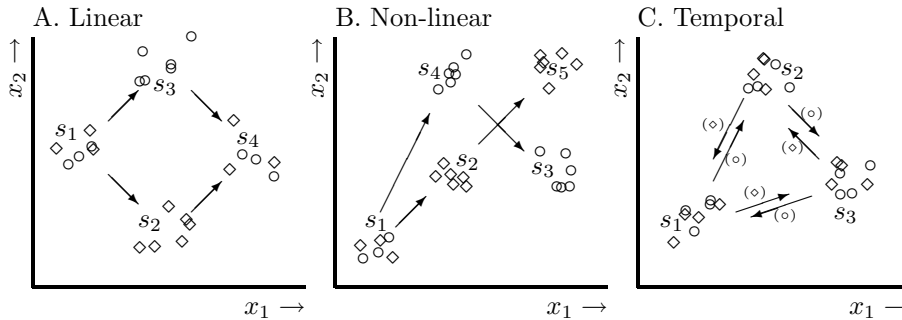


Figure 4.1: This figure illustrates different types of time structure. The graphs show observations from three fictive processes with different types of time structure. There are two conditions of the process denoted \circ and \diamond . Of each condition of each process we observe one time series of two features x_1 and x_2 . The process advances through a series of underlying states over time which results in the clusters of points. The time relation between individual points is omitted, the graph only shows the transitions between clusters over time through the arrows. The reader is referred to the text for details.

These graphs are a conceptualization of the type of data we observe. These graphs are a simplification as well as an idealization: they display perfectly discrete state transitions and well separated clusters. Furthermore the process can be well described by only two features. This figure merely serves to illustrate the different types of time structure a process may exhibit rather than to provide an accurate abstraction of the processes underlying *our* data.

The discrete states in Figure 4.1 are represented by clusters of points. In some cases clusters of points from two conditions overlap resulting in heterogeneous clusters (e.g. those in panel C.), whereas other are homogeneous (e.g. s_2 and s_3 in panel A.). Heterogeneous clusters may correspond to states of the underlying process common to both conditions; for example an idling state or silence in speech recognition. In an EEG trial, an idling state might for example be characterized by an alpha peak. If we encounter a feature vector from such a cluster without any other (time) information, it is difficult to make predictions of the condition of the process. That does not hold for a feature vector from a homogeneous cluster. Such a cluster has a high discriminative power with respect to the condition of the underlying process.

In panel A, for example, one discriminates between the two conditions mainly based on the 'high' or 'low' values of x_2 when x_1 assumes intermediate values (clusters s_2 and s_3). The clusters s_1 and s_4 in panel A. presumably represent idling states which bear no information of the underlying state. It is questionable whether in this example a model accounting for time outperforms an instantaneous method employing a horizontal separating hyperplane halfway x_2 . We denote this type of structure as *linearly separable* because theoretically the data can be separated by a horizontal hyperplane.

In the case of example panel B. of Figure 4.1 the structure is more complex. There is no plane separating the homogeneous clusters. However also in this case we may discern between conditions with an instance based classifier such as a Support-Vector Machine using radial-basis functions (RBF) [45] or with a mixture model of gaussians [46]. We will employ a variation of the latter classifier in our analysis. For that reason, this type of structure is denoted *non-linearly separable*.

In panel C. of Figure 4.1 an instance based classifier is without a chance. If the dynamics of the transitions between states differs between conditions a time model may still perform reasonable in this case. In this example the three states are visited clockwise in condition \circ , whereas in condition \diamond the states are visited counter clockwise, indicated by the symbols in parentheses near the arrows between clusters. That means that at a point in time a feature vector predicts \circ , whereas at another point in time the exact same features predict \diamond . In that case though the observation probabilities for both conditions are equal, in the final comparison of $p(\mathbf{O}|\circ)$ and $p(\mathbf{O}|\diamond)$ the transition probabilities a_{ij} will determine the difference. This type of structure is denoted *temporally separable*.

We will use this notion of different types of time structure in later sections to construct artificial data with these types of time structure. By comparing the performance of several machine-learning techniques on the artificial data and on real data we make claims about the type of time structure underlying that real data in terms of the three types of time structure we introduced above.

4.2 Hidden-Markov Model for classifying EEG

In the following we use the following symbols for observations. We use \mathbf{O} for an observed trial: a $d \times T$ matrix with d the dimensionality, and T the length of a trial. For an observed d -dimensional feature vector we use O , or specifically at time t we use O_t . For one specific observed feature we use o_u with $1 \leq u \leq d$. Thus $\mathbf{O} = \{O_1 \cdots O_T\}$, and $O = \{o_1 \cdots o_d\}^\top$. Index l is used for trials as a whole, index t for feature vectors within a trial, index u for features within a feature vector.

The Markov models consist of state transitions A , prior state probabilities π and parameters B for the observation distributions $p(O|s_i)$ for states s_i (See also Section 2.4.3). We modeled the observations $p(O|s_i)$ as a continuous distribution. An observation O is either a spectrum of one or more channels (i.e. electrode locations), or the difference in spectral powers between two channels.

A multivariate observation O consists of a vector of uni-variate (differences of) powers o_u for the separate 2Hz-wide frequency bands u . For every uni-variate observation we estimate the parameter(s) of the corresponding marginal distribution $p_u(o_u|s_i)$ in each state during the training phase. The probability of observing a feature vector is evaluated as $p(O|s_i) = \prod_{u=1}^d p_u(o_u|s_i)$. This implies we assume the features (i.e. frequencies) to be uncorrelated which saves a large amount of parameters to be estimated, namely the off-diagonal elements in the covariance matrix. This limits expressive power of the model.

We empirically determine the type of distribution for these observations. Spectral powers can only assume positive values, whereas difference in power may also assume negative values. We made quantile-quantile plots (QQ plot) of the two types of observations with a normal and a chi-square distribution. These plots are shown in Figure 4.2, with the rows the two types of observations, and the columns the types of distributions.

The normal distribution is clearly a bad fit for spectral powers. This is mainly due to the fact that the normal distribution expects values below zero and roughly spans the range $[-10, 20]$. The empirical distribution however spans $[0, 40]$. The two estimates for ν , the degrees of freedom ($\nu = \bar{x}$ and $\nu = s^2/2$) do not differ greatly. For spectral powers we will use $p_u(o_u|s_i) \sim \chi^2(\nu)$ with $\nu = \bar{x}_u$. Note that these figures are for a single frequency band of a data set and are influenced by the random initialization for the theoretical distribution. This figure merely serves as a representative example.

For difference in power over two electrode locations most prominent are the bends in the lower left parts of E. and F. This is due to the fact that the chi-square distributions are unable to account for the negative values. The theoretical normal distribution shows slightly less tapered endings but overall exhibits a good fit to the data. Therefore for differences in spectral power we will use $p_u(o_u|s_i) \sim \mathcal{N}(\mu, \sigma)$ with $\mu = \bar{x}_u$ the sample mean and $\sigma = \sqrt{s_{x_u}^2}$ the sample standard error. (s_i represents the state in the model, $s_{x_u}^2$ represents variance in the sample x_u .)

Overall the observation parameters to be estimated are $d \times N_s$ for the chi-square distribution and $2 \times d \times N_s$ for the normal distribution, with d the dimensionality and N_s the number of states. Though the normal distribution requires twice as much parameters, using the *difference* in power between chan-

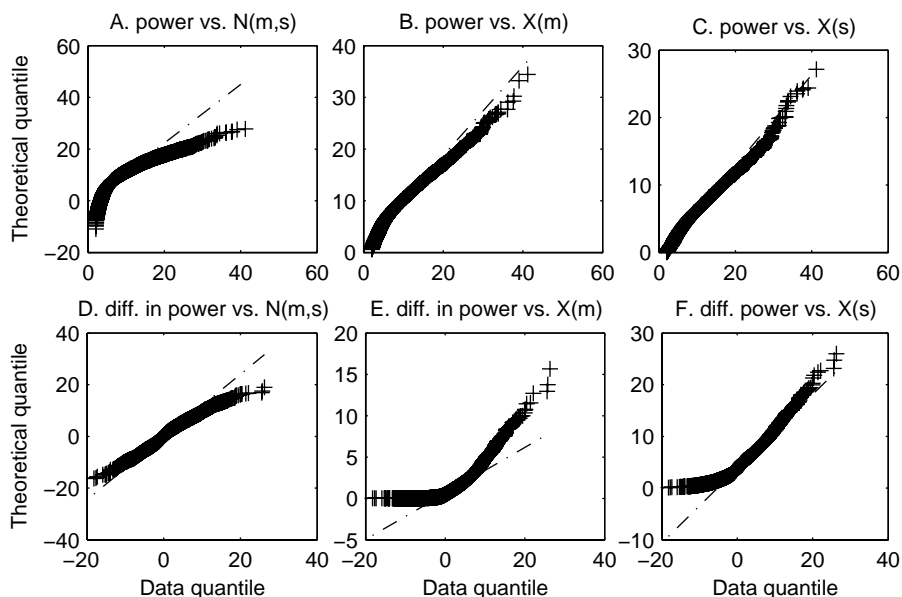


Figure 4.2: This Figure shows exemplary QQ-plots of spectral powers (Fig. A-C) and differences in spectral power (Fig. D-F) versus theoretical Normal and chi-square distributions. The parameters of the theoretical distributions are estimated from the data. We estimated the number of degrees of freedom for the chi-square distribution as the mean (Fig. B, E) and as half the variance (Fig. C, F) of the data. The dashed line represents points where empirical and theoretical quantiles correspond. Ideally, all pluses lie on this line. *Data is Subject JC, Session 2.2, Runs 1 to 5. For A.-C. Channel 3 at 10Hz. For D.-F. Channel 3 minus 7 both at 10Hz. Graphs are subject to small variations due to random initialization of the theoretical distribution.*

nels in turn reduces the dimensionality by half.

In our analysis we train the model for different numbers of hidden states. We train the model by means of the so called 'forward-backward procedure' [31]. This method starts with an initial random setting for B , A and π . In our case, we pick one sample as initial values for each cluster center, and the variance in a large subset of the data as initial estimate of variance. This is generally an overestimate of variance which prevents from zero probabilities and ensures smooth convergence of the states.

The next step is to estimate all the probabilities $\gamma_t^l(i)$ of the system being in state s_i at observation t of trial \mathbf{O}_l in the training data, given those initial parameter settings. Based on γ , points are assigned to the different states. Now, the parameters A , B and π of the states are re-estimated based on the points assigned to that state. The transition probabilities are updated similarly. The system iterates these steps until the combined probability $\prod_l p(\mathbf{O}_l|\lambda)$ of observing all the trials given the HMM converges.

We can impose restrictions on the state transitions by setting elements a_{ij} to zero as discussed in Section 2.4.3. We do not impose such restrictions because

we have no knowledge of the underlying process yet. Secondly, it is possible to inspect the transition matrices afterward to see if a similar structure emerges by itself. If such structure exists we can then improve performance by imposing it beforehand. The same goes for prior probabilities π_i . Another way to incorporate prior knowledge is by adjusting the training procedure.

4.3 Practical issues in modeling

The examples introduced in Section 4.1 disregard practical issues such as faulty or sparse data and noisy or uninformative features. This section will address advantages and disadvantages of time models compared to instantaneous classifiers, specific to our problem of classifying EEG trials.

When constructing a model of a process the training data is divided into subsets based on the conditions to be discerned. Because the store of training data is finite (indeed rather limited due to our preference for short training times), there will exist uncertainty about exact centers of the clusters in the data. Therefore the estimates of parameters B for $p(O|s_i, \lambda)$ will be subject to noise.

If a state s_i in theory should constitute a perfectly heterogeneous cluster (otherwise stated: bears no information of the condition of the process without time information), with infinite data the two distributions $p(O|s_i, \lambda_1)$ and $p(O|s_i, \lambda_2)$ of the two models should become infinitely similar as well. There are two reasons why in practice these distributions will differ. See Figure 4.3 for an impression of how the cluster centers might be estimated by the training algorithm.

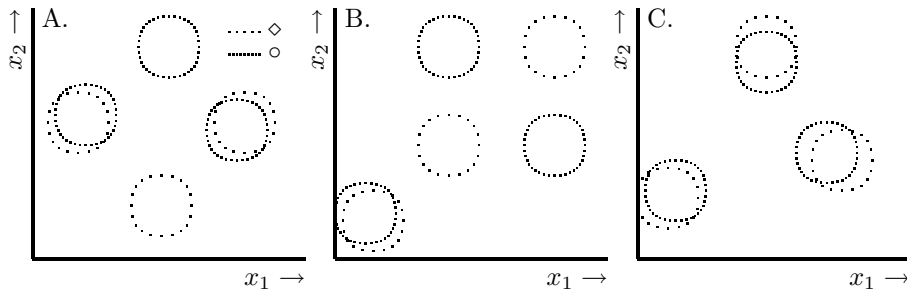


Figure 4.3: An example distribution of the cluster centers for two models (one for condition \circ and \diamond) with the appropriate number of states as might be estimated based on the data from Figure 4.1. The circles are centered at the mean, the radius indicates the standard deviation. The heterogeneous clusters should overlap perfectly in this example in theory. The slight differences in this case are man made and meant to illustrate the issues in practice.

The first reason is the imperfect data. This results in differences between the cluster centers found in disjoint subsets of the data. Though on average this does not favor one model over the other, it does introduce noise to the classification because for each observation $p(O|s_{idle}, \lambda_1) \neq p(O|s_{idle}, \lambda_2)$. This problem aggravates if through a long-term non-stationarity (e.g. learning) one of the models λ_i becomes structurally favored. The problem is also related to

the dimensionality of feature space. In high-dimensional feature space there are generally more features that bear little or no information. This is known as the curse of dimensionality.

A second reason lies in the windowing procedure. Due to windowing, our cluster centers are not as well separated as suggested by Figure 4.1. Figure 3.5 illustrates the smooth transitions in the feature vectors. Therefore the exact form of a distribution is also determined by the vicinity of the other distributions. If for example s_2 in part A. were closer to s_1 than s_3 , the variance in $p(O|s_1, \diamond)$ would be smaller than in $p(O|s_1, \circ)$.

In Section 4.4 we will address the issue of classification noise by ensuring that both models have the same structure encoded in the observation distributions but transition probabilities specific of the conditions. Another possibility is to merge clusters of which the constituent data points are not significantly different.

Generally a model accounting for time structure will require more parameters to be estimated, and therefore require more data to be trained or estimated. A model with a large number of parameters also has a higher risk of being over trained resulting in a lack of generalization.

We will assess the effect on performance of reducing the size of the training set for both time models and instantaneous classifiers. One might expect that due to the larger number of parameters to be estimated, time models will sooner be affected by a limitation of the number of training samples.

A third drawback of time models is their high dependency on the preprocessing steps and in particular feature selection. An instantaneous classifier aims at employing those features that exhibit the highest discriminative power. Least mean squares regression for example uses the correlation of a feature with the label as a measure of how informative a feature is. Support-vector machine's are also famous for their ability to enhance contrast between classes.

Generative classifiers generally do not employ such a strategy by itself. This is due to the fact that the aim of a model is not to discriminate but to describe. Discrimination is done by choosing the best description. In describing a process there is no measure of what is, and what isn't discriminative. It may also be deduced from the fact that during the training phase of the model, there is no data used from conditions other than the one being modeled.

There are methods to optimize discriminability of a series of models. In these cases the maximum likelihood scheme of estimating parameters is replaced by one that uses the cross entropy between models [47]. Applying these methods may be a next step in constructing the models. At this time it is beyond the scope of this project.

We perform feature selection by hand based on the methods described in Section 3.3 and on literature suggesting not only informative, but also meaningful features. By obeying the latter requirement we obtain a meaningful abstraction of the underlying process.

As the outcome of the model bears a well defined meaning (the probability of making the current observation), a model may provide intuitive means to reject 'bad' trials. A lower threshold on the outcome $p(\mathbf{O}|\lambda)$ coincides with our notion of the improbability of observing data under the assumption of the model. It can also help to 'switch off' the classification in phases of the trial where we know the data contain little information such as in the 'idling' states.

Finally, the high dimensionality of the data poses problems. For observations from a finite alphabet the dynamic range of observation probabilities and the a_{ij} are of the same order of magnitude. However since in our case the observation probability is the product of at least 14 probabilities from exponentials this dynamic range increases. The dynamic range of a_{ij} does not as these are still relative frequencies of state transitions. This problem is described in [48]. As the dynamic range of $p(O|s_i)$ exceeds that of a_{ij} , the effect of transition probabilities on the final $p(\mathbf{O}|\lambda)$ diminishes.

4.4 Variations on the re-estimation procedure

In the previous sections we described the implementation of a Hidden-Markov Model for our specific task. This section will introduce two adjustments in the re-estimation procedure to assess the type of time structure that exists in EEG trials. I construct three types of models each with a time structure to which they are particularly suited. The performance of a particular model on the EEG data then serves as evidence for whether the time structure to which that model is suited, underlies the observed EEG data.

4.4.1 Flat HMM

We analyze performance of a HMM *without* transition probabilities (i.e. $a_{ij} = 1/N_s \forall i, j$ also during training, with N_s the number of states). In this setting our estimation procedure is similar to the estimation of a mixture of Gaussians through expectation maximization [46]. The probability of an observation sequence \mathbf{O} originating from the model at hand λ is evaluated as $p(\mathbf{O}|\lambda) = \prod_{t=1}^T \sum_{i=1}^{N_s} p(O_t|s_i, \lambda)$.

Should the actual time structure be similar to that in panel C of Figure 4.1 we expect this model to perform worse than the original HMM. If performance does not degrade through this simplification we interpret that as a lack of information in the time structure itself.

We denote this model the Flat HMM to relate it to the other HMM approaches. This allows a comparison based on the state-space representation where both methods have a number of states. Fitting a true mixture model to the data would prohibit a comparison of the two approaches for different numbers of states in the HMM and gaussians in the mixture model respectively.

4.4.2 Common-structure HMM

The other alternative we will employ in our analysis in a sense is inverse to the Flat HMM. Where the Flat HMM neglects the a_{ij} entirely, this approach aims at exploiting as much of the time structure as possible. The method also reduces classification noise resultant from variability in the estimation of states for the two models.

In this scenario one model first is based on *all* the data, as opposed to the traditional case where only the subset of training data from a specific condition is used. A schematic overview of the training of a CSHMM is depicted in Figure 4.4. Since we use data from both conditions it might be expected that

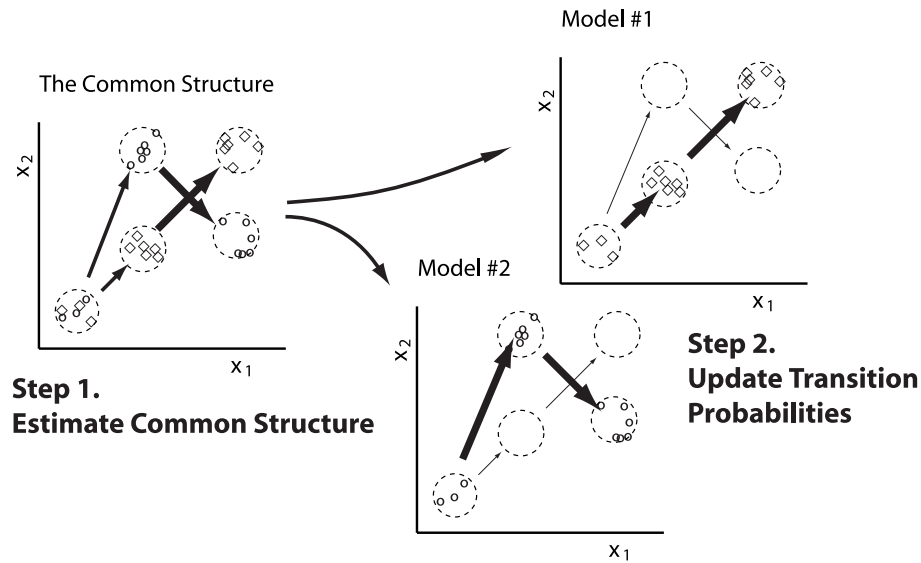


Figure 4.4: This Figure gives a schematic overview of the two steps in the training of the Common-Structure HMM for the example data from panel B. in Figure 4.1. The thickness of arrows represent the transition probability; a thick arrow indicates it is probable to make that transition, whether a thin line indicates that transition is improbable. The reader is referred to the text for details.

the required number of hidden states is higher. Only in the case of panel C of Figure 4.1 the number of states in one of the conditions is equal to that in both conditions together. However, the number of states in the model of the entire process is expected to be less than the sum of the number of states in models for the individual conditions due to idle states common to both conditions.

The model based on all the data provides us with a model of general properties of a BCI trial. From the lower left state, the middle two states are equally probably attained as indicated by the thick arrows in Figure 4.4. The states define subspaces of the feature space where EEG spectra can be expected, independent of the condition of a specific trial. Training this model requires more time than training two separate models because computation time is not linear in the number of states, and the re-estimation procedure converges more slowly.

If we now take this model to be an appropriate description of an EEG trial in feature space, we may discern between conditions based on how an unseen trial advances through this structure. This is different from considering to which one of two structures a trial complies best. We denote the model describing all the data the *Common Structure*. This approach is assumed to be more robust to noise in the exact locations of cluster centers.

In order to discern the conditions, we construct new models in Step 2 aimed at classification by making a copy of the structural model for each condition. We then update the estimates for a_{ij} of each copy to the relative frequencies of state transitions observed in the subset of the training data corresponding to the condition we want to describe. We see in Step 2 of Figure 4.4 that

for the condition \diamond the transition to the center cluster is far more likely than the transition to the upper-central cluster and vice versa for trials from the \circ condition.

If we now evaluate $p(O|\lambda_i)$ with λ_i the copy with updated a_{ij} and compare it to other models, the difference in probability of the observation reflects how probable the advancement through the common structure is in that condition.

This model would perform as well as or better than a traditional HMM if the underlying structure is similar to that in panel C of Figure 4.1. However if the structure were similar to that in part A and we observe a trial from condition \circ , the difference between λ_\circ and λ_\diamond cannot be determined by the probability of observing feature vectors from s_2 or s_3 , but must follow from the improbability of going to s_2 ($a_{12}^\circ < a_{13}^\circ$) and the difference in the recurrent probability $a_{33}^\circ < a_{33}^\diamond$. The system remaining in state s_3 is more probable in model λ_\circ .

However, this model is unable to exploit the structural information that is present in panel A. The performance of this model is therefore evidence for a relatively large amount of information present in the time structure with respect to information present in the structure of the feature space.

4.5 Experiments with HMM variants

The performance of these three time models (Traditional HMM, Common-structure HMM and Flat HMM) will be assessed in offline experiments on EEG data. We use the data sets described in Section 3.4. We compare the accuracy in classification of trials as a whole (percentage correct) of these time models and the instantaneous classifiers (See also Section 3.5). The models described here are implemented with wrapper methods similar to those for instantaneous classifiers described in Section 3.5.

The traditional HMM and Flat HMM are trained with a number of states ranging from 2 to 6. This range is empirically determined. Furthermore the window of 500ms imposes an upper bound on the number of state transitions we can discern. The Common Structure HMM is trained with twice the number of states of the aforementioned models. We iterate the model until the average over three iterations of $\log(p(O|\lambda_t)) - \log(p(O|\lambda_{t-1})) < 1$. We make six runs of every model on every data set. We report the accuracies of these different settings as the mean accuracy with error bars indicating the standard deviation around the mean $\bar{x} \pm s_x/\sqrt{6}$.

4.6 Experiments on artificial data

We also created three artificial data sets exhibiting the structures described in Section 4.1. These are data that are linearly separable, non-linearly separable and data that are only separable based on its time structure. The data sets contain two features and cluster centroids are positioned similarly to those in Figure 4.1.

We compiled the data sets by drawing sequences of fixed length ($L = 20$) from an ergodic Markov process with preset parameters A , B and π . We added random zero-mean noise to all the observations. The difficulty of the classifi-

cation can now be set through the variance of observation distributions and of the overall noise. We set parameters such that the classification is trivial nor impossible for the HMM as our goal is a comparison between the five types of classification.

We constructed three data sets with 100 trials of length 20 each. In the results we report the performance for a number of states between one and ten obtained from 3-fold cross validation. For kNN we report the performance for numbers k that are the odd numbers that are nearest to, but larger than the square of the number of states, also cross validated. For the linear model we report a single value, namely the mean value obtained from cross validation.

Per fold in cross validation, we initialized every model three times yielding an estimate of performance based on 9 experimental values. We did not remove models that converged badly. Of course this type of data is very well suited for classification by an HMM by design. But though the real data will not obey the assumptions of a Markov process as well as this data, we use the performance on the artificial data as a measuring stick for the type of structure in the real data.

Chapter 5

Results

This chapter describes in detail the results we obtained. We compiled a collection of data sets from the original raw data. Our own data were collected in differing contexts due to the fact that running participants in our setup had only just left the exploratory phase when these results were compiled. Therefore we did not yet have the disposal over large well controlled data sets. At this point however the system is ready to be deployed in order to obtain that sort of data. The data sets are described in Section 3.4.

Section 5.1 describes results on simulated data. Section 5.2 reports on the *engineering* research question regarding the performance of a BCI accounting for time structure, compared to instantaneous classification. The next Section 5.3 describes the results of experiments investigating the decrease in performance of classifiers when the size of the training set is reduced. Section 5.4 focuses on the difference in performance between the three types of Markov models we employed. This enables us to answer the more *fundamental* research question about the information present in the time structure. The Chapter concludes with qualitative results obtained from the participants through a questionnaire regarding their experience of control. We omitted the results from our methods to describe the data such as correlation over time and mutual information.

5.1 Performance of Temporal models on artificial data

We modeled the types of time structure introduced in Section 4.1 to test the actual performance of our different types of temporal models. Recapitulating, we discerned linearly separable, non-linearly separable and time-based separable data. The first two types of data can be discerned on the basis of structural differences in the centroids of underlying states, whereas the third cannot.

The performance of all five learning methods on the three data sets is shown in Figure 5.1. For linearly-separable data, the CSHMM clearly suffers from being unable to use structural information and requires more states to attain performance similar to that of the HMM and Flat HMM which have no problem in classifying. The performance of instantaneous classifiers may be expected to be slightly lower than that of the model-based classifiers since this data originates from a Markov model.

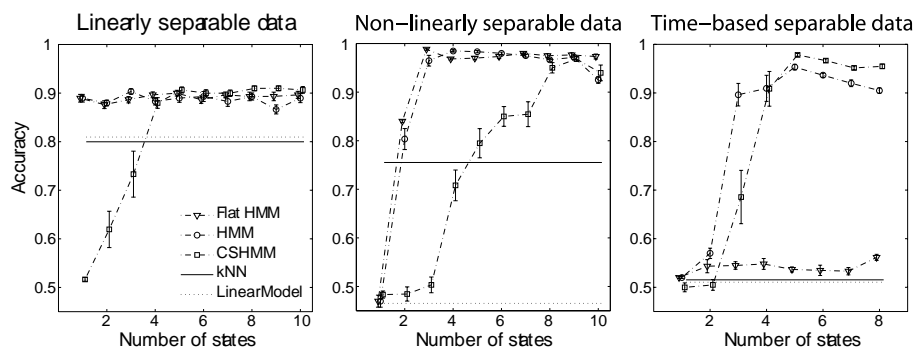


Figure 5.1: An overview of the performance of the three models on artificial data. The panels from left to right correspond to performance on linearly-separable data, non-linearly separable data and time-based separable data. Mean accuracy and standard deviation around the mean are estimated from 3-fold cross validation, with three random initializations of the model per fold. Error bars indicate the standard deviation around the mean performance.

For non-linearly separable data we again see that the CSHMM is outperformed by the other models, for smaller numbers of states. The CSHMM needs 8 states in order to perform as well as the others. On this data the Linear model performs at chance level, which stems from the impossibility to draw a straight line separating the classes.

For the time-based separable data, the Flat HMM performs only slightly over chance level and the CSHMM by times outperforms the HMM. The Linear model and kNN also perform at chance level. There is a large jump in performance from two to three states. The lag of the CSHMM stems from problematic convergence for this type of model in this case. This is reflected in the large error bar: the CSHMM either performs very well, or at chance level. For four or more states this problem no longer occurs.

5.2 HMMs compared to instantaneous classification

In this section we compare the performance of the HMM with that of instantaneous classifiers. The differences between the models HMM, CSHMM and the Flat HMM are considered in Section 5.4. We will start out with an overview of performances on the different one-dimensional data sets. In the remainder of the section more specific aspects of these results are described.

The fit of an HMM is dependent on the number of states in the model. The graphs in this section display the accuracy for a number of states ranging from two to six, see Figure 5.2. The Linear model is deterministic so we obtained only one accuracy per data set. The kNN method is deterministic given the number k of neighbors. We also report kNN performance as a single accuracy, namely the highest over k . These two values are plotted as horizontal lines in the graphs and are reported in Table 5.1.

The performance of a Hidden-Markov Model for a given number of states,

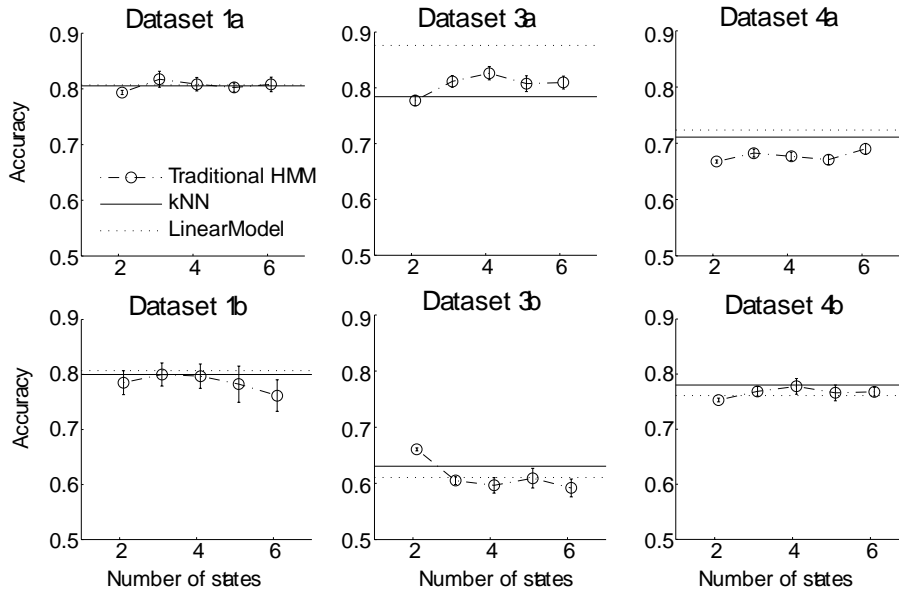


Figure 5.2: An overview of the performance of HMM set of to instantaneous classifiers on the data sets with one-dimensional tasks. Data set 3c is omitted because the graph is very similar to that of data set 3b.

is plotted as the mean accuracy with error bars of length twice the standard deviation of the mean. Variability in the performance of a model is due to the effect of random initialization and not from cross validation as described earlier. The standard deviations give an impression of the robustness to random initialization of the model.

We compare performance for the optimal number of states of the HMM to a baseline of the Linear model and to kNN with the optimal number k in Table 5.1. The methodology of choosing optimal parameters based on the results from testing is addressed in the Discussion. For all subjects except for RH, chance level was 0.5 (two targets). RH operated a two-dimensional BCI with chance level at 0.25 (four targets).

The classification accuracy of the HMM was generally worse or as good as the instantaneous baseline methods. This also follows from Figure 5.2. The difference in performance between a HMM and instantaneous classifiers is expressed by the p-values (right-most columns). We present separate p-values for each data set because the data was too heterogeneous to allow for pooling of data sets. These p-values are not corrected for the large number of tests we perform but merely serve as a normalized difference in performance. We will come back to the possibilities of generalization from these results in the next Chapter.

There is a large drop in performance on the two-dimensional task in data sets 2 and 5. For a large part this is due to the fact that chance level is only 0.25 in this case. But if control over the horizontal and vertical component were independent and accuracy was similar, we would expect performances roughly

data	subj.	HMM \pm sd (n)	kNN (k)	LM	p-kNN	p-LM
1a	JC	$0.82 \pm .014$ (3)	0.81 (4)	0.81	.22	.25
1b	JC	$0.80 \pm .021$ (3)	0.80 (16)	0.81	.50	.63
2a	RH	$0.39 \pm .017$ (2)	0.18 (4)	0.33	< .01	< .01
2b	RH	$0.28 \pm .023$ (2)	0.29 (4)	0.29	.66	.66
2c	RH	$0.50 \pm .028$ (2)	0.47 (9)	0.46	.17	.11
2d	RH	$0.32 \pm .027$ (3)	0.30 (36)	0.39	.19	.96
3a	O3VR	$0.83 \pm .012$ (4)	0.78 (4)	0.88	< .01	> .99
3b	S4	$0.66 \pm .002$ (2)	0.63 (16)	0.61	< .001	< .001
3c	X11	$0.70 \pm .009$ (2)	0.62 (16)	0.68	< .001	< .05
4a	JC	$0.69 \pm .010$ (6)	0.71 (16)	0.72	> .95	> .99
4b	JC	$0.78 \pm .014$ (4)	0.78 (4)	0.76	.57	.15
5a	RH	$0.41 \pm .024$ (2)	0.35 (36)	0.38	< .05	.13

Table 5.1: This table shows the mean performance of the HMM over six initializations with optimal number of states and the instantaneous classifiers on all datasets. The instantaneous classifiers are single values because there is no random initialization. The p-values originate from one-sided t_5 -tests comparing the mean HMM score over 6 iterations to the fixed score from kNN and LM (Linear Model) respectively ($df. = 6 - 1$). H_0 is the hypothesis that the instantaneous classifier is as good as or better than the HMM. The p-value expresses the probability of the observed or a larger difference under H_0 .

the square of that on one-dimensional tasks, i.e. around 0.55. The observed performances are lower. We can further investigate two-dimensional performance by decomposing it into the separate performance on the horizontal and vertical dimension of the task.

We show a comparison between the horizontal and vertical cursor control task in Figure 5.3. These tasks are both expected to exhibit a performance of 0.5 at chance level. It is clear that the horizontal task can be better discerned than the vertical task ($t(29) = -24.81, p < .00001, H_0 : acc_{UD} \geq acc_{LR}$ from a paired t-test over six repetitions for five numbers of states). The performance over chance level of 0.25 almost entirely results from the subjects control over the horizontal component.

There is also large variability in the performance of methods between one-dimensional data sets as can be seen in Figure 5.2. This is presumably due to difference in control over the EEG components between subjects. The larger data sets 3a and 4b show a very robust accuracy at 0.80 and 0.78 respectively. From the graphs, it is striking how small the differences are for different numbers of states.

We performed one-way analysis of variance to quantify the difference in accuracy for different numbers of states for the HMM on data sets 3a and 4b. We used the different numbers of states as an ordinal variable¹. We used 5 numbers of states and six observations per class. This results in 28 degrees of freedom in the error term.

On neither data set we found significant influence on performance from the

¹Using a nominal variable for the number of states did not alter the outcome.

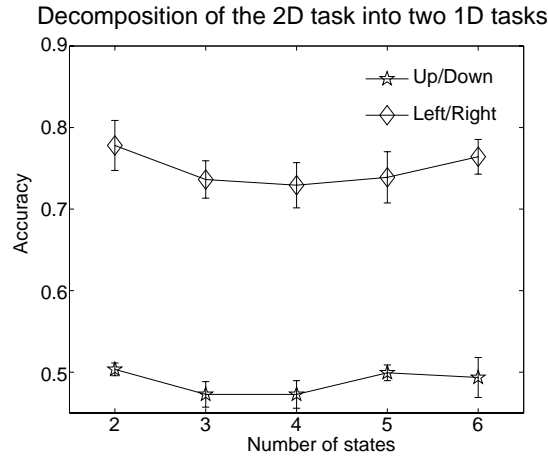


Figure 5.3: The performance on the two tasks comprising two-dimensional control. For scoring of the trials only information in one dimension is considered. In this case, should the target be the upper left (see Figure 3.4), but the trial is classified as upper right it is considered correct, since the up/down distinction was correct. We see that the left/right distinction is classified comparable to the one-dimensional task, whereas the vertical task scores no higher than chance ($= 0.5$).

number of states ($F(1, 28) = 2.38$; $p = .13$ and $F(1, 28) = .68$; $p = .41$ for data sets 3a and 4b respectively). From the graphs we may expect the effect size to be too small to be discerned with only six observations. Due to the small difference in means the power of the tests was indeed very low: .15 and .07 for data sets 3a and 4b respectively. We did not compute the discernible effect-size for our test. Upon comparing the difference in performance for the numbers of states with the difference on the artificial data, we conclude the effect to be negligibly small, if existent.

The classification accuracy acc might also be affected by the type of feature used for classification. This can be seen by comparing data sets using difference in power between two channels (data set 1a) and data sets using the concatenation of the vectors of spectral powers in both channels (data set 1b) as in Figure 5.4.

In two-factor ANOVA with the number of states (5 levels) and the type of feature (2 levels) for a HMM, neither one of the factors nor interaction between the two, is significant. For the effect of the type of feature we find $F(1, 50) = 2.76$; $p = .10$.

We also performed a paired t-test to compare the mean for both features. This also did not attain significance ($t(29) = 1.83$; $p = .07$) even though power at a difference in means of 2.5% was .999. From this we conclude there is no difference in performance between these two types of features.

In a later stadium I used the first derivative of spectral power to time as a feature (graph not included here). It will require a more thorough analysis to assess whether this feature performs significantly different. The observed difference is similar to the difference between the two features described earlier.

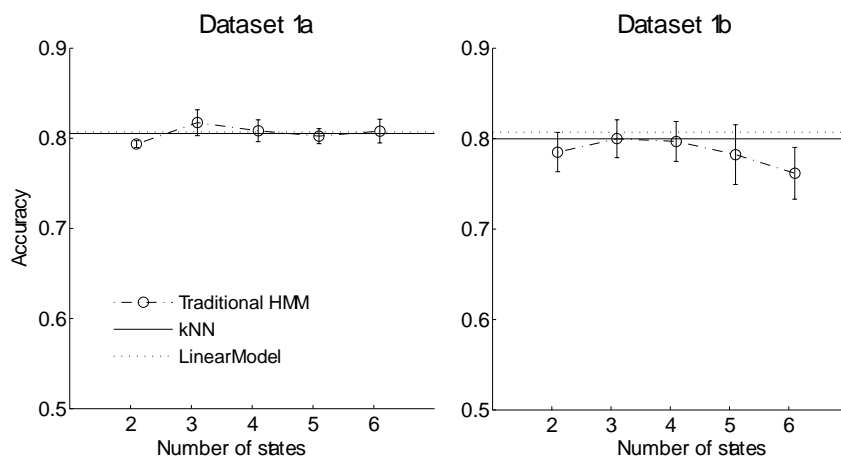


Figure 5.4: The performance on data set 1. The difference between the two is the feature used for learning and classification. Data set 1a uses the difference in spectral power between two electrodes, whereas 1b uses the spectra of both electrodes.

5.3 Performance on reduced training sets

Larger training sets generally improve performance as the learning system is able to make a more reliable estimate of the underlying distribution. This is of course under the assumption that training data and test data are similar apart from variability in individual trials. In other words, it is assumed that samples in training and testing data are drawn from one common distribution.

For practical purposes we would like to keep the training set as small as possible. Therefore it is useful to know how performance is influenced by the amount of training data available.

The evolution of performance when we alter the size of the training set is shown in Figure 5.5. The graph shows this evolution of the mean accuracy of a HMM with four states² over 4 random subsets of the data of the same size. We see no large differences in how performance degrades between methods. In fact, one may be surprised about the insensitivity of the HMMs to reduction of the training set. Only for the CSHMM one may denote the graph structurally different.

We analyzed these data in a two-way ANOVA. We described the process of drawing subsets from the larger data set as a random effect, on which we performed repeated measures for every training method. So for every fold, we assess performance of each training mechanism on the exact same subset. A subset of the data is regarded as an individual 'subject' in psychological terms. The size of such a draw is a feature of that draw (similar to 'height' of human subjects).³ We denote this factor as 'percentage' with the levels 1%, 5%, 10%,

²One HMM or Flat HMM uses 16 a_{ij} , 4 π_i and 2×4 mean- and variance vectors for the observation distributions. One model thus has 28 parameters to estimate from observations, the two models use 56 parameters. The CSHMM estimates 64 a_{ij} twice, and 2×8 mean- and variance vectors only once. Thus 144 parameters.

³This approach was suggested by dr. H. van Rijn.

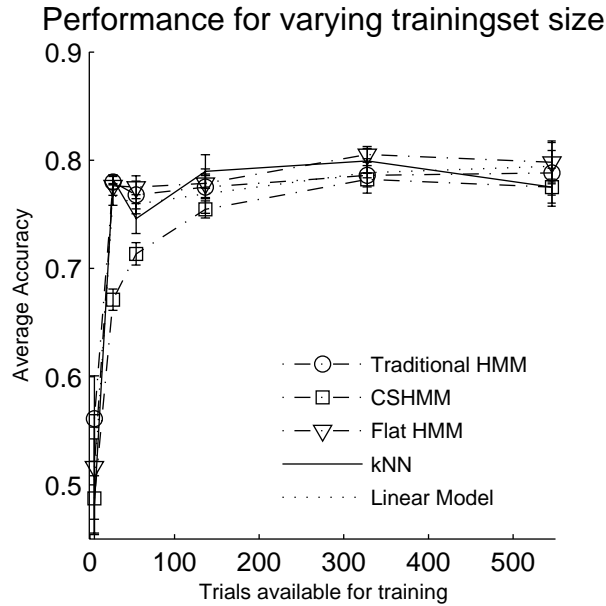


Figure 5.5: The evolution of performance for reduced training data. If we reduce the number of training data, we expect the performance to degrade. The horizontal axis indicates the number of trials available for training. The vertical axis indicates the mean accuracy from 4-fold cross validation for models with four states (HMM and Flat HMM), or eight states (CSHMM) or 17 neighbors (kNN). Data is data set 4b. The mean length of a trial is 45 feature vectors.

25%, 60% and 100%. We treat 'percentage' as a factor due to the observed non-linearity in the data.

In psychological methodology terms: we compare the effect of training method *within* subjects, and the effect of training set size *between* subjects. We have four folds and 6 levels for the factor 'percentage', resulting in 24 partly overlapping subsets of the data set on which we assess performance. The purpose of this approach is to remove correlations due to a particular subset, such as the separability of the data, prior to analysis.

A difference in how a training method is affected by a decrease in the number of training samples should result in a significant interaction between the factors 'method' and 'percentage'. From Figure 5.5, we would certainly expect an effect from the percentage of training samples, and possibly an effect from the training method.

We found a highly significant effect of percentage ($F(5, 18) = 41.20, p < .0001$) as was to be expected. We also found a significant effect of training method ($F(4, 72) = 8.58, p < .0001$). The interaction ($F(20, 72) = 1.84, p = .032$), though $p < .05$, should be treated carefully in this analysis. We did not compute the power of this test. From the graph we expect that only the CSHMM degrades structurally different from the other methods when the training set size is reduced to less than 200 trials. This is partly due to the fact that the CSHMM requires more states and therefore uses over twice as many parameters. The

hypothesized difference in degradation between the HMM and instantaneous classifiers does not follow qualitatively from the graph.

5.4 Comparison of HMM variants

To investigate the type of time structure underlying the process we compare the performance of models suited to different time structures. The traditional HMM combines the structural and temporal characteristics of the data in a probabilistic framework. As described in Chapter 4 we used the Common-Structure HMM to emphasize on temporal information and Flat HMM to emphasize on structural information.

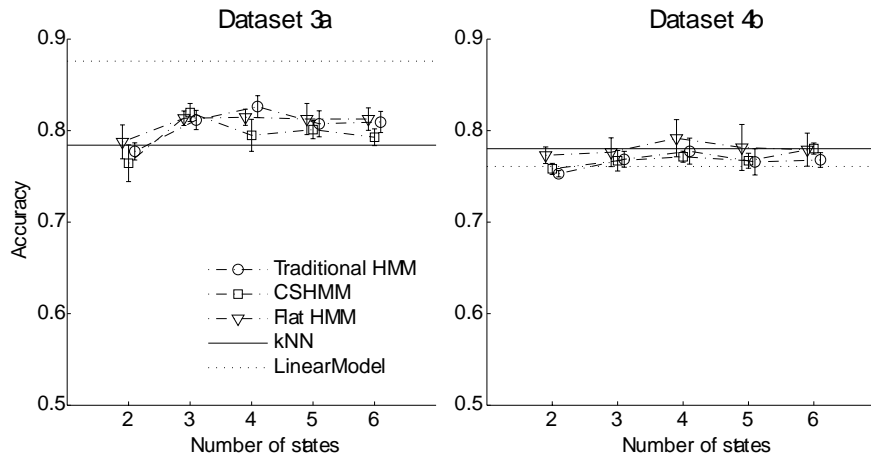


Figure 5.6: The performance of all three models on the larger training sets. The horizontal lines display the performance for comparison to instantaneous classification. There are little differences between methods and between numbers of states. The reader is referred to the text for the analysis.

The previous Section also incorporates these alternative trial models to assess the sensitivity of methods to the amount of training data. Otherwise a difference in performance between methods due to insufficient training data could lead to unjustified conclusions regarding the type of time structure. From that analysis we have no reason to expect the amount of data is limiting performance.

We see from Figure 5.6 that the three models of trials only differ slightly. The effect size of the number of states is small on all learning methods. The lines connecting points of specific methods cross constantly. This may indicate the effect is not just small, but even nonexistent. From qualitative analysis of the graph we do not expect the existence of a difference between the three types of model. If any, the mean performance of the Flat HMM structurally lies higher than that of the HMM and CSHMM for data set 4b.

When describing the larger data sets we find no significant effect from the type of model on performance ($F(2, 75) = 2.09, p = .13$ and $F(2, 75) = 2.87, p = .06$ for data sets 3a and 4b respectively). We performed two-way ANOVA with factors for the type of model (3) and the number of states (5). If we use

an ordinal variable for the number of states, the effect of the number of states is significant, but the effect of the type of learning does not change greatly.

The fact that the learning method *was* found significant in the previous section stems from the increase of power in that test. We measured performance for 6 percentages of the data set, thereby yielding far more observations. In this case we only seek a difference in performance on the total data set. Also from Figure 5.5 we cannot discern a qualitative difference between different modeling methods.

5.5 Qualitative results

Participants were given a short questionnaire on their experience of control. As also reported by Wolpaw [2] the initially instructed control by performing mental imagery was gradually replaced by a type of control where the subject thought of moving the cursor itself. The subjects experienced the cursor to be responding directly to their intentions which is a good sign; they actually experienced control.

The subjects were not able to discern correct and incorrect trials from their experience of control i.e. without feedback. If there was an unintended but conscious process at work, we would expect the user to 'proprioceptively' experience a cursor going astray. This requires further research but suggests that it is either an unconscious process, or external noise interfering.

When asked who became better over time, the system or the subject, we received different answers. One subject reported he had better control over suppression of unrelated activity and was better able to focus. The other subject reported no change in his mental activity.

Subjects were skeptical about future applications when compared to applications suggested in the literature. One subject suggested the use of games (such as pong) for severely impaired children. Subjects were especially concerned about the potential of error in a BCI decision.

The subjects did not experience an effect of mental activity on the speed of the cursor. One subject hypothesized that his amount of concentration determined speed. They only experienced control over position and not over speed.

The subjects both experienced a large effect of concentration and fatigue on the performance on the task. Whether the subjects found the operation interesting, fun, strenuous etc. differed largely over the subjects.

Chapter 6

Conclusion & Discussion

In this chapter we will first report on our conclusions regarding the research question into the performance of temporal models compared to instantaneous classifiers in Section 6.1. In the subsequent section we describe conclusions regarding the question into the type of time structure in these trials. The results are discussed in Section 6.3. Section 6.4 points out limitations of our approach partly giving rise to possible future work in Section 6.5.

6.1 Instantaneous classifiers and temporal models

Literature from speech- and handwriting recognition has shown that in well controlled conditions, HMM-based approaches outperform instantaneous methods. In that field of research time information is crucial but is it also in this field?

In answering the first engineering research question, whether models accounting for time structure outperform instantaneous classifiers, we did not find conclusive evidence for the superiority of either one. The data do not indicate that this lack of significance is due to our limited amount of data, but due to the very small effect sizes. Increasing the number of iterations of the model to attain significance does not seem sensible in this case.

To gain more insight into the matter of accuracy we need more data in order to be able to generalize beyond our individual subjects. It would be advantageous to be able to pool data from multiple subjects into one analysis. That would also allow to sensibly increase the power of the test allowing to conclude smaller differences.

Though the HMM requires more parameters than the instantaneous classifiers we tested, our analysis shows that with 200 trials the model can be well trained. This corresponds to roughly half an hour of training which seems acceptable. The training of the subject itself requires far more time.

6.2 The type of time structure

We constructed three artificial data sets with varying amounts of information in the structure of features and in the time structure. We assessed performance on these data sets of all the methods described previously: HMM, CSHMM, Flat HMM, Linear Model and kNN.

The simulated data originate from a Markov process by design. This imposes a proper time structure but also guarantees the existence of discrete states underlying the process. Therefore it is not surprising that the HMM performs well on these data.

For linearly and non-linearly separable data, the Flat HMM performs equally well as the HMM. For the non-linear data we clearly see the so-called knee in the performance at a number of states of three. Because the Flat HMM performs as well as or better than the HMM, we may hypothesize that it is primarily the structure in state space that is relevant for classification rather than time structure. This is also supported by the performance of the CSHMM which is unable to use this information and performs significantly worse.

We find far worse performance of the Flat HMM for data in which there is a structure of discrete states, but where it is uninformative. This strokes with our design goal for the three types of models. The HMM performs well as expected, as this is exactly the type of data suited to a HMM. The performances of the HMM and CSHMM also exhibit a large leap in performance when the number of states exceeds the true number of states (3). The instantaneous classifiers are without a chance on these data as was to be expected.

We also see a clear distinction between the instantaneous classifiers between the linear and non-linear data. Performance of the Linear Model compared to performance of kNN appears to be an indicator of linearity in the data. They are -as expected, unable to make use of time structure. We will now seek the aforementioned characteristics in real data to describe the type of structure in trials with an EEG based brain-computer interface.

The results on artificial data show that with these approaches it is in principle possible to discern different types of time structure. We did however not find significant differences between the three types of models on real data. The observed differences are so small, they require far more iterations to attain the power to significantly discern difference. We did not increase the number of iterations due to the small effect size ($< 2.5\%$). It would be difficult to defend conclusions from differences that small obtained from only two subjects.

From the lacking of large differences between the Linear Model and the HMM we conclude that to a large extent the data is linearly separable. Since we found no data set where the Linear Model did not perform well, we hypothesize that linearity is characteristic for this type of data to a large extent. This is also in line with Garrett et al. [49] who also found small differences when using non-linear and linear classification.

This may also explain the high performance of the Flat HMM using structural information. From these results we also conclude that time information is not as vital as it is in speech recognition. We may rule out the hypothesis of a time structure such as that in panel C of Figure 4.1. The performance of the Flat HMM compared to the HMM is partly in line with earlier work on

time-dependent neural networks [50], which also found only small improvements when using time information (effect size $\approx 2\%$ reduction of Error rate.).

The fact that the CSHMM also performs well may indicate that state space does not lack completely, as the transition probabilities obviously bear enough meaning for classification. We could check this by further investigating the structure that is learned by the models such as the difference in centroids within a model, and whether the structure of a model is constant over different initializations. If the model converges similarly every time, that is a strong indication there actually is a structure. In the early phase of this project we analyzed the data with hierarchical clustering to find clusters or states in the data. Extensions of this work may also aid in answering the question into the existence of state space.

6.3 Discussion of the results

We found only small differences in performance between the methods, but relatively large differences between data sets. We must consider the possibility that performance is primarily limited by the performance of the subject. Note that the intrinsic separability of the data is determined by how well the subject performs its task.

When comparing performance on data sets 1a and 4b from the same subject but at different moments during its training, we don't see large differences in performance of classifiers. This may suggest that the separability of the data does not improve with this type of training by the subject. The subject is incited to learn a linearly separable signal as the feedback is fixed to the difference in power over the mu-bands of C3 and C4.

The effects of the type of learning by the machine on the learning by the subject are difficult to investigate as they require large numbers of subjects trained in separate conditions. This cannot be investigated in offline analysis. The learning of this type of task is a very interesting field of research for example for movement scientists. Such a setup would require an on line implementation of the machine learning schemes in BCI2000. This is not addressed in the current project.

When we compare the artificial data and real data, the most striking difference we see is in the performance of the models (HMM, CSHMM and Flat HMM) relative to that of the instantaneous classifiers. The performance of the Linear Model and kNN are rather good on our experimental data when compared to artificial non-linearly and temporally separable data. We were also unable to establish an effect of the number of states on performance. From this we may suspect the state-space representation to be inappropriate.

It may be the case that discrete states exist, but that they are smeared too much by the 500ms window for spectral estimates. In speech recognition, frequencies are higher (speech typically occupies the kHz band). That allows for shorter integration times. As the band bearing the gross of information is around 10Hz, there is a lower bound on integration time of 100ms.

We used 5 periods of a 10Hz waveform to estimate the power based on current literature. The methods on which we based our window size however do not employ state-space methods. It may be the case that it is necessary to

use shorter windows to capture possible underlying states. An other alternative would be to use temporal features which do not impose lower bounds on integration time.

Concluding, our claims about the value of time structure are limited by our window size. We are also limited in our conclusions by the power of our tests because the observed effect sizes are very small. Time structure may exist at another level. Note that literature on ERD [19] suggests that the time structure in this phenomenon is of the order of seconds. Also the task outline with four seconds of feedback and repetition of the motor imagery at twice per second do not indicate we should expect time structure for shorter windows. We also assessed performance when using the first derivative of features which should reduce the smearing. The increase in performance is far from clear with this feature. But we leave open the hypothesis that the low time resolution limits performance.

We decided not to perform cross validation but to use a fixed training and test set. These sets are constructed as the first and last part in time of recorded signals. This violates the assumption that training and test data share a common structure, as we may expect there to be non stationarities. However it *is* a good reflection of the operation of such a system in practice where the training data is obtained first, and than the system is tested by a user on new data which may be different in structure for example due to learning.

We did check whether cross-validation has large influence on performance. Though in detail performances change slightly, the overall impression of these data remains unaltered.

Hidden-Markov models are very sensitive to proper initialization. We have seen this particularly in our artificial data sets. The EEG data sets did not suffer as much from lack of convergence. The CSHMM is more sensitive to this problem because it estimates the state centroids only once whereas the mixture of Gaussians and the HMM estimate centroids for each condition. In the latter two cases the bad convergence for one condition is masked by good convergence for the other condition. The well converged model will correctly classify the corresponding test samples, whereas the other test samples are classified at chance level. For the CSHMM all instances are classified approximately at chance level.

Originally we used the mean- and variance vectors of random subsets of the data as initial estimates for the states. The problem that may arise is that for large subsets all (or some) states are initialized at the same location. If that happens in consecutive iterations of the training algorithm, states may become coupled instead of optimally spanning the feature space.

A solution to this is to randomly select one instance from the data as an estimate for the mean of a state. However this may also pose problems as the same problem arises when two instances from the same underlying state are chosen for two different states in the model. An even more elaborate solution would be to use a variant of hierarchical clustering [28] for initialization.

From the fact that variability in performance remained small, we concluded that the problem was not as acute on the EEG data as it was on artificial data. Though it would be a valuable extension to the model, ensuring better convergence, we did not implement it yet.

There is a methodological flaw in comparing the performance in optimal conditions posterior to training. It would be more sound to partition the data in three. We could then use disjoint portions to set parameters (such as the number k in kNN and number of states in the HMM, CSHMM and Flat HMM), train the model and test the model respectively. This was noted only after the experiments were run.

We don't expect that the general impression of the results will change, only that the methods may suffer even worse from data sparseness. Moreover we found little influence on performance from these parameters. Therefore we decided to estimate these parameters from the results based on optimal performance. This approach is biased against the Linear Model because this approach has no auxiliary parameters and therefore would benefit from a larger data set than the other methods in the proper estimation, training and testing procedure.

6.4 Limitations

The fact that we were able to base our conclusions on only two subjects severely impairs our ability to generalize these results. The description of the two subjects however is very accurate as it is based on large amounts of training by the subjects. This enables us to construct hypotheses to be tested in larger-scale experiments with a larger number of subjects. Note that this project constituted the start up phase of the new research group. Therefore the informed construction of hypotheses for BCI research was one of the aims of this project.

The community of BCI researchers attributes high value to on line implementations. This method is as yet not implemented on line. However to our knowledge, almost all on line implementations operate instantaneous and no HMM approach has been implemented on line. As noted earlier, this would be a valuable addition as it would also allow for research into the effect of the learning scheme of the machine on the learning by the subject.

HMMs are generally trained in batch mode due to the nature of the Baum algorithm. It is very expensive to rerun the re-estimation procedure between every trial with the previous trial added to the training data. However a batch training between training runs (typically every 40 trials) is very well conceivable. It is also possible to re-estimate parameters from each new observed trial and then dispose of that datum. This better resembles a typical on line setting and also allows the system to *track* the learning by the subject.

A real-time implementation of Viterbi decoding used for classification *during* trials and on line is very well conceivable as the Viterbi algorithm itself is recursive over time. This would allow for the application of real time cursor control. Finally, an on line implementation for classification of a trial as a whole (for a 1 out of N selection) does not pose a problem in any way. An on line implementation sheds light on whether this approach improves the learning experience by the user. This is valuable information subsidiary to the simple amount of correctly classified trials in offline analysis.

The EEG is non-stationary in many ways. In part our approach is able to account for these non-stationarities, namely the non-stationarity that occurs

within a trial. For the rest our approach assumes the process to be stationary, in particular on the time scale *over* trials. Since learning is inherent to human behavior this assumption is clearly violated to some extent. The trials from sessions separated in time differ, though we do not quantify this difference. One may get an impression of the non-stationarity from the Figures in Appendix B.

Our performance measure is only partly affected by this assumption because the training- and test set are generally composed as the first and second half of trials in a session. If these trials would differ largely that would negatively affect the recognition of the second half of trials. It does however not account for even longer-term learning effects which clearly do exist.

The undesirable result of this lack of adaptivity is that though the amount of information in the EEG increases, the performance decreases. Since we operate in a probabilistic framework we may envision an approach deriving from Kalman filters to track structural changes in the EEG of the order expected resultant from learning. Adaptive learning has been employed more frequently in BCI research, e.g. on Kalman filtering in [36].

A far more simple solution (but less elegant) is to require a calibration phase of the system. Such a calibration may be imposed periodically (e.g. every hundred trials). Another way would be to use a self evaluation by the system of $p(O|\lambda)$. If the characteristics of the process remain stable (i.e. no calibration is necessary) we expect this probability to remain stable. If this probability decreases, the system can conclude that it is time for calibration. For deployment in real-life situations the system must be able to adapt.

6.5 Future work

This project was the start up of a BCI group in Groningen. Therefore we performed a large amount of exploratory work. In this thesis I reported on the subject of my focus: time structure in trials with the BCI. However a large number of other questions were encountered and left open. This section will describe the in my opinion most promising topics for future work.

This approach being a model-wise approach it is very interesting to explore other advantages of such a method over instantaneous methods. Examples of this are the implementation of rejection for trials or data which do not contain information that fits our model. This could be a first step towards the implementation of an asynchronous BCI, in which the BCI itself can detect whether the user has an intention at all. In that setting the CSHMM may prove beneficial as it is an expression of a trial in general.

Another improvement could be found in feature selection and -combination. Probabilistic methods are sensitive to the curse of dimensionality, resulting from uninformative features which introduce noise to the classification which the classifier is unable to remove. We found that the mu band contains most of the information. Incorporation of this a-priori knowledge by hand, or through an automatic feature-selection phase may greatly improve performance. We also trained with only the five features directly around the mu-band in combination with using the first derivative over time of features. At first sight this does not work miracles but future research should certainly include features improving time resolution.

Also the use of different features altogether should be considered. In current BCI research spectral power is but one of a number of features. We already suggested the use of temporal features allowing for a higher temporal resolution such as the Bereitschafts Potential and Slow Cortical Potentials. There is also the possibility of using features which combine temporal and spectral aspects as used in Lausanne [51]. For our group in Groningen the use of Continuity Preserving Signal Processing might be considered.

Finally, a probabilistic framework allows for the combination of heterogeneous features [22] in an elegant manner. One would still have to guard for too high a dimensionality for reasons discussed earlier.

The field of BCI research is currently occupied by (clinical-) psychologists and computer scientists. However, there is a field of research wide open for movement scientists to be developed. The initial training by the subject is difficult and theory about learning such a task does not yet exist. Also the effect of certain design choices, such as agility of the system are an unexplored topic. In the previous Section we described the upper limit of performance by the machine imposed by the performance of the subject.

This project was partly concerned with the type of control a user has or can have over the cursor. In this thesis we were however unable to completely answer this question. The movement of a cursor suggests an analog type of control, with the control signal obtained from the EEG representing a real value of the position or speed of the cursor.

It is however unclear to what extent the movement of a cursor properly reflects the information present in the EEG. For one, there is no work yet in BCI research on whether and how subjects are able to modulate the effect size of features, though there is in EEG literature [19]. Perhaps what we measure is better represented as a one out of N decision. It is also unclear whether this type of control is dependent on the location on the scalp and type of feature used.

Concluding and as suggested also by discussions on the future of BCI research and applications, it is vital to implement approaches in on line applications. We have at moments made an analogy to speech recognition where time structure *is* essential. We should take learning from the lessons learned in that field about the dangers of building systems in too well-controlled conditions. Should a BCI ever be used we must have an impression of the usability in noisy conditions. For example when envisioning a BCI application for a television remote control, is useless if attention to television programs interferes with the signal for that same BCI.

Appendix A

MatLab Manual

The signal processing module in BCI2000 is a series of filters each transforming the signal as described in 3.1.2. In offline analysis we can re-apply these filters to the original data with the BCI2000 command line tools [52]. These command line tools also allow exporting the output of any filter to a MatLab data file. In this project I have made extensive use of this functionality. For analysis of the EEG data I have implemented several tools in MatLab. This Appendix reports on these tools.

A.1 Continuous Observation HMM

I have implemented a Hidden Markov Model (HMM) with continuous observation distributions. This way we can model the EEG data as discrete states emitting continuous type data (spectra). I have implemented a general form which allows use in other applications.

The HMM is a MatLab object. In MatLab an object is implemented as a dedicated directory in the MatLab path containing all the member methods as separate MatLab scripts (.m files). The name of the directory is equal to the object name, prefixed with the 'at' symbol (@). The HMM object is called `HMMeff` ('eff' for **efficient**) therefore the directory name is `@HMMeff`. The parent directory of `/@HMMeff` should be in the MatLab path. The object has the following member variables:

- `nstates`: The number of states of this HMM. This is an integer value.
- `distrType`: The (continuous) distribution. At this moment this parameter can be either 'norm' for a normal, or 'chi2' for a χ^2 distribution. *The latter currently requires the Statistics Toolbox be installed.*
- `modelType`: The model type represents underlying limitations on the state transitions. Currently the HMM allows for two types of models: General models ('gen') where $a_{ij} \geq 0 \forall i, j$ and Bakis (or left/right-) models ('bak') where $a_{ij} = 0 \forall j < i$.
- `estProc`: The initial estimation procedure may be either entirely random or based on an initial partition of the data in time segments. The random

procedure estimates the transition probabilities a_{ij} and priors π_i as normalized random numbers $\sum_i \pi_i = 1$ and $\sum_i a_{ij} = 1$, and the mean and variance of the observation distributions as that of random subsets of the training data.

- **MAXITER**: This is the maximum number of iterations of the Expectation Maximization (EM) algorithm if the convergence criterion is not met earlier.
- **LOWERPROB**: This is the lower bound of transition probabilities a_{ij} . For a Bakis model this parameter must be set to zero. Note that zero probabilities can still occur if the observation distribution is high dimensional. In this case it is not uncommon for $p(O|\lambda)$ to attain the lower bound of double precision.
- **d**: The dimensionality of the feature vectors and therefore of the observation distributions.
- **A, B and Priors**: These are the estimated transition, observation and prior probability parameters respectively of the HMM.

The convergence criterion for the EM is not a member variable but passed to the `train()` method at run time. In evaluating the HMM I followed the HMM tutorial by Rabiner [31]. The method- and parameter- names derive from this text. The member methods for the `HMMeff` object are:

- **HMM = HMMeff(...)**: This is the constructor. The call `HMM = HMMeff(nstates,...)` results in the construction of a `HMMeff` object named `HMM`. All parameters up to **LOWERPROB** can be passed to the constructor in the order in which they are mentioned above. All parameters can be omitted resulting in a default object.
- **HMM = initRand(HMM, O)**: This method initializes the parameters **A**, **B** and **Priors** to random values. It should be passed the `HMM` object and a series of observations `O`. The variable `O` is a $n \times 1$ cell-array, with n the number of observation sequences. Every element `O{i}` must be a $d \times t$ array with d the dimensionality of the data (equal for all sequences) and t the length of the sequence (possibly different for each sequence). This is the general way to present a series of observation sequences to `HMMeff` methods. `O` may also be a single $d \times t$ matrix, which makes sense if initialization is set to 'time'. Erroneous initial settings (especially of variance) may however lead the EM algorithm astray, leading to empty states.
- **HMM = train(HMM, O, psum)**: This method trains the HMM (i.e. iterates the EM algorithm) until the average difference $p(O|\lambda_i) - p(O|\lambda_{i-1})$ over three iterations is smaller than `psum` or the number of iterations is larger than **MAXITER**. `psum` is thus the criterion for convergence, `O` is a cell array of observation sequences as described above.
- **[HMM log[p(O|λ)]] = updateABP(HMM, O)**: This function is called by `train()` and updates the model parameters **A**, **B** and **Priors**. It calls the methods `getAlpha()`, `getBeta()`, `getGamma()` and `getXsi()` to obtain the variables $\alpha_t(i)$, $\beta_t(i)$ and $\gamma_t(i)$ for state i at time t , and $\xi_t(i, j)$ for state $i \rightarrow j$

at time t (for explanation of these variables see [31, Section 3]). This method expects a cell-array of observation sequences (as described above) but calls the aforementioned helper methods with single observation sequences. This procedure is explained in [31, Section 5.B].

- `[HMM log[p(O|λ)]] = updateAP(HMM,0)`: This function is similar to `updateABP()`, except for that it does *not* update observation probabilities. This is useful when we want to update only the underlying dynamics of the model expressed by the transition probabilities, but leave the underlying structure of states -expressed by the observation probabilities- untouched.
- `[-log[p(o|λ)] γt(i)] = viterbi(HMM,o)`: This method evaluates the probability of observing the sequence o (*not* a cell array but a single $d \times t$ matrix of a sequence) given the model λ which generally was trained prior to this call. Optionally the method can return the matrix $\gamma_t(i)$: the normalized probabilities of being in state i at time t during the sequence.

Example of Continuous Observation HMM

The following describes an example problem in MatLab. We have two underlying states $\{s_1, s_2\}$ and a possible transition from $s_1 \rightarrow s_2$. It can be modeled as a two state Bakis model. We have three observed sequences of two-dimensional points which we combine into a cell array. The state transition for o_1 is between points 2 and 3, for o_2 between points 1 and 2 and for o_3 between points 4 and 5. (`>>` denotes the MatLab prompt, and should not be entered. The code is included in the object directory as `demo.m`.)

```
>> O{1} = [0.7 0.7 3.6 3.5; 1.1 0.9 2.6 2.3];
>> O{2} = [0.5 3.6 3.2 4.0 3.5; 1.0 2.4 3.0 2.0 2.5];
>> O{3} = [0.2 0.6 0.8 0.2 3.0 4.0; 1.2 0.9 1.0 0.7 2.3 2.7];
```

Next we instantiate the `HMMeff` object with two states, normal observation distributions, random initialization and at most ten iterations before convergence. (Note for this to work the directory `@HMMeff` must be in the working directory or path) Then we initialize the member variables `A`, `B` and `Priors` to random values and train the model with convergence criterion $\log[p(O|\lambda_i)] - \log[p(O|\lambda_{i-1})] < 1$.

```
>> HMM = HMMeff(2,'norm','gen','rand',10,0);
>> HMM = initRand(HMM,0);
>> HMM = train(HMM,0,1);
```

We can check what the model returns for the following probe sequences. Note that the first sequence matches the model better than the second. Therefore we expect that $-\log[p(o_1|\lambda)] < -\log[p(o_2|\lambda)]$. Check these values by entering:

```
>> p = viterbi(HMM,[.4 .6 3.4 3.6;1.1 .9 2.6 2.4])
>> p = viterbi(HMM,[2 2 2 2;1 2 3 4])
```

A.2 Wrapper method for Machine Learning

The wrapper method is implemented for all five learning schemes¹. The method expects a `TrainingStruct` containing data and settings for learning. The method invokes the appropriate methods to construct feature vectors, train and test the learning scheme, and write the results to a data file.

The most common way to invoke the method is by passing the `TrainingStruct` as an argument. This struct must contain the following fields:

- **d1** and **d2**. These are two parts of a larger data set. In general these sets are meaningfully separated. In our experiments **d1** was composed of trials prior to those is **d2**.
- **TrainingStruct.Index**. This is the structure with information about the structure of the data as constructed by the BCI2000 system.
- **Settings**. This structure contains fields for all settings for the machine learner. This is different for each learning system. For the Linear Model, there are no settings, for kNN only the field **k** is required. For the Hidden-Markov Models there are more fields such as the convergence criterion.
- **featVecType**, **feat1** and **feat2**. These are the parameter passed to the function `toFeatureVectors()` to construct feature vectors appropriate for classification. The operation of this method is described in the MatLab code itself but is a straightforward implementation of 1) selection of the appropriate points from the data set, 2) selection of the channels and frequency bands of interest (enclosed in the parameters **feat1** and **feat2**, 3) performing basic computations e.g. to obtain the difference in power and 4) compiling the result into the proper data structure for the classifier objects.

The method also requires a path relative to the current directory to save the results. It is required that in the directory passed to the method a sub-directory with the name of that learning scheme exists. For example for the kNN wrapper, if the path `/test/` is passed, the results will be saved into `/test/kNN/`. This latter directory must exist or the method will return an error.

The method can be passed an optional parameter `perm=[n i]` for cross validation. This variable is a list with two elements with $n \geq i \geq 1$. The first element represents the n in n -fold cross validation, whereas the second element represents this particular fold. In this case **d1** and **d2** are concatenated and feature vectors are computed as described above. These trials are then ordered in a random permutation. For cross validation the random number generator is re-initialized to result in the same subset each time. For training $(n - 1)/n$ of the trials is used, while $1/n$ is used for testing.

The method can also be passed a parameter `perc` representing the percentage of the training set to be used. In combination with cross validation this is the percentage of $(n - 1)/n$ of the trials.

¹The methods are available under the names `RunHMM`, `RunHMM_PL` (Common-Structure HMM), `RunMixture` (Flat HMM), `RunKNN` and `RunLMS` (Linear Model)

A.3 Data Exploration Methods

This Section describes the methods used for exploratory analysis of the data as described in Section 3.3. These methods provide the basic functionality for analysis. Most of the actual analysis is implemented in scripts to yield combined graphs for data sets. These scripts are not discussed here but should be readable by the person informed of the function of the invoked methods.

corrToTarget()

This method is invoked as `corrToTarget(Data, Index[, plot, fr, verbose, channels])`. The arguments in square brackets are optional. `Data` is data as described in `Index` directly imported, or preprocessed (e.g. after removal of the ITI recordings). The parameter `plot` is a boolean expressing whether the result should be plotted. `fr` is the range for the frequency axis in the plot. This range is linearly interpolated in the graph. If `verbose` is true, the method provides more output. `channels` allows the user to only compute correlation for a subset of the available channels.

This function computes the Pearson correlation of spectral values to the target code. It returns (and optionally plots) the result of size equal to `Index.Signal`. Each element in this variable is a row in the variable `Data`, of which this method computes correlation to the value of `Data(Index.TargetCode, .)`. In the graph, the frequency are set out on the horizontal axis, whereas different channels are represented by separate lines.

corrPerBinToTarget()

This method is invoked as `corrPerBinToTarget(Data, Index, chn, bins)`. `Data` and `Index` are as described above. This method computes the correlation of a specific (set of) frequency bands `bins` of specific channels `chn` to the target code. In this plot, the horizontal axis represents the time axis in a trial. Correlation is computed by first determining the unique values of time `unique(Data(Index.ResponseTime, :))` and then computing correlation for each of these values, thus one value per trial.

mutualInformation()

This method operates similar to `corrToTarget()`. The output is of similar form (A matrix of the same size as `Index.Signal`). The user must plot the results by hand. In this method the user can preselect the channels and frequency bands to use. Of course this can also be selected post-hoc by selecting a subset of rows or columns in the result variable.

meanTrial()

This method is invoked as `[m v] = meanTrial(Data, Index, trials, pl, featVecType, feat1, feat2)`. The resulting `m` and `v` represent the mean and variance respectively. These are matrices of size $d \times T$ with T the mode of the lengths of trials. `Data` and `Index` are as described above. `trials` is a list of the numbers of trials to be used for the computation of the mean. The method

invokes `trialsOverview(Data,Index)` to find trials in the data. `pl` indicates whether the results should be plotted. This is discouraged as in general one wants to have control over the color mapping which in this case you don't. The method invokes `toFeatureVectors()` passing it the last three parameters. This determines the composition of the rows in the resulting matrix.

Appendix B

Datasets

This Appendix contains an overview of the datasets we used in our experiments. In Table B we present the details of the datasets in terms of size, distribution of target values etc. Figures B.1 through B.12 present the mean (difference-) spectrograms (as introduced in Section 3.3.1) of all datasets.

The colormapping is different for datasets, but constant within a dataset. Red cells always indicate large values, blue cells small values. We choose the level of red and blue for a dataset such that structural details are visualized optimally. The extremas are given in the caption of the Figures.

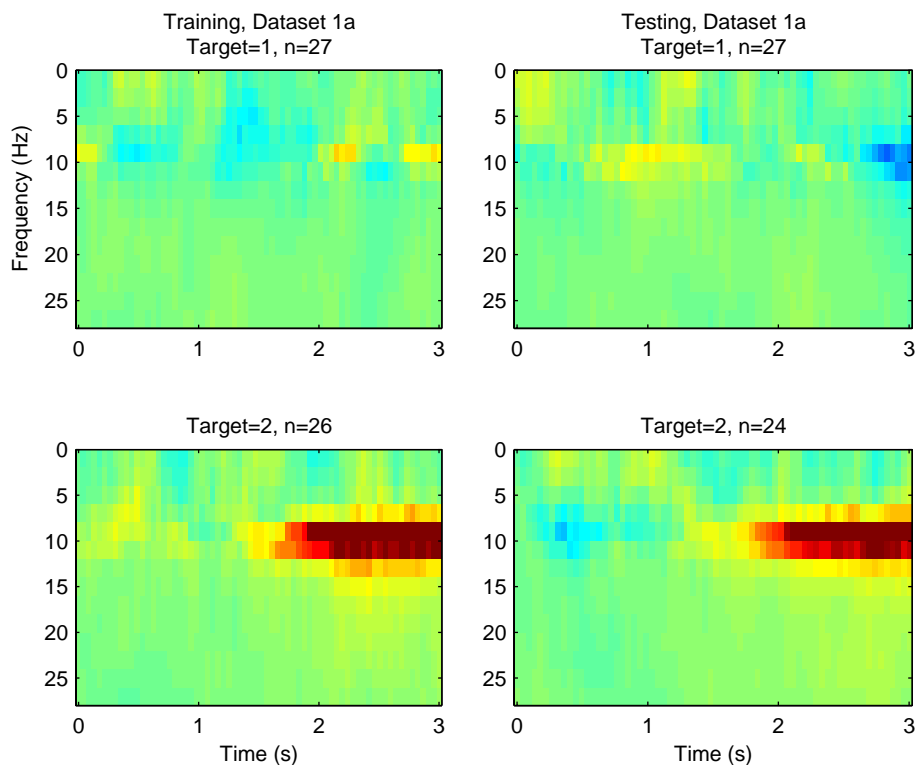


Figure B.1: Difference spectrogram for dataset 1a. Colorange: $[-5, 5]$.

	subject	targets	trials ^a	T^b	MI ^c	$\ r\ $	f_s	datafiles	features ^d	setup ^e
dataset 1a	JC	2	TR 56 (28,28) TE 57 (28,29)	3s. (48)	.043	.12	250Hz	JCS002R[6,8,9,10]	C3 - C4	MB1D
dataset 1b	JC	2	TR 56 (28,28) TE 57 (28,29)	3s. (48)	.043	.12	250Hz	JCS002R[6,8,9,10]	C3 ... C4	MB1D
dataset 2a	RH	4	TR 47 (11,12,12,12) TE 48 (12,12,12,12)	5s. (72)	.100	.13	250Hz	RHS004R0[3,4,8,10]	C1 - C2	MBB2D
dataset 2b	RH	4	TR 48 (12,12,12,12) TE 48 (12,12,12,12)	5s. (70)	.086	.10	250Hz	RHS004R0[2,5,7,9]	C2	MBB2D
dataset 2c	RH	4	TR 48 (12,12,12,12) TE 48 (12,12,12,12)	5s. (70)	.077	.11	250Hz	RHS004R0[1,2,4,5]	C3 - C4	MBB2D
dataset 2d	RH	4	TR 47 (11,12,12,12) TE 48 (12,12,12,12)	5s. (72)	.126	.13	250Hz	RHS004R0[3,4,8,10]	C2 ... C4	MBB2D
dataset 3a	O3VR	2	TR 159 (76,83) TE 161 (84,77)	4s. (62)	.019	.11	125Hz	O3VR-BCI2000	C3 - C4	MB1D
dataset 3b	X11b	2	TR 159 (76,83) TE 161 (84,77)	4s. (62)	.005	.04	125Hz	X11b-BCI2000	C3 - C4	MB1D
dataset 3c	S4	2	TR 232 (113,119) TE 308 (157,151)	4s. (62)	.005	.10	125Hz	S4b-BCI2000	C3 - C4	MB1D
dataset 4a	JC	2	TR 200 (100,100) TE 199 (100,99)	3s. (46)	.020	.10	250Hz	JCS005R[...]	C3 - F4	MB1D
dataset 4b	JC	2	TR 413 (206,207) TE 414 (206,208)	3s. (45)	.027	.14	250Hz	JCS006R[...] and JCS007R[...]	C3 - F4	MB1D
dataset 5a	RH	4	TR 120 (30,30,30,30) TE 119 (29,30,30,30)	5s. (77)	.043	.10	250Hz	RHS011R[...]	C3 - F4	MBB2D

Table B.1: Overview of the datasets used.

^aTR corresponds to the training set, TE to testset. Numbers between parentheses are the numbers of trials per target.

^bFeedback duration in seconds, and in parentheses the mean duration in the number of features vectors.

^cWe report the average mutual information and absolute correlation over all features used for the combined training and test set.

^dThe minus sign indicates the difference in power, ... indicates the concatenation.

^eMB1D: mu-based one-dimensional feedback. MBB2D: mu- and beta-based two-dimensional feedback.

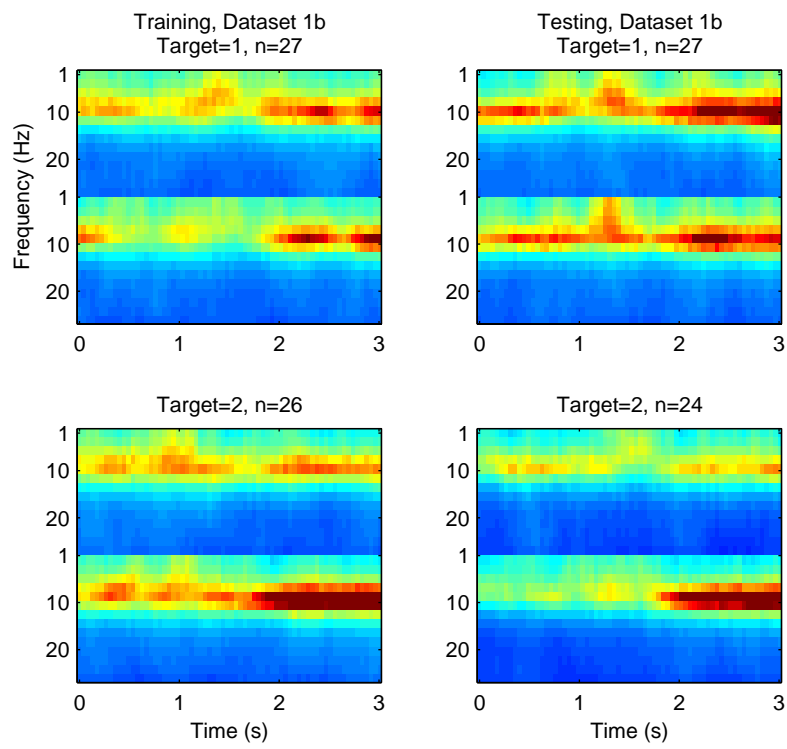


Figure B.2: Spectrogram for dataset 1b. Colorange: [0,10].

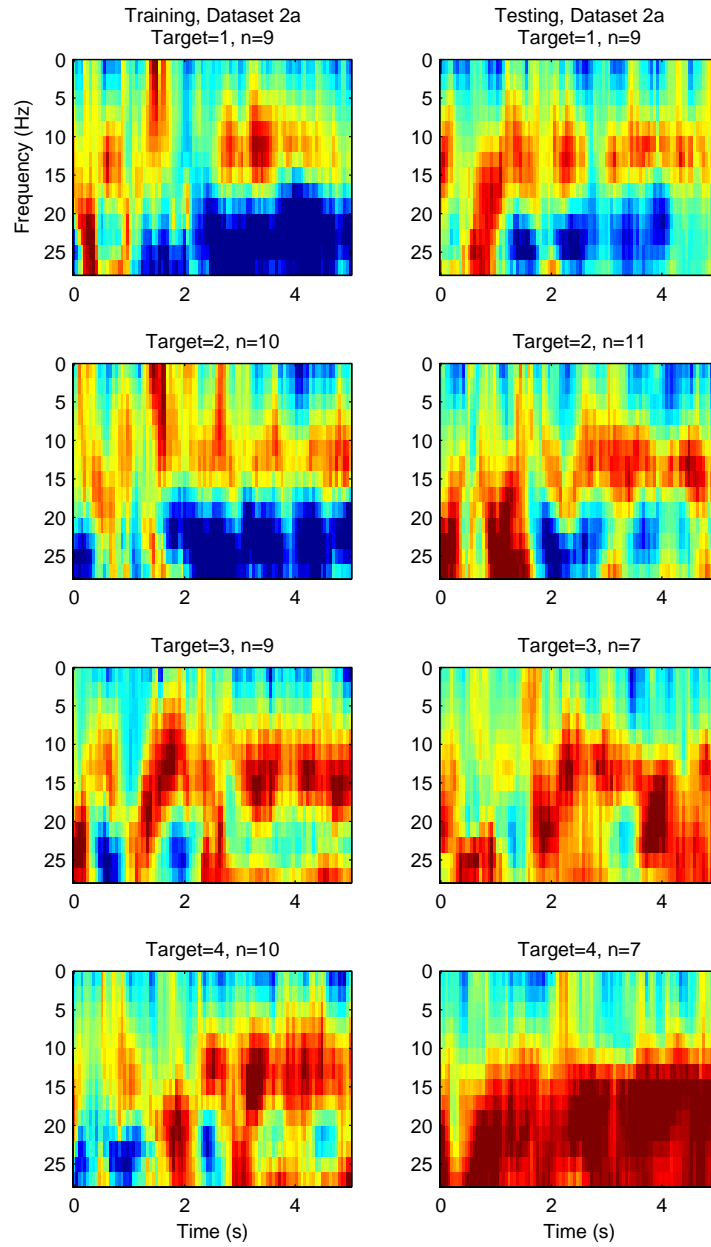


Figure B.3: Difference spectrogram for dataset 2a. Colorange: $[-5, 5]$.

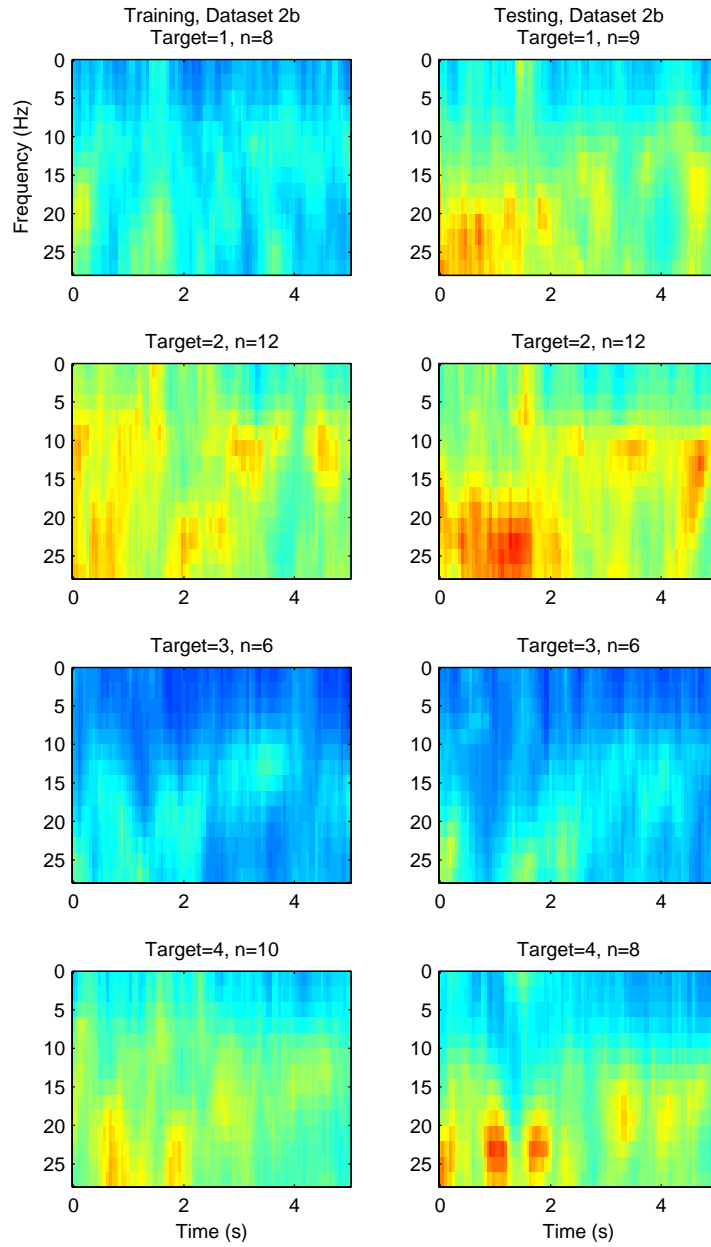


Figure B.4: Spectrogram for dataset 2b. Colorange: [0, 25].

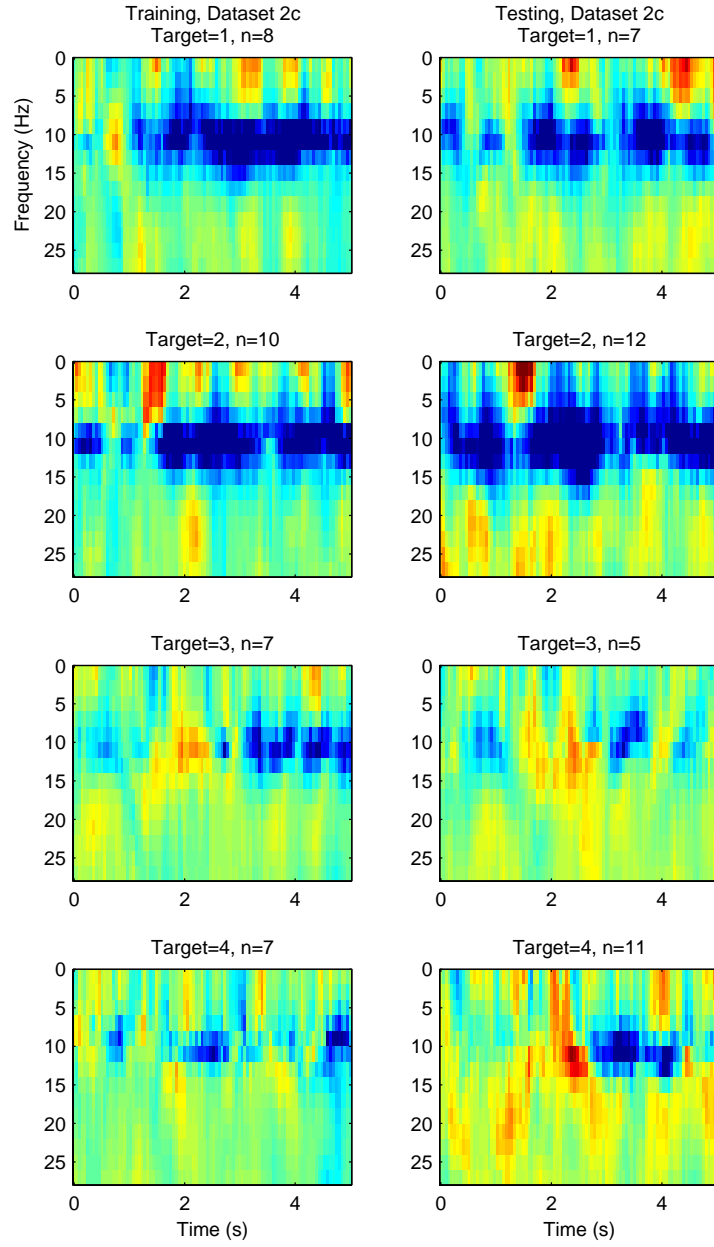


Figure B.5: Difference spectrogram for dataset 2c. Colorrange: $[-2.5, 2.5]$.

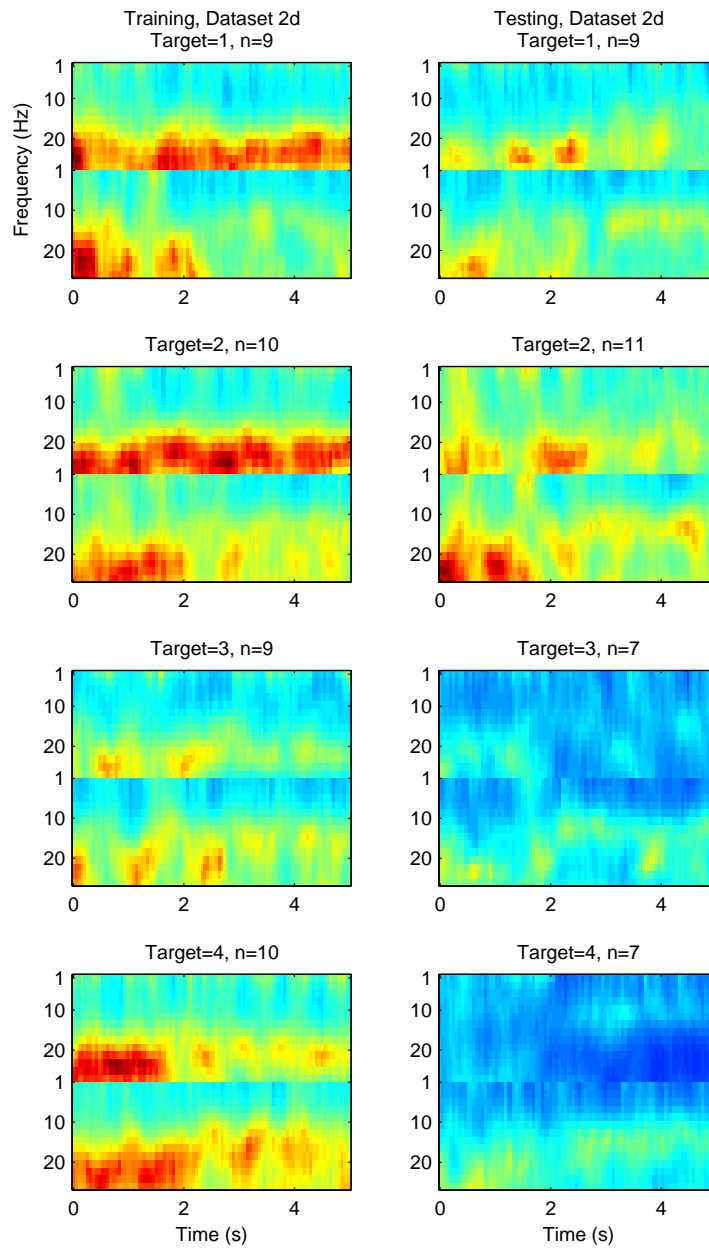


Figure B.6: Spectrogram for dataset 2d. Colorange: [0, 25].

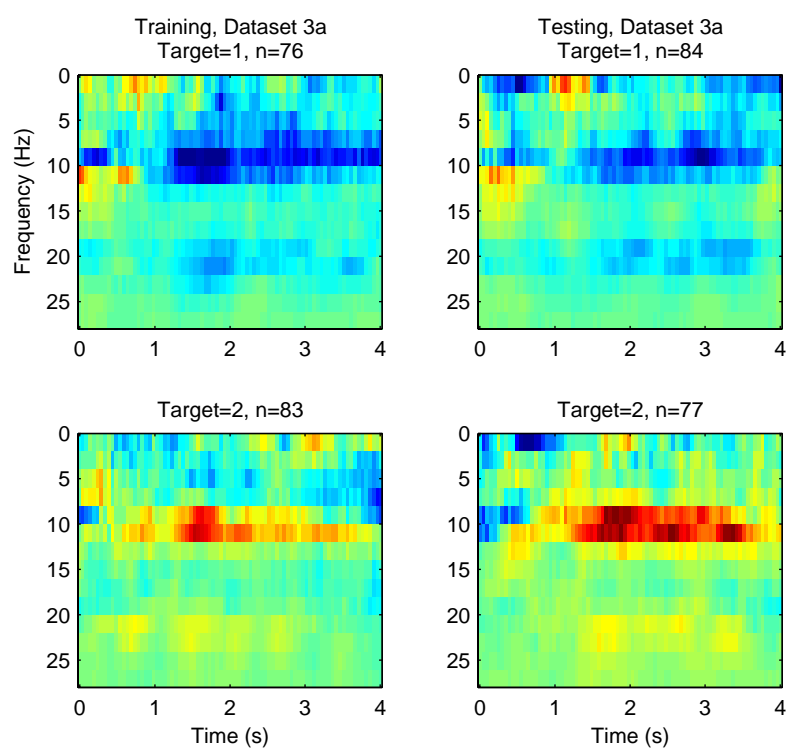


Figure B.7: Difference spectrogram for dataset 3a. Colorange: $[-1.5, 1.5]$.

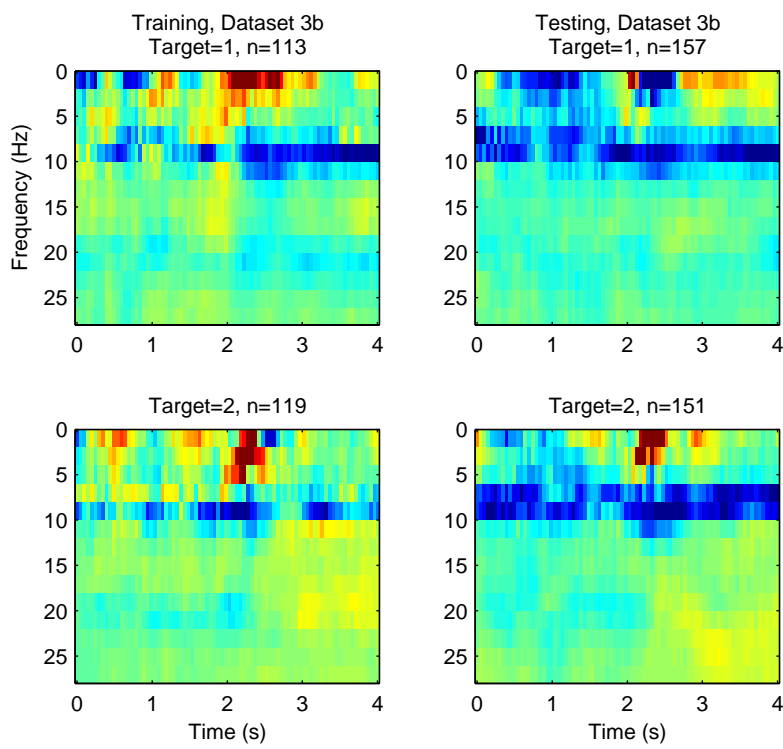


Figure B.8: Difference spectrogram for dataset 3b. Colorange: $[-1.5, 1.5]$.

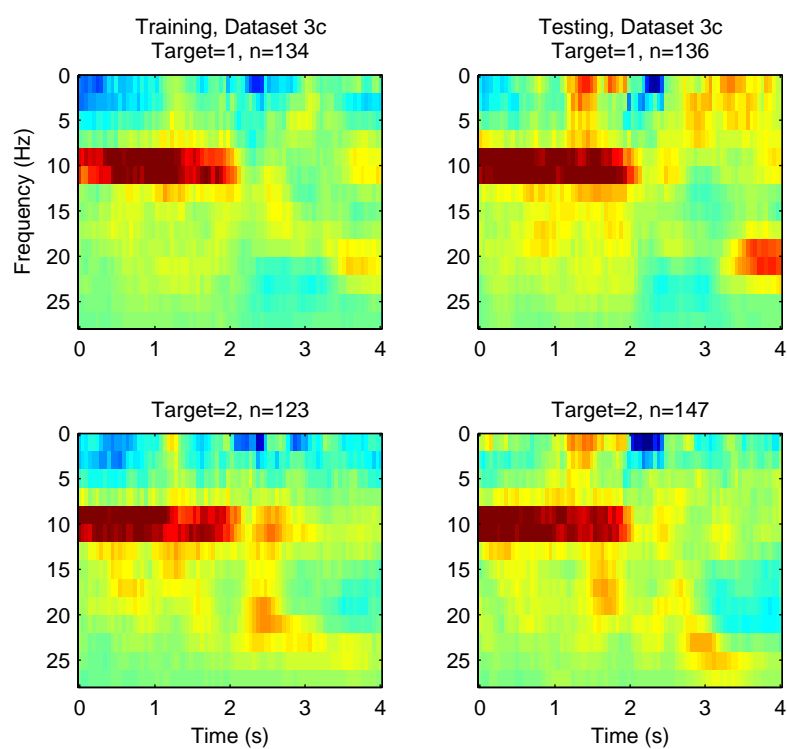


Figure B.9: Difference spectrogram for dataset 3c. Colorange: $[-1.5, 1.5]$.

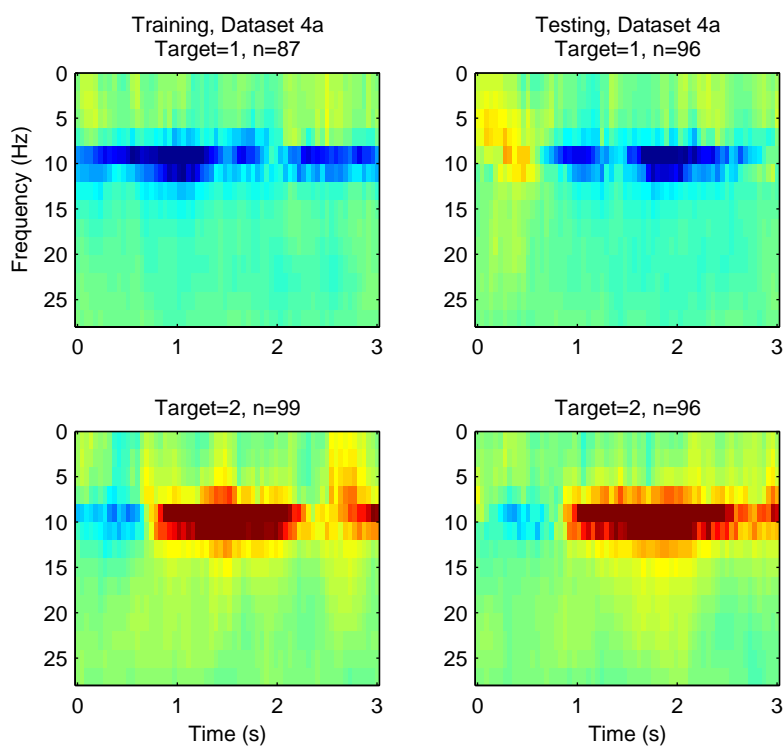


Figure B.10: Difference spectrogram for dataset 4a. Colorange: $[-20, 20]$.

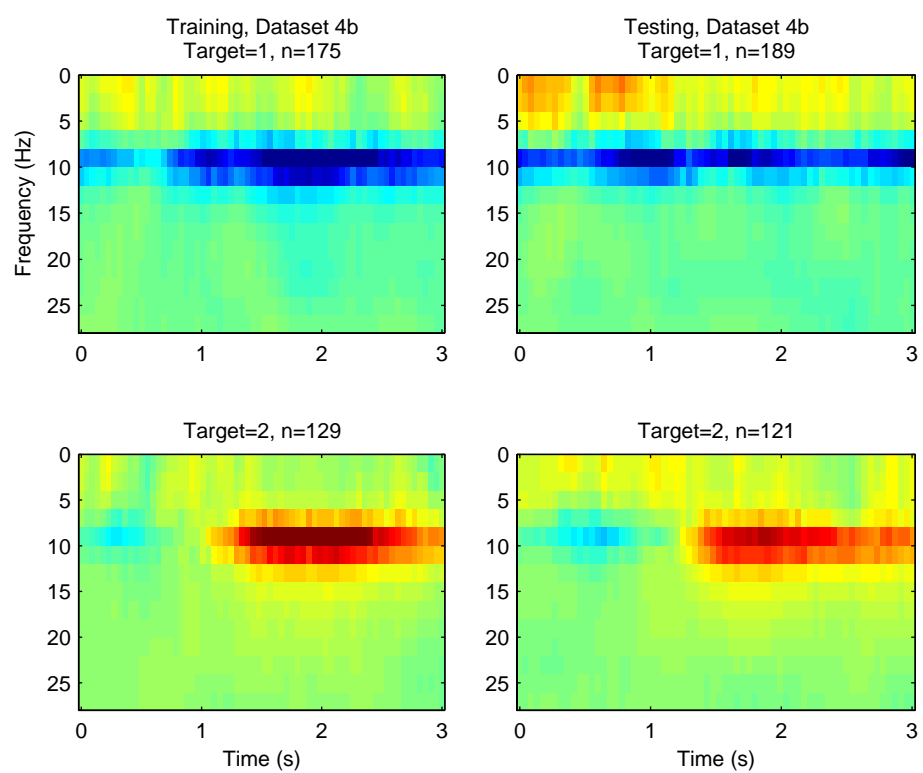


Figure B.11: Difference spectrogram for dataset 4b. Colorrange: $[-20, 20]$.

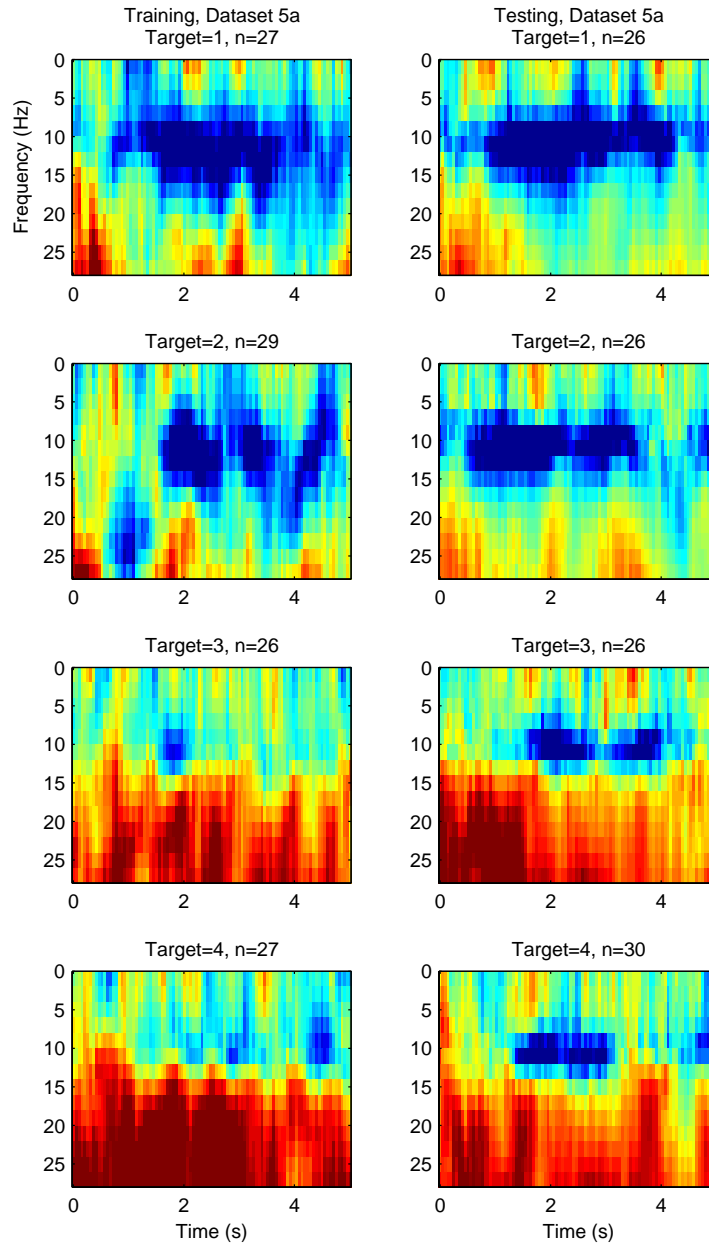


Figure B.12: Difference spectrogram for dataset 5a. Colorange: $[-15, 15]$.

List of Figures

2.1	Qualitative comparison of current neuro-imaging techniques with respect to their spatial and temporal resolution. A high resolution implies small discernible differences which is desirable.	12
2.2	The 'Homunculus' illustrates which part of the motor cortex innervates which part of the body. We refer to area's in the motor cortex as <i>representations</i> of the limb they innervate. (From Love & Webb 1992, p. 19)	14
2.3	The electrode locations according to the international 10-20 system. The horizontal line in the figure corresponds to the location of the motor cortex depicted in Figure 2.2.	16
2.4	The operation of a Laplacian spatial filter with the black electrode that of interest. For a Small and Large Laplacian, the average of the electrodes labeled <i>s</i> and <i>l</i> respectively is subtracted. In the 10-20 system only around the electrodes over the central gyrus (Cz and C1 to C8) a perfect grid as here exist. For other locations the electrodes must be placed at the appropriate distances by hand.	18
3.1	A Pattern Classification pipeline. Data acquiry typically represents an analog-to-digital converter or otherwise the link between sensor and computer. In pre-processing the signal is for example normalized through a form of detrending. Feature extraction requires knowledge about the input phenomena. Features represent the essential information from the input objects. Classification is then a rather straightforward decision function of the vector of features copmuted previously. In post-processing we may incorporate the classification result into a larger decision for example by combining multiple classifications into a single classification. In our case we use this to combine the instantaneous decisions during a trial into a single classification of a trial as a whole. . .	23
3.2	Task layout for motor imagery. The thin bar on the (in this case) left side of the middle screen indicates the desired side of motor imagery. The bar appears subsequently on either one of the two sides in a random order. The appearance indicates the onset of the task, the dissappearance the end of the task, no further information is provided to the subject through the screen.	27

- 3.3 **Task layout for feedback tasks.** The thin bar on the (in this case) right side is the target for the cursor to hit. This target appears subsequently on either one of the two sides in a random order. The target appearance is the cue for the subject to start the cognitive task. One second later the cursor appears and immediately starts moving. 27
- 3.4 **Target locations in 2D feedback task.** This figure illustrates the four target locations in the 2D feedback task. All targets are positioned along the left and right side of the screen. 28
- 3.5 Example difference spectrograms. The panels show the difference in spectral power between electrodes C3 and C4 during four typical trials from conditions L (left target) and R (right target). The differences lie between +10 ($C4 > C3$; red cells), and -10 ($C4 < C3$; blue cells). The graphs give a clear indication for time structure in the data. Also note the frequency specificity of the structure. (*Data is Subject JC, session 2.2, run 1, trials 7, 13, 16 and 23.*) 29
- 3.6 Example average difference spectrogram. The graph shows the average difference in power between electrodes C3 and C4. Again red cells indicate $C4 > C3$ and blue cells vice versa. The upper graph is the average over all left trials, whereas the lower over the right trials. The trial starts at $t = 0$, corresponding to the vertical black line, negative time corresponds to the resting prior to the trial for contrast. (*Data is Subject JC, session 1: Ball-clench imagery runs 1 to 5. The average is over the two disjoint subsets corresponding to target code (both $n = 82$).*) 30
- 3.7 Example graph of the correlation between power in frequency bands and the side (left or right) of motor imagery. The different lines correspond to distinct spatial locations. In this example best performance is found around 10Hz for spatial location C4 with $r \approx 0.2$. Note that C3 and C4 on opposite sides of the scalp correlate inversely. For details the reader is referred to the text. (*Data is subject JC, session 2.2, runs 1 to 5. Correlation is computed over $n = 6760$ (142 trials \times 47.6 points per trial) points)*) 31
- 3.8 The mutual information between the observed spectral powers in different frequency bands, and the side of motor imagery. One bit of information would imply perfect information about the intended side of motor imagery. In this figure C4 bears the most information. (*Data is Subject JC, Session 2.2, runs 1 to 5. $n = 6760$ (142 trials \times 47.6 points per trial) for every frequency band, 250 bins in histogram to estimate the pdf.*) 33
- 3.9 This graph shows the evolution of the correlation with the side of motor imagery over time *within a trial* for the two electrodes over the hand representations. Note the variability over time and the qualitative difference between the graphs. The dotted line denotes the level of significance at $\alpha = .05$ for these correlations. (*Data is Subject JC, Session 2.2, runs 1 to 5. $n = 142$ trials for every point.*) 34

- 4.1 This figure illustrates different types of time structure. The graphs show observations from three fictitious processes with different types of time structure. There are two conditions of the process denoted \circ and \diamond . Of each condition of each process we observe one time series of two features x_1 and x_2 . The process advances through a series of underlying states over time which results in the clusters of points. The time relation between individual points is omitted, the graph only shows the transitions between clusters over time through the arrows. The reader is referred to the text for details. 40
- 4.2 This Figure shows exemplary QQ-plots of spectral powers (Fig. A-C) and differences in spectral power (Fig. D-F) versus theoretical Normal and chi-square distributions. The parameters of the theoretical distributions are estimated from the data. We estimated the number of degrees of freedom for the chi-square distribution as the mean (Fig. B, E) and as half the variance (Fig. C, F) of the data. The dashed line represents points where empirical and theoretical quantiles correspond. Ideally, all pluses lie on this line. *Data is Subject JC, Session 2.2, Runs 1 to 5. For A.-C. Channel 3 at 10Hz. For D.-F. Channel 3 minus 7 both at 10Hz. Graphs are subject to small variations due to random initialization of the theoretical distribution.* 43
- 4.3 An example distribution of the cluster centers for two models (one for condition \circ and \diamond) with the appropriate number of states as might be estimated based on the data from Figure 4.1. The circles are centered at the mean, the radius indicates the standard deviation. The heterogeneous clusters should overlap perfectly in this example in theory. The slight differences in this case are man made and meant to illustrate the issues in practice. 44
- 4.4 This Figure gives a schematic overview of the two steps in the training of the Common-Structure HMM for the example data from panel B. in Figure 4.1. The thickness of arrows represent the transition probability; a thick arrow indicates it is probable to make that transition, whether a thin line indicates that transition is improbable. The reader is referred to the text for details. . . . 47
- 5.1 An overview of the performance of the three models on artificial data. The panels from left to right correspond to performance on linearly-separable data, non-linearly separable data and time-based separable data. Mean accuracy and standard deviation around the mean are estimated from 3-fold cross validation, with three random initializations of the model per fold. Error bars indicate the standard deviation around the mean performance. . . 52
- 5.2 An overview of the performance of HMM set of to instantaneous classifiers on the data sets with one-dimensional tasks. Data set 3c is omitted because the graph is very similar to that of data set 3b. 53

5.3	The performance on the two tasks comprising two-dimensional control. For scoring of the trials only information in one dimension is considered. In this case, should the target be the upper left (see Figure 3.4), but the trial is classified as upper right it is considered correctly, since the up/down distinction was correct. We see that the left/right distinction is classified comparable to the one-dimensional task, whereas the vertical task scores no higher than chance ($= 0.5$).	55
5.4	The performance on data set 1. The difference between the two is the feature used for learning and classification. Data set 1a uses the difference in spectral power between two electrodes, whereas 1b uses the spectra of both electrodes.	56
5.5	The evolution of performance for reduced training data. If we reduce the number of training data, we expect the performance to degrade. The horizontal axis indicates the number of trials available for training. The vertical axis indicates the mean accuracy from 4-fold cross validation for models with four states (HMM and Flat HMM), or eight states (CSHMM) or 17 neighbors (kNN). <i>Data is data set 4b. The mean length of a trial is 45 feature vectors.</i>	57
5.6	The performance of all three models on the larger training sets. The horizontal lines display the performance for comparison to instantaneous classification. There are little differences between methods and between numbers of states. The reader is referred to the text for the analysis.	58
B.1	Difference spectrogram for dataset 1a. Colorange: $[-5, 5]$	75
B.2	Spectrogram for dataset 1b. Colorange: $[0, 10]$	77
B.3	Difference spectrogram for dataset 2a. Colorange: $[-5, 5]$	78
B.4	Spectrogram for dataset 2b. Colorange: $[0, 25]$	79
B.5	Difference spectrogram for dataset 2c. Colorange: $[-2.5, 2.5]$	80
B.6	Spectrogram for dataset 2d. Colorange: $[0, 25]$	81
B.7	Difference spectrogram for dataset 3a. Colorange: $[-1.5, 1.5]$	82
B.8	Difference spectrogram for dataset 3b. Colorange: $[-1.5, 1.5]$	83
B.9	Difference spectrogram for dataset 3c. Colorange: $[-1.5, 1.5]$	84
B.10	Difference spectrogram for dataset 4a. Colorange: $[-20, 20]$	85
B.11	Difference spectrogram for dataset 4b. Colorange: $[-20, 20]$	86
B.12	Difference spectrogram for dataset 5a. Colorange: $[-15, 15]$	87

Bibliography

- [1] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan, “Brain-computer interface technology: A review of the first international meeting,” *IEEE Transactions on rehabilitation engineering*, vol. 8, pp. 164–173, 2000.
- [2] J. R. Wolpaw and D. J. McFarland, “Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 51, pp. 17849–17854, 2004.
- [3] N. Birbaumer, N. Ghanayim, T. Hinterberger, I. Iversen, B. Kotchoubey, A. Kübler, J. Perelmouter, E. Taub, and H. Flor, “A spelling device for the paralyzed,” *Nature*, vol. 398, pp. 297–298, 1999.
- [4] Y. Kamitani and F. Tong, “Decoding the visual and subjective contents of the human brain,” *Nature Neuroscience*, vol. 8, no. 5, pp. 679–685, 2005.
- [5] T. M. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman, “Learning to decode cognitive states from brain images,” *Machine Learning*, vol. 57, pp. 145–175, 2004.
- [6] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O’Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, “Learning to control a brain-machine interface for reaching and grasping by primates,” *PLoS Biology*, vol. 1, no. 2, pp. 1–16, 2003.
- [7] O. G. Patil, J. M. Carmena, M. A. Nicolelis, and D. A. Turner, “Ensemble recordings of human subcortical neurons as a source of motor control signals for a brain-machine interface,” *Neurosurgery*, vol. 55, pp. 27–38, 2004.
- [8] A. K. Engel, C. K. E. Moll, I. Fried, and G. A. Ojemann, “Invasive recordings from the human brain: clinical insights and beyond,” *Nature Reviews Neuroscience*, vol. 6, no. 1, pp. 35–47, 2005.
- [9] T. E. Gladwin, *Time-frequency domain EEG activity during the preparation of task sets and movements*. PhD thesis, Rijksuniversiteit Groningen, 2006.
- [10] E. Curran, P. Sykacek, M. Stokes, S. Roberts, W. Penny, I. Johnsrude, and A. Owen, “Cognitive tasks for driving a brain-computer interfacing system: a pilot study,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 12, no. 1, pp. 48–54, 2003.

- [11] B. Blankertz, G. Dornhege, M. Krauledat, K. R. Müller, and G. Curio, "The berlin brain-computer interface: report from the feedback sessions," tech. rep., Fraunhofer institut für Rechnerarchitektur und Softwaretechnik, 2005.
- [12] G. Pfurtscheller, C. Neuper, G. R. Müller, B. Obermaier, G. Krausz, A. Schlögl, R. Scherer, B. Graimann, C. Keinrath, D. Skliris, M. Wortz, G. Supp, and C. Schrank, "Graz-bci: state of the art and clinical applications," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 11, pp. 177–180, 2003.
- [13] W. Bechtel, ed., *Philosophy and the neurosciences: a reader*. Blackwell Publishers, Malden, MA, 2001.
- [14] E. Kandel, J. H. Schwartz, and T. M. Jessell, *Principles of neural science*. McGraw-Hill, New York, NY, 2000.
- [15] D. J. McFarland, L. A. Miner, T. M. Vaughan, and J. R. Wolpaw, "Mu and beta rhythm topographies during motor imagery and actual movements," *Brain Topography*, vol. 12, no. 3, pp. 177–186, 2000.
- [16] J. R. Wolpaw, D. J. McFarland, G. W. Neat, and C. A. Forneris, "An eeg-based brain-computer interface for cursor control," *Electroencephalography and clinical Neurophysiology*, vol. 78, pp. 252–259, 1991.
- [17] P. A. Lynn and W. Fuerst, *Introductory digital signal processing with computer applications*. John Wiley and Sons Ltd., Chichester, UK, 1998.
- [18] G. Pfurtscheller and F. H. L. da Silva, "Event-related eeg/meg synchronization and desynchronization: basic principles," *Clinical Neurophysiology*, vol. 110, pp. 1842–1857, 1999.
- [19] C. Neuper and G. Pfurtscheller, "Event-related dynamics of cortical rhythms: frequency-specific features and functional correlates," *International journal of psychophysiology*, vol. 43, pp. 41–58, 2001.
- [20] G. N. G. Molina, T. Ebrahimi, and J. M. Vesin, "Joint time-frequency-space classification of eeg in a brain-computer interface application," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 713–729, 2003.
- [21] T. C. Andringa, *Continuity preserving signal processing*. PhD thesis, Rijksuniversiteit Groningen, 2001.
- [22] G. Dornhege, B. Blankertz, G. Curio, and K. R. Müller, "Combining features for bci," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 15, 2003.
- [23] S. Haykin, *Adaptive filter theory*. Information and system sciences series, Prentice Hall, Englewood Cliffs, NJ, 1986.
- [24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The art of scientific computing*. Cambridge University Press, 1986.

- [25] D. J. McFarland, L. M. McCane, S. V. David, and J. R. Wolpaw, "Spatial filter selection for eeg-based communication," *Electroencephalography and clinical Neurophysiology*, vol. 103, pp. 386–394, 1997.
- [26] B. Blankertz, K. R. Muller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroder, and N. Birbaumer, "The bci competition 2003: Progress and perspectives in detection and discrimination of eeg single trials," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1044–1051, 2004.
- [27] P. Sajda, A. Gerson, K. R. Muller, B. Blankertz, and L. Parra, "A data analysis competition to evaluate machine learning algorithms for use in brain-computer interfaces," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 11, no. 2, pp. 184–185, 2003.
- [28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley Interscience, New York, NY, 2nd ed., 2001.
- [29] R. F. Tate, "Correlation between a discrete and a continuous variable. point-biserial correlation," *The Annals of mathematical statistics*, vol. 25, pp. 603–607, 1954.
- [30] D. J. McFarland and J. R. Wolpaw, "Sensorimotor rhythm-based brain-computer interface (bci): feature selection by regression improves performance," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 13, no. 3, pp. 372–379, 2005.
- [31] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [32] S. Zhong and J. Ghosh, "Hmms and coupled hmms for multi-channel eeg classification," in *International Joint Conference on Neural Networks (IJCNN)*, 2002.
- [33] W. D. Penny and S. J. Roberts, "Gaussian observation hidden markov models for eeg analysis," tech. rep., Neural systems research group, Imperial College of Science, Technology and Medicine, 1998.
- [34] T. Hinterberger, N. Neumann, M. Pham, A. Kübler, A. Grether, N. Hofmayer, B. Wilhelm, H. Flor, and N. Birbaumer, "A multimodal brain-based feedback and communication system," *Experimental Brain Research*, vol. 154, pp. 521–526, 2004.
- [35] G. Dornhege, B. Blankertz, G. Curio, and K. R. Müller, "Increase information transfer rates in bci by csp extension to multi-class," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 16, 2004.
- [36] P. Sykacek, S. Roberts, and M. Stokes, "Adaptive bci based on variational bayesian kalman filtering: an empirical evaluation," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 5, pp. 719–727, 2004.

- [37] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "Bci2000: A general-purpose brain-computer interface system," *IEEE Transactions on biomedical engineering*, vol. 51, no. 6, pp. 1034–1042, 2004.
- [38] G. Schalk, T. Hinterberger, D. J. McFarland, and J. Mellinger, *Software design document for a specific implementation of BCI2000*, July 2004.
- [39] J. R. Wolpaw and D. J. McFarland, "Multichannel eeg-based brain-computer communication," *Electroencephalography and clinical Neurophysiology*, vol. 90, pp. 444–449, 1994.
- [40] G. Pfurtscheller, C. Neuper, C. Brunner, and F. L. da Silva, "Beta rebound after different types of motor imagery in man," *Neuroscience Letters*, vol. 378, pp. 156–159, 2005.
- [41] C. Neuper, R. Scherer, M. Reiner, and G. Pfurtscheller, "Imagery of motor actions: differential effects of kinesthetic and visual-motor mode of imagery in single-trial eeg," *Cognitive Brain Research*, vol. 25, no. 3, pp. 668–677, 2005.
- [42] T. Lan, D. Erdogmus, A. Adami, and M. Pavel, "Feature selection by independent component analysis and mutual information maximization in eeg signal classification," in *Proceedings of International Joint Conference on Neural Networks*, 2005.
- [43] A. Schlögl, C. Neuper, and G. Pfurtscheller, "Estimating the mutual information of an eeg-based brain-computer interface," *Biomedizinische Technik*, vol. 47, pp. 3–8, 2002.
- [44] P. McCullagh and J. A. Nelder, *Generalized Linear Models*. Chapman & Hall, London, second ed., 1992.
- [45] N. Christianini and J. Shawe-Taylor, *An introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2001.
- [46] A. Dempster, N. M. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, 1977.
- [47] Y. Ephraim, A. Dembo, and L. R. Rabiner, "A minimum discrimination information approach for hidden markov modeling," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87*, vol. 12, pp. 25–28, 1987.
- [48] B. H. Juang and L. R. Rabiner, "Hidden markov models for speech recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.
- [49] D. Garrett, A. Peterson, C. W. Anderson, and M. H. Thaut, "Comparison of linear, nonlinear and feature selection methods for eeg signal classification," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 11, pp. 141–144, 2003.

- [50] E. Haselsteiner and G. Pfurtscheller, "Using time-dependent neural networks for eeg classification," *IEEE Transactions on rehabilitation engineering*, vol. 8, pp. 457–463, 2000.
- [51] G. Garcia and T. Ebrahimi, "Time-frequency-space kernel for single eeg-trial classification," in *Proceedings of the NORSIG conference*, 2002.
- [52] J. Mellinger, *BCI2000 Command Line Interface*, July 2005. Part of BCI2000 documentation.