Reinmar Josef Kobler, BSc

Transfer Learning with Restricted Boltzmann Machines for an online sensorymotorrhythm Brain-Computer Interface

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieur

Master's degree programme
Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Ass.Prof. Dipl.-Ing. Dr.techn. Reinhold Scherer
Institute for Knowledge Discovery, Laboratory of Brain-Computer Interfaces

Head of Institute

Univ.-Prof. Dipl.-Ing. Dr. techn. Gernot R. Müller-Putz
Institute for Knowledge Discovery, Laboratory of Brain-Computer Interfaces

Graz, September 2015

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

The text document uploaded to TUGRAZonline is identical to the presented master's thesis dissertation.

Place	Date	Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Ort	Datum	Unterschrift

Acknowledgments

This work has been carried out at the Institute for Knowledge Discovery as part of the Master's program in Information and Computer Engineering in Graz, Austria as well as part of a 2 month internship at the Brain Imaging Technology Laboratory in Kobe, Japan.

At this point I would like to thank all those people who supported me kindly and answered all my questions with great pleasure during the last months. I am very grateful for this terrific company.

Foremost I would like to thank Dr. Reinhold Scherer. Without him this thesis would not have been possible. He ignited my interest in this topic and brought me back on track whenever I was about to drift away. His support was also of great value when I was planning to go to Japan.

There, Dr. Naruse, Head of the Brain Imaging Technology Lab, let me not only work on this thesis but also supported me with valuable hints and hardware to conduct studies. For all this a special thank you.

Back in Graz I would also like to thank Dr. Müller-Putz, Head of the Institute for Knowledge Discovery, for providing the necessary hardware and support so that I was able to conduct final experiments.

Lastly, I also want to express my sincere gratitude to my partner, parents, siblings and friends for their unconditional support enabling me to focus on this work.

Abstract

Calibration of a Sensory Motor Rhythm (SMR) based Brain-Computer Interface (BCI) is often found to be tedious and lengthy by naive users. The lack of feedback during training is assumed to be a major cause. This thesis addressed the issue by utilizing a generative model entitled Restricted Boltzmann Machine (RBM). We show that it is capable of extracting representations from logarithmic bandpower features which generalize well across users. That is to say a pre-trained *RBM* can be transfered to a new user and, consequently, feedback-training started immediately.

Although the model was not trained on an individual's patterns, a mean accuracy of 73% with a standard-error of 4% was achieved for a two-class problem covering 9 participants in a simulated experiment.

In a subsequent online experiment discriminative fine-tuning of a pre-trained *RBM* yielded even significantly better results. 8 of the 10 naive users reached a criterion level above 70% within a single co-adaptive training-session. The median accuracy achieved was 84% with a standard-error of 7%. Faller et al. 2012 reached $80 \pm 3\%$ in 2-3 sessions with a comparable co-adaptive system.

In this thesis, the online training lasted for approximately 45 minutes. Feedback was already presented after a 1 minute setup stage, whose purpose was to estimate initial statistics and train an online artifact detection system.

Keywords. transfer learning, co-adaptive training, restricted boltzmann machines, artificial neural networks, online brain computer interface

Kurzfassung

Das Kalibrieren von Gehirn-Computer Schnittstellen, die auf Modulationen von Oszillationen basieren, wird von neuen Probanden oft als mühsam und langwierig empfunden. Es wird angenommen, dass das Fehlen von Feedback beträchtlich dazu beiträgt. Diese Arbeit adressiert das Problem durch die Verwendung eines generativen Modells mit der Bezeichnung Restricted Boltzmann Machine (*RBM*). Wir zeigen dass dieses Modell Muster aus logarithmischen Bandleistungs-Features von Signalen mehrere Benutzer extrahiert, die gut generalisieren. Das heißt, eine vor-trainierte *RBM* kann auf neue Benutzer transferiert und daher auch sofort mit Feedback-Training begonnen werden.

Obwohl das Modell nicht auf Daten eines Individuums trainiert wurde, hat ein simuliertes Experiment für 2 Klassen basierend auf Daten von 9 Benutzern eine mittlere Genauigkeit von 73% mit einem Standardfehler von 4% ergeben.

Darüber hinaus konnte in einem nachfolgenden online Experiment gezeigt werden, dass diskriminatives Adaptieren der vor-trainierten *RBM* zu einem signifikant besseren Ergebnis führt. 8 von 10 naiven Probanden erreichten eine Genauigkeit von über 70% in einer co-adaptiven Trainings-Sitzung. Der Median beträgt $84 \pm 7\%$. In Faller et al. 2012 wurde für ein vergleichbares co-adaptives System ein Median von $80 \pm 3\%$ in 2-3 co-adaptiven Trainings-Sitzungen erreicht.

Für diese Arbeit betrug die tatsächliche Trainingszeit circa 45 Minuten pro Proband. Feedback wurde bereits nach einem 1 minütigen Setup, während dem initiale Statistiken geschätzt und ein Artefakt-Detektions System trainiert wurde, präsentiert.

Stichwörter. Transfer-Learning, co-adaptives Training, Restricted Boltzmann Machines, künstliche neuronale Netzwerke, online Gehirn-Computer Schnittstelle

Contents

List of Figures

List of Tables

List of Acronyms

AAR	Adaptive Autoregressive
ALS	Amyotrophic Lateral Sclerosis
ANN	Artificial Neural Network
BCI	Brain-Computer Interface
CD	Contrastive Divergence
CSP	Common Spatial Patterns
EEG	Electroencephalography
EMG	Electromyography
EOG	Electrooculography
ERD	Event-Related Desynchronization
ERDS	Event Related (De-)Synchronization
ERP	Event-Related Potential
ERS	Event-Related Synchronization
FIR	Finite Impulse Response
GBRBM	Gaussian-Bernoulli Restricted Boltzmann Machine
IIR	Infinite Impulse Response
PCD	Persistent Contrastive Divergence
PT	Parallel Tempering
RBM	Restricted Boltzmann Machine
RLS	Recursive Least Squares
sLDA	Shrinkage Linear Discriminant Analysis
SMR	Sensory Motor Rhythm

1 Introduction

Contents

Breakthroughs in cognitive neuroscience and brain imaging technologies enabled us with the ability to interface directly with the human brain. This is made possible through the use of sensors that can monitor physiological processes that occur within the brain and correspond with certain aspects of intent. In the last decades researchers have used these technologies to build Brain-Computer Interface (BCI). They are communication systems that do not depend on the brain's normal output pathways of peripheral nerves and muscles. In these systems, users explicitly manipulate their brain activity instead of using motor movements to produce signals that can be used to control computers or communication devices [46].
The described systems are often the only means of communication for individuals with severe motor disabilities such as Amyotrophic Lateral Sclerosis (ALS). [30]. Consequently, research groups worldwide are working on systems with better performance in terms of accuracy, convenience, reliability and robustness.

1.1 Background and Related Work

Back in 1924 Dr. Berger discovered the Electroencephalography (EEG). It is defined as electrical potential, recorded at the scalp, which is caused by physiological processes in the brain. He was also first to observe different rhythms present in the *EEG* and their modulation through closing the eyes for example [8]. Since then there has been increasing research on the *EEG* and ways to decode intent.

In the 1990s Pfurtscheller et al.[37, 38] as well as Wolpaw et al.[35, 54] worked extensively on methods to measure changes in the *EEG* during execution and later *imagination* of specific movements. Pfurtscheller et al. entitled the observed task-dependent changes in power of frequency bands Event Related (De-)Synchronization (ERDS) [38]. Their work yielded the first so called Sensory Motor Rhythm (SMR) *BCI*s [39]. Since then scientists put effort in improving them to get the technology out of the lab-environment to real-life situations. The common aim lead to increased collaboration among the labs and a general definition of the parts a *BCI* consists of [53]. Their interplay is depicted in Figure 1.1. The *Signal Acquisition* block deals with the recording of brain signals such as the *EEG*. Signal processing tools are employed to extract useful information in the *Feature Extraction* stage. In the *Feature Translation* block they are translated into a control signal for the *Application*, which provides *Feedback* to the *User*.

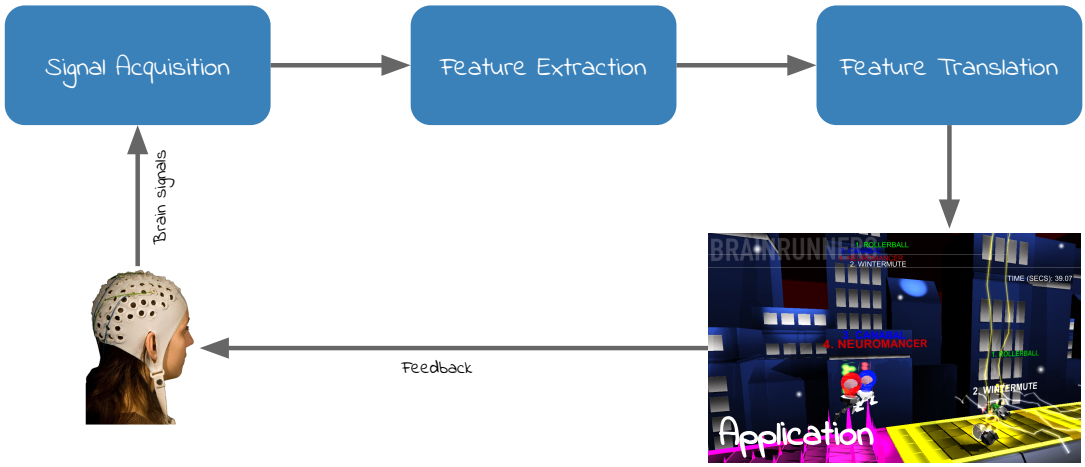


Figure 1.1: Building blocks of a *BCI* system. All components starting from the user, who modulates brain signals, up to the application, which provides feedback, form a closed loop system. Screenshot of the application taken from the brainrunners game designed for [41].

Still, there are several hurdles to master. The following gives a selection of current limitations formulated in [1]:

1. **Bandwidth:** The number of reliably distinguishable classes for non-invasive BCIs based on brain oscillations is low (2-4) and varies between subjects and time [21].

2. **BCI inefficiency:** There is a fraction of up to 30 % of users for which the necessary *ERDS* patterns cannot be detected within *EEG* [2, 50]. Scientists have not identified all reasons for this phenomenon yet.

3. **Training:** Brain-oscillations based *BCI*s initially require recording sessions during which the participant does not get any feedback. In other words, the user is unaware

of how well he/she is performing and might lose focus and attention. According to [49] this contributes also to the *BCI* inefficiency issue.

In recent years there has been extensive research on increasing the bandwidth. Algorithms, intended for two class problems, such as Common Spatial Patterns (CSP) were extended to multi-class versions [20] and numerous classifiers, inherently supporting multiple classes, were applied [34, 45].

Other researchers focus on the exploration of mental tasks alternative or additional to motor imagery [17, 19] to tackle the first and second challenge. The BCI inefficiency itself is attracting rising attention. Scientists try to find predictors for BCI performance to identify possible reasons [13, 22].

The community puts also effort in pushing the training time down. It's only lately that impressive results were achieved with co-adaptive systems by Faller et al. [18]. They proposed an adaptive system which employs only very basic feature extraction methods. The key idea was to present feedback already after minutes of auto-calibration. The result: 10 out of 12 naive participants were able to operate a 2-class system above 70 % accuracy after 2 to 3 training-session.

Another very recent approach is to transfer classifiers, learned on data recorded from multiple users, to a new user. It has been successfully applied to an Event-Related Potential (ERP) based BCI [28] and also to some extent to *SMR* based BCIs [51]. In [51] Vidaurre et al. argue that this approach could even reduce BCI inefficiency, since some users have difficulty in performing well in the absence of feedback. Their reported results support this view since 5 out of 10 users who were unable to control a *BCI* before exceeded the 70 % threshold within a single training-session.

This thesis aims to exploit a generative classifier to extract representations of a mixture of users. It is then transfered and fine-tuned on an individual in a co-adaptive training-session. The experimental results, discussed in detail below, match the findings of Faller et al. [18] and Vidaurre et al. [51] suggesting that immediate feedback and continuous adaption not only shortens training-time but also contributes to a reduction of BCI inefficiency. Furthermore, the transfer learning approach enables researchers to exploit previously recorded data and might, therefore, lever up the model complexities as well as the information transfer rate of *BCI*s.

To conclude, the benefits of transfer learning across users are: (1) One is able to utilize much more data for training the classifier. Consequently, more complex models can be trained without risking severe overfitting. (2) The training time during which the user does not receive feedback can be skipped because the pre-trained system is used initially. (3) Adapting the classifier to a new individual allows to either stay close to the prior[1] if the data is hard to fit or change to a specific model if the new data is good to fit.

[1] Model learned from other users.

1.2 Structure of the Thesis

The remainder of this thesis is divided into five chapters. The first 3 present studies whose results were prerequisites for the design of the online system discussed in detail in chapter 5.

The next chapter answers the question whether the generative Restricted Boltzmann Machine (RBM) introduced in section 1.4.4 is able to perform as well as two standard machine learning approaches used in BCI-literature when it is trained only on an individual user's data[2]. This is assessed with a publicly available dataset of recorded *EEG* signals of 9 subjects. As expected the major finding was that the discriminative models are superior.

In chapter 3 the concept of transfer learning is exploited to enlarge the data available for training. Based on the same dataset the question which classifier generalizes best on data of a previously unseen user is answered. Here, the generative training-criterion of the *RBM* yielded best results.

Chapter 4 questioned whether the weights and biases learned by the *RBM* in transfer mode are better seeds than random ones. Consequently, the *RBM* of chapter 3 was discriminatively fine-tuned to the individual users data as described in section 1.4.5. Using the *RBM*'s weights and biases yielded higher mean performance and less variation across users.

All findings of the offline studies were considered during the development of the online, adaptive *BCI*. Its underlying paradigm, methods as well as results of a co-adaptive training study covering 12 participants are summarized in chapter 5. The very last chapter is intended to present general conclusions and potential future work.

[2] Only dozens of observations available.

1.3 Contributions of the Thesis

Before presenting the basic concepts of *RBM*s in the next section, the major contributions of this thesis to the field of non-invasive *BCI*s are highlighted.

Firstly, in [4] the applicability of *RBM*s on detecting oscillatory *EEG* components was shown in an offline study. *RBM*s are known to require datasets with many observations to achieve superior generalization than discriminative models. However, typically there are only dozens of trials available for a single user [33]. In [4] they tried to overcome this issue through extraction of multiple strongly correlated feature vectors of a single trial with some success. Here the idea of transfer learning is exploited to boost the pool of available training-data.

Secondly, the relevant baseline power of spontaneous *EEG* varies within and between subjects and is believed to be modulated through activity of other non-task relevant networks within the brain [21]. Here an adaptive normalization technique in feature space is employed to handle this issue and standardize the data.

Thirdly, based on findings of simulations on a publicly available dataset, an adaptive online *BCI* was developed. Its aim is to provide feedback from the first trial onwards. The feedback initially originates from a pre-trained *RBM* which is then adapted to the user's specific patterns. As a consequence the training can be immediately conducted within the closed loop system displayed in Figure 1.1.

Lastly, during the course of the thesis numerous extensions to the *RBM*-toolbox created by David Balderas were implemented. Among others they comprise of algorithms for obtaining a learning signal such as Persistent Contrastive Divergence (PCD) [48] and Parallel Tempering (PT) [15], extensions for Gaussian-Bernoulli Restricted Boltzmann Machine (GBRBM)s [14] and hybrid optimization criteria [7, 32].

1.4 Restricted Boltzmann Machines

1.4.1 Motivation

With regard to [11] pattern recognition methods can be divided into *generative* and *discriminative* models. The former learn each class' distribution and can be used to choose the most likely for new observations e.g via Bayes' rule. The latter do discrimination directly through maximizing the posterior[3] of all samples in the dataset. As a consequence, discriminative models require labeled data.

Many algorithms applied to EEG data are of discriminative nature [33]. An intuitive reason is that datasets are typically small (dozens of trials per subject) [33]. Therefore, estimates for the more complex joint distribution tend to be poorer than direct ones of the posterior. Also, with regard to [11], in practice, generalization of generative models is often found to be worse than for discriminative ones.

However, in recent years generative models – also based on Restricted Boltzmann Machines (RBMs)– showed to outperform discriminative ones in other research fields e.g. object [25], [31] and speech recognition [24]. Their key advantages are that they can learn from unlabeled data as well as make use of internal representations through learning the joint distribution.

The major application is found to be feature extraction for supervised learning algorithms [25], [27]. This is typically done through training of a *RBM* on a dataset. Subsequently, the learned weights are applied as seeds for an Artificial Neural Network (ANN), which is fine-tuned through backpropagation learning with a small learning rate.

1.4.2 Definition

*RBM*s as a useful machine learning tool were first introduced by Hinton et al. in 2002 [26]. They are defined as a two layer neural network with *stochastic* activation functions, *binary* states, and *symmetric* weight connections. The two layers form a bipartite graph of visible $\mathbf{v} = (v_1, \ldots, v_i, \ldots)^T$ and hidden $\mathbf{h} = (h_1, \ldots, h_j, \ldots)^T$ units – see Figure 1.2.

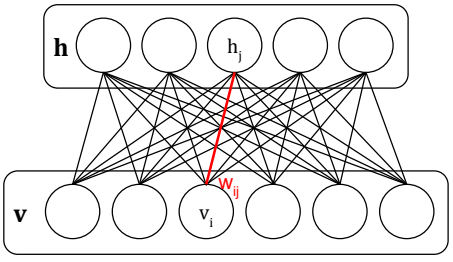


Figure 1.2: Visualization of a *RBM* consisting of a visible layer $\mathbf{v}$ and a hidden layer $\mathbf{h}$. Note that there are no connections within the same layer.

[3] Conditional distribution of labels given the observations.

The joint probability of the configuration being active $p(\mathbf{h} = \mathbf{1}, \mathbf{v} = \mathbf{1})$ is defined by the Boltzmann distribution:

$$p(\mathbf{h}, \mathbf{v}; \Theta) = \frac{e^{-E(\mathbf{v},\mathbf{h};\Theta)}}{\sum_{\hat{\mathbf{v}},\hat{\mathbf{h}}} e^{-E(\hat{\mathbf{v}},\hat{\mathbf{h}};\Theta)}} \tag{1.1}$$

Which is itself defined by an energy function with parameters $\Theta = (\mathbf{W}, \mathbf{b}, \mathbf{a})$ consisting of the weight matrix $\mathbf{W}$, visible $\mathbf{b}$ and hidden biases $\mathbf{a}$ for binary units.

$$E(\mathbf{v}, \mathbf{h}; \Theta) = -\mathbf{h}^T \mathbf{W} \mathbf{v} - \mathbf{b}^T \mathbf{v} - \mathbf{a}^T \mathbf{h} \tag{1.2}$$

The bipartite structure of the *RBM* results in conditionally independent units of the same layer. This enables sampling of the entire layer at once. The energy function of binary units in visible and hidden layer yields

$$p(h_j = 1|\mathbf{v}) = \sigma(a_j + \sum_i v_i \, w_{ij}) \tag{1.3}$$

$$p(v_i = 1|\mathbf{h}) = \sigma(b_i + \sum_j h_j \, w_{ij}) \tag{1.4}$$

with activation function $\sigma(\cdot)$ being the logistic function. The state or output of unit h_j or v_i is either 0 or 1 and is *sampled* from a Bernoulli distribution based on these conditional probabilities.

Since the hidden variables are not observed, the generative objective function is the model's marginal distribution $p_{model}(\mathbf{v})$ which is fitted to the data's $p_{data}(\mathbf{v})$.

$$p_{model}(\mathbf{v}; \Theta) = \sum_{\mathbf{h}} p(\mathbf{h}, \mathbf{v}; \Theta) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h};\Theta)}}{\sum_{\hat{\mathbf{v}},\hat{\mathbf{h}}} e^{-E(\hat{\mathbf{v}},\hat{\mathbf{h}};\Theta)}} \sim p_{data}(\mathbf{v}) \tag{1.5}$$

This is usually done by optimizing a function called Contrastive Divergence (CD) – not the likelihood directly because it would be computationally intractable [26]. The basic learning scheme is to compute statistics of the model's states when the visible units are clamped to training-data and subtract so called negative statistics calculated from 1 step reconstructions, derived though Gibbs Sampling from the conditional probabilities. The learning process is elaborated in detail in [26].

1.4.3 Gaussian-Bernoulli Restricted Boltzmann Machines

The original *RBM* consists of neurons which have binary states. That is, each neuron's output is either 0 or 1. However, in nature real valued data, like *EEG*, is dominant. Researchers have worked extensively on expanding the framework to continuous data [14, 29, 52].
This lead to *GBRBM*s which include Gaussian distributed visible units with diagonal covariance matrix and binary hidden units. They are defined via the following energy function:

$$E(\mathbf{v}, \mathbf{h}; \Theta) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_i \sum_j w_{ij}\, h_j\, \frac{v_i}{\sigma_i^2} - \sum_j a_j\, h_j \tag{1.6}$$

Under the modified energy function, new conditional probabilities of visible and hidden neurons can be derived:

$$p(h_j = 1|\mathbf{v}) = \sigma(a_j + \sum_i \frac{v_i}{\sigma_i^2}\, w_{ij}) \tag{1.7}$$

$$p(v_i = v|\mathbf{h}) = \mathcal{N}(v \,|\, b_i + \sum_j h_j\, w_{ij}\,;\, \sigma_i^2) \tag{1.8}$$

With $\mathcal{N}(\cdot|\mu, \sigma^2)$ being the Gaussian probability density function with mean μ and variance σ^2. The conditional probability of the visibles tells us that the model parameters ($\mathbf{b}$ and $\mathbf{W}$) can be utilized to learn the mean, whereas the variance is solely used to model observation noise. In literature it is recommended to normalize the variance of each feature to 1.0 [23].
Based on the work of [4] and a comparison of standard deviations of the features, we concluded that observed small differences[4] in noise level can be neglected. Hence, we followed the recommendation for our experiments.
*GBRBM*s can be trained similarly to *RBM*s. The update rules for gradient descent of the likelihood change slightly [14]. Due to the intractability problem the gradient of the likelihood is also approximated via *CD* learning.

[4]The variation changes mainly across sessions and subjects and might, therefore, mostly be caused through electrode mounting.

1.4.4 RBMs for discrimination

So far we have discussed *RBM*s as methods for modeling distributions of binary and real valued data. However, they can also be employed for classification.
There are three straight forward ways [23]. (1) the output of the hidden neurons can be used as features for some other standard discriminative method. (2) For each class a separate *RBM* is trained. A test vector is assigned to the class whose RBM computes highest probability under the model. (3) A single *RBM* is used to train the joint density model. This requires a combination of class labels and feature vectors as visible units. The structure is depicted in Figure 1.3.

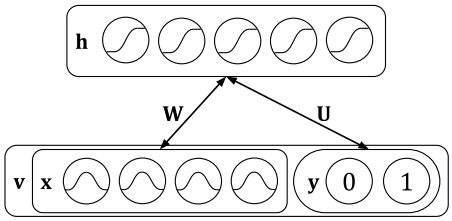


Figure 1.3: A self-contained *GBRBM* for discrimination between 2 classes. The model consists of binary neurons in the hidden layer **h**, Gaussian for the features **x** and a softmax unit for the labels **y**.

With reference to the findings of [31] and results of the offline *EEG* study conducted in [4], we decided to use the latter approach. The energy function of the model is

$$E(\mathbf{x}, \mathbf{y}, \mathbf{h}; \Theta) = \sum_i \frac{(x_i - b_i)^2}{2\sigma_i^2} - \sum_i \sum_j \frac{x_i}{\sigma_i^2} w_{ij} h_j - \sum_j a_j h_j - \sum_k d_k y_k - \sum_k \sum_j d_k u_{kj} h_j \tag{1.9}$$

where the parameters $\Theta = (\mathbf{W}, \mathbf{b}, \mathbf{a}, \mathbf{d}, \mathbf{U})$ are the weight matrix $\mathbf{W}$, visible $\mathbf{b}$, hidden $\mathbf{a}$ and class bias $\mathbf{d}$ as well as the class weight matrix $\mathbf{U}$. In this setup the class labels are one-out-of-K encoded. This is, the vector $\mathbf{y} = (1_{y=i})_{i=1}^K$ consists of zeros except a single one at the label y. This energy function yields the following conditional probabilities:

$$p(h_j = 1|\mathbf{x}, y) = \sigma(a_j + u_{jy} + \sum_i \frac{v_i}{\sigma_i^2} w_{ij}) \tag{1.10}$$

$$p(x_i = x|\mathbf{h}) = \mathcal{N}(x \,|\, b_i + \sum_j h_j w_{ij} \,;\, \sigma_i^2) \tag{1.11}$$

$$p(y|\mathbf{h}) = \frac{e^{d_y + \sum_j u_{jy} h_j}}{\sum_{y^*} e^{d_{y^*} + \sum_j u_{jy^*} h_j}} \tag{1.12}$$

Because of the conditional independence of the visibles given the hidden states, the equation for the feature vector $\mathbf{x}$ is similar to 1.8. Gibbs Sampling is run on these equations

to create a learning signal in a similar fashion to the models elaborated before.

1.4.5 Discriminative Fine-Tuning

As a generative model, *RBM*s are suitable for extracting useful representations of the data. These representations are usually also of discriminative nature. To enforce this, the labels can be incorporated into the visible layer, as described before.

In literature, however, even better results in terms of classification accuracy are reported when the learned weights are fine-tuned with a discriminative objective function [27, 29, 31].

An intuitive way is to employ the extracted weights and biases as seeds[5] for an *ANN*. The pendant to the model displayed in Figure 1.3 is shown in Figure 1.4.

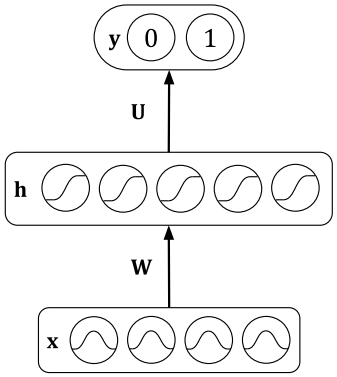


Figure 1.4: A shallow *ANN* with real valued inputs **x**, one hidden layer **h** with sigmoid activation functions and a softmax output **y**.

As discriminative objective function the cross-entroy error is minimized. Similarly to [27] the method of conjugate gradient descent with Polak-Ribière updates [40] is applied to standard backpropagation learning [10]. One advantage of the conjugate method over steepest descent is that gradients of previous iterations are incorporated into the current update step. Consequently, it is not prone to criss-cross patterns in valleys.

Instead of a line search the Armijo rule for sufficient decrease is used to determine the step size in each iteration [3]. The key idea is that for a larger step towards the current iteration's descend direction[6] the residual error of the *ANN* must decrease sufficiently. If this is not the case the step size is reduced in exponentially decaying steps until the criterion is fulfilled or convergence.

[5] Initial weights and biases.

[6] The descend direction is computed via the conjugate gradient method.

2

Model evaluation: Standard Mode

2.1 Introduction

The goal of this experiment was to compare *RBM*s to standard machine learning algorithms in a setup for which little ($\sim$ 100 trials/subject) training-data is available. The mode is entitled *standard* because for each participant an individual classifier is learned. Which is the standard way to set up a brain oscillations based *BCI* [33].

Shrinkage Linear Discriminant Analysis (sLDA) and an *ANN* were chosen as reference classifiers. LDA and its regularized version through shrinkage, on the one hand, are wide spread in the *BCI* community [4, 12, 45]. Moreover, they serve as a kind of gold standard to test new methods. The *ANN* model for pattern recognition on the other hand is very similar to the *RBM* for classification. The key difference is in how weights and biases are trained. The former tries to minimize the classification error and is therefore a discriminative method. Whereas the latter aims to learn the distribution of the visible units[1] and can therefore be called a generative method. The distinction in objective function but similarity in model complexity[2] makes the *ANN* model an ideal candidate for these investigations. The differences are elaborated based on previously recorded data. To maintain comparability to actual online experiments, the final signal processing methods were designed to be causal.

2.2 Methods

2.2.1 Dataset

We decided to use dataset 2a of BCI Competition IV for the following reasons. (1) A number of 72 trials per class and session is moderate to high. (2) It contains data of 9 participants with different performance levels. This variety is also beneficial for the transfer

[1] Observations $\mathbf{x}$ and labels $\mathbf{y}$ in this case.

[2] If the same number of hidden units with sigmoid activation functions are used.

learning approach discussed later. (3) The dataset is publicly available and therefore easy to retrieve.

The cue-based paradigm is depicted in Figure 2.1. It defines four motor imagery tasks, namely imagination of the movement of the left hand, right hand, both feet and tongue. The participants were instructed to start the imagery process as soon as they realized the target class indicated by a visual cue.

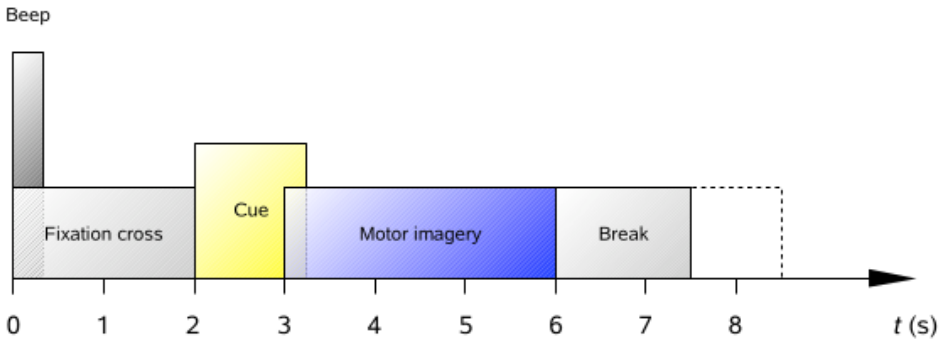


Figure 2.1: Timing scheme of the cue-based paradigm. Image taken from [47]. A trial starts with a green fixation cross at the center of a black screen. At second 2 a cue indicates the mental task. A trial ends when the green cross disappears at second 6. The break in between trials lasts for 1.5 to 2.5 seconds.

For each study-member 2 sessions were recorded on different days. Every session consists of 6 runs with short breaks in between. One run in particular is comprised of 12 trials per class, yielding 72 trials per class per session.

In addition, the experimental protocol includes around 5 minutes of recordings to estimate the influence of Electrooculography (EOG) – see Figure 2.2 for the timing of one session.

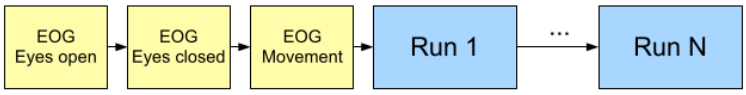


Figure 2.2: Sequence of recording blocks during one session. Image taken from [47]. First, 3 short blocks of about 5 minutes in total were recorded for *EOG* estimation. Subsequently, 6 runs interrupted by short breaks were captured.

22 Ag/AgCl electrodes were used to record *EEG* around central and parietal areas according to the international 10-20 system. The signals were sampled at a rate of 250 Hz and band pass filtered between 0.5 and 100 Hz. A notch filter at 50 Hz was employed to reduce line noise. Moreover, an expert visually inspected all data and annotated trials which were corrupted by artifacts. Further details, such as the exact electrode positions, are listed in [47].

Without loss of generality we decided to use 2 classes out of the 4 possible. The three major arguments in favor of this decision are. (1) We are interested in elaborating the basic principles. (2) An extension of *RBM*s to support multiple classes is straight forward. (3) Most importantly, *RBM*s are usually applied to datasets comprised of thousands up to even billions of observations. As this is certainly not the case here, we try to keep the number of classes[3] and features low[4].

After initial experiments, left hand and both feet motor imagery were chosen because they showed highest classification accuracy.

2.2.2 Feature Extraction

Out of the 22 channels available we used 13 namely FC3, FCz, FC4, C5, C3, C1, Cz, C2, C4, C6, CP3, CPz and CP4. As spatial filtering technique Laplacian derivations around C3, Cz and C4 were applied. They extract local activations, which match with the associated areas of hands and feet on the motor and somato-sensory humunculus. Simultaneously, the noise – especially inherent on all channels – is suppressed [55]. After this process 3 signals remained.

They were, subsequently, filtered by 2 *causal* Infinite Impulse Response (IIR) bandpass filters. The Butterworth filter-type was chosen because it does not show ripples in pass and stop band. The order was set to be minimum to fulfill the specifications listed in Table 2.1. As filter implementation a concatenation of second order structures (SOS) was chosen. The MATLAB (Mathworks Inc., Natick, USA) provided functions `fdesign` and `design` of the DSP-Toolbox were applied to match exactly the passband specifications. For numerical stability reasons the coefficients were scaled to lie in the interval $[-1, 1]$.

filter	f_{stop1}	f_{pass1}	f_{pass2}	f_{stop2}	A_{stop}	A_{pass}
-	Hz	Hz	Hz	Hz	dB	dB
1	6	8	15	17	20	0.5
2	14	16	30	35	20	0.5

Table 2.1: Specifications of stop-band edge frequencies f_{stop} and associated minimum attenuation A_{stop} as well as pass-band edge frequencies f_{pass} and maximum attenuation A_{pass}.

For all bands and trials a 2 s long window starting from second 3.0 and ending at second 5.0 in Figure 2.1 was employed to compute logarithmic bandpower features. Therefore, each filtered signal within the time-window was squared and summed. The base-10 logarithm of the sum resulted in the final feature-value.

All in all, this method yields 6 features as well as 1 feature-vector/observation per trial. Moreover, the feature extraction chain itself does only depend on the relevant electrode positions. Thus, it can be readily applied across subjects which is done in the next chapter.

[3] The applied RBM has to learn the joint density of classes and observations.

[4] In the setup depicted in Figure 1.3 the labels are also features.

2.2.3 Normalization

As already pointed out in chapter 1, the non-stationarity of *EEG* has a severe impact on *BCI* performance. One method to overcome this issue is adaptive normalization of the signals. At first the question where to estimate the statistics arises? In an initial experiment we compared standardization of raw, spatially filtered signals and features. The results were in favor of normalizing the features. From a theoretical perspective one can argue that the feature extraction steps reduce the signal to noise ratio. Thus, when normalization is done at last the impact of noise on the estimates of mean and standard-deviation is minimal.

The next question was to find a proper time scale for adaptation. Therefore, two time spans were chosen. Either all trials of the selected classes of one session, which took about 30 minutes of recording time, or one run, lasting approximately 4 minutes, were utilized to compute mean and standard-deviation per feature. The session's/run's associated observations where then standardized to have zero-mean and unit-variance.

As this approach is clearly non-causal, exponentially weighted estimates for mean μ and standard-deviation σ were implemented to retrieve results for simulated online experiments. The estimated mean of trial k and feature i is

$$\hat{\mu}_{k,i} = \lambda\, \hat{\mu}_{k-1,i} + (1-\lambda)\, x_{k,i} \tag{2.1}$$

with forgetting factor λ and the trial's feature vector $\mathbf{x}_k$. Along, the standard deviations' estimates are given by

$$\hat{\sigma}_{k,i} = \sqrt{\lambda\, \hat{\sigma}^2_{k-1,i} + (1-\lambda)\, (x_{k,i} - \hat{\mu}_{k,i})^2} \tag{2.2}$$

For the sake of simplicity the same forgetting factor λ was applied for all estimates. The following theoretical considerations help to determine suitable values for λ. Equation 2.1 can be transformed to[5]:

$$\hat{\mu}_{k,i} = (1-\lambda) \sum_{j=0}^{\infty} \lambda^j\, x_{k-j,i} \tag{2.3}$$

The sum of the last N observations' weights defines a truncated geometric series for $\lambda \in (0,1)$ an has therefore a closed form solution:

$$(1-\lambda) \sum_{j=0}^{N-1} \lambda^j = 1 - \lambda^N \stackrel{!}{=} p \tag{2.4}$$

Since the sum of the weights of all observations is 1.0, this equation expresses the last N weights' fraction p. For given N and p the desired weighting factor is

[5] This holds also for the variance's estimates.

$$\lambda = \sqrt[N]{1-p} \tag{2.5}$$

Besides λ the initial estimates are also crucial. This issue was addressed by computing feature vectors with data of the *EOG* estimation blocks – see Figure 2.2. In particular, the first 1 minute block with eyes open condition was exploited, since the subjects were instructed to look at the center of the screen. This fits best to the motor imagery phase during a trial. The block's mean and standard-deviation per feature were applied as initial estimates.

2.2.4 Data Partitioning

The standardized data are then separated into training and test set according to Figure 2.3. That is, the algorithms are evaluated on each subject individually. They are trained on the first session and tested on the second.

Figure 2.3: The training-set consists of participant i's data of the first session (highlighted green) while the test-set is comprised of all runs of the second session (highlighted red).

2.2.5 Hyper-Parameter Estimation

sLDA The optimal regularization parameter can be computed analytically. Hence, this method can be applied to the dataset right away. For details see [12].

RBM During all subsequently presented experiments the architecture was fixed to a single layer double-entry RBM depicted in Figure 1.3. The model's hyper-parameters (initial weights, biases, momentums, ...) were chosen with reference to guidelines published by Bengio and Hinton [5, 23].

The parameters which all subsequent experiments have in common are *CD* with 1 step of Gibbs-Sampling as generative training algorithm. The size of a mini-batch was set to

cover 16 observations. The initial weights were drawn from a normal distribution[6] with parameters $\mathcal{N}(0, 0.5^2)$ for Softmax-Binary and Gaussian-Binary connections, while the biases were set to 0. On all weights and hidden bias updates a L2 regularization term of 10^{-3} , penalizing large values, was imposed. A momentum, to speed up gradient descent, was set to 0.5 first and linearly increased to 0.9, whereas with regard to [23] the learning rates were decreased to half their initial values over the course of the learning-process.

The remaining (hyper-)parameters were determined using a grid search over a range of possible values. Details see section A.1 in the appendix. For this experiment the initial learning rate of weights $\mathbf{W}$ was set to 10^{-3} and for class weights $\mathbf{U}$ to 10^{-2} respectively. The number of hidden units $\mathbf{h}$ was found to be 50. The epochs, standing for how often the algorithm iterates over the entire training-set, were set to 500. Every feature vector's association to a mini-batch was randomly selected during an epoch.

ANN The model was determined through unrolling the above described *RBM* into a directed graph as displayed in Figure 1.4. The implementation employs conjugate gradient descent introduced in chapter 1. To ease comparability, the aim was to change as few hyper-parameters as possible.

Consequently, the initial weights and biases, number of hidden units, mini-batch size and L2 regularization were the same. The learning rate was also annealed towards half the initial value. In each epoch the current value was used as starting step-size for the Armijo rule. The rule's other parameters were set to common values in literature; $\beta = 0.5$, $\sigma = 10^{-3}$ and $n = 7$. Solely, the number of epochs and initial learning rate were optimized similarly to the *RBM* using the same range of values. It yielded 100 iterations and $\lambda = 10^{-2}$ respectively.

2.2.6 Performance Measure

There are many criteria for comparing classification performance. The most common in *BCI* research is the observed accuracy $\hat{p}$. However, for this work the choice fell on Cohen's Kappa [16] because it also incorporates the off-diagonal elements of the confusion matrix. They are utilized to calculate the expected accuracy of random agreement p_e between targets and model assigned labels. Kappa is defined as the following ratio

$$\hat{\kappa} = \frac{\hat{p} - p_e}{1 - p_e} \tag{2.6}$$

which can lie within the range $[-1.0, 1.0]$. 1.0 stands for perfect agreement and -1.0 for total disagreement respectively. A value of 0.0 indicates random agreement. Since the number of available trials for one subject was limited, the two-sided confidence interval

[6] The variance was determined empirically so that initial conditional probabilities of the hidden units were around 0.5. In this regime the sigmoid activation is not saturated, which speeds up initial learning.

for chance level was calculated applying the method discussed in [9]. The interval's limits are

$$\kappa_{u,l} = \frac{\hat{p} - p_e}{1 - p_e} \pm z_{1-\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1-\hat{p})}{N \cdot (1-p_e)^2}} \tag{2.7}$$

where $z_{1-\frac{\alpha}{2}}$ is the $1-\frac{\alpha}{2}$ quantile of the standard-normal distribution. For an unbiased test-set[7] the expected level of chance is 0.5, which is also the expected accuracy of a random classifier $\hat{p}$. The 95 % one-sided confidence interval's limit are then $\kappa_{u,l} = \pm 0.16$ for $N = 2 \cdot 72 = 144$ observations per session. That is, the practical level of chance lies in the interval $[0.16, -0.16]$ with a probability of 95%.

2.3 Results and Discussion Normalization-Methods

For exponential weighted moving average normalization the initial estimates are important. Therefore, the difference in first and second order statistics between the first run and the block with eyes open condition of the same session was calculated in feature space. Figure 2.4 depicts boxplots across subjects, sessions and features for a selection of time-spans. Each takes the samples from the beginning of the eyes open block until the time-stamp displayed on the x-axis into account.

From Figure 2.4a follows that the variance of estimated means across the factors participant, session and feature is quite high. Additionally, the chance that the estimate is biased is high i.e. the median of μ_{open} is larger than μ_{run1}. The estimation time itself has little influence.

On the contrary, the effect on the standard-deviation, shown in Figure 2.4a, exhibits a trend. One can observe improvement until a window length of 30 seconds, which covers 15 consecutive feature vectors[8] for each estimate.

Altogether, for longer windows the improvement of the standard-deviation can be neglected, while the variance of mean estimates tends to increase. As a trade-off a 20 s long window was selected.

Table 2.2 lists kappa values for a *sLDA* classifier trained on the data of the first session and evaluated on the second. The table compares the presented methods versus no standardization at all. *sLDA* was chosen at this early stage, since there were no hyper parameters to identify.

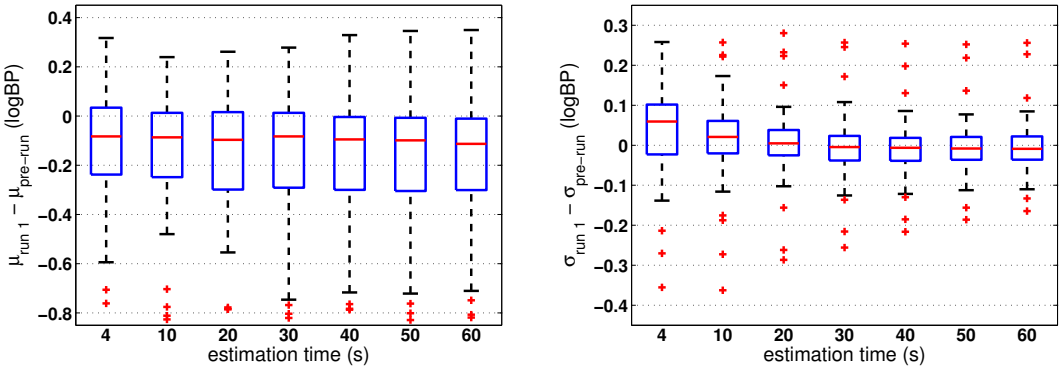
The results point out that normalizing the features offline improves classification performance across sessions for almost every individual in the group. Moreover, standardizing run-wise yields a 0.01 larger Kappa value than session-wise. In Figure 2.5 the course of raw feature vectors across runs and sessions for study-member 7 is displayed. One can see that the statistics vary not only across but also during sessions.

[7] The amount of observations of both classes are similar.

[8] A 2 s long window is used to estimate band power.

Consequently, the adaptive method's forgetting factor λ was chosen such that the last run's trials, which are 24, possess 90% of the weights. Putting this into equation 2.5 yields a value of 0.90.

On the one hand, the adaptive, *causal* approach shows better overall kappa values than no normalization. Participants 3, 6 and 7 exhibit even considerable improvement. On the other hand, it can not perform as well as the non-causal methods. Because of that succeeding simulated online experiments used acausal `run` normalization for the training-set and the causal `adaptive` method for the test-set.



(a) Differnece in means.

(b) Difference in standard-deviations.

Figure 2.4: Boxplots of the difference between a session's first run (`run1`) and the preceding block with eyes open condition (`open`) over all subjects, sessions and features. The blue box covers 50 percent of the values, while the red line within the box indicates the median. A red cross highlights outliers. Some are not displayed due to better visualization of the boxes. The whiskers end at the most extreme data point or 1.5 times the upper or lower bound of the box. This would cover 99% of normal distributed data.

normal-ization	participant 1	2	3	4	5	6	7	8	9	**overall**
no	0.87	**0.65**	0.57	0.39	0.09	0.42	0.76	0.50	0.86	**0.57±0.25**
session†	0.90	0.65	0.79	**0.42**	0.04	**0.60**	0.93	**0.52**	0.86	**0.63±0.28**
run†	**0.94**	0.61	**0.81**	0.41	0.09	0.59	**0.94**	**0.52**	**0.91**	**0.64±0.29**
adaptive	0.86	0.59	0.73	0.37	**0.11**	0.55	0.90	0.48	0.78	**0.60±0.25**

Table 2.2: Kappa values of the test-set for an *sLDA* classifier learned on session 1 and evaluated on session 2. The normalization techniques applied are: `no` for no standardization, `session`, `run` and `adaptive` as described above. The overall column summarizes mean and standard-deviation across participants.
† Non-causal methods.

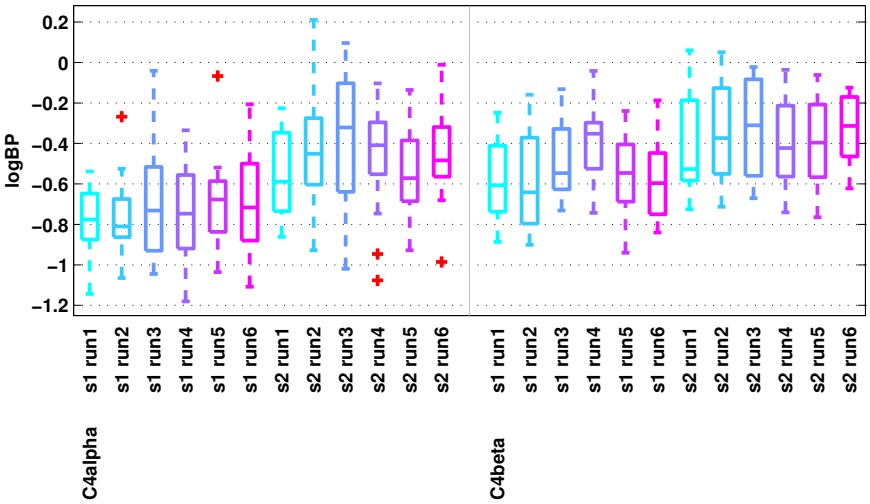


Figure 2.5: Boxplots of 2 selected features for all observations of subject 7. They illustrate the change during as well as across sessions. Every boxplot summarizes the statistics of a single run.

2.4 Results

After fixing the normalization method and its parameters, the 3 models were evaluated. The obtained Kappa values are presented in Table 2.3. *sLDA* is a deterministic method. Hence, repeats result in the same value. Whereas, *ANN*s and *RBM*s depend on random initial weights. Moreover, the *RBM*'s units are stochastic by nature. To estimate the variance introduced, the experiments were repeated 64 times for both methods. Complementary, the mean Kappa values together with their standard-error are depicted in Figure 2.6.

classifier		participant 1	2	3	4	5	6	7	8	9	**overall**
sLDA		**0.88**	0.58	**0.78**	**0.44**	0.03	0.58	0.89	**0.53**	**0.89**	**0.62±0.28**
ANN	μ	0.84	**0.61**	**0.78**	0.36	0.07	**0.61**	**0.90**	0.49	0.88	**0.62±0.28**
	σ	0.02	0.01	0.01	0.03	0.05	0.02	0.01	0.02	0.01	0.02±0.01
RBM	μ	0.80	0.59	0.64	0.38	**0.14**	0.51	0.85	0.44	0.76	**0.57±0.23**
	σ	0.02	0.04	0.03	0.09	0.04	0.04	0.02	0.05	0.02	0.04±0.02

Table 2.3: Comparison of classifiers applied to the two class problem standard-mode. Since ANNs and RBMs started with random weights and RBMs are stochastic by nature, their mean Kappa value μ and its standard-deviation σ were computed based on 64 repetitions. The last column states mean and standard-deviation overall participants.

As expected, the combination of run and adaptive standardization methods slightly improved the results by 0.02 for *sLDA* – see Table 2.2 versus 2.3. It is closely followed

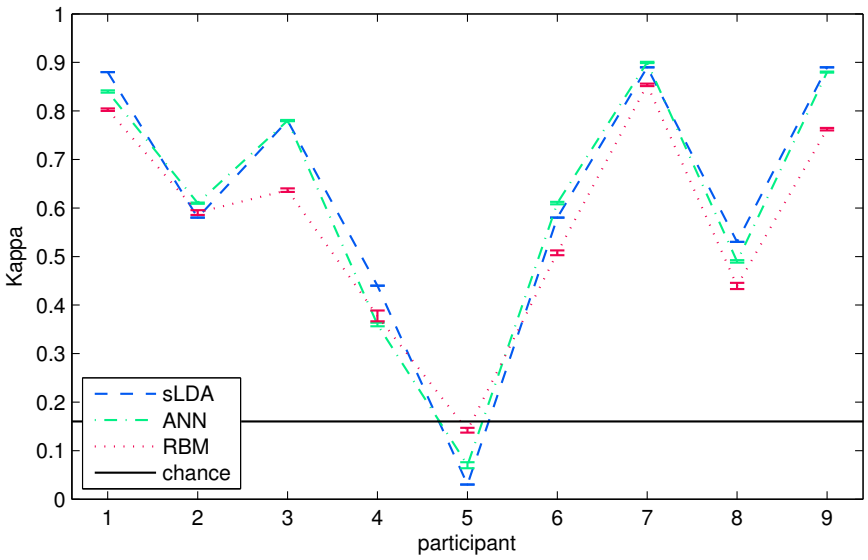


Figure 2.6: Visualization of the mean Kappa values across participants presented in Table 2.3. The bars indicate the standard-error of the mean for $N = 64$ repeats. The solid black line stands for the practical level of chance.

by the *ANN*, which produces very similar results for all members of the population – see Figure 2.6. Both methods, however, outperform the *RBM*. Its mean of 0.57 across participants is smallest. A paired Wilcoxon signed rank test resulted in a p-value $p = 0.1484$ (Bonferrioni corrected) for the difference between Kappa values of ANN and RBM.

2.5 Discussion

During this study we investigated the competitiveness of the *RBM* model in a typical *BCI* setup with few trials. Based on a participant's recordings 6 log-bandpower features were computed, normalized and session-wise split into training- and test-set.

The first goal was to find a causal normalization procedure which improves inter-session classification performance. Theoretical considerations and initial experiments favored normalization in feature space. Thereupon, a suitable time-span for first and second order statistics estimation was assessed. The results listed in Table 2.2 and depicted in Figure 2.5 indicate that standardizing run-wise seems to be a good trade-off between accurate estimates and keeping track of changes.

Still the adaptive approach does not perform as well. We assume that a major part of the performance loss is caused due to the biased initial estimates – see Figure 2.4a. The bias itself might be created due to a different mental state since it can be observed across all participants, sessions and features. Because its sign is negative we can infer that on average the power in the covered frequency bands (α and β) was lower during motor

imagery. I.e. the participants were more engaged.

The second goal pursued was assessing whether the *RBM*-model is competitive in this scenario. The applicability to successfully classify *SMR* patterns was already shown in [4]. They derived multiple correlated feature vectors of a single trial to boost the dataset. Here, we did not consider this method, since we wanted to compare the effects if the training-set is little (= dozens of trials).

In Table 2.3 and Figure 2.6 one can clearly see that the model's optimizing a discriminative criterion achieve higher Kappa values. Moreover, the *ANN* and *RBM* share the same model complexity and use similar L2 regularization. Nevertheless, the *RBM* performs worse[9]. We reason that the limited observations do not suffice the *RBM* to learn the real distribution underlying the data accurately enough. With reference to [11], generative models are only competitive or superior to discriminative ones if they achieve exactly this. Another way of boosting the training-set – transfer-learning – is presented and applied in the next chapter. Since the feature extraction methods do not change the results of both approaches can be compared.

Lastly, participant 5's Kappa values, presented in Figure 2.6, of *sLDA* and *ANN* are within the 95% confidence interval of chance level, while the *RBM*'s value is close to its border. One could therefore argue that the user would perform at chance level. To find out whether there is a task relevant change in activity *ERDS*-maps for both sessions and classes were computed. They are displayed in Figure A.1 in the appendix and show significant change in alpha- and beta-band of C3. One can also see that the Event-Related Desynchronization (ERD) is more pronounced in the second session. For that reason we decided to use the recordings of participant 5 also for training in subsequent chapters.

[9] Based on the trend displayed in Figure 2.6 the difference might turn significant if more participants would be evaluated.

3 Model evaluation: Transfer Mode

3.1 Introduction

The findings of the previous chapter point out that for limited data discriminative models are superior to the *RBM*. Here, we boosted the data available for training by a factor of up to 8[1] by applying transfer learning. Similar to [28] the classifiers were trained on a mixture of participants.

On the one hand, we were interested in the impact on classification performance across models. We assumed that it would decrease because no data of the participant under test was presented. Consequently, they were not able to identify and adapt to a participant's individual patterns. Nonetheless, they could make use of patterns discovered from other study members.

On the other hand, the question whether the *RBM* is able to make more use of the larger training-set than the discriminative classifiers arose. Based on results presented in other disciplines such as image-processing we suspect that this should be the case. If so, it urges the question which effect the composition of participants in the training-set have. Should it consist of good performers, poor ones or a mixture?

3.2 Methods

The feature extraction and normalization methods presented before did not changed since they were already designed to work in this mode as well. To maintain comparability with the previous experiment, the same classes (left hand versus both feet motor imagery) were selected.

[1] The dataset consists of 9 participant. Hence, the data of 8 can be used for training.

3.2.1 Data Partitioning

For this experiment leave one participant out cross validation was employed. Therefore, the data was separated into training- and test-set according to Figure 3.1. The first session of the tested participant was not considered to maintain comparability with the results of the previous chapter. This procedure was repeated until every participant was in the test-set once.

Figure 3.1: The training-set consists of all recordings except participant i's data (highlighted green) while the test-set is comprised of all runs of the second session (highlighted red).

Recordings of distinct groups of study-members were used to investigate effects of the training-set's composition. The participants were split into groups of three, denoted poor (Kappa values from 0.03 to 0.53), fair (0.58 to 0.73), good (0.88 to 0.89) and mixed (participants 1,6,8), based on their Kappa values of the previous chapter – see *sLDA* row of Table 2.3. If a group-member was selected to be in the test-set, its data was not considered for training.

3.2.2 Hyper-Parameter Estimation

RBM Almost all parameters of the preceding chapter were kept. These include, *CD*-1 learning and the parameters: mini-batch size, initial weight variance, initial biases, L2 regularization, *number of hidden units*.

The same grid-search procedure, described in appendix A.1, was applied to find optimal values for the remaining hyper-parameters. The initial learning rates of weights $\lambda_{weights}$ and class-weights λ_{class} were found to be 10^{-3}. The training epochs were determined to be 50.

ANN Similar to the *RBM*-model only the learning rate λ and number of epochs were selected from the same range of values. The optimization of the average Kappa value across all subjects yielded $\lambda = 10^{-3}$ and `#epochs` $= 25$.

For the training-set composition evaluation the data available for learning was shrunk to about a third. As compensation the number of epochs was tripled.

3.3 Results

To demonstrate the importance of normalizing the features across participants, mean raw feature vectors per run were computed. Their distributions across the 9 users are displayed in Figure 3.2. Compared to Figure 2.5 the baseline variation across study-members is huge. Consequently, one can not expect competitive generalization without proper standardization.

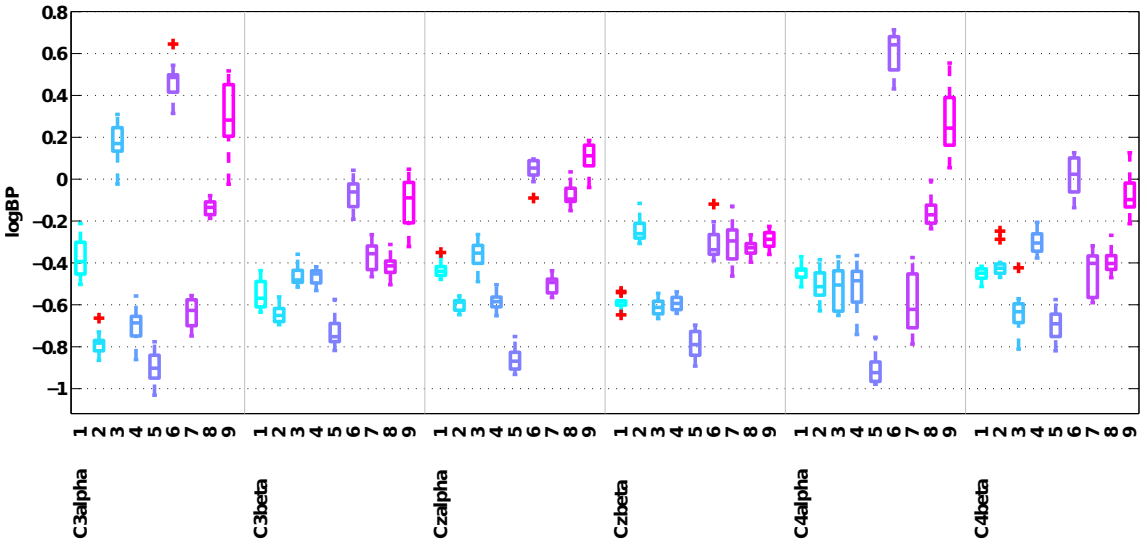


Figure 3.2: Boxplots of run-wise mean log-bandpower features per participant for non-normalized data. The subjects are color-coded.

Learning the 3 models on standardized features resulted in overall Kappa values of 0.44 for *sLDA*, 0.45 for the *ANN* and 0.46 for the *RBM* respectively. Details are listed in Table 3.1. The simulations were again repeated 64 times to estimate the variances introduced by the models. The *RBM* achieved highest mean and most peak Kappas. It is closely followed by the *ANN* and *sLDA*. However, the overall differences are too small to testify whether the *RBM* is superior. The listed results are also presented graphically in Figure 3.3. For most participants the variation of classifier means is lower than 0.10, only for participants 4 and 7 it is higher.

classifier		participant 1	2	3	4	5	6	7	8	9	**overall**
sLDA		**0.86**	0.17	0.60	0.27	**0.13**	**0.43**	0.42	0.27	0.77	**0.44±0.26**
ANN	μ	0.85	0.17	**0.62**	0.35	0.10	0.39	0.50	**0.28**	0.76	**0.45±0.26**
	σ	0.01	0.02	0.02	0.02	0.01	0.02	0.01	0.02	0.02	0.02±0.01
RBM	μ	0.81	**0.19**	0.61	**0.39**	**0.13**	0.39	**0.63**	0.21	**0.79**	**0.46±0.26**
	σ	0.02	0.03	0.03	0.01	0.02	0.02	0.01	0.02	0.02	0.02±0.01

Table 3.1: Comparison of classifiers applied to the two class problem in transfer-mode. μ and σ were estimated based on 64 repetitions. The last column states mean and standard-deviation overall participants.

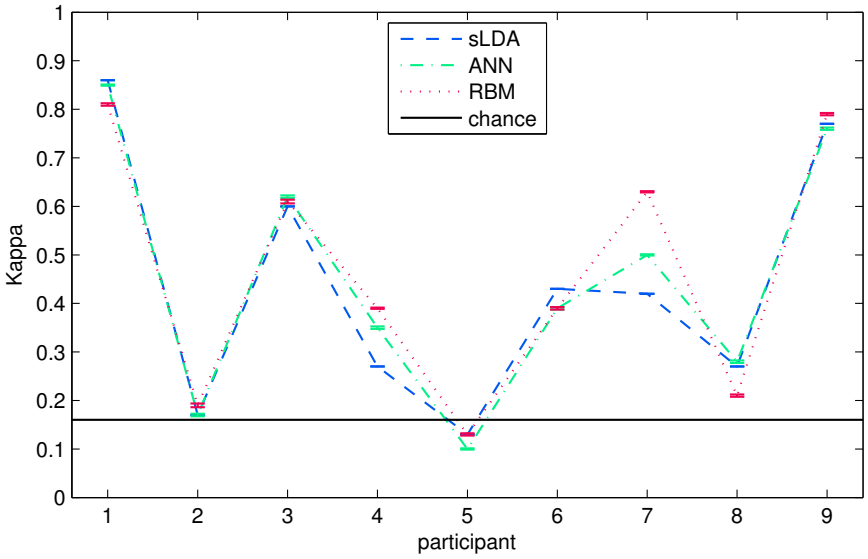


Figure 3.3: Mean kappa values across participants presented in Table 3.1. The bars indicate standard-errors of the means for $N = 64$ repeats. The solid black line is the practical level of chance.

Next, the training-set composition was evaluated. Table 3.2 summarizes the simulation outcomes for *sLDA* and *RBM*. For every tested group the *RBM* works overall better than *sLDA*. The gap was larger when the models were trained on poor (0.47 versus 0.41) or fair (0.46 versus 0.41) performers, while it shrunk for good (0.45 versus 0.45) and mixed (0.48 versus 0.47) ones. Interestingly, the difference of the *RBM*'s Kappas are smaller than 0.09 across groups for every tested participant. I.e. the *RBM* was able to extract representations that generalize well on other participants independent of the dataset's composition. The *sLDA*, however, exhibits considerable changes up to 0.44 – see columns of participant 2, 3, 6, 7 and 8 for example.

classifier		participant 1	2	3	4	5	6	7	8	9	**overall**
		poor									
sLDA		0.81	**0.43**	0.46	0.29	0.09	**0.58**	0.20	0.12	0.71	**0.41±0.26**
RBM	μ	0.82	0.25	0.61	0.35	0.16	0.46	0.60	0.23	0.74	**0.47±0.24**
	σ	0.02	0.02	0.03	0.02	0.03	0.02	0.03	0.02	0.02	**0.02±0.01**
		fair									
sLDA		**0.87**	0.10	0.33	0.37	**0.17**	**0.58**	0.20	**0.45**	0.65	**0.41±0.25**
RBM	μ	0.83	0.21	0.57	**0.39**	0.13	0.42	0.62	0.25	0.75	**0.46±0.25**
	σ	0.01	0.01	0.03	0.01	0.01	0.03	0.01	0.02	0.01	**0.02±0.01**
		good									
sLDA		0.80	-0.01	0.64	0.31	0.06	0.28	**0.72**	0.33	**0.89**	**0.45±0.33**
RBM	μ	0.81	0.16	0.59	0.36	0.12	0.37	0.62	0.21	0.83	**0.45±0.27**
	σ	0.01	0.01	0.02	0.02	0.01	0.01	0.01	0.02	0.01	**0.01±0.01**
		mixed									
sLDA		0.74	0.25	**0.67**	0.24	0.16	0.49	0.46	0.42	0.77	**0.47±0.23**
RBM	μ	**0.87**	0.25	0.62	0.37	0.14	0.46	0.62	0.24	0.75	**0.48±0.25**
	σ	0.01	0.01	0.03	0.01	0.02	0.02	0.01	0.02	0.01	**0.02±0.01**

Table 3.2: Transfer mode results for various compositions of participants in the training-set. The 9 study-members were categorized into `poor` (4,5,8), `fair` (2,3,6) and `good` (1,7,9). The `mixed` group consists of a blend (1,6,8).

3.4 Discussion

In this chapter we investigated the effects if data of other participants is used for training instead of an individual's first session. The large variations in baseline activity, displayed in Figure 3.2, became a severe issue compared to the previous chapter. Unification through standardization turned out to yield good results, considered that the classifiers did not see data of the subject tested. The overall kappa values in standard-mode are approx. 0.15 larger. That corresponds to a difference in accuracies of 7.5% for an unbiased test-set.

The difference varies among individuals, if one compares Figure 2.6 with Figure 3.3. It is marginal for 1, 4 and 5, moderate for 3, 6 and 9 and considerable for 2, 7 and 8. For further investigations *ERDS*-maps of participants 2 and 7 were computed. Subject 2's patterns, for example, show Event-Related Synchronization (ERS) instead of expected *ERD* which contradicts with data of the others – see Figure A.3. This explains why the performance decreased that much. The maps of participant 7, depicted in Figure A.4, show strong *ERD* for both classes on C3 and C4, which is not that pronounced in any other subject.

These findings indicate that there were no stereotypical matching subjects in the training-set, whose patterns would match to the tested one. We argue that if there were many more subjects in the dataset, the chance of a matching stereotype would be higher. Consequently, the performance of the classifiers in transfer mode should not drop that much. Furthermore, more complex models could be learned on a larger set of features.

Finally, we investigated whether the composition of the training-set has an effect on classification performance. Again, the *RBM*-model performed better than *sLDA*. Fascinating is that its results hardly vary no matter what group it was trained on. Nonetheless, the mixed group achieves best overall Kappa. It is even higher than the value of the *RBM* trained on all other subjects. A possible explanation could be that the very good performers' patterns might be too dominant. The *sLDA* accomplishes higher peak Kappas but depends considerably on the constitution of the training-set.

All in all, the presented results are encouraging to employ transfer-learning for *SMR* based *BCI*s. As expected the mean accuracy is lower than in standard-mode. However, the difference is not large and a pre-trained classifier could be used to give feedback to new participants from the first run onwards. The simulations showed that out the 3 possible models the *RBM* obtains best classification performance. It is not significant, however.

4 Standard vs. Transfer Mode

4.1 Introduction

In this chapter we assessed whether the information encoded in a pre-trained *RBM* can be exploited for a classifier trained in standard-mode. Therefore, we incorporated the weights and biases learned by the *RBM* in transfer-mode as seeds for an *ANN*, which was subsequently trained in standard-mode. The choice fell on the *ANN*-model because it outperformed the *RBM* in the standard-mode simulations.

Moreover, fine-tuning the network discriminatively is a common approach [7, 27]. The assumption is that the *RBM* has found a solution which is already close to a good local minimum for classification. Initializing the neural network with its weights rather than random ones should result in less variation and slightly better results.

4.2 Methods

The fine-tuning approach, discussed in section 1.4.5, was applied. A trade-off between fine-tuning the *RBM*'s weights and exploring the adjacent energy-landscape to find a better minimum for the individual subject but also risking divergence had to be found.

As a consequence, most parameters of the *ANN* were fixed to the same values as in chapter 2. To reduce the step sizes in each update, the (mini-)batch size was increased to span the entire training-set and the learning rate was reduced to 0.005 (= half its value).

The seeds were extracted from the *RBM*'s trained for Table 3.1 applying leave one subject out cross validation. The left out participant's first session was employed as training-set.

4.3 Results

The classification performance is displayed in the last row of Table 4.1. The first row is taken from Table 2.3 to ease comparison of the individual's Kappa values.

The *ANN* with generatively pre-trained seeds achieves not only best overall mean Kappa (0.64) but also smallest standard-deviation (0.25). Whereas, the model with random initial weights yielded larger variation across subjects (0.03) and repetitions (0.01). A Wilcoxon signed rank test between the mean kappa values across participants resulted in a p-value of 0.125. That is, the difference is not significant based on this dataset.

classifier		participant									**overall**
		1	2	3	4	5	6	7	8	9	
ANN	μ	0.84	0.61	0.78	0.36	0.07	0.61	**0.90**	0.49	0.88	**0.62±0.28**
	σ	0.02	0.01	0.01	0.03	0.05	0.02	0.01	0.02	0.01	0.02±0.01
RBM + ANN	μ	0.84	0.61	0.78	**0.44**	**0.12**	**0.64**	0.89	**0.53**	0.88	**0.64±0.25**
	σ	0.00	0.01	0.00	0.02	0.03	0.00	0.00	0.01	0.00	0.01±0.01

Table 4.1: Comparison of classifiers applied to the two class problem in standard-mode. Similar to before, 64 repetitions were used to estimate the models' μ and σ. The last column states mean and standard-deviation overall participants.

4.4 Discussion

For this experiment we employed weights and biases of the *RBM* learned in transfer-mode as initial values for an *ANN*. The *ANN* was subsequently learned discriminatively on the participant's first session and tested on the second respectively. The simulation results indicate that this method of choosing initial weights works better random seeds. This means that the generatively pre-trained *RBM* found a local minimum (for a mixture of subjects) which helped in finding a good local minimum for a new individual. This is especially true for the previously as poor (4,5,8) or fair (6) classified participants, since their Kappa values are higher – see Table 4.1.

This method is even overall better than *sLDA* in single-subject mode for the investigated dataset and extracted features. A key advantage is that its variance across individuals is *lower* while the mean is *higher*. However, one would have to assess whether the difference is significant utilizing a larger dataset.

Furthermore, the (hyper-)parameters of the *ANN* and *RBM* were optimized on this dataset using cross-validation. Thus, estimates of the true classification performance are biased. However, since both method's parameter-sets were optimized, the relations between *ANN* and *RBM* should be accurate.

5 Online System

5.1 Introduction

The findings of the preceding chapters indicate that the *RBM* based transfer learning approach is applicable for *SMR-BCI*s. A combination of generative pre-training and discriminative fine-tuning achieved very good results on a previously recorded dataset, although very simple feature-extraction methods were employed. In this chapter we went one step further to an actual online *BCI* system. The goal was to assess whether the combination works online as well as to quantify its performance on naive participants.

Since the models learned in transfer mode achieved high mean accuracies[1], we decided to exploit this and present feedback from the first trial onwards with the aim to engage participants more into the experiment. To achieve this the entire offline dataset was employed for pre-training. The *RBM* was chosen as model because it performed best in the transfer-mode scenario. Also, the paradigm of the experiment to estimate initial statistics was changed with the aim to reduce the bias of baseline activity estimates.

Moreover, the results listed in chapter 4 tell us that adaptation of the pre-trained system to the individual's patterns is possible and leads to improved classification performance. Therefore, each new participant's recorded data is employed for discriminative online-adaptation of the model. That is, during recording the training of the classifier is continued. Other co-adaptive *BCI* systems presented in [18, 51] reported results that let us feel positive about what to expect.

Essential for effective adaptive learning is to identify and discard outliers. Therefore, an online Electromyography (EMG) and *EOG* detection system was implemented. An inverse filter, learned adaptively, represents its core.

[1] A kappa value of 0.46 corresponds to an accuracy of 73% for a balanced two-class test-set.

5.2 Methods

5.2.1 Experimental Setup

A single session experiment was designed to assess if the combination of pre-training and adaptation works for a new subject. The session's timing scheme is depicted in Figure 5.1. At first, a one minute pre-run is employed to estimate initial statistics of features as well as to train the artifact detection model. Thereafter, 4 runs – each lasting roughly 5 minutes – are recorded for co-adaptive training. The number of trials per run was set to 40. Based on Equation 2.7 and [36] the practical level of chance lies below 65%[2] with a probability of 95%. Hence, the performance of the pre-trained *RBM* on the first run can be assessed with reasonable certainty.

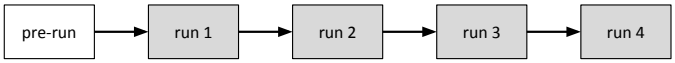


Figure 5.1: Timing of a training-session. In a single pre-run the first and second order statistics per feature are estimated. This lasts about 1 minute. Subsequently, 4 runs are employed to estimate a subject's performance. Each run takes about 5 minutes. The runs were interrupted by short breaks.

5.2.2 Experimental Paradigm

Figure 5.2 displays the sequence of events for a trial of runs 1 to 4. It is inspired by Figure 2.1 because the *RBM* is trained on recordings generated by that paradigm. There are deviations though. The break between consecutive trials lasts about 0.5 seconds longer. More interestingly, at second 5 the current classifier's output is presented to the subject for 1 second.

We decided to present the feedback in a discrete way for the following reasons. (1) The participants can focus better on the motor imagery task. (2) A 2 s long window is used for estimation. That means the lag between displayed feedback and input would be relatively high for continuous presentation. (3) It is not straightforward for the participant to interpret the presented feedback because it is an output of a non-linear transformation.

The feedback was explained to participants as the classifier's certainty that it detected the type of thought. The visual presentation is depicted in Figure 5.3. It consists of a white frame pointing from the center to the preceding cue's direction. The classifier's probability for the current target exceeding chance level (50%) is displayed. If it does not exceed 50% no bar appears. I.e. only the magnitude of positive feedback is presented. If the classifier was 100% certain to choose the correct target, the bar would have touched the other end of the frame.

[2] Or a Kappa value of 0.3 for an equal number of trials for the 2 classes.

The participants were asked to avoid producing artifacts as long as they saw the green fixation cross on the screen. Any type of movements (neck, eyes, tongue, jaw, ...) and eye blinks were classified as such. In addition, an artifact detection system was implemented. A white circle above the green fixation cross was displayed above the feedback for the current trial, if an artifact was detected between seconds 2 to 5. This information told the participant that the feedback presented was corrupted – see Figure 5.3b for example.

Regarding the type of thought, the subjects were instructed to either think of making a fist or squeezing an anti-stress ball for imagination of left hand movement. For both feet they were told to imagine either pedal movements with their feet or pressing against the floor. For details please see the study information sheet in Appendix B .

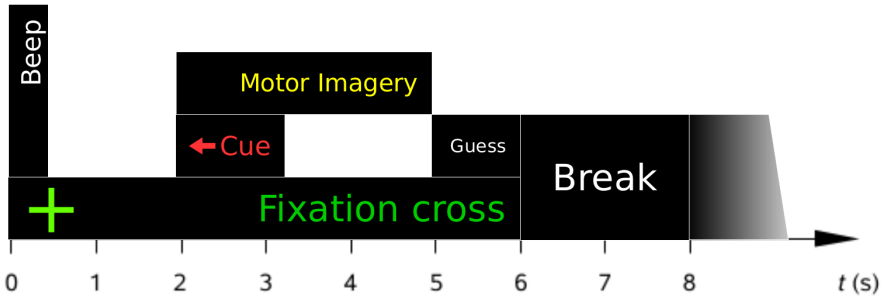
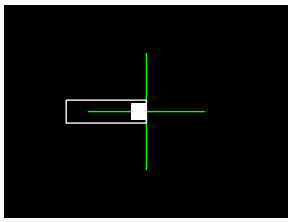
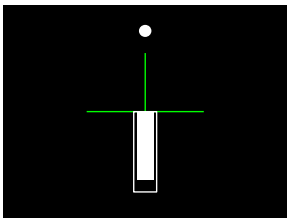


Figure 5.2: Timing of one trial in particular. The cues were presented as red arrows pointing to the left for imagination of left hand and down for the feet respectively. The break between two consecutive trials was chosen randomly to last 2 to 3 seconds.



(a) Uncertain; left hand.



(b) Certain; both feet; detected artifact.

Figure 5.3: Visualization of the way how feedback was provided. A white frame indicates the target class. The fraction of the classifier's output probability for the target exceeding 0.5 was displayed as a white bar. A white dot above the green fixation cross tells the participant that an artifact was detected.

During electrode montage videos of a pre-run, a run with artificially generated random *EEG* and a good performer were presented to a new participant. He/she was instructed to use the time for training and memorizing the sequence of events.

5.2.3 Data Recording

Thirteen Ag/AgCl electrodes were used to record *EEG*. They were mounted according to the international 10/20 system at positions FC3, FCz, FC4, C5, C4, C1, Cz, C2, C4, C6, FC3, FCz and FC4. The signals were recroded monopolarly with the left mastoid as reference and the right one as ground. This setup was chosen because the offline dataset was recorded in this fashion [47]. The data were sampled at a frequency of 512 Hz and band-pass filtered between 0.5 Hz and 100 Hz. An additional 50 Hz notch filter was employed to suppress line noise. An additional *EOG* electrode was placed at the center of the forehead to ease detection of eye blinks.

5.2.4 System Overview

A simplified block diagram of the adaptive system's core is sketched in Figure 5.4. The logic, which creates the sequence of events for each run, is also employed to trigger individual blocks. As already mentioned, a single feature vector is extracted per trial. It is sampled at $t = 5.0\,s$ – see Figure 5.2. In addition, the trial's label as well as whether there was an artifact in the relevant time window is stored. This information is used to update the class' statistics if no artifact happened.

The sampled observation is temporarily stored in a buffer. Every 10th trial the `adapt` event is issued. In that case the buffer's content is sent to the adaptation algorithm. It incorporates the new data into its trianings-set. A new classifier is sent back before the next trial starts.

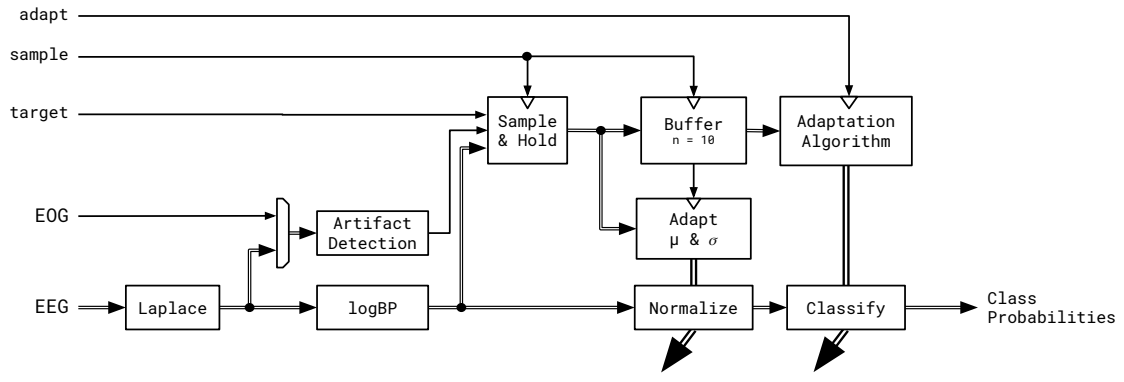


Figure 5.4: Block diagram of the online BCI system. The events, triggering the blocks, are derived from the paradigm displayed in Figure 5.2. The `sample` event, for instance, is generated at $t = 5.0$. `target` stands for the type of thought for the current trial. The `adapt` event is triggered every 10 trials when the buffer gets full.

5.2.5 Feature Extraction

As already pointed out, a *RBM* was pre-trained on the offline dataset used in the previous chapters. To maintain comparability, the feature extraction chain was not altered. Hence, the 13 *EEG* channels were spatially filtered using Laplacian derivations for C3, Cz and C4. Similar bandpasses as described in Table 2.1 were employed to extract 2 frequency-bands for each spatially filtered signal. Their band-power was estimated with a 2 s long window.

5.2.6 Normalization

The results presented in Figure 2.4a indicate that the estimation of baseline activity based on a preceding block with rest and eyes open condition tends to be biased. This issue is tackled through the introduction of a short pre-run experiment. It consists of 2 trials (one for each class in random order) following the timing scheme displayed in Figure 5.5.

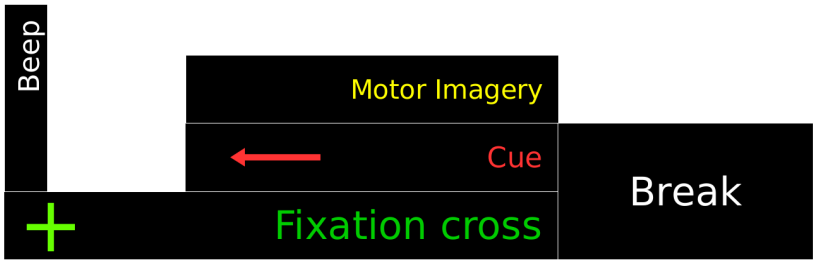


Figure 5.5: Pre-run experiment: Timing. At $t = 0\,s$ an auditory warning and green fixation cross mark the beginning of a trial. At $t = 2.0\,s$ a red arrow indicates the type of thought. It stays on the screen for 17 s. A black screen indicates a 5 s long break.

The participants were asked to start with motor imagery of the indicated class as soon as they see the cue. A 16 s long window starting at $t = 3.0\,s$ was extracted to compute 8 successive feature vectors for each type of thought. Rather than estimating the overall mean and standard-deviation, the within-class values were calculated.

Adaptive estimation of first and second order statistics per class was already successfully applied in [50]. The advantage for training is that if randomly several consecutive trials of just one class are generated, the overall method will get biased towards the mean of this class. If the class means are estimated instead, the order of trials has no influence. Consequently, slightly better performance is expected.

Equations 2.1 and 2.2 can be readily adapted to compute μ and σ per feature and class. Since this is a supervised method, the label of a new trial decides which class-statistics to update.

Based on equation 2.5 the forgetting factor λ for adaptive class estimates can be determined. For this experiment the number of trials per class and run has roughly doubled compared to the offline dataset but also the number of classes has halved. That is, the trials per run stay roughly constant. And so does the time a run lasts. Hence, λ was again

chosen so that a run's trials cover 90 % of the weight fraction. This results in a value of 0.9 for 20 trials per class.

The averages of within-class means and standard-deviations were computed to standardize a new feature vector.

5.2.7 Adaptation Algorithm

In essence the algorithms used for fine-tuning in chapter 4 were applied here as well. That means, an *ANN* was initialized with the weights of the pre-trained *RBM* in the first run. After recording 10 trials[3], the model was trained by backpropagating the errors exploiting conjugate gradient descent and Armijo's step-size selection rule. The resulting classifier was sent back to the online system.

For all subsequent blocks of 10 trials, the composition of the training-set changed a bit. Early simulations showed that if previous blocks are also used for training, better results can be achieved. Hence, the new block was appended to the older ones. However, the network was already trained on them in an earlier stage. To compensate this effect, error weights were introduced. They scale the contribution of a single observation's error to the total cross entropy error[4]. Consequently, exponentially decaying weights were applied. A block's weights $w_{err}(b)$ are given by

$$w_{err}(b) = \lambda_{err}^{(N_{blocks}-b)} \tag{5.1}$$

where b stands for the block's index, N_{blocks} the number of blocks received so far and λ_{err} for the decaying factor. For example, the newest block's observations' error weights are 1.0. Simulations on the offline dataset resulted in choosing $\lambda_{err} = 0.7$ and removing blocks for which $w_{err}(b) \leq \lambda_{err}^4$ holds. I.e. only the 4 most recent blocks were kept in the training-set. This procedure was also continued across runs.

Almost all model parameters of chapter 4 were taken over. Solely, the number of epochs was reduced from 100 to 25 since the *ANN* is updated 4 times during a run.

5.2.8 Artifact Detection

Inverse filtering is a common technique to detect muscular activity. As pointed out in [43] the *EEG* can be described by an Adaptive Autoregressive (AAR) process. Its parameters can be identified through a Finite Impulse Response (FIR) adaptive linear prediction filter. In this setup the adaptive filter is also called inverse filter. Here, Recursive Least Squares (RLS) was employed as estimation algorithm.

Consequently, the filter's error signal contains the part which it is unable to explain. When the *EEG* is corrupted with muscle activity the power of the error signal rises. Simple

[3] Before a block was added to the training-set, its features were standardized using the most recent estimates for μ and σ.

[4] The total error over the entire training-set is used for backpropagation.

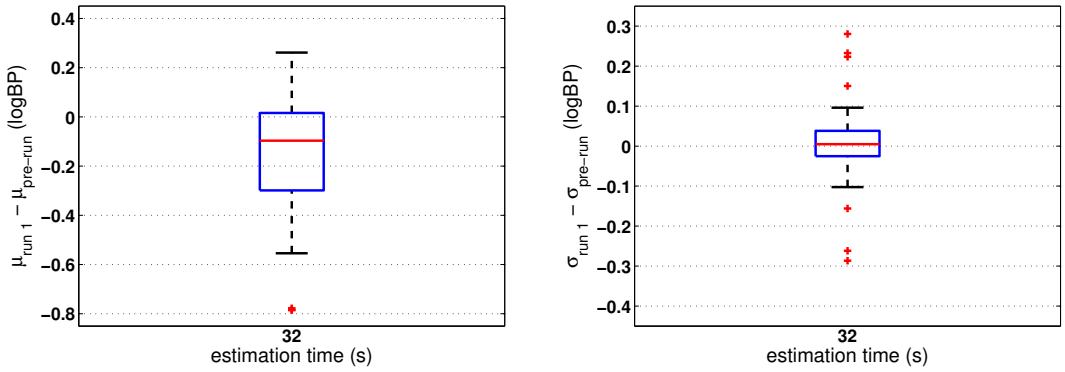
thresholding can then be applied to detect it. For further theoretical and technical details we refer to [43].

For this *BCI*, the *RLS*-algorithm was trained on the pre-run. In particular, the spatially filtered signals of the two 16 s blocks were used. Once trained, its adaptation rate was reduced significantly so that it would adapt slowly during the 4 training-runs. If the power of the filter's error exceeds 3 times its average level, an artifact event is triggered. The threshold was chosen based on simulations with a previously recorded and annotated dataset.

For eye blink detection the single *EOG* channel was utilized. The procedure is similar except that no adaptive filter was used and the threshold was set to 4 times the average power level. Its purpose is mainly for an easier computation of *ERDS*-maps because the artifact signal was also recorded and saved. Hence, trials with eye blinks in the reference interval or motor imagery phase could be discarded readily.

5.3 Results

The *RBM* was trained on standardized data. I.e. its decision boundary is optimal for zero-mean and unit-variance features. Thus, ideally the initial estimates of mean and standard-deviation per feature are accurate. In this chapter the paradigm for their estimation was slightly adapted. Figure 5.6 depicts box-plots for the differences in first and second order statistics of non-normalized features between first run and pre-run. They summarize the factors participant and feature similar to Figure 2.4.



(a) Differnece in means. **(b)** Difference in standard-deviations.

Figure 5.6: Boxplots of the difference between raw feature values (logBP) of the training-session's first run (`run1`) and the 32 s long block extracted from the preceding pre-run (`open`) over all 12 participants and features.

The performance of the presented artifact detection method was assessed using a previously recorded dataset . The recordings of 6 participants with a total of $N = 431$ trials were manually inspected and artifacts annotated. Thereafter, the system was applied to do the same. Table 5.1 lists their confusion matrix. After inspecting the raw signals, we can say that the 33 false negatives were mainly eye blinks. Nevertheless, sensitivities and accuracies of around 80% are acceptable [44].

TN	FP	FN	TP	N	Sensitivity	Accuracy
245	31	33	122	431	0.79	0.83

Table 5.1: Confusion matrix of the artifact detection system versus the manually annotated ground truth. $N = 431$ trials of 6 participants were utilized. `TP` sums the matches for the case that both detected a trial corrupted by an artifact. The `sensitivity` states the probability that the system identifies an artifact successfully given it was manually marked.

In the online study the artifact detection mechanism discarded on average around 10 out of $4 \times 40 = 160$ trials per user – see Table 5.2. The table lists the individual's Kappa values as well. They are summarized over runs in the columns and subjects in the rows

respectively. Where the last row considers only naive users. They achieved a mean Kappa value of 0.57 ± 0.35[5]. The 95% confidence intervals for chance level are 0.0 ± 0.31 for a single run and 0.0 ± 0.16 overall runs. Hence, participants 7 and 10 performed at chance level. The 8 other naive users' Kappa values exceeded the limit of 0.4 (= 70% accuracy in this setup), in the 3rd and 4th run.

Boxplots of naive subject's Kappa values were computed to estimate the underlying distribution – see Figure 5.7. One can see that the performance increased during runs. The last box summarizes the values of the entire session. One can observe that the session's distribution is skewed. Its median is 0.68, which is considerably higher than the mean (0.57).

The experiment was repeated again in an offline simulation. Instead of adapting the model the pre-trained *RBM* was utilized as classifier. The results of this retrospective simulation are listed in the last column of Table 5.2. A paired Wilcoxon signed rank test between the adaptive and non-adaptive method resulted in a p-value of $p = 0.016$. I.e. the adaptive method performed significantly better.

Figure 5.8 depicts colored boxplots of normalized log-bandpower values for all subjects of the offline dataset (both sessions) and the online experiment (first run). The black-gray pairs were generated through sampling from the pre-trained *RBM*. One is able to see that the power tends to be smaller for imagination of left hand condition across all features and subjects (offline and online). Moreover, the *RBM* models the offline dataset's distribution per feature very well.

This capability is reduced during discriminative adaptation. Figure 5.9 visualizes the evolution of samples drawn from the fine-tuned model for subject 8. For him/her one is also able to see a learning effect. The distance between the distributions of observations for the two classes increased during the training-session.

However, if the discriminative learning signal is small i.e. when the features generated by the user contain little discriminative information the model adapts only slowly – see Figure 5.10. The Figure summarizes the evolution of the C4 beta-band feature of user 7, who performed at chance level. For 1st and 2nd run the cyan and magenta boxplots overlap considerably. That is, the user was not capable of modulating the C4 beta-band. As a result the distance between the distribution of samples shrunk slightly – see `rbm0` vs. `ann1` vs. `ann2`. For runs 3 and 4 the modulation worked a bit better resulting in an increased distance between the samples' medians.

[5] Which corresponds to an accuracy of $79 \pm 18\%$.

user	type		run 1	2	3	4	# trials	**overall adaptive**	**overall RBM**
1	exp		0.89	1.00	1.00	1.00	153	0.97	0.93
2	nav		0.83	0.95	0.82	0.53	140	0.78	0.71
3	nav		0.84	0.79	0.65	0.85	156	0.78	0.67
4	exp		0.85	0.90	0.78	0.95	152	0.87	0.72
5	nav		0.32	0.90	0.80	0.90	157	0.73	0.73
6	nav		0.21	0.15	0.42	0.54	148	0.33	0.01
7	nav		0.15	-0.16	0.03	0.16	141	0.04	0.02
8	nav		0.79	0.78	0.94	1.00	145	0.88	0.79
9	nav		0.55	1.00	0.95	0.85	154	0.84	0.56
10	nav		-0.29	0.04	-0.08	-0.06	143	-0.09	-0.10
11	nav		0.35	0.47	0.58	0.64	144	0.54	0.44
12	nav		0.76	0.95	0.73	0.69	150	0.78	0.80
	all	μ	0.52	0.65	0.64	0.67	149	**0.62**	**0.52**
		σ	0.36	0.40	0.33	0.32	5.9	0.35	0.35
	nav	μ	**0.50**	**0.59**	**0.58**	**0.61**	**148**	**0.57**	**0.46**
		σ	0.37	0.41	0.34	0.32	6.2	0.35	0.35

Table 5.2: Kappa values for all 12 participants of the online study. The 10 naive subjects are marked by `nav`; experienced ones by `exp`. Each user's Kappa per run as well as the mean (`overall adaptive`) is listed. Mean and standard-deviation across all or only naive subjects are stated in the last rows. The column `overall RBM` was computed offline. It states a users's overall Kappa value if the pre-trained *RBM* would not have been adapted.

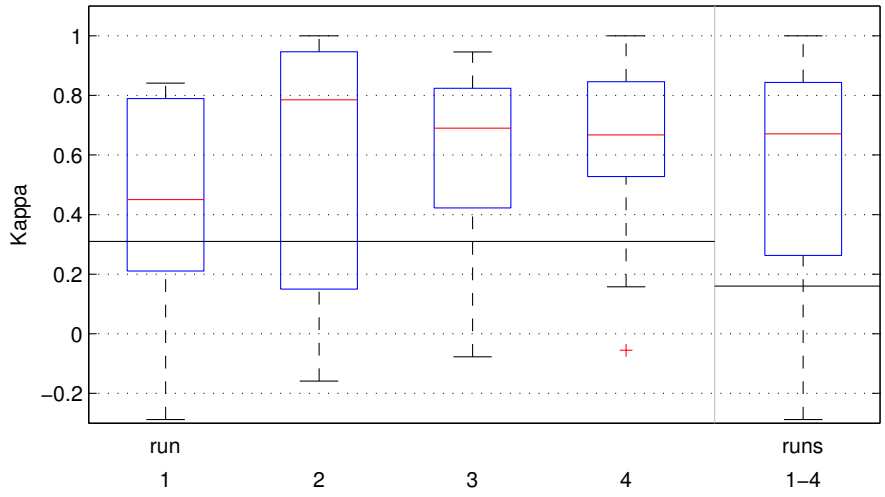


Figure 5.7: Visualization of the Kappa values of Table 5.2 across **naive** subjects. The first 4 boxplots summarize a single run, while the last combines all. Black horizontal lines highlight the 95% confidence interval limits for chance level.

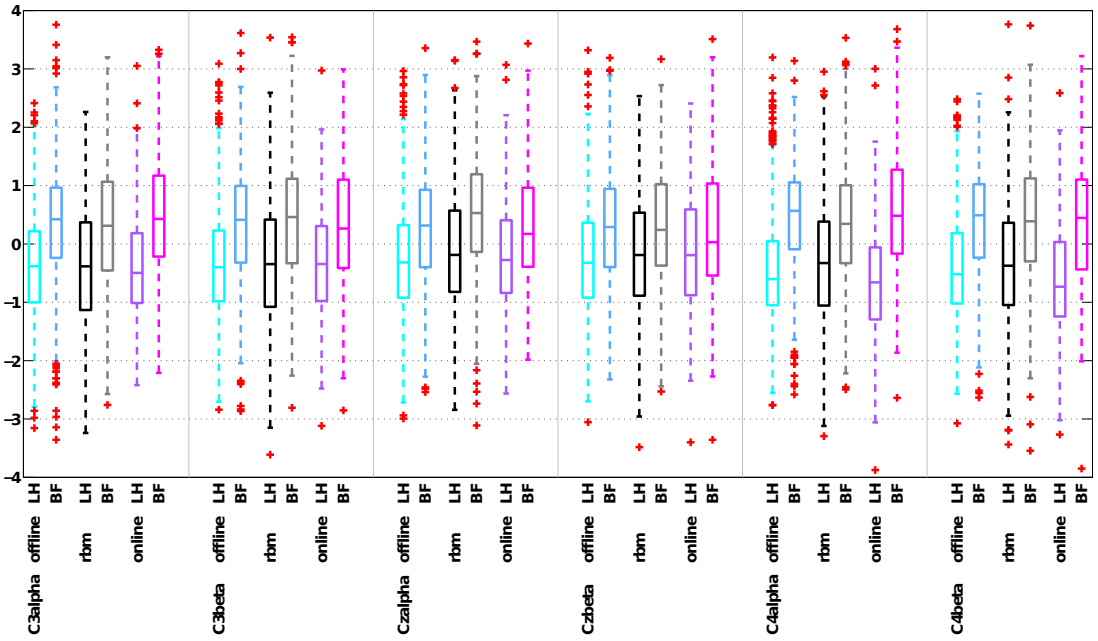


Figure 5.8: Boxplots of standardized log-Bandpower values. For each feature (`C3alpha` to `C4beta`) 6 boxplots are displayed. The first two summarize the offline-dataset's values across all sessions and subjects for both classes (`LH` Left Hand, `BF` Both Feet). The next 2 were computed from samples of the *RBM* trained on the offline dataset. The last 2 stand for the 1st run of the online experiment across all users.

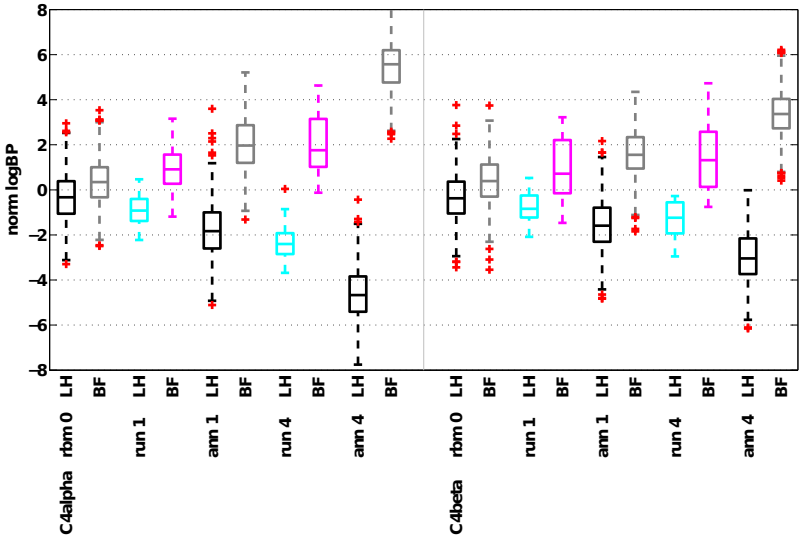


Figure 5.9: Evolution of the samples generated through discriminative training based on participant 8's recordings. Two features were selected for demonstration. For each 3 black-gray pairs of boxplots summarize the samples of the pre-trained *RBM*, *ANN* after run 1 and 4. Whereas the cyan-magenta pairs show the adaptively normalized log-Bandpower values of run 1 and 4.

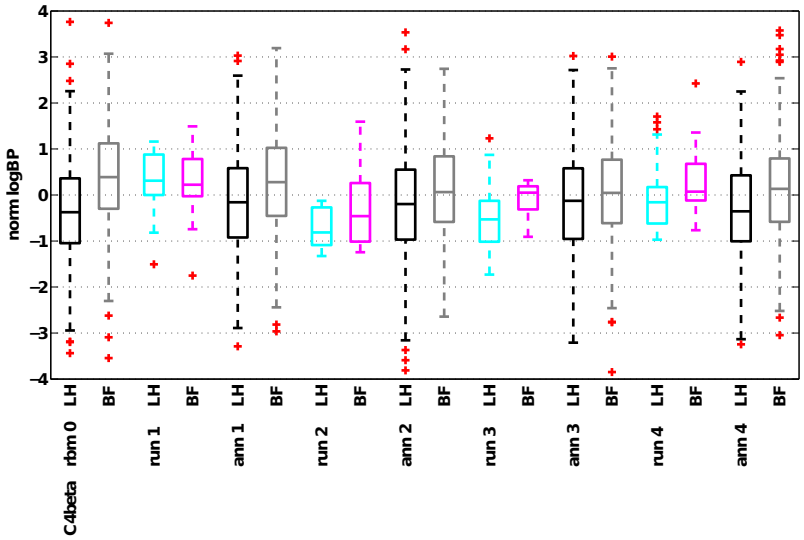


Figure 5.10: Evolution of features for the C4 beta-band generated by participant 7. For whom the system performed at chance level. His/her features per class are plotted across runs 1 to 4. Similar to Figure 5.9 samples generated by the pre-trained model (`rbm0`) as well as the updated model after each run are summarized by black and gray colored box-plots.

5.4 Discussion

In this chapter an online *BCI* system, incorporating the findings of preceding chapters, was presented. Key concepts were (1) generative pre-training of a *RBM* on an offline dataset (2) adaptive fine-tuning it discriminatively on an individual user (3) exploiting the pre-training to give feedback from the first trial onwards.

Since the system's feedback is only as good as its weakest part, accurate initial estimates for mean and standard-deviation are of importance. Additional to gazing at the center of the fixation cross, continuous imagination of both tasks was used in the pre-run. The effect caused by the changed paradigm can be neglected, if one compares Figure 5.6a to 2.4a and 5.6b to 2.4b. To conclude, performing motor imagery rather than only looking at the green cross with open eyes did not result in better initial estimates.

However, the participants were also asked to avoid artifacts (including eye-blinks), so that the artifact detection model could be trained on the pre-run. This could have forced participants to focus on not to blink with their eyes rather than motor imagery. Another possible explanation for the bias of the estimator could be that the participants got more involved as soon as they were confronted with the system's feedback.

Starting with feedback of a subject-independent classifier and co-adaptive training has recently achieved promising results [50]. Although different methods were employed here, one can also see steady improvement over the course of co-adaptive training across participants – see Figure 5.7. This results in a skewed distribution of the session with a median Kappa value of 0.68 (= 0.84% accuracy).

The two users who performed at chance level are 7 and 10. Their *ERDS*-maps, displayed in Figures A.6 and A.7, show that the co-adaptive training did not result in distinguishable patterns in the relevant time window $t = [1.0\,s, 3.0\,s]$ after the cue. Immediately, after the experiment all users were asked for feedback. User 7 answered that he/she conducts autogenetic training regularly and associated the mental tasks with it. Which resulted in getting in a relaxed and sleepy mental state. Participant 10's answer was that he/she had a hard time to relax his/her jaw because the associated muscles corrupted the signals of electrodes C5 and C6 substantially.

Very interesting are also the result of user 6. If the model would not have been adapted, he/she would have performed on chance level too. Figure A.5 displays the *ERDS*-maps. The upper β-band of Cz is the most discriminative band. However, the broadband feature extraction filter combines mid and upper β-band which is not optimal for this subject. Nevertheless, co-adaptative learning improved the performance substantial over runs.

Another interesting observation is that for the average subject (online and offline) left hand motor imagery results in lower band-power (α and β) than both feet – see Figure 5.8. The samples drawn from the *RBM* model this accurately due to the generative training criterion.

A beneficial property of the pre-trained and adapted model is the dependency of adaptation on the strength of the learning signal. On the one hand, if there is a lot of discriminative information like for user 8 – see Figure 5.9 – the classifier adapts quickly. On the other hand, if there is little information like for user 7 – see Figure 5.10 – the classifier adapts slowly. That is, the system is somewhat patient with poor performers and encourages them to generate already known patterns[6] through positive feedback.

[6]Extracted from the users in the offline dataset by generative training.

6 Conclusions & Future Work

6.1 Conclusions

Although *SMR* based *BCI*s have been introduced already in the 1990s, they still have not succeeded to get out of the laboratory environment to real world applications. We referred to 3 major hurdles which have to be overcome. (1) *BCI*-training can take a long time until a user generates stable patterns. (2) Due to the *BCI* inefficiency issue a non-negligible fraction of users might not be able to get control at all. (3) The bandwidth of *EEG* based systems is relatively low.

In recent years scientists put effort into pushing the training time down. It is only lately that impressive results were achieved with co-adaptive systems. For example by (1) Faller et al. in [18] – 10 out of 12 naive participants were able to operate a 2-class system above 70 % accuracy after 2 to 3 training-session. (2) Vidaurre et al. in [51] – 5 out of 10 users, who were unable to control a *SMR BCI* before, exceeded the 70 % threshold within a single-training session.

In this work, *generative* transfer learning in the context of co-adaptive training was investigated. Based on the findings of Chapter 3 the *RBM* was utilized to extract representations of a mixture of users. In Chapter 4 we have demonstrated that this model can be fine-tuned to an individual. Even better, it benefits from the information of other users. A *discriminative* optimization criterion was chosen since better results were achieved for limited data in Chapter 2.

Lastly, all these observations were exploited to construct a 2-class online *BCI* that is capable of giving feedback immediately. Within a single co-adaptive training-session 8 out of 10 naive participants exceeded the 70 % accuracy threshold. Furthermore, the results reported in the simulated transfer experiments match with the Kappa values of the non-adaptive online system. That is, for (9 + 12) users studied, the pre-trained *RBM* achieved a mean Kappa value of 0.46. In other words, the transfered classifier worked with an accuracy of 73% on average. The adaptive fine-tuning improved this even significantly.

6.2 Future Work

The results of this work suggest that the transfer learning-approach based on *RBM*s is useful for brain oscillations based *BCI*s. An obvious next step would be to expand the training-set to even more subjects of other studies. Consequently, more complex models could be trained on more or different features without an increased risk of overfitting.

The combination of generative transfer learning and co-adaptive fine-tuning could also be utilized to train users to generate pronounced patterns for more than 2 mental states and, therefore, contribute to improve the bandwidth issue. Moreover, extending *RBM*s to support multiple classes is straightforward. This property was utilized to detect 3 distinct mental states in [4].

Very recently, [45] reported that a non-linear method was superior to *sLDA* for *SMR-BCI*s. Because of that, the non-linearity (and complexity) could be elevated by training deeper models such as Deep Belief Networks [25], Deep Boltzmann Machines [42] or Generative Stochastic Networks [6]. Also other types of generative models could be applied. Density forests, for example, since [45] exploited random forests which are a close relative.

In the long run, co-adaptive training is aimed to help handicapped individuals to reach a satisfactory level of *BCI* control in a faster and more motivating way.

Bibliography

[1] Allison, B. Z., Dunne, S., Leeb, R., Millán, J. D. R., and Nijholt, A. (2012). Recent and Upcoming BCI Progress: Overview, Analysis and Recommendations. In *Towards practical brain-computer interfaces: bridging the gap from research to real-world applications*, pages 1–13. Springer Science & Business Media. (page 2)

[2] Allison, B. Z. and Neuper, C. (2010). Could anyone use a BCI? In *Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction*, pages 35–54. Springer. (page 2)

[3] Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3. (page 10)

[4] Balderas, D., Zander, T., Bachl, F., Faller, J., Neuper, C., and Scherer, R. (2011). Restricted Boltzmann Machines as Useful Tool for Detecting Oscillatory EEG Components. In *5th International Brain-Computer Interface Conference*, pages 68–71. (page 5, 8, 9, 11, 21, 46)

[5] Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer. (page 15)

[6] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828. (page 46)

[7] Bengio, Y. and Lamblin, P. (2007). Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19(1). (page 5, 29)

[8] Berger, H. (1929). Über das elektrenkephalogramm des menschen. *European Archives of Psychiatry and Clinical Neuroscience*, 87(1):527–570. (page 1)

[9] Billinger, M., Daly, I., Kaiser, V., Jin, J., Allison, B. Z., Müller-Putz, G. R., and Brunner, C. (2013). Is it significant? Guidelines for reporting BCI performance. In *Towards Practical Brain-Computer Interfaces*, pages 333–354. Springer. (page 17)

[10] Bishop, C. (2006). *Pattern Recognition and Machine Learning*, volume 4. Springer. (page 10)

[11] Bishop, C. and Lasserre, J. (2007). Generative or discriminative? getting the best of both worlds. *Bayesian Statistics*, 8:3–24. (page 6, 21)

[12] Blankertz, B., Lemm, S., Treder, M., Haufe, S., and Müller, K.-R. (2011). Single-trial analysis and classification of ERP components-a tutorial. *NeuroImage*, 56(2):814–825. (page 11, 15)

[13] Blankertz, B., Sannelli, C., Halder, S., Hammer, E. M., Kübler, A., Müller, K. R., Curio, G., and Dickhaus, T. (2010). Neurophysiological predictor of SMR-based BCI performance. *NeuroImage*, 51(4):1303–1309. (page 3)

[14] Cho, K. (2011). *Improved learning algorithms for restricted Boltzmann machines*. PhD thesis, Aalto University. (page 5, 8)

[15] Cho, K., Raiko, T., and Ilin, A. (2010). Parallel tempering is efficient for learning restricted Boltzmann machines. In *Proceedings of the International Joint Conference on Neural Networks*. (page 5)

[16] Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46. (page 16)

[17] Curran, E., Sykacek, P., Stokes, M., Roberts, S. J., Penny, W., Johnsrude, I., and Owen, A. M. (2004). Cognitive tasks for driving a brain-computer interfacing system: a pilot study. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 12(1):48–54. (page 3)

[18] Faller, J., Vidaurre, C., Solis-Escalante, T., Neuper, C., and Scherer, R. (2012). Autocalibration and recurrent adaptation: towards a plug and play online ERD-BCI. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 20(3):313–319. (page 3, 31, 45)

[19] Friedrich, E. V. C., Scherer, R., and Neuper, C. (2012). The effect of distinct mental strategies on classification performance for brain-computer interfaces. *International Journal of Psychophysiology*, 84(1):86–94. (page 3)

[20] Grosse-Wentrup, M. and Buss, M. (2008). Multiclass common spatial patterns and information theoretic feature extraction. *Biomedical Engineering, IEEE Transactions on*, 55(8):1991–2000. (page 3)

[21] Grosse-Wentrup, M. and Schölkopf, B. (2013). A Review of Performance Variations in SMR-Based Brain? Computer Interfaces (BCIs). In *Brain-Computer Interface Research*, pages 39–51. Springer. (page 2, 5)

[22] Hammer, E. M., Halder, S., Blankertz, B., Sannelli, C., Dickhaus, T., Kleih, S., Müller, K. R., and Kübler, A. (2012). Psychological predictors of SMR-BCI performance. *Biological Psychology*, 89(1):80–86. (page 3)

[23] Hinton, G. (2010). A practical guide to training restricted Boltzmann machines. *Momentum*, 9(1):926. (page 8, 9, 15, 16)

[24] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., and Kingsbury, B. (2012). Deep Neural Networks

for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97. (page 6)

[25] Hinton, G., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18:1527–1554. (page 6, 46)

[26] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800. (page 6, 7)

[27] Hinton, G. E. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507. (page 6, 10, 29)

[28] Kindermans, P.-J., Tangermann, M., Müller, K.-R., and Schrauwen, B. (2014). Integrating dynamic stopping, transfer learning and language models in an adaptive zero-training ERP speller. *Journal of neural engineering*, 11(3):035005. (page 3, 23)

[29] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. (page 8, 10)

[30] Kübler, A., Kotchoubey, B., Kaiser, J., Wolpaw, J. R., and Birbaumer, N. (2001). Brain-computer communication: unlocking the locked in. *Psychological bulletin*, 127:358–375. (page 1)

[31] Larochelle, H. and Bengio, Y. (2008). Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543. (page 6, 9, 10)

[32] Larochelle, H., Mandel, M., Pascanu, R., and Bengio, Y. (2012). Learning Algorithms for the Classification Restricted Boltzmann Machine. *Journal of Machine Learning Research*, 13:643–669. (page 5)

[33] Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., and Arnaldi, B. (2007). A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of neural engineering*, 4:R1–R13. (page 5, 6, 11)

[34] Mason, S. G., Bashashati, A., Fatourechi, M., Navarro, K. F., and Birch, G. E. (2007). A comprehensive survey of brain interface technology designs. *Annals of biomedical engineering*, 35(2):137–169. (page 3)

[35] McFarland, D. J., McCane, L. M., David, S. V., and Wolpaw, J. R. (1997). Spatial filter selection for EEG-based communication. *Electroencephalography and clinical Neurophysiology*, 103(3):386–394. (page 2)

[36] Müller-Putz, G., Scherer, R., Brunner, C., Leeb, R., and Pfurtscheller, G. (2008). Better than random: A closer look on BCI results. *International Journal of Bioelectromagnetism*, 10:52–55. (page 32)

[37] Pfurtscheller, G. and Aranibar, A. (1977). Event-related cortical desynchronization detected by power measurements of scalp EEG. *Electroencephalography and clinical neurophysiology*, 42(6):817–826. (page 2)

[38] Pfurtscheller, G. and Lopes Da Silva, F. H. (1999). Event-related EEG/MEG synchronization and desynchronization: Basic principles. (page 2)

[39] Pfurtscheller, G. and Neuper, C. (2001). Motor imagery and direct brain- computer communication. *Proceedings of the IEEE*, 89(7):1123–1134. (page 2)

[40] Polak, E. and Ribiere, G. (1969). Note sur la convergence de méthodes de directions conjuguées. *Revue française d'informatique et de recherche opérationnelle, série rouge*, 3(1):35–43. (page 10)

[41] Riener, R. and Seward, L. J. (2014). Cybathlon 2016. In *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, pages 2792–2794. IEEE. (page 2)

[42] Salakhutdinov, R. and Hinton, G. (2009). Deep Boltzmann Machines. *Artificial Intelligence*, 5:448–455. (page 46)

[43] Schlögl, A. (2000). *The electroencephalogram and the adaptive autoregressive model: theory and applications*. PhD thesis, Graz University of Technology. (page 36, 37)

[44] Schlögl, A., Keinrath, C., Zimmermann, D., Scherer, R., Leeb, R., and Pfurtscheller, G. (2007). A fully automated correction method of EOG artifacts in EEG recordings. *Clinical neurophysiology*, 118(1):98–104. (page 38)

[45] Steyrl, D., Scherer, R., Förstner, O., and Müller-Putz, G. R. (2014). Motor Imagery Brain-Computer Interfaces: Random Forests vs Regularized LDA-Non-linear Beats Linear. *Proceedings of the 6th International Brain-Computer Interface Conference*. (page 3, 11, 46)

[46] Tan, D. and Nijholt, A. (2010). Brain-computer interfaces and human-computer interaction. In *Brain-Computer Interfaces*, pages 3–19. Springer. (page 1)

[47] Tangermann, M., Müller, K.-R., Aertsen, A., Birbaumer, N., Braun, C., Brunner, C., Leeb, R., Mehring, C., Miller, K. J., Mueller-Putz, G., Nolte, G., Pfurtscheller, G., Preissl, H., Schalk, G., Schlögl, A., Vidaurre, C., Waldert, S., and Blankertz, B. (2012). Review of the BCI Competition IV. *Frontiers in Neuroscience*, 6. (page 12, 34)

[48] Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. (page 5)

[49] Vidaurre, C. and Blankertz, B. (2010). Towards a cure for BCI illiteracy. *Brain Topography*, 23(2):194–198. (page 3)

[50] Vidaurre, C., Sannelli, C., Müller, K.-R., and Blankertz, B. (2011a). Co-adaptive calibration to improve BCI efficiency. *Journal of neural engineering*, 8(2):025009. (page 2, 35, 42)

[51] Vidaurre, C., Sannelli, C., Müller, K.-R., and Blankertz, B. (2011b). Machine-Learning-Based Coadaptive Calibration for Brain-Computer Interfaces. (page 3, 31, 45)

[52] Welling, M. and Sutton, C. (2005). Learning in Markov random fields with contrastive free energies. *Artificial Intelligence and Statistics*, pages 397–404. (page 8)

[53] Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology*, 113:767–791. (page 2)

[54] Wolpaw, J. R., McFarland, D. J., Neat, G. W., and Forneris, C. A. (1991). An EEG-based brain-computer interface for cursor control. *Electroencephalography and clinical neurophysiology*, 78:252–259. (page 2)

[55] Wolpaw, J. R. and Winter Wolpaw, E. (2011). *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press. (page 13)

A Supplementary Material

A.1 Model evaluation

Grid Search

Due to the high dimensionality of the parameter space implying long simulation times, a less time consuming greedy approach was chosen. That is, all parameters but one were clamped. Then this one was varied within certain values. The optimal value, yielding highest Kappa across all participants, was chosen. Thereupon values of the next parameter were varied. After the last parameter the iterative procedure started all over again until convergence. The value ranges are listed in Table A.1.

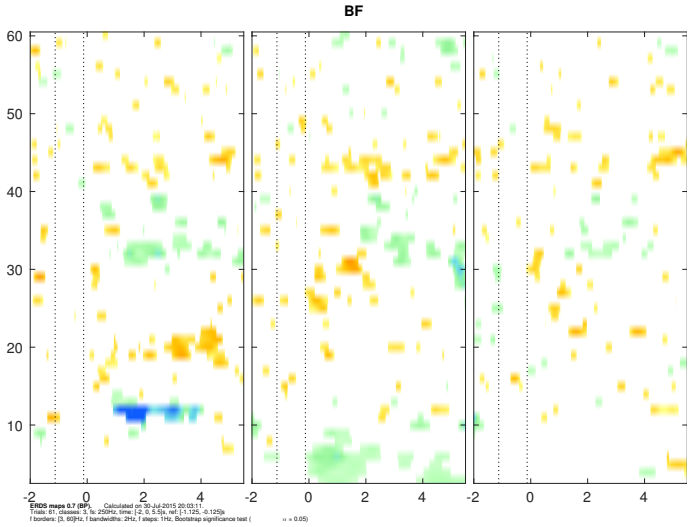
parameter	value range
$\lambda_{weights}$	0.01, 0.0075, 0.005, 0.0025, 0.001
`# units`	10, 50, 100, 500
λ_{class}	0.01, 0.0075, 0.005, 0.0025, 0.001
`# epochs`	25, 50, 100, 250, 500

Table A.1: Value ranges of model parameters used for optimization.

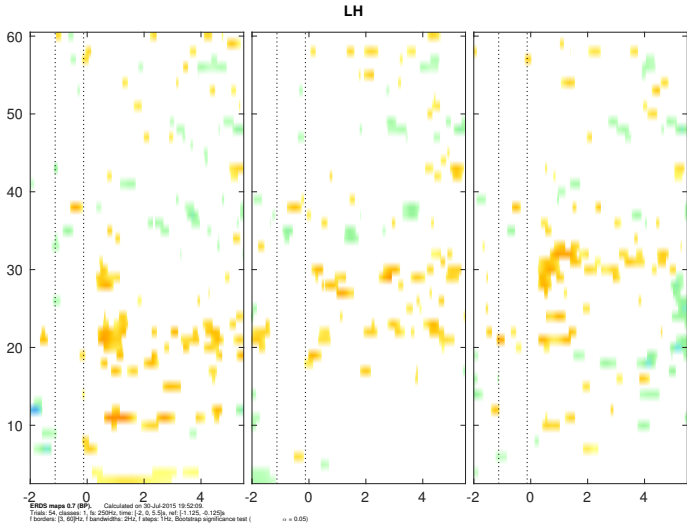
A.2 ERDS-maps

BCI Competition IV Dataset 2a

The subsequent ERDS-maps were computed for large Laplacian derivations around C3 (left), Cz (center) and C4 (left). At second 0 the cue was presented. It triggered either imagination of left hand (LH) or both feet (BF). The reference power was computed for the interval $t = [-1.125\,s, -0.125\,s]$ and *averaged* across all trials (LH & BF). The break after a trial started at $t = 4.0\,s$. Bootstrapping was applied to test the mean statistics. A significance level of $\alpha = 0.05$ was chosen.

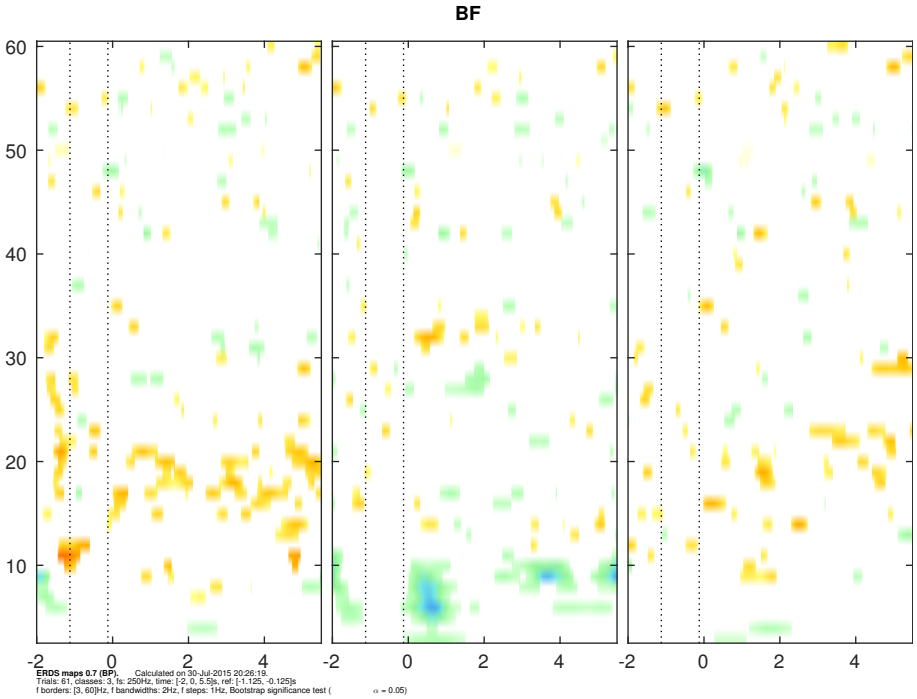


(a) Session 1 BF.

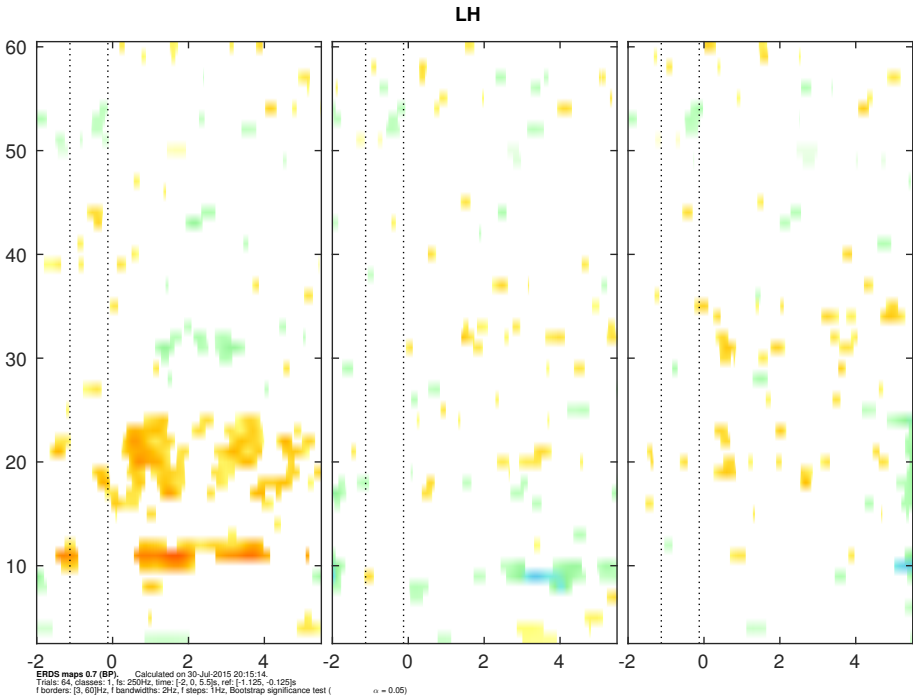


(b) Session 1 LH.

Figure A.1: ERDS-maps of session 1 for participant 5.

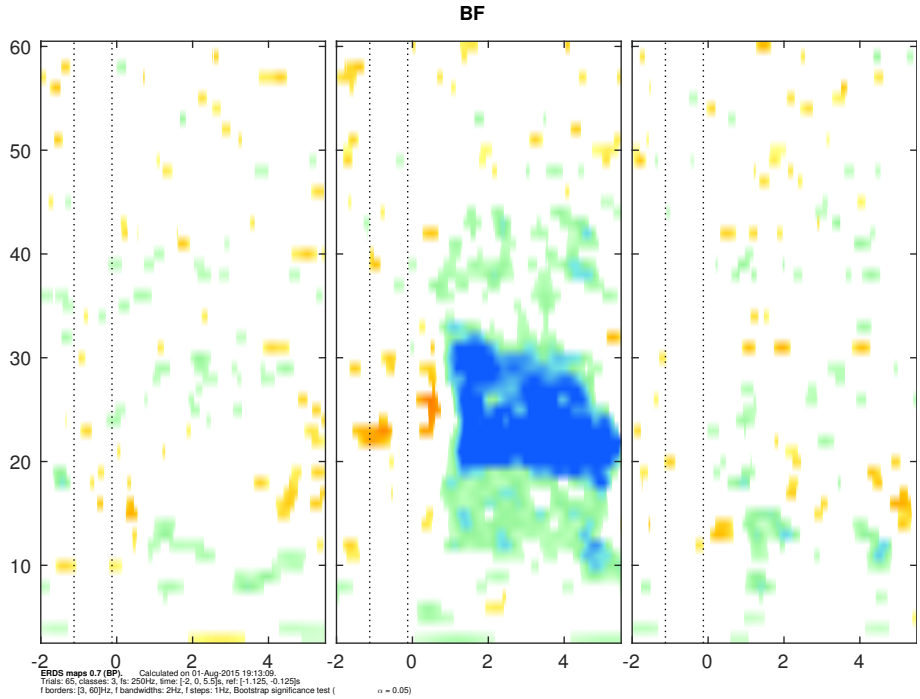


(a) Session 2 BF.

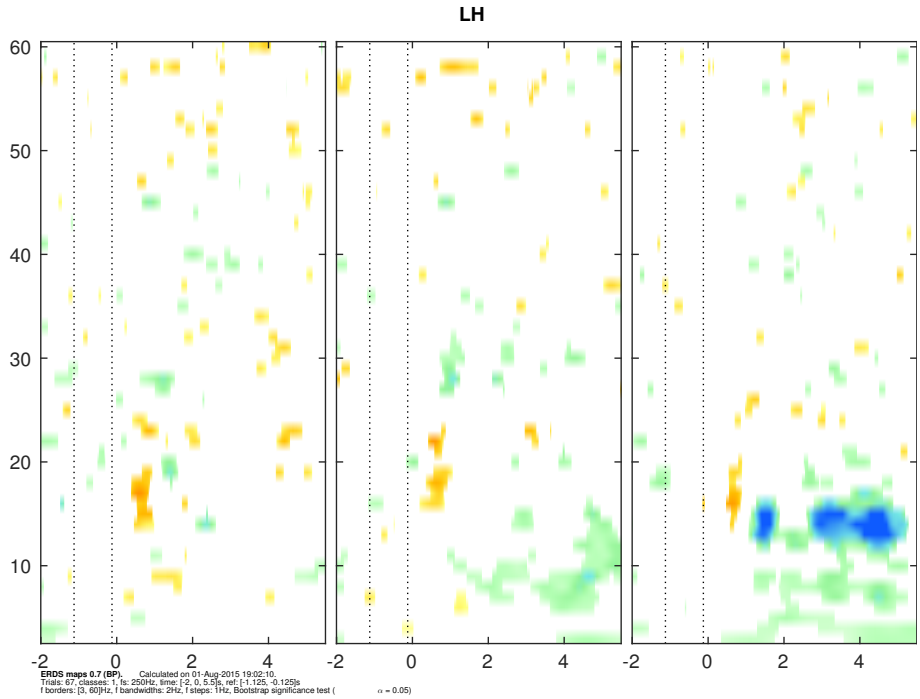


(b) Session 2 LH.

Figure A.2: ERDS-maps of session for participant 5.

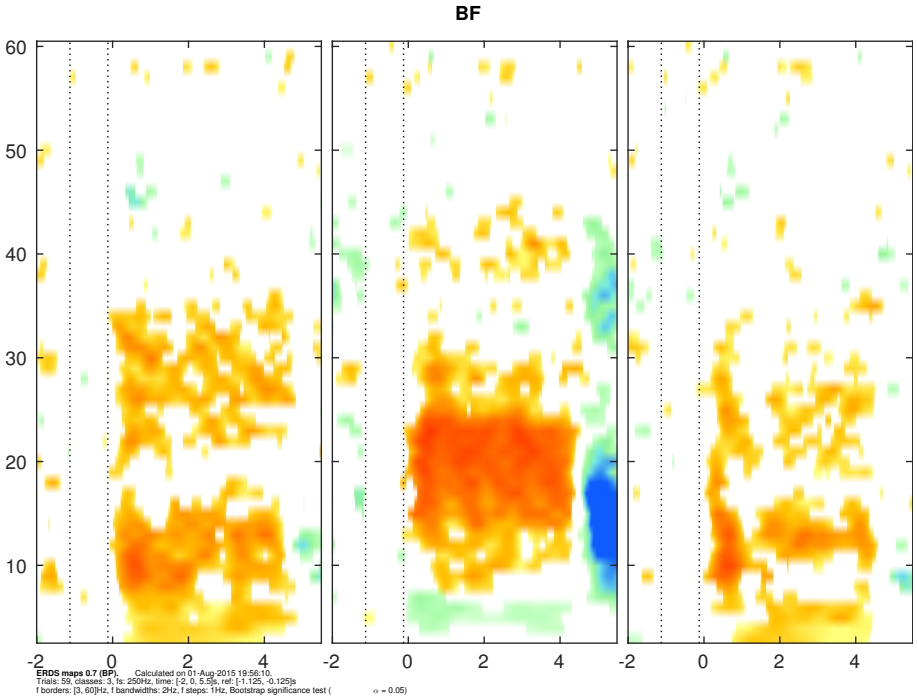


(a) Session 2 BF.

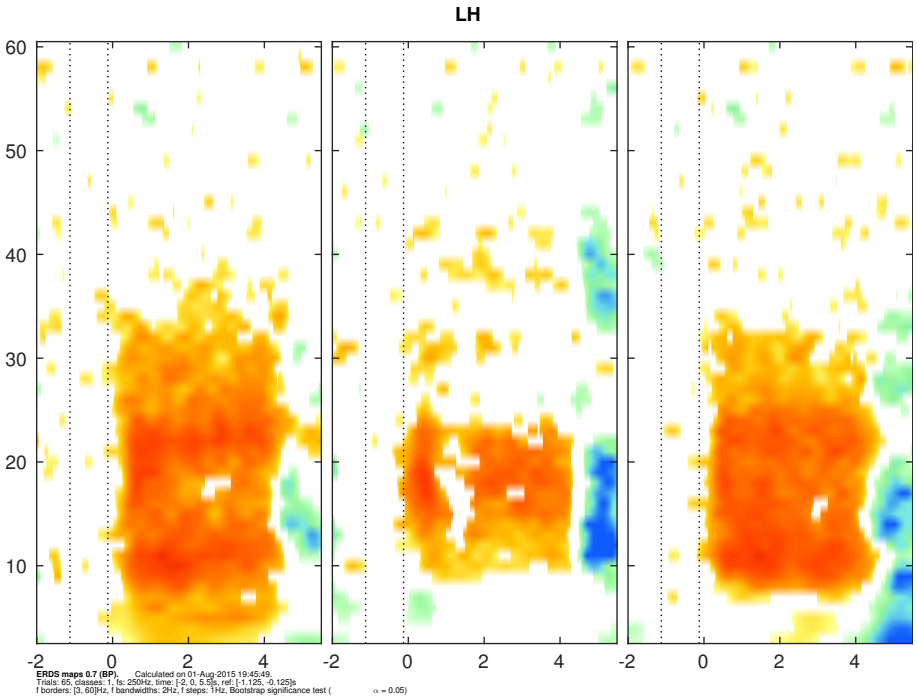


(b) Session 2 LH.

Figure A.3: Participant 2's ERDS-maps of session 2.



(a) Session 2 BF.

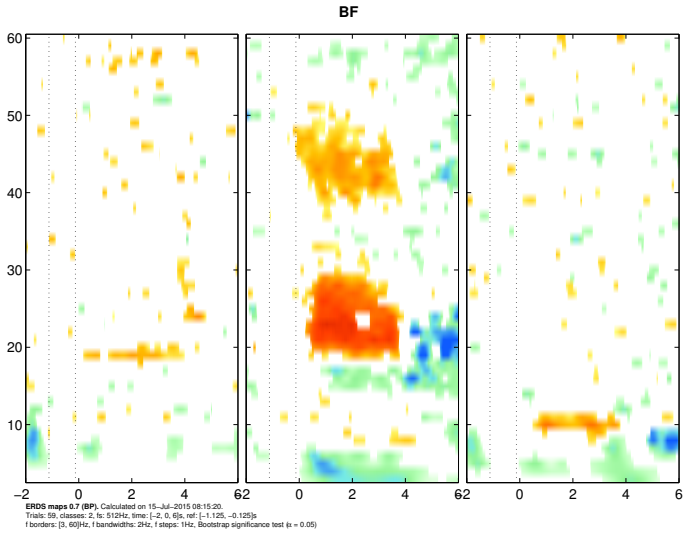


(b) Session 2 LH.

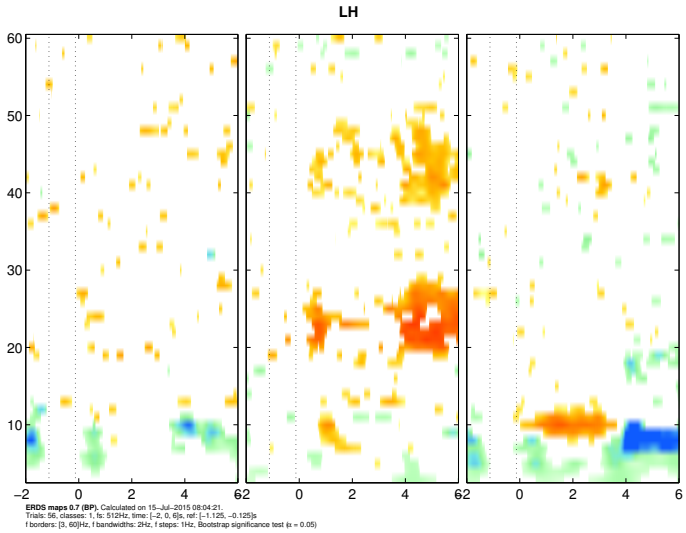
Figure A.4: Participant 7's ERDS-maps of session 2.

Online Study

The following ERDS-maps were computed for large Laplacian derivations around C3 (left), Cz (center) and C4 (left). At second 0 the cue was presented. It triggered either imagination of left hand (LH) or both feet (BF). The reference power was computed for the interval $t = [-1.125\,s, -0.125\,s]$. The break after a trial started at $t = 4.0\,s$. **The feedback was presented in the interval** $t = [3.0\,s, 4.0\,s]$. Bootstrapping was applied to identify significant changes. A significance level of $\alpha = 0.05$ was chosen.

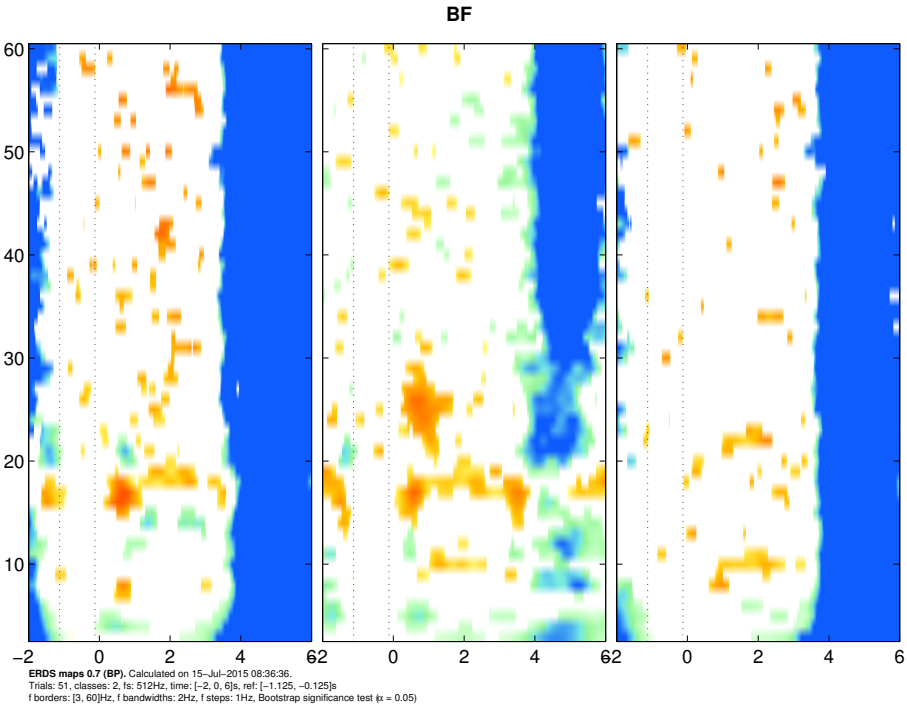


(a) Both Feet.

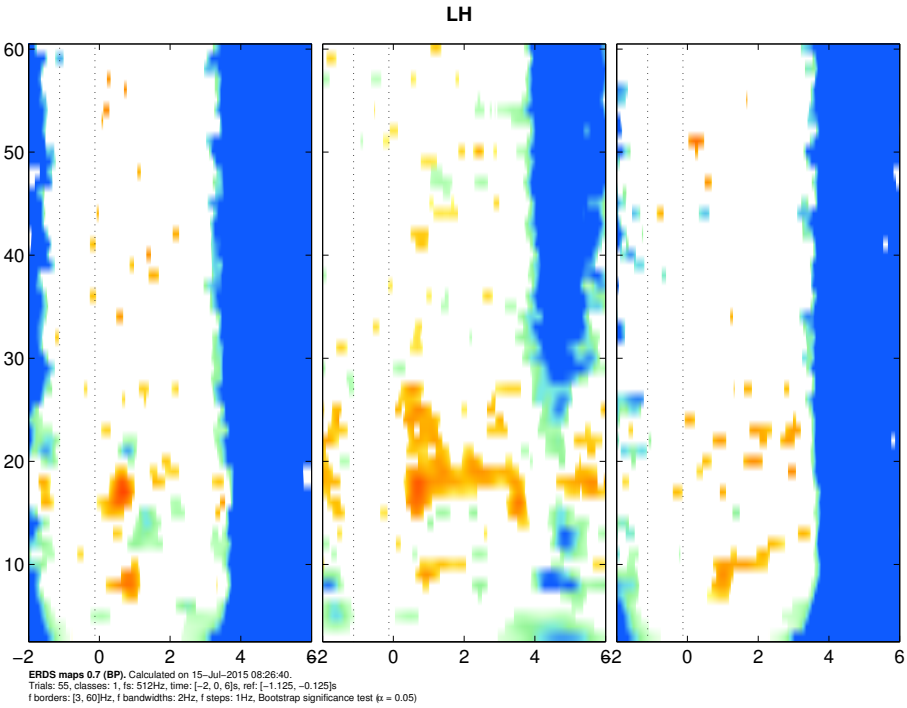


(b) Left Hand.

Figure A.5: Participant 6's ERDS-maps.

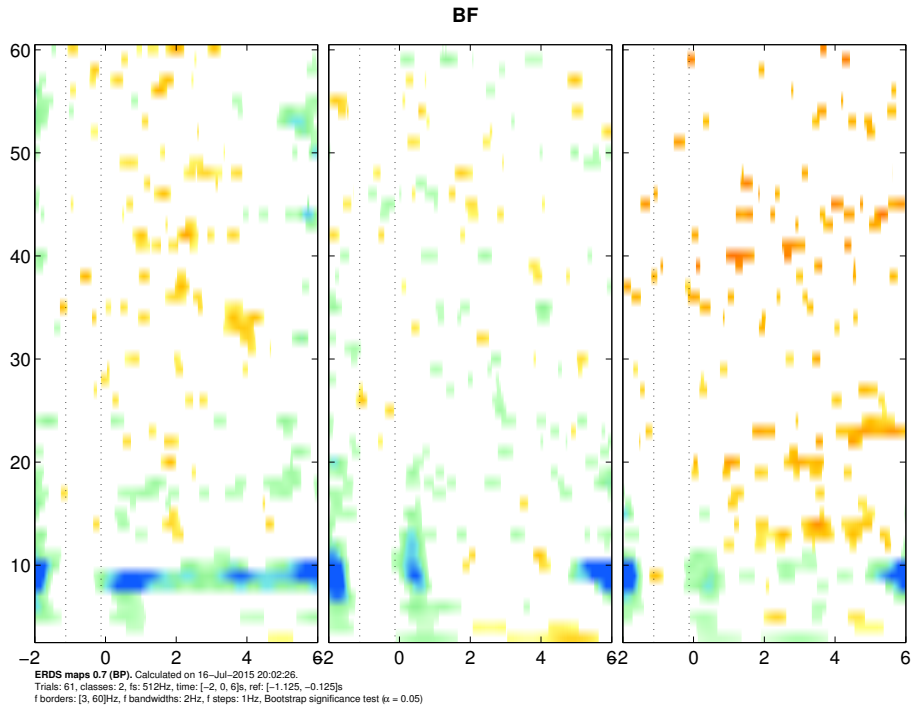


(a) Both Feet.

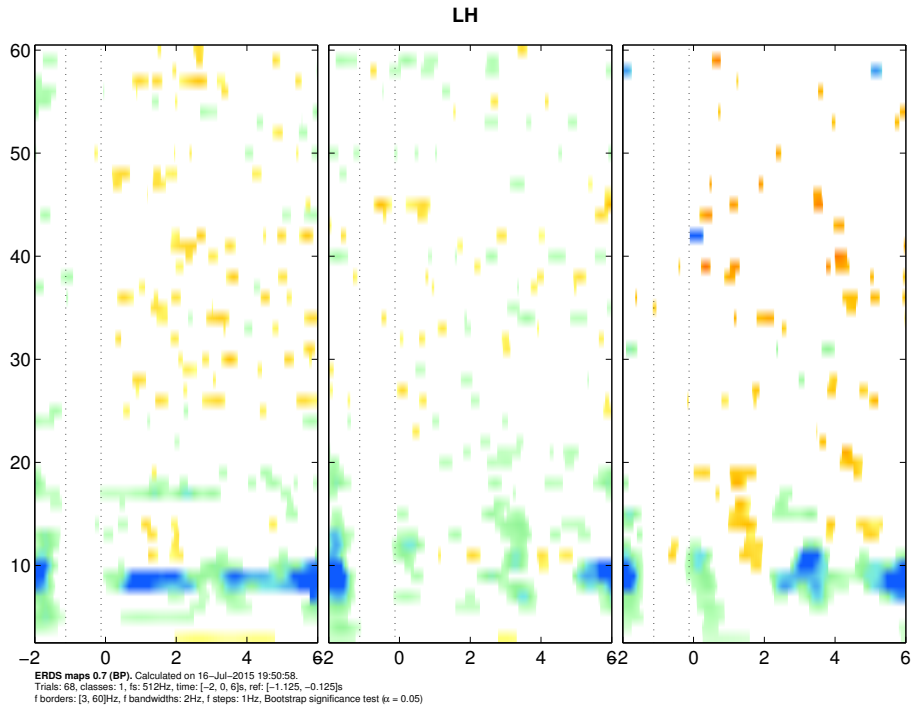


(b) Left Hand.

Figure A.6: Participant 7's ERDS-maps. He/she swallowed very often in the breaks between trials.



(a) Both Feet.



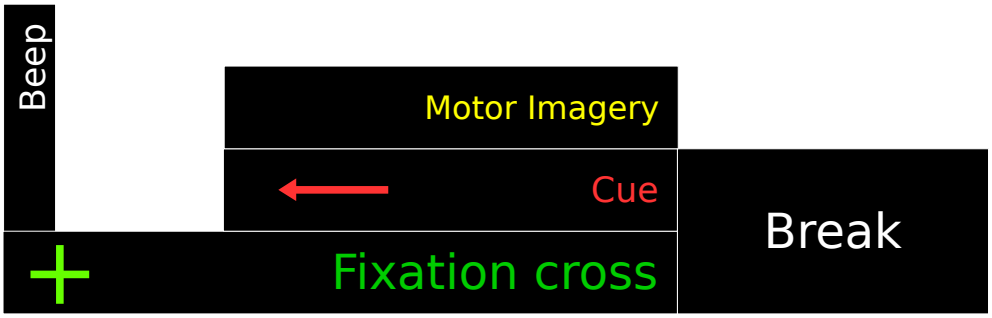
(b) Left Hand.

Figure A.7: Participant 10's ERDS-maps.

B

Excerpt of the Study-Information Sheet

Pre-Run: One Trial in particular



1. After starting the **pre-run** the screen will become black for a short time.

2. A green cross will appear in the middle of the screen. Simultaneously you will hear a short audio warning.

3. Additionally to the green cross a **red** arrow will appear on the screen. This arrow specifies the required body part/ imagination of movement.

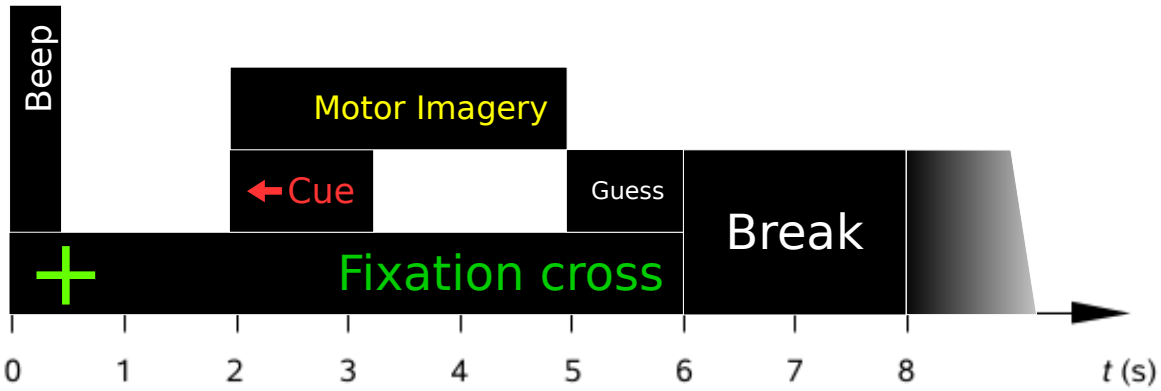
4. As soon as the arrow appears, you should start to imagine the movement of the required body part. The arrow will stay at the screen for ~15 seconds. Try to keep focused on the mental task during this period.

5. In addition, during you see the red arrow avoid producing artifacts. Artifacts are any type of **movements** (neck, eyes, tongue, jaw,...), clenching your teeth, **eye blinks.**

6. The green cross and white arrow will disappear – the screen will become black again – this indicates a short break which has a length of ~5 seconds.

7. This procedure will be repeated 1 time for the other type of thought.

Run 1-4: One Trial in particular



1. After starting this experiment the screen will become black for a short time.

2. A green cross will appear in the middle of the screen. Simultaneously you will hear a short audio warning.

3. Additionally to the green cross a **red** arrow will appear on the screen. This arrow specifies the required body part/ imagination of movement.

4. As soon as the arrow appears, you should start to imagine the movement of the required body part.

5. The red arrow will disappear after 1 second – the green cross will still be on the screen- continue the imagination until a white frame appears

6. The white frame/bar indicates the system's guess for the current trial. If the frame is empty, the system was not able to detect your intention. If there is a white bar in the frame the system was able to detect your intention. The length of the white bar indicates the system's certainty.

7. In addition, during you see the white bar a white circle will pop up if the system identified artifacts during the motor imagery phase. When you see a white circle the system's guess is corrupted!

8. The green cross and white frame will disappear – the screen will become black again – this indicates a very short break which has a length between 2 and 3 seconds.

9. This procedure will be repeated 40 times until a run is over.

Which movement should be imagined?

Here are some examples:

Right/left hand: making a fist; pressing an anti-stress ball

Legs/both feet: pressing and releasing a pedal; pressing against the floor

Execute different **voluntary** (not pre-programmed) movements for each body part. **Then imagine** the movements you executed before. Select the movement for each body part which you can imagine best in terms of vividness, concentration, repeatability.
You will soon get some minutes to execute and imagine several movements.

Attention

Start the imagination of the movement as soon as the red arrow shows you the required part of the body.
To precise the type of thought a little for you, we have a few tips:

→ Imagine a sustained (ununterbrochen) movement
→ Focus on kinesthetic/tactile sensation (put attention on what you (would) feel in your hand/feet)
→ It might also be helpful to picture the movement simultaneously from your perspective (1st person perspective)

You found the right one, if you think you can recall the thought/mental task often easily and that it is life-like in your brain.

It is IMPORTANT to maintain the chosen movements over ALL runs!!!

As already mentioned, the developed system is pre-trained on a dataset recorded of 9 different subjects. We try to exploit the learned patterns via transformation and adaption to your individual one during the entire 4 runs.

The **experiment** itself can be somewhat compared to the following situation: You try to **teach a baby** (the system) its first word (e.g. to say 'mama'). Your strategy is to repeat the word always in a similar way (**repetition**).
After each utterance you listen to what the baby says. Most likely it takes some time until you hear something similar like 'ma' or 'm' or 'mam' (this corresponds to occasional correct **but maybe random** guesses of the system). **Occasional hits** are okay but we want to get the entire word. So you stay focused on the pronunciation (type of thought) so that the baby hears the same over and over again (**patience**).
We are happy when we hear 'mama' for the first time (= a dozen of subsequent correct system guesses), however, as our baby (the system) is **forgetful** and **listening to all what you say** we stay focused on the task until we are absolutely sure (= end of experiment).

Code of behavior

•Try to sit comfortable and **avoid** movements during the measurement.

•Always **look at the center of the green cross** during each trial

•Put your hands on your lap/ or specific place.

•**Relax the muscles** of your face, neck, shoulders and lower jaw!

•Avoid clenching your teeth!

•Please reduce blinks and swallowing to a minimum during each trial- for those actions you should **use** the short **breaks** between the trials (black screen)!