Brein-computer-interfaces met kunstmatige intelligentie: een symbiotisch ontwerp

Brain-Computer Interfaces with Machine Learning: A Symbiotic Approach

Thibault Verhoeven

Promotoren: prof. dr. ir. J. Dambre, dr. ir. P.-J. Kindermans, dr. ir. P. van Mierlo
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: computerwetenschappen

UNIVERSITEIT
GENT

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. R. Van de Walle
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2016 - 2017

ISBN 978-94-6355-026-0
NUR 984, 954
Wettelijk depot: D/2017/10.500/61

Department of Electronics and Information Systems
Faculty of Engineering and Architecture
Ghent University

Internet Technology and Data Science Lab
iGent, Technologiepark-Zwijnaarde 15
B-9052 Ghent
Belgium

Medical Imaging and Signal Processing
Ghent University Hospital, Block B, De Pintelaan 185
B-9000 Gent
Belgium

Promotors:
Prof. dr. ir. Joni Dambre
Dr. ir. Pieter-Jan Kindermans
Dr. ir. Pieter van Mierlo

Examination board:
Em. prof. dr. ir. Daniël De Zutter, Ghent University, chairman
Prof. dr. ir. Pieter Simoens, Ghent University, secretary
Prof. dr. ir. Willem Waegeman, Ghent University
Prof. dr. Daniele Marinazzo, Ghent University
Dr. rer. nat. Michael Tangermann, University of Freiburg

This work was funded by a PhD grant of the Special Research Fund - Ghent University

Acknowledgement

The process of obtaining the PhD degree is a journey. In its literal sense it takes you to several places all over the world. The greater journey is however the expedition you undertake to the absolute edge of human knowledge, and beyond. During my journey I received generous support from many people, to whom I want to express my sincere gratitude.

First of all I would like to thank my promotor prof. dr. ir. Joni Dambre, for giving me the opportunity and freedom to perform this research in my own way. My deep appreciation goes to my co-promotor dr. ir. Pieter-Jan Kindermans for his endless support in every aspect of my PhD, for the interesting discussions, motivational talks and for giving me such a warm welcome in Berlin. I want to thank my co-promotor dr. ir. Pieter van Mierlo, for keeping me motivated with his enthusiasm and for introducing me to so many interesting people in the epilepsy research domain. I am grateful to the members of the examination board em. prof. dr. ir. Daniël De Zutter, prof. dr. ir. Pieter Simoens, prof. dr. ir. Willem Waegeman, prof. dr. Daniele Marinazzo and dr. rer. nat. Michael Tangermann for their advice and suggestions to improve this book. Furthermore I owe thanks to dr. ir. Pieter Buteneers, who guided me during the first years of my PhD, and to prof. dr. Stefaan Vandenberghe for introducing me into the Medisip research group.

Several people have contributed to the scientific content of this

PhD. In particular, I want to thank David Hübner and dr. rer. nat. Michael Tangermann, from the University of Freiburg, for the fruitful collaboration we had in the past years, for their hard work to run experiments so swiftly, for the valuable feedback on results and manuscripts and for making this project a success, now and in the future. I owe thanks to prof. dr. Klaus-Robert Müller, from the Technical University of Berlin, for giving me the opportunity to do part of my research in Berlin and for his insightful comments and suggestions. In addition, I would like to thank dr. Ana Coito from the University of Geneva for the pleasant cooperation on the epilepsy project and for the interesting (and sometimes philosophical) conversations we had.

Although the creative development of novel techniques is a 24/7 job, the most memorable moments of the PhD process are the ones you share with colleagues in and outside of the office. A special thanks to Willeke for being my friend, study buddy and colleague during the past seven years. We shared laughter and tears, but we made it (to Kempenhaeghe). Thank you for laughing with all my silly jokes (and for making even more silly jokes), and for being the best networking companion at conferences. We will never know if Michael Carmichael truly exists. I want to thank Gregor, for our occasional deep conversations, I truly enjoyed them. You are an inspiring person and I wish you all the best with Epilog. I am very grateful to Kim, Jens (mr. practical joker) and Emma, who brightened up the office in the final years of my PhD. I wish we could have spent more time together, although that would have probably endangered my deadlines. You are all very talented, make Medisip great again! Thank you Inge, for helping anywhere you could and for your eternal smile. Amir, for your kindness and your hard work. Stijn, for jumping in the black box of machine learning. I hope you will continue the integration of artificial intelligence in the medical field. Society counts on you (no pressure). Furthermore I would like to thank all the other colleagues from the Medisip, bioMMeda and the IDLab research groups for the interesting and amusing conversations during lunchtime, for the afterwork parties and for being such great colleagues.

De steun van familie en vrienden is voor mij van onschatbare waarde geweest bij het succesvol voltooien van mijn doctoraat. Om deze reden zou ik in de eerste plaats mijn ouders, grootouders en bij uitbreiding de rest van de familie willen bedanken voor hun onvoorwaardelijke steun. Ook zou ik Mathias en Raf, mijn voormalige huisgenoten, willen bedanken voor alle steun en om voor de nodige ontspanning te zorgen na het werk. Ik zou ook mijn vrienden van het FaculteitenKonvent willen bedanken, in het bijzonder Sam, Camille, Pieter-Jan, Helena en Jasper om al die mooie herinneringen te creëren en ze regelmatig nog eens op te halen, opdat we ze nooit meer zouden vergeten. Joost en Elise, om nog steeds van elke avond er eentje te maken om nooit te vergeten. Maarten, mijn studiegenoot, voor alle mooie momenten die we samen hebben beleefd. Ook de Moraalridders (Maaike, Kamie, Jens, Yoerik en Aäron), de Vrienden van de Vrijdag (Wim, Chanel, Anouk, Jordy, Jens, Vero, Matthias en Arvid), Pauline en Laura, bedankt voor alle gezellige momenten en om op tijd eens alles in het nodige perspectief te plaatsen.

Als laatste, maar belangrijkste persoon in dit hele verhaal zou ik Frisine willen bedanken. Je bent de enige die echt weet wat het gekost heeft om dit doctoraat tot een goed einde te brengen. Bedankt om mij te vergezellen op de vele trips naar het buitenland, voor alle steun, voor het geduld wanneer ik nooit tijd had en om steeds bereid te zijn te luisteren en naar oplossingen te zoeken.

Thibault Verhoeven
Ghent
May 2017

Nothing is impossible, at most improbable

Summary

The aim of this dissertation is to improve self-learning brain-computer interfaces by means of a symbiotic design approach in which the different components are co-adapted to each other. The second goal is to demonstrate how machine learning can contribute to the recognition of pathological brain activity and as such form a powerful tool for a fast and accurate diagnosis of neurological diseases.

Brain-computer interfaces

A brain-computer interface is a system that creates a direct communication link between the brain and a computer. Since their inception in 1973 by Jacques J Vidal, BCIs have been developed for many applications. BCIs that infer the user's intention from his/her brain activity allow patients with paralysis to control a communication system or wheel chair. Other BCIs monitor the brain state, for example to identify neurological diseases such as epilepsy.

A BCI is composed of three components: the user, the computer application and a decoder that translates the brain activity to control commands. BCI systems suffer from several issues that keep them from being used on a daily base. A major problem is the tuning of the decoder to the specific properties of the subject's brain activity. In traditional BCIs this is done in a separate calibration session before actual use of the system, which is very tedious for the subject.

The growing BCI community has put a lot of effort in improving the usability of BCIs. However, each proposed improvement still

regards the application, user and decoder as separate entities and optimises these subsystems individually. The interaction between these components is never considered, which limits the level of performance that can be obtained.

Event-related potential based brain-computer interfaces

To inform the computer about his/her intention, the subject can use neural control signals. In BCIs based on event-related potentials (ERP), the user is presented a sequence of different visual, auditory or tactile stimuli, each of which is linked to a command. The user has to focus on the stimulus corresponding to the desired control command. This causes his/her brain to respond differently when this so-called *target stimulus* appears. The decoder in the BCI has the task to classify the recorded ERPs as target or non-target responses and subsequently infer the desired command.

The most well-known ERP-BCI is the speller system introduced by Farwell and Donchin in 1988. In this BCI for communication, the stimuli are linked to the symbols of the alphabet. These symbols are highlighted on a screen to form visual stimuli. By choosing to attend different symbols, the user can spell words and sentences, giving him/her the opportunity to communicate solely by making use of brain signals.

A tunable stimulus presentation paradigm

The key element in the ERP speller application is the stimulus presentation paradigm. It determines how much stimuli are presented to infer the target symbol and, for each stimulus, which symbols are simultaneously highlighted on the screen. In this way, it dictates the speed of spelling and strongly influences the decoder's accuracy by determining the quantity and quality of the recorded stimulus responses. Several stimulus presentation paradigms have been proposed that achieve either a high accuracy or a faster spelling. There clearly is a trade-off between speed and accuracy that has to be made when choosing the stimulus presentation paradigm.

We propose a tunable paradigm for the ERP speller that provides us flexibility in choosing the speed of spelling while maximising the

spelling accuracy through optimisation of the highlighting scheme. We present the results of a spelling experiment with 24 subjects in which our paradigm is compared to basic paradigms. For the decoding, we use a state of the art self-learning decoder that avoids the calibration session by tuning its parameters online. With our new paradigm, a higher number of correct symbols can be spelled per minute. What's more, the experiment illustrates the influence of the paradigm settings on the learning and decoding performance obtained with the self-learning decoder. This confirms the strong interaction between application and decoder. With the tunable paradigm, this interaction can be employed to improve the overall BCI performance. This is exploited in the second contribution of this thesis.

A reliable self-learning decoder

Traditional supervised BCI decoders are tuned by means of labelled data, recorded during a calibration session in which the user is asked to perform a predefined task. This is exhausting and makes the BCI very unattractive. The BCI community has proposed several methods to reduce the need for labelled data, for example by recycling data from previous users or by retuning the decoder on data recorded during actual use. Recently, a method was proposed that completely avoids the calibration procedure. The decoder starts from scratch, with random parameter values and uses the expectation maximisation (EM) algorithm to tune the parameters to the data recorded during use of the BCI. The method was shown to learn very quickly how to discriminate between target and non-target responses. Nevertheless, it does not have the theoretical guarantee to always achieve excellent classification performance and as such is not fully reliable.

Recently, the learning from label proportions (LLP) idea was proposed to estimate the class-conditional mean feature vector from unlabelled data. It requires the data to be observed in two groups with known proportion of samples in both classes. The LLP-based decoder is theoretically guaranteed to converge to a traditional decoder, calibrated with labelled data. In collaboration with the Technical University of Berlin and the University of Freiburg, we propose the application of the LLP idea in ERP-BCI to obtain the first reliable

self-learning decoding method. We use our tunable paradigm to merge two sequences of stimuli with different relative frequencies of target stimuli to obtain the two required groups of data.

An online spelling experiment with 13 subject, conducted at the University of Freiburg, confirms that the LLP-based decoder is capable of achieving an accuracy that converges to the supervised solution. Its robust performance however comes at the cost of a slower learning process. The obtained accuracy level is lower compared to the EM-based decoder.

A reliable and effective self-learning decoder

The two self-learning decoding methods for ERP-BCI clearly show complementary strengths and weaknesses. The LLP-based method is robust while the EM-based method has the potential to obtain higher performance levels. Inspired by this observation, we propose a method to combine these two self-learning decoders analytically, thereby adopting the benefits of each. We propose to estimate its parameters as a mixture of the estimates found by the two existing decoders. We present an analytical formula to obtain the optimal mixing coefficient.

An offline simulation of a spelling experiment with 13 subjects allows us to compare the mixing method to the two aforementioned self-learning decoders on the same data. The simulation illustrates that the classification performance obtained with the mixing method is higher than any of the self-learning decoders it is made from. On top of that, the method is as robust as the original LLP-based decoder.

We present the results of an online spelling experiment that compares the original self-learning decoders based on LLP, EM and their mixture for six subjects, conducted in collaboration with David Hübner and Michael Tangermann at the University of Freiburg. The exceptional performance is confirmed by this experiment. We obtain the first self-learning decoding method that is capable of classifying ERP responses very accurately and reliably without observing any labelled data. This provides us with a new generation of BCI systems where the online data, without explicit knowledge of the user's intention, is as valuable as labelled calibration data. This was only possible by

tuning the application to the decoder with our flexible paradigm.

An automatic diagnosis system for temporal lobe epilepsy

A final contribution of this thesis is the development of a computer-aided diagnosis and lateralisation system for temporal lobe epilepsy (TLE). We use machine learning to discriminate between patients with TLE and healthy subjects, based on a measure of the functional connectivity between different regions in the brain. These connectivity values can be computed from very short scalp EEG recordings. Candidates for epilepsy surgery usually undergo a long monitoring to record pathological brain activity and determine the localisation of the epileptogenic zone. These patients can benefit greatly from this very efficient system.

Short recordings from 40 TLE patients (20 left TLE and 20 right TLE) and 35 healthy subjects are used to compute the functional connectivity between the different regions in their brain. Two classifiers are built on these connectivity measures, one for diagnosis and one for lateralisation of TLE. To avoid learning false decision rules, we automatically select the subset of connectivities that contain the most relevant information for classification.

We present the classification performance of this system in a leave-one-out procedure. One subject is left out of the data while the others are used to select the relevant features and tune the classifier. This classifier is then tested on the subject that was left out. Both classifiers are capable of accurately classifying subjects (90 % accuracy). The automatic selection of connectivities demonstrates which connections most strongly indicate the neurological disease and its lateralisation. Some of these connectivities did not show statistically significant differences between groups in previous studies. This demonstrates the power of machine learning in computer-aided diagnosis. By considering the interaction between the connectivities, it goes beyond standard statistics to find an accurate solution for the classification problem

Samenvatting

In dit proefschrift wordt een nieuwe ontwerpswijze ontwikkeld voor brein-computer-interfaces waarin kunstmatige intelligentie wordt toegepast om de hersenactiviteit te vertalen naar bruikbare informatie.

Brein-computer-interfaces

Brein-computer-interfaces zijn systemen die een directe verbinding creëren tussen de hersenen en een computer. Sinds de introductie van het concept in 1973 zijn er BCI's ontwikkeld voor verscheidene toepassingen. Sommige BCI's geven de gebruiker de mogelijkheid om zijn/haar hersensignalen actief te gebruiken om bijvoorbeeld een tekstverwerker of een rolstoel aan te sturen. In andere BCI's blijft de gebruiker passief en vormt de computer een beeld van de toestand van het brein, bijvoorbeeld om emoties of stoornissen zoals epilepsie of de ziekte van Alzheimer te detecteren.

Een BCI bestaat doorgaans uit drie componenten: de gebruiker, de applicatie en de decoder. Deze laatste interpreteert de hersenactiviteit van de gebruiker en vertaalt deze naar controlesignalen om de applicatie aan te sturen. Er wordt veel onderzoek gevoerd naar het verbeteren van deze componenten om de BCI te kunnen integreren in ons dagelijks gebruik. In huidig onderzoek worden de gebruiker, applicatie en decoder echter nog steeds als afzonderlijk werkende systemen beschouwd. Dit staat in schril contrast met de invloed die zij hebben op elkaars werking. Het buiten beschouwing laten van deze wisselwerking beperkt de mogelijke verbetering van het systeem.

Het doel van dit werk is het herontwerpen van een BCI, waarbij de drie componenten op elkaar worden afgestemd. Op deze manier proberen we BCI's dichter bij de uiteindelijke gebruiker te brengen.

Een BCI op basis van gebeurtenis-gerelateerde potentialen

Eén van de meest gebruikte en bekende types BCI is diegene gebaseerd op gebeurtenis-gerelateerde potentialen (in het Engels event-related potential, ERP). Een ERP is een respons die in de hersenen gecreëerd wordt wanneer men een visuele of auditieve stimulus waarneemt. In ERP gebaseerde BCI's wordt een reeks stimuli herhaaldelijk gepresenteerd aan de gebruiker. Hij/zij kan een bepaalde stimulus kiezen. Wanneer deze doelstimulus wordt waargenomen zal de respons in de hersenen anders zijn. Door deze doelresponses te herkennen kan de decoder achterhalen welke stimulus gekozen werd. Wanneer de verschillende stimuli aan verschillende controlesignalen worden gelinkt kan de gebruiker dus actief een computer aansturen, louter door middel van hersensignalen.

De eerste, en nog steeds meest gekende, ERP gebaseerde BCI is de ERP speller ontworpen door Farwell en Donchin in 1988. In dit systeem worden de letters van het alfabet voorgesteld in een raster op een computerscherm. Groepen symbolen worden na elkaar opgelicht in het raster. Door te concentreren op een bepaald symbool kan de gebruiker woorden spellen.

Het herkenen van de doelrespons in de gemeten hersenactiviteit is een uitdagende taak voor de decoder. De respons verschilt namelijk sterk tussen personen en kan zelfs sterk variëren voor eenzelfde persoon ten gevolge van vermoeidheid, emoties enz. Bijgevolg moet de decoder steeds opnieuw afgestemd worden bij ieder gebruik van de BCI. In de huidige generatie BCI's worden geavanceerde algoritmes gebruikt waarmee de computer zelf leert om de doelrespons te herkennen op basis van voorbeelden van hersensignalen waarin deze respons gemarkeerd wordt. Het opnemen van deze gemarkeerde data gebeurt in een afzonderlijke kalibratiesessie die zeer vermoeiend is voor de gebruiker. Dit maakt de BCI uiterst onaantrekkelijk voor dagelijks gebruik. Er wordt dan ook actief gezocht naar methodes om

deze nood aan gemarkeerde data te verminderen. Recent werd zelfs een eerste kalibratieloze decoder voorgesteld die de doelrespons leert herkennen door middel van ongemarkeerde activiteit, gemeten tijdens het eigenlijke gebruik van de BCI.

Een regelbaar paradigma voor de presentatie van stimuli

Het belangrijkste onderdeel van de applicatie in ERP gebaseerde BCI's is het stimulus presentatie paradigma. Dit algoritme bepaalt welke stimuli achtereenvolgens getoond worden aan de gebruiker. Het paradigma beïnvloedt de snelheid waarmee het controlsignaal kan geselecteerd worden, alsook ook hoe sterk de doelrespons is en dus hoe nauwkeurig de decoder het controlesignaal kan selecteren. In het huidige aanbod paradigma's moet steeds een afweging gemaakt worden tussen snelheid en nauwkeurigheid. We presenteren een oplossing voor dit probleem door een nieuw paradigma te ontwikkelen waarmee de snelheid kan ingesteld worden. Terzelfdertijd wordt de nauwkeurigheid van de BCI gemaximaliseerd door de stimuli te optimaliseren.

Het nieuwe paradigma wordt vergeleken met bestaande basisparadigma's in een experiment met 24 personen die zinnen spellen met de kalibratieloze ERP speller. De resultaten tonen aan dat, met het nieuwe paradigma, een hoger aantal symbolen correct gespeld kunnen worden per tijdseenheid. Bovendien maakt het experiment duidelijk dat het ingestelde paradigma een sterke invloed heeft op hoe goed de decoder de doelresponses leert herkennen tijdens de sessie. Bijgevolg kan de kalibratieloze decoder nog krachtiger gemaakt worden door het flexibele paradigma af te stemmen.

Een betrouwbare zelflerende decoder

De bestaande kalibratieloze decoder voor ERP gebaseerde BCI kan de doelrespons goed herkennen eens er voldoende data is opgenomen. De methode heeft echter niet de garantie om deze hoge prestaties steeds opnieuw te bereiken voor elke gebruiker en elke sessie. Dit maakt de BCI onbetrouwbaar.

In samenwerking met David Hübner en Michael Tangermann (Universiteit van Freiburg), Pieter-Jan Kindermans en Klaus-Robert Müller (Technische Universiteit Berlijn) stellen we de eerste zelflerende

decoder voor die gegarandeerd de prestatie van een traditionele gekalibreerde decoder bereikt tijdens het gebruik van de BCI. De techniek vereist dat de gemeten ERP's worden verzameld in twee afzondelijke groepen met een verschillende (en gekende) verhouding doelresponses. Daarvoor maken we gebruik van het regelbare paradigma. De applicatie wordt dus afgesteld op de noden van de decoder.

De nieuwe methode werd ingebouwd in de ERP speller en getest aan de Universiteit van Freiburg in een experiment met 13 personen. De resultaten bevestigen dat de zelflerende decoder zeer robuust werkt en de prestatie effectief convergeert naar die van een traditionele decoder met kalibratie. De hoge betrouwbaarheid van deze methode komt echter ten koste van een trager leerproces. Gemiddeld gezien is de nauwkeurigheid van deze decoder lager dan die van de bestaande zelflerende decoder.

Het combineren van betrouwbare en nauwkeurige zelflerende decoders

Met onze vorige bijdrage beschikken we nu over twee zelflerende decoders die de kalibratiesessie vermijden. De ene decoder heeft het potentieel om zeer snel te leren tijdens het gebruik van de BCI, maar heeft niet de garantie om de juist oplossing te vinden. De andere methode is wel betrouwbaar, maar leert minder snel. Een derde bijdrage van dit werk is een techniek om twee verschillende zelflerende decoders te combineren en zo de voordelen van beide over te nemen.

De parameters van de nieuwe decoder zijn een combinatie van deze van de bestaande decoders. We geven een analytische formule om het gewicht te berekenen dat aan elk van de bestaande decoders moet gegeven worden in deze combinatie.

De methode wordt getest in een gesimuleerd speller experiment met 13 personen. Dit laat ons toe de drie zelflerende decoders te testen op exact dezelfde data. De gecombineerde decoder leert sneller dan elk van de voorgaande decoders. Bovendien zijn de behaalde prestaties zeer robuust. Deze resultaten worden bevestigd door een werkelijk experiment met zes personen, uitgevoerd aan de Universiteit van Freiburg. Door het afstemmen van de applicatie op de decoder en het combineren van verschillende methodes bekomen we de eer-

ste ERP gebaseerde BCI zonder kalibratie die zowel nauwkeurig als uiterst betrouwbaar is.

Automatische diagnose van epilepsie in de temporale hersenkwab

Een laatste bijdrage van dit werk is de ontwikkeling van een systeem dat de hersenactiviteit interpreteert voor de diagnose en lateralisatie van epilepsie in de temporale hersenen (in het Engels temporal lobe epilepsy, TLE).

Een derde van de patienten met epilepsie kan niet geholpen worden met medicatie. Zij zijn kandidaten voor een chirurgische ingreep waarin het deel van de hersenen dat verantwoordelijk is voor de epileptische aanvallen wordt weggenomen. Daarvoor moet de patient een lange meetprocedure ondergaan waarin afwijkende hersenactiviteit moet gevonden worden om de diagnose te kunnen stellen en nauwkeurig de epileptogene zone te kunnen bepalen. Deze procedure is zeer belastend voor de patient.

Wij stellen een systeem voor waarbij zeer korte metingen van de hersenactiviteit gebruikt kunnen worden om de connectiviteit tussen verschillende regio's in de hersenen te berekenen en op basis daarvan de diagnose te laten stellen door een computer. Een database van patienten en gezonde personen wordt gebruikt als voorbeelddata om de computer zelf te laten leren welke connectiviteiten relevant zijn en hoe zij de correcte diagnose kunnen aanwijzen.

We testen dit computergestuurd diagnosesysteem op een database van 20 personen met TLE in de linker hersenhelft (LTLE), 20 personen met TLE in de rechterhersenhelft (RTLE) en 35 gezonde personen. Zowel de diagnose (gezond vs. TLE) als de lateralisatie (LTLE vs. RTLE) kan correct gesteld worden voor 90 % van de personen. Bovendien brengt de automatische selectie van relevante verbindingen nieuwe kennis over het ziektebeeld.

List of Acronyms

ACC	Anterior Cingulate Cortex
ALS	Amyotrophic Lateral Sclerosis
Amyg	Amygdala
AUC	Area Under Curve in a Receiver Operator Characteristic
BCI	Brain-Computer Interface
BLDA	Bayesian Linear Discriminant Analysis
CB	Checkerboard
CNN	Convolutional Neural Network
CSM	Correct Symbols per Minute
DTI	Diffusion Tensor Imaging
EEG	Electroencephalogram
EM	Expectation Maximisation
ERP	Event-Related Potential
fMRI	Functional Magnetic Resonance Imaging
GMM	Gaussian Mixture Model
Hipp	Hippocampus
HS	Hippocampus Sclerosis
IED	Interictal Epileptiform Discharges
iEEG	intra-cranial EEG
IID	Independently and Identically Distributed

ISI	Inter-Stimulus Interval
ITR	Information Transfer Rate
LDA	Linear Discriminant Analysis
LIS	Locked-In Syndrome
LLP	Learning from Label Proportions
LOOCV	Leave-One-Out Cross Validation
LSR	Least Squares Regression
LTLE	Left Temporal Lobe Epilepsy
MEG	Magnetoencephalography
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MRI	Magnetic Resonance Imaging
PCC	Posterior Cingulate Cortex
PDC	Partial Directed Coherence
PHipp	Parahippocampus
RC	Row-Column
RF	Random Forest
ROI	Region Of Interest
RTLE	Right Temporal Lobe Epilepsy
SC	Single Cell
SNR	Signal-to-Noise Ratio
SP	Switching Paradigm
SSVEP	Steady State Visually Evoked Potential
STD	Standard Deviation
SVM	Support Vector Machines
TL	Transfer Learning
TLE	Temporal Lobe Epilepsy
TPMid	Medial Temporal Pole
WSR	Written Symbol Rate

Contents

1 Introduction

The interaction between a human being and his environment is controlled by his brain. It is the most complex organ in our body and many of its aspects are yet to be fully understood. When changes in our surroundings trigger us through sight, hearing, smell, touch and taste, the brain analyses the signals received from the senses and decides how to react to the observed changes. Subsequently, the brain informs our body to perform the desired action, impacting our environment. This standard procedure may limit the speed or complexity that can be achieved in the interaction. Even more, it is impeded for those suffering from blindness, deafness, (partial) paralysis etc. In these cases, the interaction pathway through the human body can be bypassed with a *brain-computer interface* (BCI). A BCI is a system that creates a direct link between the brain and a computerised system (Wolpaw et al., 2002; Dornhege et al., 2007). Cochlear implants, for example, transform recorded sound to signals that are sent directly to the brain, thereby enabling the deaf to hear again (Crosby et al., 1985). Interaction happens in the opposite direction when the brain sends signals to the computer. The computer then interprets the brain activity and executes the desired command. Examples are a brain-controlled robotic arm, wheel chair or drone and text spelling systems for communication (Farwell and Donchin, 1988; Wolpaw et al., 1991; Chapin et al., 1999; Birbaumer, 2006). Even systems that diagnose neurological diseases, such epilepsy or Alzheimer's disease, based on recorded brain activity, can be considered brain-computer interfaces (Górriz et al., 2008; Kerr et al., 2013).

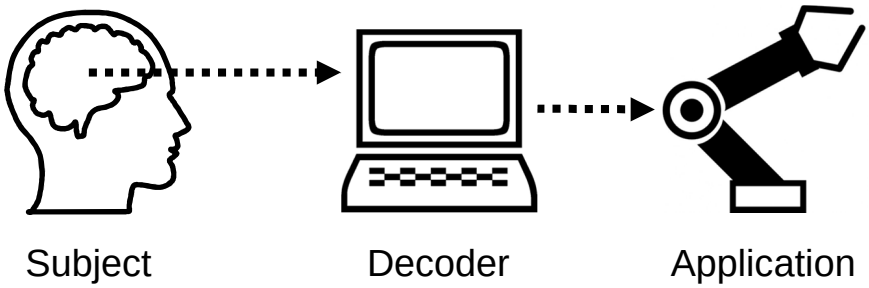

Figure 1.1: Schematic presentation of a brain-computer interface. The three components generally found in BCI are indicated. The subject's brain activity is sent to a decoder. The decoder interprets the brain signals and translates the subject's intention or brain state to a command that controls the application, represented here by a robotic arm.

The BCI concept was first introduced by Jacques J. Vidal (1973). A BCI generally consists of three components, as shown in Figure 1.1 (Wolpaw et al., 2002; Vallabhaneni et al., 2005; Dornhege et al., 2007). First, the user's brain activity is measured, for example with electroencephalography (EEG). Second, the decoder interprets the recorded activity. Its task is to extract useful information from the brain signals to determine the user's intention or brain state. Subsequently, it selects the appropriate control command. Finally, the application executes the command, for instance moving the robotic arm or wheel chair in the desired direction, pronouncing words, presenting a diagnosis etc.

In this doctoral thesis I present the optimisation of a BCI by improving the synergy between the different components. With this new approach, I intend to facilitate the integration of BCI in the daily life for those who could benefit from these systems.

To understand the significance of my contributions, it is important to define which requirements a BCI has to meet. To begin with, the system needs to be effective. The user's intention or brain state must be detected with high accuracy. In addition, the BCI is required to work efficiently. Certain applications may demand an almost instant translation of the recorded brain activity to a control command (e.g.

the brain-controlled wheel chair). Finally, the system needs to provide user satisfaction in a specified context of use. Overall this means that a BCI is required to be usable. Here, *usability* is defined by the International Organization for Standardization as:

"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use"

As the definition indicates, usability is strongly related to effectiveness and efficiency but also describes for instance how pleasant and satisfying the system is to use, how tolerant it is to errors made by the user and how easy the user can learn to control it.

In current BCI design we face two major challenges. First, there is a constant trade-off between effectiveness and efficiency (Polikoff et al., 1995; McFarland et al., 2003; Mak et al., 2011; Schreuder et al., 2011). A fast translation of brain activity limits the time to record brain data and as such reduces the amount of information that is available to accurately define the user's intention or state. Second, the decoder needs to be tuned to the specific user to understand his/her individual brain activity. In traditional BCIs this is done by means of a calibration recording before each session (Müller et al., 2004; Lotte et al., 2007; Müller et al., 2008; Lemm et al., 2011). This is exhausting for the user and as such degrades the usability drastically. Even more, frequent recalibration is required as the properties of a user's brain activity may change over time (Shenoy et al., 2006; Krusienski et al., 2011). A lot of effort has been put into developing methods that reduce or even eliminate the need for calibration (Shenoy et al., 2006; Krauledat et al., 2008; Fazli et al., 2009; Lu et al., 2009; Vidaurre et al., 2011a; Kindermans et al., 2012, 2014a). Nevertheless, they make the BCI less effective or less robust.

This dissertation details the development of methods that tackle the aforementioned challenges by means of a symbiotic BCI design. In contrast to previous work, I consider the interaction between the different components of the BCI. They are tuned to each other in order to improve the performance of the BCI in a way that is not possible by optimisation of the individual parts as was done previously. In addition, I demonstrate how machine learning can contribute to

recognising pathological brain activity and as such form a powerful tool for fast and accurate diagnosis of neurological diseases.

This chapter continues with a more detailed introduction to brain-computer interfaces. Afterwards, the main type of BCI used in this dissertation, those based on event-related potentials, is introduced. Then, an overview of previous work in the field is given. Finally, the content of the dissertation is outlined.

1.1 The brain-computer interface

The first recording of human brain activity was performed in 1924 by Hans Berger (Berger, 1929). He measured electric brain signals in a so-called *electroencephalogram*. With the development of personal computers in the mid-seventies, the question arose if these brain signals could serve as carriers of information in man-computer communication. In 1973, the Belgian Jacques J. Vidal introduced the term *brain-computer interface* for systems that employ brain signals in man-computer dialogue (Vidal, 1973). Since then, BCI systems have been developed for a wide variety of applications e.g. to restore sensory-motor functions, interpret and diagnose brain activity and much more.

This first section commences with a short portrayal of the human brain and how its activity is generated and measured. Next, I will specify how information can be obtained from brain activity and how this is applied in several types of BCIs.

1.1.1 The human brain

The spinal cord and the brain constitute the central nervous system of the human body. While the spinal cord is responsible for basic responses (e.g. reflex movement), the brain is the centre of voluntary control using advanced information processing. It senses changes in its surroundings and coordinates the interaction with its environment through sensory input, motor control, memory etc.

The brain consists of the cerebrum, divided in a left and right hemisphere, the brain stem and cerebellum, as illustrated in Fig-

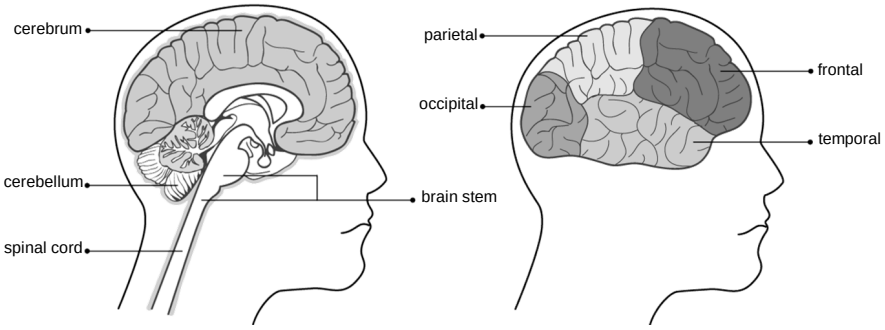

Figure 1.2: Anatomy of the human brain. Left: main parts of the central nervous system. Right: the division of the cerebrum into four brain lobes (frontal, temporal, parietal and occipital). Figure based on `http://www.cancer.ca`

ure 1.2. The cortex is the outer layer of the cerebrum where information is received, analysed and interpreted. The cerebrum is divided into four lobes. The frontal lobe contains various functions such as abstract thinking, behaviour and personality as well as problem solving capabilities, motor control and speech production. Functionality for spatial sense, navigation and touch is contained in the parietal lobe. The occipital lobe covers visual reception and processing. The temporal lobe takes care of memory as well as hearing and understanding language. Each lobe is subdivided in several cortical regions named according to their location and specific functionality (Tzourio-Mazoyer et al., 2002). The two hemispheres are largely symmetric, although some functions are more developed in one hemisphere (e.g. language is in most people located in the left hemisphere) (Davidson and Hugdahl, 1996). With magnetic resonance imaging (MRI), an image of the anatomical structure of the head can be obtained that visualises the different types of tissue.

The core component of the nervous system is the nerve cell or *neuron*. The brain contains on average 86 billion neurons, 16 billion of which are located in the cortex. The nerve cell itself consists of a cell body (soma) and typically one axon and multiple dendrites. Neurons receive, process and send electrical pulses called *action potentials* to other neurons via synapses. In this way they can form complex neural networks.

Recording brain activity

In order to make use of brain activity for the control of computer systems, we must record it. Capturing brain activity is possible because the initiation and inhibition of action potentials in the neurons creates a separation of electrical charge around them. This in turn creates a very small electric field. When several neurons are activated together, the compound electric field is measurable outside the brain by means of electrodes. Electrodes are placed on the scalp of the head in standardised locations, e.g. the 10-20 electrode system (Klem et al., 1999) as shown in Figure 1.3. The bottom panel in Figure 1.3 illustrates the collection of electrical signals recorded in all electrodes which forms an electroencephalograph (EEG). Due to its high temporal resolution (on the order of milliseconds), ease of use, mobility and relatively low cost, this is the most common method used in BCIs for real-time recording of brain activity. Nevertheless, the quality of EEG signals is low, which makes the decoding of information from these signals a true challenge. This is a major issue that we have to address in our design of brain-controlled systems.

When the electrodes are placed inside the head, directly on the surface of the brain it is called *intracranial EEG* (iEEG). These signals contain less noise as the electric field does not need to propagate through the different types of tissue in the head (e.g. the very low-conductive skull tissue) before being recorded. However, this method requires surgery for electrode placement and as such is only convenient for long-term BCI use.

Several other medical imaging methods have been used to record brain activity. Functional MRI uses the same technology as MRI to measure activity in the brain by imaging changes in blood flow and the level of oxygen in the blood. Magnetoencephalography (MEG) measures the magnetic field that is created by the separation of charge in neurons. It requires the recording to take place in a magnetically shielded room. For this reason, it is less convenient to be used in BCIs.

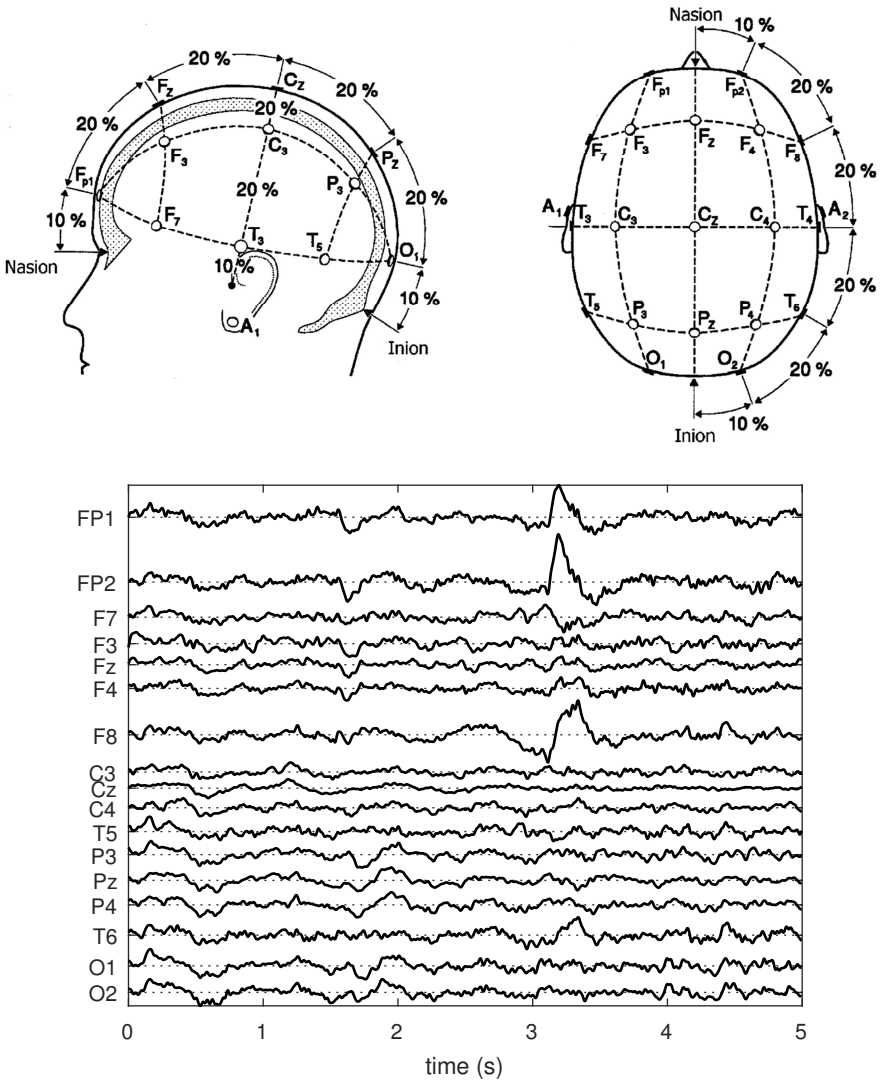

Figure 1.3: The 10-20 electrode placement and an example of recorded EEG. The peak just after the 3 s mark is caused by an eye blink.

Brain damage and diseases

The brain is protected physically by the skull and cerebrospinal fluid and chemically by the blood-brain barrier. Nevertheless, several forms of damage may occur to the brain and an entire range of neurological

diseases has been discovered. Two of them are within the direct scope of this dissertation and therefore described here.

The *locked-in syndrome* (LIS) is a condition in which the patient suffers from paralysis of nearly all voluntary muscles (except for eye movement) (Laureys et al., 2005). Patients are however conscious and cognitively intact. They are said to be *locked* inside their own body. LIS is caused by damage to the lower brain or brain stem due to a stroke, a traumatic brain injury or diseases like multiple sclerosis and amyotrophic lateral sclerosis, which causes death of neurons that conduct voluntary muscles (Rowland and Shneider, 2001). A BCI that infers the user's intention for communication and control can give these patients the possibility to interact again with their surroundings and as such improves their standard of living greatly (Kübler et al., 2005; Sellers et al., 2006; Birbaumer and Cohen, 2007). The main contribution of this dissertation is the optimisation of this specific type of BCIs.

Epilepsy is a neurological disorder that affects about 1 % of the world's population (Fisher et al., 2005). It is identified by recurrent abnormal electrical discharges in the brain called *seizures*. Depending on the location of the brain at which the seizure occurs, the externally observed symptoms can be the loss of consciousness, contraction of muscles in the limbs, a visual disturbance, a specific smell that is only observed by the patient etc. Seventy per cent of patients with epilepsy can be successfully treated with the range of anti-epileptic drugs that has been developed. For the other 30 %, neurostimulation or surgery is considered (French, 2007). This requires a long and for the patient exhaustive procedure to accurately diagnose epilepsy and determine the epileptogenic zone that needs to be removed. The second contribution of this dissertation is the development of a fast and accurate computer-aided diagnosis system for epilepsy.

1.1.2 Decoding information from the brain

After recording the brain signals, they must be interpreted by the decoder to determine the desired control command. For that purpose, BCI systems rely on information in the brain activity that is indicative of the user's intention or brain state.

To inform the computer about his/her intention, the user can employ *neural control signals*. These are distinctive patterns in the brain activity that are generated voluntarily by the user, or elicited by the observation of a stimulus. By linking different patterns to a different command, the user can select the command of choice. Several types of neural control signals exist. For instance, in BCIs based on *motor imagery*, the user alters the activity recorded in different regions of the scalp by imagining the movement of the left or right hand (Pfurtscheller and Da Silva, 1999). Another example is the *steady state visually evoked potential* (SSVEP). This is a brain response with a specific frequency that is evoked when visually observing an image that flashes at this same frequency (Middendorf et al., 2000). Finally, the *event-related potential* (ERP) is a specific pattern in the brain signal that is elicited by a visual, auditory or tactile stimulus. The pattern is different when a certain infrequent stimulus is observed that requires a response from the user (Sutton et al., 1965). BCIs based on ERPs as neural control signal will be the main focus in this dissertation. The ERP-based BCI will be explained in more detail in Section 1.2.

The decoding of the brain activity is a two-step process. First, informative features are extracted from the recorded brain signals. Next, the feature values are used to categorise the recorded activity as a specific neural control signal or brain state. Features can be obtained directly from the EEG electrode signals, or by imaging the source of the recorded activity within the brain.

Information from the sensors

Neural control signals are decoded by recognising their specific pattern in the recorded brain signals. For example, for ERP responses, the EEG that is recorded in a specific time window after the stimulus event is selected for classification. In this case, the feature vector is obtained by simply concatenating the signals recorded in the different electrodes. For motor imagery, the signals are usually transformed to the frequency domain. The signal power computed in the different electrodes and several frequency bands is then used to discriminate between left and right hand imagined movement.

Information from the source space

In addition to the features of the time-varying electric field at the scalp, it may be useful to know the location in the cortex where this electric field was generated. The goal of *source localisation* is to estimate the location and distribution of the sources in the brain from the EEG recorded at the scalp (Michel et al., 2004). It uses a three-dimensional electromagnetic model of the head that describes the geometry and electrical conductivity of the various tissues as well as the exact location of the electrodes.

Many different source distributions can generate the same electric field at the scalp. Hence, the solution is non-unique and solving this *inverse problem* is challenging. Several methods have been developed to solve the problem by imposing a priori information: Minimum Norm Estimate, LORETA, sLORETA etc. (Jatoi et al., 2014). Source localisation is for example used to accurately define the epileptogenic zone prior to epilepsy surgery (Staljanssens et al., 2016). It is also used in more fundamental research on brain activity, for instance to examine cognitive processes (e.g. the recognition of faces (Herrmann et al., 2005)) or to unravel changes in the brain due to a neurological disorder such as Alzheimer's disease (Dierks et al., 1993) or migraine (Clemens et al., 2008).

Once the activity in the different brain regions is computed, the connection between these regions can be investigated. The functional connectivity between a pair of brain regions is measured as the correlation between their source activity signals. The whole brain functional connectivity network shows significant differences between healthy subjects and those with a certain neurological disorder such as Alzheimer's disease (Supekar et al., 2008) or schizophrenia (Lynall et al., 2010). Hence, it can be used to diagnose these disorders. Furthermore, connectivity measures have been exploited to predict if a patient will respond to a specific treatment or not, for example vagus-nerve stimulation for epilepsy (Wostyn et al., 2016). Functional network measures are mainly used in BCIs that monitor a particular state of the brain.

Classifying brain activity

Once the informative features are obtained, they are processed by the decoder to discriminate between different neural control signals or brain states. For this purpose, we have to find a link between the feature values and the desired output of the decoder. This is a challenging task for several reasons. First, uninformative brain activity is recorded simultaneously, which creates a large amount of noise on the feature values. Second, the dimensionality of the feature vector is usually high, which makes the classification very complicated. For example, computing the connectivity between every pair of 82 brain regions results in 6724 features. Finally, brain activity varies greatly from subject to subject. Even within a single subject, the pattern of a neural control signal may change over time, for example due to increasing fatigue (Shenoy et al., 2006; Von Bünau et al., 2009). Overall, this makes it a laborious, if not impossible, task to design and program an algorithm that is capable of accurately classifying brain activity.

To tackle the challenges in decoding brain activity, the BCI community has adopted the benefits of machine learning. Machine learning refers to techniques that enable a computer to learn from data. It involves the training of a classification model by applying specific algorithms on a set of samples, each consisting of several feature values, and the corresponding correct label (the output class to which the sample belongs). The algorithm tries to find relations between features that are exclusively associated with the given label, which is not always obvious from visual inspection of the data. Afterwards, it assigns a new input sample to one of these classes based on its feature values and the learned relations. Machine learning is for instance used in social media to present advertisements that are expected to be interesting for you, based on your past activity on the Internet.

The machine learning decoder requires example data to learn from. For automatic diagnosis systems, a database containing patients and healthy subjects is required. Obtaining such a database is generally expensive and labour intensive. For BCIs based on neural control signals, the decoder needs to be tuned to each specific user. In traditional BCI systems, example data is recorded during a calibration

session prior to BCI use, in which the user is instructed to perform predefined tasks. This is exhausting for the user and drastically lowers the usability of these systems. Furthermore, the user's brain response to the same stimulus may change over time, which requires a recalibration before every session of use (Shenoy et al., 2006; Krusienski et al., 2011). This is a major challenge in traditional BCI systems. It makes them very unattractive for use on a daily basis. This dissertation will contribute to the design of BCI systems that avoid the calibration session by learning from the data that is recorded during actual use of the system.

Although the machine learning decoder is capable of solving complex classification tasks, the categorisation of neural controls signals is still challenging due to the low quality of the recorded EEG. For that purpose, BCIs typically elicit neural control signals multiple times and average the recorded response signals to reduce noise. While this improves the decoding accuracy, the repetitive recording increases the time that is needed to translate the user's intention to a control command. A trade-off has to be made between speed and accuracy. This is a specific challenge for the design of BCIs that determine the user's intention and will be addressed in this dissertation.

In Chapter 2 we will give a comprehensive overview of machine learning techniques and several methods that are used to tackle the challenges in decoding brain activity.

1.1.3 Brain-computer interface applications

The definition of BCI involves a wide variety of systems. I will categorise them according to the type of information that is acquired from the brain data. BCIs monitoring the brain state include, among many others, detectors of brain tumours based on anatomical brain imaging (Sharanreddy and Kulkarni, 2013) or recorded activity (Clark et al., 1998), automated diagnosis systems for neurological disorders such as epilepsy (Focke et al., 2012; Cantor-Rivera et al., 2015) and sleep disorders (Koch et al., 2013), early detectors of epileptic seizures (Gotman, 1982; Liang et al., 2010) and even systems that are capable of predicting when a seizure is likely to occur (Cook et al., 2013; Brinkmann et al., 2016).

BCIs that rely on motor imagery as neural control signal have been developed to control wheel chairs (Choi and Cichocki, 2008; Li et al., 2013) or a cursor on a computer screen (Wolpaw et al., 1991) and to assist in the rehabilitation of movement after a stroke (Pfurtscheller et al., 2008; Pichiorri et al., 2015). Systems for communication are mostly based on SSVEP (Cheng et al., 2002; Müller-Putz et al., 2005; Yin et al., 2015), visual or auditory ERP (Farwell and Donchin, 1988; Furdea et al., 2009) or a combination of both (Yin et al., 2014). Furthermore, BCIs based on neural control signals have been developed to control a robotic arm and other neuroprosthetics (Chapin et al., 1999; Taylor et al., 2002).

The applications of BCI are not limited to the clinical environment (Blankertz et al., 2010). BCIs that recognise the emotional state (Garcia-Molina et al., 2013) are for example used in neuromarketing to get a measurement of product appeal that is assumed to be more objective than the usual consumer surveys (Vecchiato et al., 2009). Besides that, BCIs have also found their way into the gaming industry (Ahn et al., 2014) and are expected to gain a lot of interest in combination with virtual and augmented reality.

In this dissertation we will focus on BCIs based on the ERP as neural control signal. We will use an ERP-based communication system as a case study for our symbiotic design approach. The next section will describe this system in more detail. In addition, the development of an EEG-based automated diagnosis system for epilepsy will be discussed at the end of this dissertation.

1.2 Event-related potential based BCI

This section gives a more thorough explanation of one specific neural control signal: the event-related potential. I explain how it is used to control a computer application and describe the challenges in designing such an ERP-based BCI.

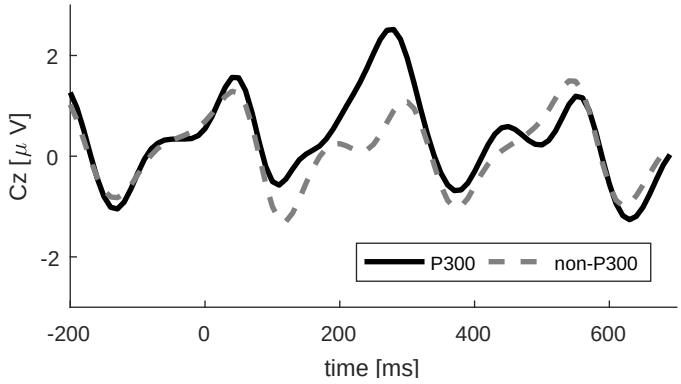

Figure 1.4: Event-related potential with and without P300 component. The waveforms are obtained by averaging the response of an individual subject on several visual stimuli.

1.2.1 The event-related potential

An event-related potential (ERP) is an involuntary response of the brain when a sudden external stimulus is observed (Sutton et al., 1965). This stimulus can be visual (e.g. a flash of light), auditory or even tactile. An ERP is composed of several components that can be recognised as positive and negative deflections in the EEG at specific time points after the stimulus event. One component is of particular interest in BCI: the P300 wave. It is marked as a positive peak about 300 ms after the event and is most clearly observed in the electrodes covering the parietal and occipital regions. It is elicited when a certain infrequent stimulus is observed that requires a response from the user. Figure 1.4 shows two event-related potential responses, one with and one without the P300 component.

Besides its application as neural control signal, the P300 component of the event-related potential has been used as a biomarker for dementia, depression or schizophrenia (Pfefferbaum et al., 1984) and to predict the receptiveness of a patient to certain treatments such as vagus-nerve stimulation for epilepsy (Wostyn et al., 2016).

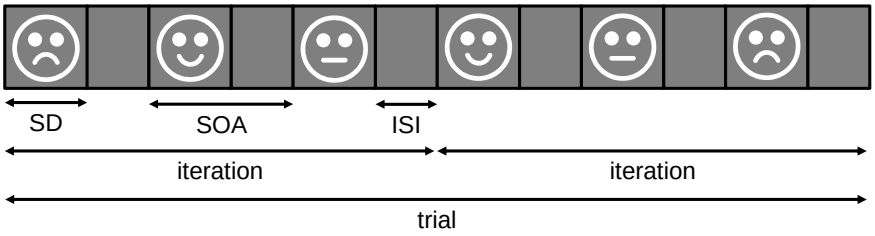

Figure 1.5: Example of an oddball paradigm. The sequence of stimuli is illustrated and the several parameters of the paradigm are indicated. The horizontal length of the squares denotes the duration of the stimulus or the blank screen shown to the subject.

ERP as neural control signal

The BCI based on the ERP as neural control signal was introduced by Farwell and Donchin (1988). The use of the ERP to control a computer application is explained with the following example. A subject is positioned in front of a screen that repeatedly presents faces with different expressions, as illustrated in Figure 1.5. The subject is instructed to count silently the appearances of the face that corresponds to his/her current emotion. Each presentation of a face is a visual stimulus and will elicit an ERP in the user's brain. However, only the infrequently occurring target stimulus, chosen by the user, requires the user to respond by counting. This process of applying a sequence of stimuli in which infrequently occurring target stimuli require a task from the user is called an *oddball paradigm* and is known to elicit the P300 component (Sutton et al., 1965; Donchin, 1981; Fabiani et al., 1987). By detecting the presence or absence of this component in the responses on the different stimuli, the decoder can infer the target stimulus and express the correct emotion. This system can give patients with the locked-in syndrome the freedom to express their mood solely by making use of their brain signals.

At this point I want to remark that the difference between target and non-target response can also be found in other components of the ERP. A decoder that is trained with example data will use all the components that are provided in the feature vector to discriminate

between the two classes of stimulus responses.

Challenges in decoding ERP

The complete sequence of stimuli that is presented to select one control command is called a *trial*. For each stimulus, the EEG signal recorded during a specific time window after the stimulus event is selected as stimulus response. At the end of a trial, the decoder predicts for each stimulus response how likely it is to be a target response. The predictions for the individual responses are then aggregated to select the most likely control command. The challenges in decoding brain activity, described in Section 1.1.2, apply in particular to the decoding of ERP responses.

First of all, discriminating between target and non-target stimulus responses is challenging due to the low signal-to-noise ratio (SNR) of the recorded data. For this reason, the stimuli are repeatedly presented in several iterations to reduce the effect of noise. This affects how fast the user's intention can be translated. Other influencing factors are the stimulus duration (SD, typically on the order of tens or hundreds of milliseconds), the time between the onset of two consecutive stimuli (stimulus onset asynchrony, SOA) and the time between the end of the previous stimulus and the onset of the next one (inter-stimulus interval, ISI), see Figure 1.5. Reducing the time between two target stimuli is known to reduce the amplitude of the target response (Gonsalvez and Polich, 2002; McFarland et al., 2011). Hence, speeding up the spelling process reduces the quantity and quality of the responses that are available to accurately determine the control command. This results in an inevitable trade-off between speed and accuracy that has to be made when designing ERP-BCI.

Secondly, the ERP response pattern can differ greatly between subjects. Also, for a single subject the response pattern can differ between several recordings. For example, the amplitude and latency of the P300 peak are determined by several factors such as the relative frequency of the target stimulus, the stimulus duration and intensity, the inter-stimulus interval and even the motivation of the subject (Carrillo-De-La-Pena and Cadaveira, 2000; Sellers et al., 2006; McFarland et al., 2011). This is the reason why traditional BCI sys-

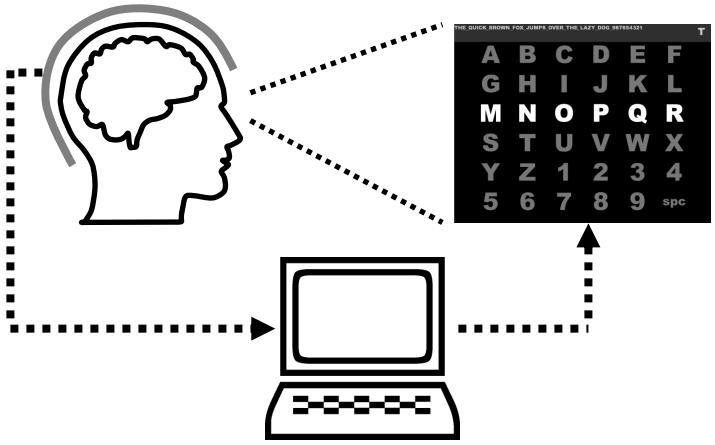

Figure 1.6: Schematic overview of the ERP spelling system. The application consists of a screen presenting a grid of symbols as well as the text that is spelled by the subject. A visual stimulus (the highlighting of the third row of symbols in the grid) is shown on the screen.

tems require a calibration procedure to be carried out for each new user and each session of use. Reducing the need for these inconvenient calibration sessions is a major challenge that has gained a lot of interest from the BCI community.

1.2.2 The ERP speller

We will use the ERP speller as a case study in this thesis. In this ERP-BCI, the commands are linked to the symbols from the alphabet. The speller was first described by Farwell and Donchin (1988). As illustrated in Figure 1.6, the original version presented a 6×6 grid of symbols on a screen. The subject focuses on the target symbol he or she wants to spell. Next, the rows and columns in the grid are highlighted in random order (the third row is highlighted in Figure 1.6). The subject is asked to count silently when his/her target is highlighted. The desired symbol is selected as the one on the intersection of the row and column that elicited a target response. In this way, symbol by symbol, words and sentences can be spelled.

1.2.3 Evaluating performance

Several measures will be used in this dissertation to assess the accuracy of the decoder and the speed of spelling in the ERP speller. They are discussed here.

Spelling accuracy

In the ERP speller, a trial consists of the complete procedure of presenting stimuli and classifying the elicited response signals in order to select a single symbol. The spelling accuracy is defined as the percentage of symbols that are selected correctly at the end of a trial. This reflects how many times the user gets positive feedback from the speller and as such influences the motivation of the user. Therefore, a high spelling accuracy at the beginning of the spelling session can have a positive influence on further performance (Kleih et al., 2010).

Respelling accuracy

Some BCI decoders tune their parameters with the data recorded during actual use of the BCI (Kindermans et al., 2014b) (see section 1.3.3 on adaptive methods). At the end of each trial, this new decoder can reclassify the data from former trials, thereby potentially correcting symbols selected in the past. The respelling accuracy is the percentage of correct symbols in the spelled text, after respelling every symbol with the decoder tuned to all recorded data. This corrected text is shown to the user. For adaptive decoders, this metric gives a more correct view on the quality of the speller output compared to the spelling accuracy.

Area under the curve

The AUC is the area under the ROC curve, plotting the true positive rate of the decoder against the false positive rate at various classification threshold values. It is the probability that the decoder will rank a randomly chosen target response higher than a randomly chosen non-target response (Ling et al., 2003). For example, an AUC of 0.5 indicates that the decoder classifies randomly. The AUC measures the decoder quality in terms of classifying individual ERP response

signals. On the contrary, the (re-)spelling accuracy is a rather coarse measure since it aggregates the classification of several stimuli before selecting the target symbol.

Symbols per minute

The speed of spelling can be expressed as the number of symbols spelled per minute. It is calculated as the inverse of the trial duration in minutes.

$$SM = \frac{60s}{e * n * SI + (e * n - 1) * ISI + P} \tag{1.1}$$

with e the number of iterations per trial, n the iteration length, SI the stimulus interval duration, ISI the inter-stimulus interval duration and P the pause between two trials.

Correct symbols per minute

Because of the trade-off between accuracy and speed of spelling, we need a measure that incorporates both performance metrics to evaluate the BCI. Thompson et al. (2014) proposed the *utility* as an intuitive measure for the performance of BCIs that infer a user's intention. It is formulated as:

$$U = \frac{E[b_k]}{E[\Delta t_k]} = \frac{\sum b_k}{\sum \Delta t_k} \tag{1.2}$$

To compute the utility, we measure the time Δt_k that was needed to translate the k^{th} intention to a control command and the gain b_k that is achieved with this action. The utility is maximised when the maximum benefit is achieved in the shortest period of time. In the context of the ERP speller, a gain of +1 symbol is obtained with each correctly selected symbol. In this case, $\sum b_k$ is the number of symbols respelled correctly at the end of the spelling session and $\sum \Delta t_k$ is the total duration of that session. This is called the *correct symbols per minute* (CSM) and has been used before to measure how quickly symbols can be spelled correctly (Schreuder et al., 2013; Kindermans et al., 2013, 2014b). Please note that common other measures are the *information transfer rate* (ITR) and *written symbol rate* (WSR)

(Billinger et al., 2012). However, as described by Thompson et al. (2014) the ITR is only valid under a list of assumptions, some of which are violated in the ERP speller. WSR, on the other hand, is more suited for systems in which the subject himself/herself is involved in correcting wrong symbols.

1.3 Prior work

This section gives an overview of the current state and limitations of BCI design, mainly focused on ERP-BCI and the application of machine learning to decode ERP response signals. For readers unfamiliar with the fundamentals of machine learning it might be useful to read Chapter 2 first.

1.3.1 Stimulus presentation paradigms

The key element in the ERP speller application is the stimulus presentation paradigm. It determines for each visual stimulus the group of symbols that is highlighted simultaneously in the grid. As explained before, sequences of stimuli are presented repeatedly in several iterations to reduce the effect of noise. Hence, the stimulus presentation paradigm dictates the spelling speed through the number of sequence iterations that is presented to spell a symbol and the number of stimuli in each iteration (further denoted by the parameter n). In addition, the paradigm has a significant influence on the accuracy of the speller. For example, increasing the spelling speed by presenting less iterations decreases the amount of data that is available to accurately infer the target symbol. Alternatively, the iteration length can be decreased. However, this increases the relative frequency of target stimuli and as such degrades the SNR of the target responses (Gonsalvez and Polich, 2002; McFarland et al., 2011). There clearly is a trade-off between speed and accuracy that has to be made when choosing the stimulus presentation paradigm.

In the original speller from Farwell and Donchin (1988), the rows and columns in the grid are highlighted in random order. For the original 6×6 grid, this row-column paradigm (RC) has an iteration

length $n = 12$. The number of times each symbol is highlighted during an iteration will be further denoted by r. For the RC paradigm, every symbol is highlighted twice per iteration, so $r = 2$. The spelling accuracy achieved with this paradigm is limited due to two types of spelling errors. First, *adjacency distraction* (Fazel-Rezai, 2007) is the phenomenon in which the user is distracted by the flash of a row (column) next to the target symbol. This causes the generation of a target response on a non-target stimulus. Together with the response generated on the target column (row) this results in the selection of a neighbouring symbol in the grid. Second, *double flash errors* occur when the target row and column are highlighted shortly after each other. This causes the target response generated on the second flash to overlap with the first one and have a lower amplitude (Gonsalvez and Polich, 2002; Martens et al., 2009) which can again result in a wrong symbol selection.

An entire spectrum of stimulus presentation paradigms has been designed as an alternative to the original RC paradigm, optimising either the spelling speed or accuracy (Mak et al., 2011). One end of the spectrum is the single cell (SC) paradigm (Guan et al., 2004), which highlights just one, randomly chosen, symbol in each stimulus. Due to the longer interval between two target stimuli, this most simple paradigm was shown to achieve a high spelling accuracy. Nevertheless, it makes the speller very slow as the iteration length is equal to the number of symbols in the grid ($n = 36$).

Alternatively, stimulus presentation paradigms have been proposed that highlight quasi-random groups of symbols simultaneously, thereby reducing the iteration length n. For instance, highlighting multiple rows or columns simultaneously has been considered by Allison and Pineda (2006). A major contribution is the checkerboard paradigm (CB) from Townsend et al. (2010). It actively avoids the adjacency distraction and double flash errors by optimising the highlighting scheme. For that purpose, the symbol grid is divided in black and white cells like a checkerboard, invisible to the subject. Adjacency distraction is avoided in this paradigm by only allowing symbols on the white/black cells to highlight simultaneously. Although designed for a 9×8 grid of symbols, the paradigm is converted easily to the original 6×6 grid as follows. The grid is virtually superimposed on a

checkerboard (not seen by the subject), see Figure 1.7. The symbols corresponding to white cells are secluded in a virtual 4×5 *white* matrix, leaving two cells blank. The same is done for the other symbols in a *black* matrix. The symbols are shuffled in their respective matrices before each iteration. An iteration of stimuli shown to the subject is then constructed by going top-down over the rows of the white matrix, collecting the symbols from each row and highlighting them in a separate stimulus in the original grid, followed by the rows of the black matrix, left-to-right the columns of the white matrix and finally the columns of the black matrix. As a result the subject sees random groups of symbols being highlighted in the grid. The reported higher accuracy however comes at the price of a longer stimulus iteration ($n = 18$) compared to the RC paradigm.

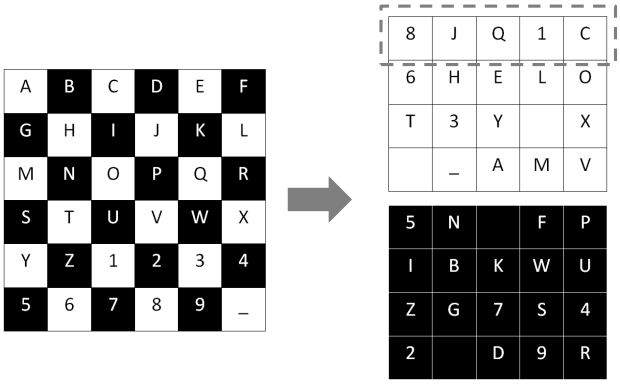

Figure 1.7: Construction of an iteration of stimuli for a 6 × 6 grid according to the checkerboard paradigm by Townsend et al. (2010). The dashed line encircles the symbols that will be highlighted in the grid during the first stimulus.

In the work by Jin et al. (2011), double flashes are avoided by preventing two consecutive stimuli from highlighting a common symbol. Fixed stimulus sequences of several lengths are constructed. As expected, the shortest iteration length was found to elicit less discriminable ERP patterns due to the higher relative frequency of target stimuli. This confirms once again the trade-off between speed and accuracy. Furthermore, the optimal iteration length differed from subject to subject, indicating the importance of tuning the application to the user's desire.

paradigm	SC	CB	RC
n	36	18	12
r	1	2	2
target frequency	0.03	0.11	0.17

Table 1.1: Application settings for the single cell (SC), row-column (RC) and checkerboard (CB) paradigm for a 6 × 6 spelling grid. The resulting relative frequency of target stimuli is given in the last row.

Inspired by Jin et al. (2011), Townsend et al. (2012) proposed a paradigm in which the iteration lengths n and relative frequency of target stimuli r can be chosen freely. To avoid double flashes and adjacency distraction, a number of constraints are applied. In this paradigm, each stimulus is created by randomly selecting and reselecting a group of symbols until compliant with all constraints.

Table 1.1 shows the different parameters of the basic single cell, row-column and checkerboard stimulus presentation paradigms for a 6 × 6 grid of symbols. The iteration length directly influences the speed of spelling. With increasing speed, the relative frequency of target stimuli increases. This influences the discriminability of the target response and as such decreases the accuracy of spelling.

A different way to balance between speed and accuracy in ERP-based BCI is by determining the number of stimulus sequence iterations presented in a trial. This parameter can be altered during use with *dynamic stopping* (Schreuder et al., 2011, 2013; Kindermans et al., 2014b). With this technique, the stimulus presentation stops when the desired command can be selected with a certain level of confidence. Confidence is measured for instance as the likelihood of the most likely command (Kindermans et al., 2014b) or its difference with the second most likely command (Schreuder et al., 2011).

Other modifications have been made to the stimulus presentation in order to improve the raw SNR of the response signals such as changing the matrix size and inter-stimulus interval (Sellers et al., 2006), highlighting symbols in different colours (Salvaris and Sepulveda, 2009; Takano et al., 2009), changing the size of the symbols (Gib-

ert et al., 2008), moving the symbols during highlighting (Jin et al., 2012), or a combination of brightness enhancement, rotation, enlargement and a trichromatic grid overlay (Tangermann et al., 2011). Furthermore, by flashing famous faces over the symbols, Kaufmann et al. (2011) improved the ERP response by introducing additional ERP components, related to the processing of these faces.

1.3.2 Machine learning for ERP classification

Several machine learning techniques were shown to be successful in discriminating between target and non-target ERP responses (Krusienski et al., 2006; Lotte et al., 2007; Müller et al., 2008). As described in Section 1.1.2, the challenges are the low SNR, the large variability between users and the high dimensionality of the data as EEG is measured simultaneously in several electrodes with a high temporal resolution.

Support vector machines (SVM) are known to cope well with classification tasks that involve a limited set of train data in combination with a high number of features per sample. Consequently, this machine learning technique has been successfully applied multiple times for the classification of ERP response signals. The best result on the BCI competition III dataset II, a commonly used dataset to evaluate ERP decoders, is still achieved with an ensemble of linear SVM classifiers (Rakotomamonjy and Guigue, 2008). Each SVM is trained on a different part of the calibration dataset and their outputs are aggregated to select the target symbol.

Linear discriminant analysis (LDA) assumes the ERP feature vector in each class to be normally distributed with a class-wise mean and shared covariance matrix. As ERPs closely follow the normality assumption (Blankertz et al., 2011), this simple technique has been widely applied and shown to be competitive with more complex methods for the classification of brain signals (Lotte et al., 2007; Müller et al., 2008). Blankertz et al. (2011) showed that the high dimensionality of the data causes the estimation of the covariance matrix to be distorted. The model is regularised by shrinking the covariance matrix, for which an analytical formula was proposed (Ledoit and Wolf, 2004; Blankertz et al., 2011).

Cecotti and Graser (2011) proposed a convolutional neural network (CNN) for the classification of ERPs. However, neural networks require rather large amounts of training data. As mentioned before, the scarcity of labelled data is a common problem in BCI and ERP-based BCIs in particular. Furthermore, Kindermans et al. (2012) has shown that CNNs do not result in a better classification accuracy compared to much simpler methods.

All the classification methods above require the recording of a labelled dataset during a calibration session. Several methods have been developed to reduce or even eliminate calibration, thereby improving BCI usability. They can be divided into two groups. Transfer learning methods recycle data from previous sessions and users, while adaptive machine learning methods use unlabelled data recorded during actual use. These two approaches are discussed in the next sections.

1.3.3 Towards self-learning brain-computer interfaces

Eliminating the need for calibration has been the subject of extensive research in the BCI community. The amount of calibration data can be reduced by obtaining discriminative information from data recorded during actual use of the BCI.

Dähne et al. (2011) used a reduced set of labelled data to determine the class-conditional mean ERP response and shared covariance matrix for a LDA classifier. As no labels are needed to estimate the pooled covariance matrix, this parameter can be updated during actual use. Several other methods use a classifier trained on a reduced set of calibration data and label the data recorded during actual use. The classifier is then retrained with this data and the predicted labels. For example, Li et al. (2008) proposed a self-training SVM. Panicker et al. (2010) used a co-training technique, in which two different classifiers label the data for each other. Kindermans et al. (2011) presented the results of class-reweighted ridge regression classifier that uses its own predictions on online recorded data as extra labelled information for updating. In the work by Xu et al. (2011), the subset of responses that can be most confidently classified are used for further calibration

of a Bayesian LDA classifier[1]. All these methods still use a (reduced) set of labelled data for training. The calibration procedure is not eliminated completely.

Kindermans et al. (2012) introduced the first unsupervised method for ERP-based BCI that does not use any labelled example. It uses a pseudo-generative model to describe the recorded target and non-target responses. The constraints imposed by the speller application are included in the algorithm that finds maximum likelihood estimates for the parameters of this model. The data is separated in two groups so that two responses assigned to the same group show, on average, more similarity than two responses assigned to different groups. The stimulus presentation paradigm determines the proportions of target and non-target responses. From this information and the size of each group, the classifier can depict which group contains the target responses.

It is important to emphasise that this last method learns from scratch. The parameters of the classifier are randomly initialised and continuously updated during use of the BCI. This classification method was shown to achieve a performance comparable to the best calibrated classifiers once enough data is recorded. Besides that, the classifier is capable to adapt to changes in the recorded responses (e.g. due to fatigue or changing background activity) (Kindermans et al., 2014a). The main claim against the method is the warm-up period at the very beginning of use, when data is still scarce. The performance of the classifier is unreliable in this phase as it depends strongly on the random initialisation of the classifier parameters. A significant part of this work extends this classification method. For this reason, it will be explained in exquisite detail in Chapter 2.

1.3.4 Transfer learning

With transfer learning (TL), data recorded in previous sessions or from other users is recycled for the calibration of a new decoder. For that purpose, it is necessary to overcome the session-to-session or subject-to-subject differences in the neural control signal. Several

[1]Note that Bayesian LDA is considered a false name as the model is in fact least squares regression.

methods have been proposed to tackle this challenge.

First of all, we can look for discriminative information that can be mutually applied between subjects or sessions. Krauledat et al. (2008) introduced the concept of *zero training* for motor imagery BCI. In their work, calibration data from previous sessions was used to determine a spatial filter that generalises better to new sessions for long-term BCI users. Subject to subject transfer of these spatial filters and classifiers was developed by Fazli et al. (2009) and several others after him (Devlaminck et al., 2011; Lotte and Guan, 2011; Samek et al., 2012). The overall objective of these methods is to create a decoder that is session or subject independent.

Other methods adapt to the new session or subject and employ historic information when user specific data is still scarce. Lu et al. (2009) presented the first application of transfer learning in ERP-BCI. Labelled data from a pool of previous subjects was used to build a subject independent classifier. The user specific data recorded during actual use was then labelled with this classifier and employed to calibrate a subject specific classifier. Next, recorded data is labelled by one of these classifiers (the one that is most confident) and added to the data set of subject specific calibration data for retraining. The performance achieved with this method was shown to depend on how much the new subject relates to the average historic subject. Besides that, this method still requires labelled data from a large pool of subjects.

Kindermans et al. (2014b) included transfer learning in his pseudo-generative model for the self-learning ERP decoder. The classifier parameters are regularised towards the classifier trained for previous subjects. Not a single labelled ERP response is used as these historic classifiers are also trained with the unsupervised learning technique. This transfer learning method was applied to motor imagery BCI by Jayaram et al. (2016).

Even though the results achieved with these methods are remarkable, none of the transfer learning techniques exploit the true cause of inter-subject differences: the source of the brain response, cortical and head anatomy and the exact location of electrodes.

Ahn et al. (2011) introduced the conjecture that in the source space, discriminative information might be more consistent over ses-

sions and users. Projecting the recorded EEG into the source space reduces the influence from differences in electrode placement and the distortion effect by the low-conductive skull. A template spherical head model and the beam forming method were used for source imaging. The classifier, constructed on features in the source space from historic subjects, did however not perform sufficiently well on new subjects.

Wronkiewicz et al. (2015) used an individual head model per subject, based on T1-weighted MRI and coregistrated electrode positions. Activity in the source space was obtained with minimum norm estimation for each subject in the pool. Next, a transformation matrix was calculated to morph the signal from each historic subject to the source space of the new subject. Finally, the forward model of the new subject projects the data from this source space to the scalp, where it is used for calibration. The method achieved results comparable with a supervised model on simulated EEG and data from an attentional modulation task. The method is however not tested in a practically useful BCI set-up, e.g. an ERP-BCI.

1.4 Research contributions and structure

1.4.1 Contributions to the BCI community

The contribution of this dissertation is two-fold. First of all, we present a new design approach for ERP-BCI. Since the introduction of the ERP as neural control signal in 1988, the BCI community has gone to great lengths to enhance the speed, accuracy and ease of using ERP-based BCI. Each modification that has been proposed optimises either the user, the application or the decoder individually. Nevertheless, regarding these components as separate entities limits the improvement that can be obtained. We present the first symbiotic design approach that considers the interaction between the subsystems. By tuning them to each other, we increase the BCI performance to a level that was unreachable before. The development of a symbiotic

ERP-BCI covers most of this dissertation. It is divided into the first three contributions described below. Secondly, we develop an automated diagnosis and lateralisation system for temporal lobe epilepsy. We demonstrate how machine learning can create a powerful tool to analyse the brain state in this promising new application of brain-computer interfacing.

A tunable stimulus presentation paradigm

A multitude of stimulus presentation paradigms has been proposed to improve the speed or accuracy of ERP-BCI. In Section 1.3 we described the checkerboard paradigm from Townsend et al. (2010). It increases the accuracy of a visual ERP speller by setting constraints to the highlighting scheme and as such actively avoiding the most common causes of spelling errors. However, the augmented accuracy comes at the cost of a slower spelling.

We propose a new paradigm for the ERP speller in which the spelling speed and relative frequency of target stimuli can be chosen freely. At the same time, the accuracy is maximised by optimising the highlighting scheme as much as possible. The tunable paradigm is set to its maximum speed and validated in an online experiment with 24 subjects performing a spelling task with the visual ERP speller. The self-learning decoder from Kindermans et al. (2012, 2014a) is used, which avoids the tedious calibration session by tuning its parameters online. The novel paradigm is compared to the basic row-column and checkerboard paradigms in terms of AUC, symbol selection accuracy and correct symbols spelled per minute.

The true value of this experiment is that it demonstrates how the learning and classification performance of the unsupervised decoder differs for different paradigms. The experiment reveals the strong underlying mechanism of interaction between the application and the self-learning decoder. This confirms how important it is to take this interaction into account when designing BCI systems. Our flexible paradigm can be tuned to empower the decoding, which is essential to unlock our second contribution.

Unsupervised decoding with theoretical guarantees

Traditional supervised BCI decoders require the recording of labelled data in a separate calibration session which is exhausting for the user. Alternatively, transfer learning and adaptive methods have been proposed to reduce the need for labelled data. The unsupervised adaptive decoder from Kindermans et al. (2012, 2014a) surpasses the calibration even completely. It initialises its parameters randomly and updates them with the expectation maximisation (EM) algorithm during actual use of the BCI. Although this method was shown to obtain impressive accuracy levels in practice, it does not have the theoretical guarantee to always achieve excellent classification performance. This makes the EM-based self-learning decoder unreliable.

In collaboration with David Hübner and Michael Tangermann (University of Freiburg), Pieter-Jan Kindermans and Klaus-Robert Müller (Technical University of Berlin) we propose the application of the learning from label proportions (LLP) idea, by Quadrianto et al. (2009), to ERP-BCI for the classification of responses without labelled data. The LLP method requires the data to be observed in two groups with known proportion of target and non-target stimulus responses. For that purpose, we use our flexible paradigm to merge two stimulus paradigms with a very different relative frequency of target stimuli. The parameters of the LLP-based decoder are guaranteed to converge to the supervised solution when more data is collected. As such, we obtain the first self-learning decoder that is robust and capable of classifying ERPs reliably.

We present the results of an online spelling experiment with 13 subjects, conducted in collaboration with the University of Freiburg. The results confirm that the LLP-based unsupervised decoder is robust and achieves an accuracy level that converges to the level obtained with a traditional supervised classifier. Nevertheless, it learns slower and therefore is less effective compared to a well-initialised EM decoder.

Mixing unsupervised model estimators

With the previous contribution, two self-learning decoding methods are available that demonstrate contrasting behaviour during use of the

ERP-BCI. The EM-based decoding method from Kindermans et al. (2012, 2014a) has the potential to learn very fast and quickly achieve a high classification accuracy. However, due to the random initialisation of its parameters and their update with the EM algorithm, the obtained performance is highly variable. In contrast, our LLP decoder is very robust. Yet, the high reliability comes at the cost of a slower learning process.

We propose a method that combines both unsupervised decoders theoretically. In this way, we adopt the benefits of each method and obtain the first self-learning decoder that is both reliable and effective. The novel decoder's parameter values are acquired as a statistical mixture of the parameter estimations obtained with the LLP and EM method. We come up with an analytical formula to compute the optimal mixing coefficient from the data collected during use.

The mixing method is compared to both aforementioned methods in an offline simulation of a spelling experiment with 13 subjects. The results reveal the impressive performance that is obtained with this new self-learning decoder in terms of AUC and symbol selection accuracy. It learns faster than any of the methods it is made from. In addition, it is as robust as the LLP decoder. These results are confirmed by an online experiment on six subjects, carried out by David Hübner and Michael Tangermann at the University of Freiburg.

We finally obtain a calibrationless ERP-BCI that is robust and capable of learning exceptionally fast from unlabelled data recorded during use of the BCI. It makes the unlabelled data as valuable as labelled data recorded during a separate calibration session. This was only possible by adapting the application of the ERP-BCI to the requirements of the decoder with our tunable stimulus presentation paradigm.

An automated diagnosis and lateralisation system for temporal lobe epilepsy

A relatively new application of BCIs is the identification of neurological diseases from recorded brain activity. The last contribution of this dissertation is the development of a data-driven tool for the diagnosis and lateralisation of temporal lobe epilepsy (TLE) that can be easily

applied in any clinical environment.

The traditional presurgical procedure for TLE patients involves a long monitoring of the brain to record pathological activity. This is expensive and very stressful for the patient, but necessary for accurate diagnosis and determination of the epileptogenic zone in the brain. These patients could benefit greatly from a system that is capable of making a fast and accurate diagnosis.

It has been shown recently that neurological diseases can be indicated by irregularities in the functional relationship between the activity in different regions of the brain. These directed connectivity values can be computed from very short scalp EEG recordings and do not require the presence of visible pathological activity in the recording. We propose a system that can classify new subjects as left-TLE, right-TLE or healthy, based on these connectivity values.

We build two random forests classifiers: one for diagnosis (TLE vs. healthy subjects) and one for lateralisation (LTLE vs. RTLE). They are trained on a database of subjects for which the correct classification is known. The subset of connectivities that contains the most relevant information is selected automatically and takes the interaction between connectivities into account. In this way, we go beyond standard statistical methods that consider the different connectivities individually.

We use a dataset including 20 left-TLE patients, 20 right-TLE patients and 35 healthy controls. The results of a leave-one-out procedure are presented, in which the system is built on all but one subject and tested on the subject that was left out. The automatic feature selection reveals which connectivities can serve as biomarkers for the disease. It illustrates the importance of taking the interaction between these connectivities into account to obtain a system that can diagnose and lateralise subjects accurately based on their connectivity values.

1.4.2 Outline of this book

The next chapter gives an overview of machine learning techniques and concepts that are relevant for this work. In Chapter 3 we present our new stimulus presentation paradigm and use it to determine the

underlying mechanism of interaction between the application and the state of the art calibrationless decoding method in ERP-BCI. Afterwards, in Chapter 4, a new calibrationless decoding method is presented that is more reliable. The benefits of both decoders are combined with the mixing method proposed in Chapter 5. This concludes the symbiotic design of ERP-BCI. In Chapter 6, we use machine learning to develop a completely different BCI: an automated diagnosis and lateralisation system for temporal lobe epilepsy. Finally, our conclusions and ideas for future work are presented in Chapter 7.

1.5 List of publications

Journal publications

1. **Verhoeven T.**, Buteneers P., Wiersema R.J., Dambre J., Kindermans P.-J. (2015). Towards a symbiotic brain-computer interface: exploring the application-decoder interaction. *Journal of Neural Engineering*, 12(6)

2. **Verhoeven T.**, Hübner D., Tangermann M., Müller K.R., Dambre J., Kindermans P.-J. (2017). Improving zero-training brain-computer interfaces by mixing model estimators. *Journal of Neural Engineering*, 14(3)

3. Hübner D., **Verhoeven T.**, Schmid K., Müller K.R., Tangermann M., Kindermans P.-J. (2017). Learning from label proportions in brain-computer interfaces: online unsupervised learning with guarantees. *PLoS ONE*, 12(4)

4. Korshunova I., Kindermans P.-J., Degrave J., **Verhoeven T.**, Brinkmann B. H., Dambre J. (2017). Towards improved design and evaluation of epileptic seizure predictors. *IEEE Transactions on Biomedical Engineering*, in press

5. **Verhoeven T.**, Coito A., Plomp G., Thomschewski A., Pittau F., Trinka E., Wiest R., Schaller K., Michel C.M., Seeck M., Dambre J., Vulliemoz S., van Mierlo P. (2017). Automated

diagnosis of temporal lobe epilepsy in the absence of interictal spikes. *NeuroImage: Clinical*, under review

6. Idzhar Ismail L., **Verhoeven T.**, Dambre J., wyffels F. (2017). Leveraging robotics research for children with autism: a review. *IEEE Reviews in Biomedical Engineering*, under review

Conference abstracts

1. Buteneers P., **Verhoeven T.**, Kindermans P.-J., Schrauwen B. (2013). A case study demonstrating the pitfalls during evaluation of a predictive seizure detector. *In Proceedings of the 6st International Workshop on Seizure Predictions.*

2. **Verhoeven T.**, Buteneers P., Schrauwen B., Kindermans P.-J. (2014). An unsupervised plug and play BCI with consumer grade hardware. *Machine Learning Summer School Beijing*

3. **Verhoeven T.**, Buteneers P., Schrauwen B., Kindermans P.-J. (2014). An unsupervised plug and play BCI with consumer grade hardware. *Berlin BCI Winter School*

4. **Verhoeven T.**, Kindermans P.-J.,Buteneers P., Schrauwen B. (2014). Switching characters between stimuli improves P300 speller accuracy. *In Proceedings of the 6th International Brain-Computer Interface Conference.*

5. **Verhoeven T.**, Strobbe G., van Mierlo P., Buteneers P., Vandenberghe S., Dambre J. (2015). A Bayesian model to estimate individual skull conductivity for EEG source imaging. *In Proceedings of the 7th International Workshop on Seizure Predictions.*

6. **Verhoeven T.**, Strobbe G., van Mierlo P., Buteneers P., Vandenberghe S., Dambre J. (2015). A Bayesian model to estimate individual skull conductivity for EEG source imaging. *International Conference on Basic and Clinical Multimodal Imaging*

7. **Verhoeven T.**, Kindermans P.-J., Vandenberghe S., Dambre J. (2016). Reducing BCI calibration time with transfer learning:

a shrinkage approach. *In Proceedings of the Sixth International Brain-Computer Interface Meeting: BCI Past, Present, and Future.*

8. **Verhoeven T.**, Coito A., van Mierlo P., Seeck M., Michel E.M. Plomp G., Dambre J., Vulliemoz S. (2016). Using Random Forest for Diagnosis and Lateralization of Temporal Lobe Epilepsy from EEG-based Directed Functional Connectivity. *12th European Congress on Epileptology*

9. **Verhoeven T.**, Coito A., van Mierlo P., Seeck M., Michel E.M. Plomp G., Dambre J., Vulliemoz S. (2016). Automatic diagnosis of temporal lobe epilepsy and its lateralization using EEG-based directed functional connectivity. *Organization for Human Brain Mapping.*

10. Hübner D., **Verhoeven T.**, Kindermans P.-J., Tangermann M. (2017). Mixing two unsupervised estimators for event-related potential decoding: an online evaluation. *In Proceedings of the 7th International Brain-Computer Interface Conference.*

11. Hübner D., Kindermans P.-J., **Verhoeven T.**, Tangermann M. (2017). Improving learning from label proportions by reducing the feature dimensionality. *In Proceedings of the 7th International Brain-Computer Interface Conference.*

2
Machine learning

In this chapter I will explain machine learning in general and describe the specific techniques that are used in this dissertation. For illustration purposes I will use the following example. Suppose we want to make a computer application that determines the price of a house from several of its features, e.g. its location, the year it was built, the total living area, the number of rooms etc. We have an example dataset that contains 150 houses for which the feature values and selling price are given. One approach to solve this task is to explicitly program the application and return a price for each possible combination of feature values:

```
if YearBuilt > 1980:
  if LivingArea < 150:
    if bedrooms = 1:
      price = $ 250000
    elif bedrooms = 2:
      price = $ 300000
    elif bedrooms = 3:
      ...
  elif LivingArea < 300:
    ...
  else:
    ...
elif YearBuilt < 1980:
  ...
```

It is clear that this quickly becomes a tedious task for a large dataset

of houses with a lot of features. Furthermore, we cannot expect the price to be very accurate when the house that is valued isn't already present in the example dataset.

Machine learning (ML) is the general term for methods and techniques that solve a task without programming the computer explicitly. The task is described generally as finding the desired output value for a given input. With ML, a computer can learn this input-output model from data. The model is then applied to new, unseen input samples. In the example above, the model's input is the set of features that describes a house and the continuous output value is an estimate of its price. This type of problem is called a *regression* task. In *classification* tasks, the model assigns the input to a certain category or class.

ML techniques can be categorised by the type of data that is used to train the model. First of all, in supervised ML the training data contains pairs of input samples and their desired output, as in the example above. Secondly, unsupervised ML models are only provided with input data, not labelled with a specific desired output. The task is to find hidden structure in this data. For example, this technique has been used to differentiate between types of tissue in medical images. Finally, reinforcement learning techniques are mainly used in robotics. By taking actions and observing the subsequent penalty or reward, the robot learns from trying out things.

The first section in this chapter will explain the core ML concepts that are important to understand the challenges we face in this work: generalisation, over-fitting and regularisation. Next, linear regression and linear discriminant analysis will be discussed. These supervised ML techniques are commonly used to decode brain activity. Afterwards, we will focus on the Gaussian mixture model, an unsupervised machine learning method, and the expectation maximisation algorithm that is used to train this model. Finally, I will explain an unsupervised classification method that was specifically designed for ERP-based BCIs by Kindermans et al. (2012). This method learns from the data recorded during actual use of the BCI, thereby avoiding the calibration session. The methods developed in the following chapters will extend this unsupervised decoding method.

The mathematical notation in this book follows the convention in

research literature. Vectors are denoted by lower case bold symbols, e.g. $\boldsymbol{x}$, and are assumed to be column vectors. A superscript ^{T} indicates the transpose of a matrix or vector, so $\boldsymbol{x}^T$ is a row vector. Capital bold symbols denote matrices, e.g. $\boldsymbol{X}$.

2.1 The benefits and challenges of learning from data

It is easy to make an input-output model that performs well on the available set of training data. In our house price estimation, the model can be programmed to replicate for each house its price from the training data. However, this system is unable to determine the price for a new house. The power of ML is that the trained model is able to generalise from a limited set of examples to completely new inputs.

The generalisation property does not emerge automatically when applying ML on example data. In the house price estimation, it is obvious that certain features, e.g. the colour of the front door, do not influence the price in general. However, in our limited dataset it could happen by coincidence that the average price of houses with a white front door is just a little higher than for those with a different colour. Consequently, the trained model can slightly overvalue houses with a white front door. This problem of learning irrelevant details from the training data is called *over-fitting*. It decreases the model's performance on new samples. Regularisation techniques avoid the risk of over-fitting by limiting the complexity of the model. This can for example be done by including only those features that have the strongest link with the desired output. On the contrary, when the model is regularised too much, it is not able to include all the relevant information from the training data. This is called *under-fitting*. The challenge in solving a task with ML is to determine the model complexity that optimises the generalisation to new samples. Figure 2.1 graphically presents the relation between the complexity of the model and the error it makes when applied to the training data and new test samples. The complexity that corresponds with optimal generalised performance is indicated by the vertical line.

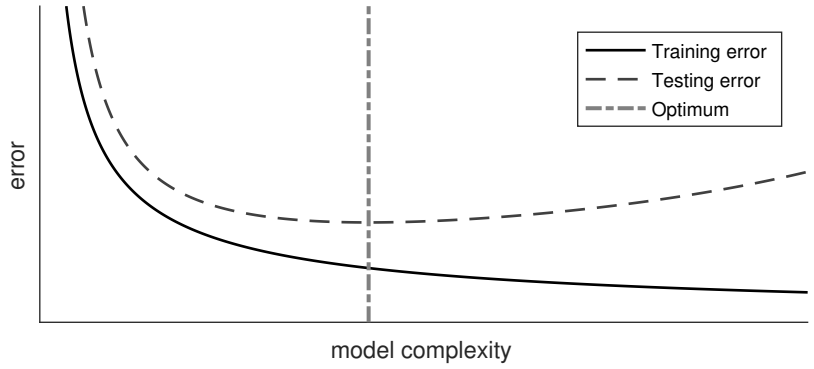

Figure 2.1: Model error on training data and unseen test data as a function of increasing model complexity. To the left of the vertical line the model is not complex enough to incorporate all the relevant knowledge: under-fitting. This is indicated by a high error on both training and test data. To the right of the vertical line the model is learning details that are irrelevant to the task: over-fitting. This is indicated by a decreased error on training data but an increased error on test data. The vertical line denotes the model complexity that optimises the generalised performance on new samples.

Only the training data is available to build the model. As illustrated in Figure 2.1, minimising the error on this data leads inevitably to over-fitting. Furthermore, the desired output is not known for new samples, so these cannot be used to measure the generalised performance. To estimate how well a model will generalise to unseen samples, we use a technique called *cross-validation*. In k-fold cross-validation, the training data is split up in k parts of equal size. One part is excluded and the other data is used to train the model. The model is then validated on the data that was left out, see Figure 2.2. This is repeated as each part is left out once. The average performance of the model on the validation data is an estimate for the performance on new samples. This technique can be used to test different models and as such determine te optimal level of regularisation.

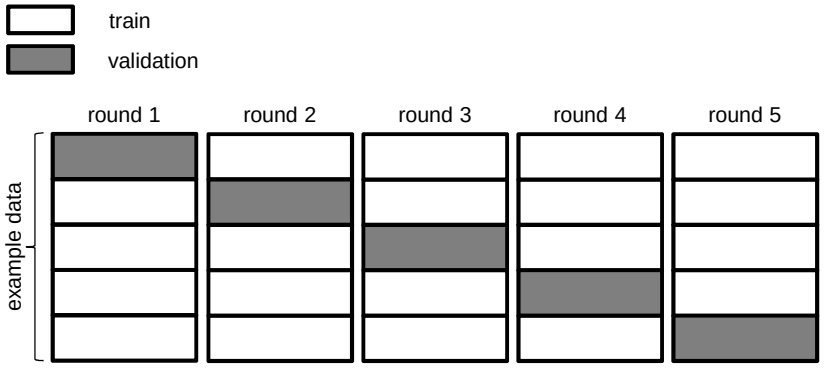

Figure 2.2: Schematic representation of the 5-fold cross-validation technique. The training data is split in five parts of equal size. In each round, one part is left out and the remaining data is used to train the model. This model is then validated on the part that was left out. The average validation result is an estimate of the model's performance on unseen samples.

2.2 Supervised machine learning

The most straightforward tasks involve examples of input-output pairs. A large number of supervised ML techniques have been developed to train and regularise a model with this type of data. Methods to solve regression as well as classification problems in a supervised way will be described in this section.

2.2.1 Linear regression

One of the most simple, yet most commonly used, machine learning techniques is linear regression. Linear regression assumes a linear relation between the features of input $\boldsymbol{x}$ and the corresponding output y. Consider the example of the price estimation for houses. The graph in Figure 2.3 shows one feature, the total living area in square meters, on the horizontal axis and the price of the house on the vertical axis. Every dot in the graph represents a house from the set of training data. There is a clear linear trend, which is represented by the straight line that is drawn through the dots. This linear relation can also be

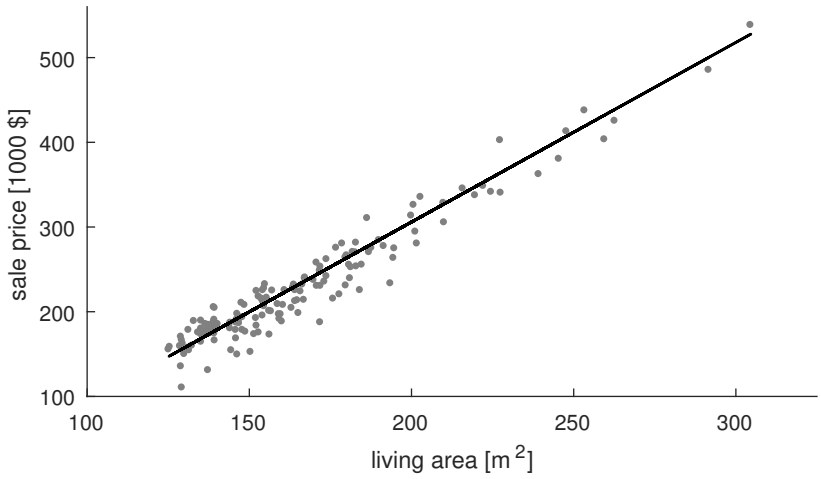

Figure 2.3: Example of a linear regression model: valuing a house. The 150 houses from the training set are represented by dots, indicating the sale price on the vertical axis and the living area on the horizontal axis. The straight line indicates the linear relation between price and living area, learned from the training data. Data obtained from: `https://www.kaggle.com/c/house-prices-advanced-regression-techniques`

represented by the following formula:

$$y = w_0 + w_1 x.$$

y is the price of the house and x is the living area. w_0 and w_1, respectively, depict the intercept of the line with the vertical axis and the slope of the line. These model parameters are determined by means of the training data. The price of a new house can be estimated by substituting x for its total living area and calculating the corresponding value of y.

In general, the linear regression method models the relation between the output y and the D features of the input $\boldsymbol{x} = [x_1, ..., x_D]^T$ as follows:

$$y = w_0 + w_1 x_1 + w_2 x_2 + ... + w_D x_D.$$

the bias term w_0 can be interpreted as the weight given to an extra feature with constant value 1 in each input sample $\boldsymbol{x} = [1, x_1, ..., x_D]^T$.

The formula can then be rewritten in vector notation:

$$y = \boldsymbol{w}^T \boldsymbol{x}$$

The regression problem is solved by optimising $\boldsymbol{w}$. To optimise $\boldsymbol{w}$, we need to measure its quality by using a *loss function*. The most commonly used loss function for linear regression is the quadratic loss function explained next.

Least squares solution

We denote the N input samples of the training data as $\boldsymbol{x}_n$ ($n = 1, ..., N$) and their desired output as t_n. The output of the trained model on $\boldsymbol{x}_n$ will be denoted by y_n. The optimal model parameters are those that make the example data fit the model optimally. As a measure of fit we define the sum of squared errors between t_n and y_n:

$$\begin{aligned} E_{sse}(\boldsymbol{w}) &= \sum_{n=1}^{N} (t_n - y_n)^2 \\ &= \sum_{n=1}^{N} (t_n - \boldsymbol{w}^T \boldsymbol{x}_n)^2 \end{aligned}$$

In vector notation this becomes:

$$E_{sse}(\boldsymbol{w}) = \|\boldsymbol{t} - \boldsymbol{X}\boldsymbol{w}\|^2$$

with $\boldsymbol{t} = [t_1, ..., t_n]^T$ and $\boldsymbol{X}$ the $N \times D$ design matrix with the input samples $\boldsymbol{x}_n^T$ in its rows. The optimal weight vector $\boldsymbol{w}^*$ is the one that minimises this quadratic loss function. It is obtained by setting the gradient of the loss function with respect to $\boldsymbol{w}$ equal to zero:

$$\begin{aligned} \nabla_{\boldsymbol{w}} E_{sse}(\boldsymbol{w}^*) &= 2\boldsymbol{X}^T(\boldsymbol{t} - \boldsymbol{X}\boldsymbol{w}^*) \\ &= \boldsymbol{0} \end{aligned}$$

Solving for $\boldsymbol{w}^*$ yields the optimal solution for least squares regression (LSR):

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{X}^T\boldsymbol{t}) \tag{2.1}$$

This approach can over-fit the training data. Therefore we need to regularise it.

Ridge regression

As discussed in Section 2.1, the complexity of the model is regularised in order to optimise the model's generalisation performance. Remark that the weight w_i of a feature $x_{n,i}$ determines the influence of this feature on the output of the model. Consequently, the effective model complexity can be measured as the L_2-norm of the weight vector: $\|\boldsymbol{w}\|^2$. Features that are considered completely unrelated to the output have a coefficient value close to zero in the optimal model and as such do not contribute to the complexity. In order to regularise the training of the model, this effective model complexity is added to the objective function:

$$E_{sse,reg}(\boldsymbol{w}) = \|\boldsymbol{t} - \boldsymbol{X}\boldsymbol{w}\|^2 + \lambda\|\boldsymbol{w}\|^2$$

With this extra term, a higher complexity is only allowed if this significantly reduces the squared error of the model. How significant this reduction has to be is determined by the regularisation constant λ. A higher value of λ results in a stronger regularisation.

Optimising this new objective function leads to the regularised solution known as *ridge regression*:

$$\boldsymbol{w}^* = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}(\boldsymbol{X}^T\boldsymbol{t}) \tag{2.2}$$

Ridge regression searches for the model with the best trade-off between minimising the complexity and fitting the observed data. The cross-validation technique described in Section 2.1 can be used to find the optimal trade-off coefficient λ.

Interpretation from probability theory

The least squares solution for linear regression can also be obtained from a probabilistic point of view. We denote the error of our model output y with respect to the true output t as ϵ and assume this error

to be normally distributed with zero mean and precision β:

$$\begin{aligned} t &= y + \epsilon \\ \epsilon &\sim \mathcal{N}(0, \beta^{-1}) \end{aligned}$$

β^{-1} is the size of the error on the model. With this assumption, a probability distribution for the target output t, given the input $\boldsymbol{x}$ and the model $\boldsymbol{w}$, can be defined:

$$\begin{aligned} p(t|\boldsymbol{x}, \boldsymbol{w}) &= \mathcal{N}(\boldsymbol{w}^T\boldsymbol{x}, \beta^{-1}) \\ &= \sqrt{\frac{\beta}{2\pi}} \exp\left(-\frac{\beta}{2}(t - \boldsymbol{w}^T\boldsymbol{x})^2\right) \end{aligned}$$

The vector $\boldsymbol{w}$ is again chosen to fit the model to the training data. This time the quality of $\boldsymbol{w}$ is not measured using the sum of squared errors but as how likely the prediction is, given the model $\boldsymbol{w}$. The likelihood of the data, given $\boldsymbol{w}$, is defined as follows:

$$\mathcal{L}(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}) = \prod_{n=1}^{N} p(t_n|\boldsymbol{x}_n, \boldsymbol{w})$$

Maximising this likelihood is equivalent to minimising its negative logarithm. The objective function then becomes:

$$\begin{aligned} E &= -\log \mathcal{L}(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}) \\ &= -\sum_{n=1}^{N} \log p(t_n|\boldsymbol{x}_n, \boldsymbol{w}) \\ &= -\frac{N}{2} \log\left(\frac{\beta}{2\pi}\right) + \sum_{n=1}^{N} \frac{\beta}{2}(t_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2 \end{aligned}$$

Excluding terms that are independent of $\boldsymbol{w}$ we obtain the sum of squares error function:

$$E \propto \sum_{n=1}^{N} (t_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2$$

This explains the choice for this objective function in LSR. Minimising the sum of squared errors maximises the likelihood of the observed

data under the assumption that the noise on the model's output is normally distributed with zero mean. The weight vector $\boldsymbol{w}$ that optimises this objective is given by Equation 2.1.

To regularise the model, we define the prior probability distribution on the vector $\boldsymbol{w}$ as a multivariate normal distribution with zero mean and isotropic covariance matrix: $\alpha^{-1}\boldsymbol{I}$:

$$p(\boldsymbol{w}) = \mathcal{N}(\mathbf{0}, \alpha^{-1}\boldsymbol{I})$$

This prior indicates that less complex models, with a weight vector closer to $\mathbf{0}$, are more likely to describe the generalised input-output relation. The optimal weight vector is again the one that is most likely, given the observed data. Using Bayes' rule we can find the posterior probability on $\boldsymbol{w}$:

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{t}) = \frac{p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w})p(\boldsymbol{w})}{p(\boldsymbol{t}|\boldsymbol{X})}$$

Now the maximum a posteriori estimate is found by minimising the negative logarithm of this likelihood. Dropping terms that do not depend on $\boldsymbol{w}$ and writing in vector notation we get:

$$\begin{aligned} E &= -\log p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}) - \log p(\boldsymbol{w}) \\ &= \sum_{n=1}^{N} \frac{\beta}{2}(t_n - \boldsymbol{w}^T\boldsymbol{x}_n)^2 + \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} \\ &\propto \|\boldsymbol{t} - \boldsymbol{X}\boldsymbol{w}\|^2 + \frac{\alpha}{\beta}\|\boldsymbol{w}\|^2 \end{aligned}$$

We obtain the same objective function as in ridge regression, for which the optimal weight vector is given in Equation 2.2. The regularisation constant is replaced by the ratio of the precision α on the prior of the weight vector and the precision β on the noise. Optimal values for these hyper-parameters can be found again with the cross-validation technique.

2.2.2 Linear regression for classification

In a binary classification problem, the task is to assign a class label $y \in \{t_+, t_-\}$ to an input $\boldsymbol{x}$. In linear classification methods, the

assignment is done by calculating the dot product of the input vector $\boldsymbol{x}$ with a weight vector $\boldsymbol{w}$ and comparing the result to a threshold value, which can be assumed to be zero without loss of generality:

$$y = \begin{cases} t_+ & \text{if } \boldsymbol{w}^T\boldsymbol{x} + w_0 > 0 \\ t_- & \text{otherwise} \end{cases}$$

This product can be interpreted as the projection of the sample $\boldsymbol{x}$ into a single dimension. The direction of this projection is determined by the weight vector $\boldsymbol{w}$.

Consider for example the problem of classifying a pigmented skin mark as a benign birthmark or a cancerous melanoma, based on several features such as the size, shape and colour of the mark. A skin mark is given the label $t = +1$ when it is considered malignant (cancerous) and $t = -1$ when it is classified as benign (a harmless birthmark). Figure 2.4 presents a toy example dataset containing 200 benign and 20 malignant skin marks. In contrast with Figure 2.3, the horizontal and vertical axis now each represent a feature: the size of the mark and a measure of its symmetry. The data points from the two classes are indicated by different markers. From this dataset, the computer needs to learn how to classify new skin marks as benign or malign.

Although developed for regression problems, least squares regression can be used as a linear classification method. For this purpose, the class labels in the example dataset are interpreted as continuous output vales. In our example case, the least squares solution is a weight vector $\boldsymbol{w} = [w_1, w_2]^T$ and a bias term w_0. The border between the two classes is described by the equation $w_0 + w_1x_1 + w_2x_2 = 0$ and represented by the dashed line on the graph in Figure 2.4 that separates the input space in two regions. A new sample is assigned to the positive class $t = +1$ if it falls above this line and to the negative class $t = -1$ otherwise. The direction of $\boldsymbol{w}$ is indicated by the full line. The dotted lines show the projection of samples on this direction. Comparing this one-dimensional projection to the bias w_0 simplifies the classification procedure for high-dimensional data. The bias w_0 can be altered to optimise either the fraction of malignant marks classified correctly (sensitivity), the fraction of benign marks

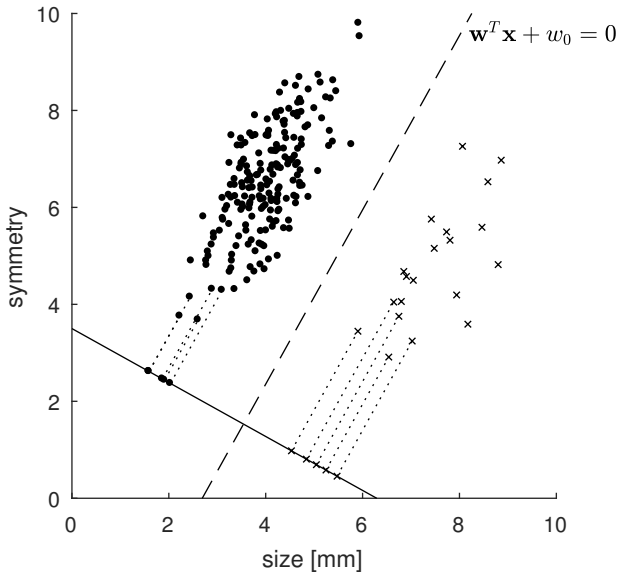

Figure 2.4: Linear classification of skin marks as benign • or malignant ×. The horizontal and vertical axis denote two features of the mark: the size in millimetres and a measure of symmetry in its shape. The dashed line shows the border between the two classes as learned by a linear classifier. Classification is simplified by projecting the samples in the direction of the weight vector $\boldsymbol{w}$, represented by the full line.

classified correctly (specificity) or the total classification accuracy.

The output of the LSR classifier is binary: one of the two classes. In contrast, some tasks require a certain measure of confidence when assigning a sample to a specific class. For example, it might be interesting to know the probability that a skin mark is malignant. If this probability is around the 50 % chance level, a second opinion can be acquired from a medical expert. Other classification methods, such as logistic regression and linear discriminant analysis, make the output interpretable as a probability value.

2.2.3 Linear discriminant analysis

Linear discriminant analysis (LDA) is a machine learning technique for classification (Bishop, 2007). It differs from LSR in the way the op-

timal weight vector $\boldsymbol{w}$ is determined. The model is generative, which means that we model how the data is generated, i.e. the probability distribution $p(\boldsymbol{x})$. In LDA, it is assumed that the input samples are normally distributed with a covariance structure $\boldsymbol{\Sigma}$ and a mean $\boldsymbol{\mu}_k$ that depends on the class C_k to which the sample belongs. For a two-class problem we get:

$$\begin{aligned} n &= 1, ..., N \\ k &\in 0, 1 \\ p(t_n = k) &= \pi_k \\ p(\boldsymbol{x}_n | t_n = k) &= \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}) \\ &= (2\pi)^{-\frac{D}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)} \end{aligned}$$

With π_k the prior probability that a sample belongs to class C_k. Using Bayes' rule we obtain an expression for the probability that a sample $\boldsymbol{x}_n$ belongs to class C_1.

$$\begin{aligned} p(t_n = 1 | \boldsymbol{x}_n) &= \frac{p(\boldsymbol{x}_n | t_n = 1) p(t_n = 1)}{p(\boldsymbol{x}_n)} \\ &= \frac{\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \pi_1}{\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \pi_1 + \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}) \pi_0} \\ &= \frac{1}{1 + e^{-(\boldsymbol{w}^T \boldsymbol{x} + w_0)}} \end{aligned} \tag{2.3}$$

with:

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \\ w_0 &= -\frac{1}{2} \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_0 + \log \pi_1 - \log \pi_0 \end{aligned}$$

The model parameters are found again by fitting the model to the training data. The likelihood of the observed data, given the LDA model is:

$$\begin{aligned} E_{mle} &= \prod_{n=1}^{N} p(\boldsymbol{x}_n, t_n | \pi_0, \pi_1, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \\ &= \prod_{n=1}^{N} (\pi_1 \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}))^{t_n} (\pi_0 \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}))^{1 - t_n} \end{aligned}$$

Minimising the negative log-likelihood results in the following estimates for the model parameters:

$$\pi_k = \frac{N_k}{N}$$
$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{\boldsymbol{x}_n \in C_k} \boldsymbol{x}_n$$
$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{k=0}^{1} \sum_{\boldsymbol{x}_n \in C_k} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T$$

Where N_k is the number of samples in the training data that belong to class C_k.

The generative model naturally leads to an output $y = p(t_n = 1|\boldsymbol{x}_n)$ that is interpretable as a probability value. A sample is assigned to the specific class C_1 if it is more likely to belong to this class than to the other, i.e. when $p(t_n = 1|\boldsymbol{x}_n) > 0.5$. Equation 2.3 shows that this is the case when $\boldsymbol{w}^T\boldsymbol{x} + w_0 > 0$. Hence, LDA is an alternative linear classifier.

Relation to least squares regression

We will now show that the LDA solution to the two-class classification problem can be obtained as a specific case of the LSR solution. We have shown that LSR assumes the projection of the input sample to be normally distributed. This assumption is in fact less strict than LDA, which assumes the input samples themselves to be normally distributed. We now make two extra assumptions. The target labels are rescaled to N/N_1 for class C_1 and $-N/N_0$ for class C_0. In addition, we assume that the average sample $\boldsymbol{\mu}$ is subtracted from every sample in the design matrix $\boldsymbol{X}$. $\boldsymbol{X}$ is said to be *centred* to zero mean.

We start from Equation 2.1, the original LSR solution for the weight vector $\boldsymbol{w}$:

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{t}$$

By filling in the rescaled labels, the second factor in this formula can

be written as follows:

$$\begin{aligned}\boldsymbol{X}^T\boldsymbol{t} &= \sum_{C_1} \boldsymbol{x}_n t_n + \sum_{C_0} \boldsymbol{x}_n t_n \\ &= N\frac{\sum_{C_1}\boldsymbol{x}_n}{N_1} - N\frac{\sum_{C_0}\boldsymbol{x}_n}{N_0} \\ &= N(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\end{aligned}$$

This yields:

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}N(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \tag{2.4}$$

Multiplying both sides of Equation 2.4 with $\boldsymbol{X}^T\boldsymbol{X}$ we obtain:

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} = N(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \tag{2.5}$$

The left hand side of Equation 2.5 can be expanded using the assumption that $\boldsymbol{X}$ is centred, i.e. $\boldsymbol{\mu} = (N_1\boldsymbol{\mu}_1 + N_0\boldsymbol{\mu}_0)/N = \mathbf{0}$:

$$\begin{aligned}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} &= \left[\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^T\right]\boldsymbol{w} \\ &= \left[\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^T - N\boldsymbol{\mu}\boldsymbol{\mu}^T\right]\boldsymbol{w} \\ &= \sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^T\frac{1}{N}\left(N_1^2\boldsymbol{\mu}_1\boldsymbol{\mu}_1^T + N_0^2\boldsymbol{\mu}_0\boldsymbol{\mu}_0^T + N_0N_1\boldsymbol{\mu}_1\boldsymbol{\mu}_0^T + N_0N_1\boldsymbol{\mu}_0\boldsymbol{\mu}_1^T\right)\boldsymbol{w} \\ &= \left[\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^T - N_1\boldsymbol{\mu}_1\boldsymbol{\mu}_1^T - N_0\boldsymbol{\mu}_0\boldsymbol{\mu}_0^T\right]\boldsymbol{w} \\ &\quad + \frac{N_1N_0}{N}\left[\boldsymbol{\mu}_1\boldsymbol{\mu}_1^T + \boldsymbol{\mu}_0\boldsymbol{\mu}_0^T - \boldsymbol{\mu}_1\boldsymbol{\mu}_0^T - \boldsymbol{\mu}_0\boldsymbol{\mu}_1^T\right]\boldsymbol{w} \\ &= \left[\sum_{C_1}(\boldsymbol{x}_n - \boldsymbol{\mu}_1)(\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T + \sum_{C_0}(\boldsymbol{x}_n - \boldsymbol{\mu}_0)(\boldsymbol{x}_n - \boldsymbol{\mu}_0)^T\right]\boldsymbol{w} \\ &\quad + \frac{N_1N_0}{N}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^T\boldsymbol{w} \\ &= N\boldsymbol{\Sigma}\boldsymbol{w} + A(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\end{aligned}$$

With $\boldsymbol{\Sigma}$ the covariance matrix as defined before and A a constant

scalar factor. Substituting this result in Equation 2.5 yields:

$$N\boldsymbol{\Sigma}\boldsymbol{w} + A(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = N(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

Remark that a constant factor does not change the solution. Only the direction of $\boldsymbol{w}$ determines the projection, so any vector proportional to $\boldsymbol{w}$ results in the same solution. Consequently, with the rescaled labels and centred data, the least squares solution is equivalent to LDA:

$$\boldsymbol{w} \propto \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$$

The bias w_0 can again be chosen to maximise either the specificity, sensitivity or total accuracy of classification.

Regularised linear discriminant analysis

When dealing with high dimensional data, the amount of available training data can be insufficient to make an accurate estimate of the covariance matrix (Blankertz et al., 2011). Regularising the LDA classifier can be done by shrinking the covariance matrix towards the identity matrix $\boldsymbol{I}$:

$$\boldsymbol{\Sigma}_{reg} = (1 - \lambda)\boldsymbol{\Sigma} + \lambda\boldsymbol{I}$$

An analytical formula was proposed by Ledoit-Wolf (Blankertz et al., 2011; Ledoit and Wolf, 2004; Bartz and Müller, 2014) to find the optimal value for the regularisation coefficient λ. This avoids the more computationally intensive cross-validation procedure.

2.3 Unsupervised machine learning

Labelled data may not be available to solve the task because it is expensive or labour intensive to obtain. With unsupervised machine learning methods, patterns or structure can be found in unlabelled data. In this case, the categorisation of data is called a *clustering problem*. This is more challenging as the number of clusters is generally not known and not a single ground truth assignment is available for any cluster.

2.3.1 Gaussian mixture model

A Gaussian mixture model (GMM) is a generative model that is used to model the distribution of unlabelled data. Each input sample $\boldsymbol{x}$ is assumed to belong to one out of K clusters and generated by a multivariate normal distribution with cluster-dependent mean $\boldsymbol{\mu}_k$ and covariance structure $\boldsymbol{\Sigma}_k$. The cluster to which the sample $\boldsymbol{x}_n$ belongs is described by another random variable $z_n \in 1, ..., K$ that is unobserved or *latent*. The probability mass function of this variable $p(z_n = k) = \pi_k$ denotes the prior probability that a sample belongs to cluster k. We will use the vector $\boldsymbol{z}$ to denote the collection of latent variables z_n for $n = 1, ..., N$. The complete model is summarised as follows:

$$
\begin{aligned}
k &= 1, ..., K \\
n &= 1, ..., N \\
p(z_n = k) &= \pi_k \\
\sum_{k=1}^{K} \pi_k &= 1 \\
p(\boldsymbol{x}_n | z_n = k) &= \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
p(\boldsymbol{x}_n) &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)
\end{aligned}
$$

The generative model used in LDA is in fact a GMM with $K = 2$ components and a shared covariance matrix. In LDA, the parameter estimates are calculated as sample statistics on a set of labelled training data. The observed labels t_n that are used in LDA are replaced by the latent variable z_n in the GMM. GMMs can model more complex distributions with more components having different covariance structures. An example for $K = 3$ is illustrated in Figure 2.5.

To train the GMM we need to determine the parameters of each model component and the prior distribution on the latent variable $p(z)$. Bayes' rule can then be applied to infer $p(z = k|\boldsymbol{x}_n)$ and assign the sample $\boldsymbol{x}_n$ to the most likely cluster. The expectation maximisation method is used commonly to obtain maximum likelihood estimates for these parameters. This method and its application to GMM

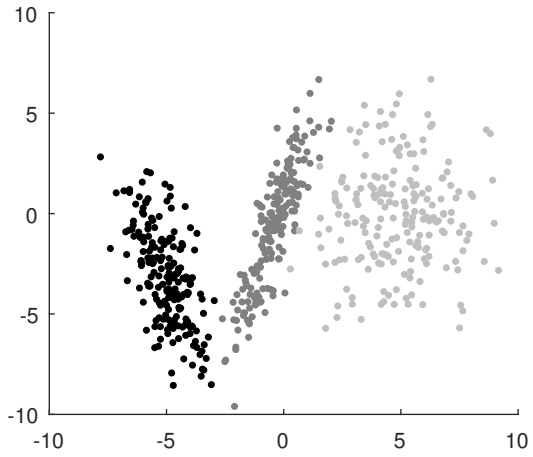

Figure 2.5: A set of samples generated by a Gaussian mixture model with $K = 3$ components. Each dot represents a single sample with two features, shown on the vertical and horizontal axis. The data in each cluster is normally distributed with a cluster-dependent mean and covariance structure.

training will be explained next.

Expectation maximisation

Expectation maximisation (EM) is an iterative algorithm to find a maximum likelihood estimate for a set $\boldsymbol{\Theta}$ of model parameters (Dempster et al., 1977; McLachlan and Krishnan, 2007). It is especially suitable in cases where it is difficult to directly maximise the likelihood of the observed data $p(\boldsymbol{X}|\boldsymbol{\Theta})$, but where a latent variable z can be defined for which the joint likelihood $p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta})$ is easier to optimise.

The algorithm starts by initialising $\boldsymbol{\Theta}$ randomly or by means of an algorithm that can give a fast inaccurate guess that is better than random (e.g. the K-means algorithm (Lloyd, 1982)). Next, the algorithm iteratively executes the following two steps:

- **E-step**: In the expectation step, the current estimate $\boldsymbol{\Theta}$ of the model parameters is used to calculate the probability $p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta})$ for each possible value of the latent variables in $\boldsymbol{z}$. In this way, the expected value of the log-likelihood function can be defined

with respect to the conditional distribution $p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta})$:

$$\begin{aligned} E_{mle} &= \boldsymbol{E}_{\boldsymbol{z}|\boldsymbol{X},\boldsymbol{\Theta}} \left[\log p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta})\right] \\ &= \sum_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}) \log p(\boldsymbol{X}, \boldsymbol{z}|\boldsymbol{\Theta}) \end{aligned}$$

- **M-step**: In the maximisation step, this expected value is maximised by updating the model parameters to a new estimate $\hat{\boldsymbol{\Theta}}$:

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\hat{\boldsymbol{\Theta}}} \sum_{\boldsymbol{z}} p(\boldsymbol{z}|\boldsymbol{X}, \boldsymbol{\Theta}) \log p(\boldsymbol{X}, \boldsymbol{z}|\hat{\boldsymbol{\Theta}})$$

These two steps are repeated until the parameter estimates converge.

EM for Gaussian mixture model training

The GMM defines a latent variable z_n for each sample $\boldsymbol{x}_n$. Therefore, the EM algorithm can be applied directly to find estimates for the model parameters. The set of parameters contains the mean and covariance matrices for each Gaussian component, as well as the prior distribution of the latent variable:

$$\boldsymbol{\Theta} = \{\pi_1, ..., \pi_K, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K, \boldsymbol{\Sigma}_1, ..., \boldsymbol{\Sigma}_K\}$$

In the E-step, the probability $p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta})$ that $\boldsymbol{x}_n$ belongs to cluster k is calculated for each sample $\boldsymbol{x}_n$ and each cluster. This probability can be interpreted as a measure of how well the parameters of the k^{th} cluster succeed at explaining the observation of the sample $\boldsymbol{x}_n$. Application of Bayes' rule yields:

$$\begin{aligned} p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta}) &= \frac{p(\boldsymbol{x}_n|z_n = k, \boldsymbol{\Theta}) p(z_n = k|\boldsymbol{\Theta})}{p(\boldsymbol{x}_n|\boldsymbol{\Theta})} \\ &= \frac{\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\pi_k}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \end{aligned}$$

The posterior probability on the latent variables is then used in

the M-step to maximise the expected value of the log-likelihood:

$$E_{mle} = \sum_{z_1=1}^{K} \sum_{z_2=1}^{K} ... \sum_{z_N=1}^{K} \left(\prod_{n=1}^{N} p(z_n|\boldsymbol{x}_n, \boldsymbol{\Theta}) \right) \sum_{n=1}^{N} \log p(\boldsymbol{x}_n, z_n|\boldsymbol{\Theta})$$

As each term $\log p(\boldsymbol{x}_n, z_n|\boldsymbol{\Theta})$ only depends on one latent variable z_n, this expression can be simplified:

$$\begin{aligned} E_{mle} &= \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta}) \left[\log p(\boldsymbol{x}_n|z_n = k, \boldsymbol{\Theta}) + \log p(z_n = k)\right] \\ &= \sum_{n=1}^{N} \sum_{k=1}^{K} p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta}) \left[\log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \log \pi_k\right] \end{aligned}$$

Computing the gradient with respect to each of the parameters π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ and setting it equal to zero yields the following formula for the updated parameters:

$$\begin{aligned} \hat{\pi}_k &= \frac{1}{N} \sum_{n=1}^{N} p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta}) \\ \hat{\boldsymbol{\mu}}_k &= \frac{\sum_{n=1}^{N} p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta}) \boldsymbol{x}_n}{\sum_{n=1}^{N} p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta})} \\ \hat{\boldsymbol{\Sigma}}_k &= \frac{\sum_{n=1}^{N} p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta})(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^{N} p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\Theta})} \end{aligned}$$

With these updated parameter values, the algorithm returns to the E-step until convergence is reached. The relation with LDA can be recognised by considering a mixture of two components. If the class/cluster is known for each sample, the posterior probabilities $p(z_n = k|\boldsymbol{x}_n)$ become 1 or 0 depending on the observed class label. In this case, the formula for $\hat{\pi}_k$ yields the fraction of samples in the example dataset that belongs to each class C_k. Likewise, $\hat{\boldsymbol{\mu}}_k$ and $\hat{\boldsymbol{\Sigma}}_k$ estimate the class-wise mean and covariance matrix.

2.3.2 Unsupervised training for ERP-based BCI

We will now return to brain-computer interfaces based on event-related potentials. The visual ERP speller will be used to make the discussion more tangible. However, the methods presented in this section are applicable to any type of ERP-BCI.

As described in Chapter 1, the visual ERP speller is a BCI for spelling text. The user is presented a grid of symbols on a screen. To spell one symbol, he/she focuses on this *target symbol*. Next, groups of symbols are repeatedly highlighted in the grid and the user is asked to count silently when the desired symbol is highlighted. Each simultaneous highlight of a group of symbols in the grid is called a *stimulus*. A sequence of stimuli in which each symbol is highlighted an equal number of times in the grid is called an *iteration*. The complete sequence of stimuli, consisting of several iterations, presented to spell one symbol is called a *trial* and will be denoted by t, the target symbol during this trial by c_t. The EEG signal recorded in a specific time window after the stimulus is the *stimulus response*. This response is different when the user sees his/her target symbol being highlighted. Hence, the recorded stimulus responses can be subdivided into two classes: target and non-target responses. The feature vector extracted from the response on the i^{th} stimulus during trial t will be denoted by $\boldsymbol{x}_{t,i}$. $y_{t,i}$ is the (unknown) class label of this response. It takes the value $+1$ for a target response and -1 for a non-target response. At the end of a trial, each recorded stimulus response is classified as target or non-target. Using the class predictions and the knowledge of which symbols were highlighted in each stimulus, the target symbol c_t is inferred. The following notations will be used in addition. $\boldsymbol{X}_t$ is the matrix with the responses $\boldsymbol{x}_{t,i}$ recorded during trial t in its rows. $\boldsymbol{X}$ contains the recorded responses from all trials in its rows. The vector $\boldsymbol{y}$ contains the (unknown) class labels of all these observed responses and $\boldsymbol{c}$ contains the desired symbols from all trials.

Blankertz et al. (2011) have shown that the time-domain features of an ERP response are approximately normally distributed with a class-conditional mean and a shared covariance structure. We have described LDA in this chapter as a method that is especially suitable for the classification of normally distributed data. Hence, LDA has

been widely applied for supervised ERP classification and was even shown to be competitive with more complex methods (Lotte et al., 2007; Müller et al., 2008; Blankertz et al., 2011; Kindermans et al., 2011).

In Chapter 1 we explained that the recording of a labelled dataset prior to actual use drastically reduces the usability of BCI systems. For this reason, several methods have been developed that reduce or even eliminate the need for labelled data. When no label information is available, the LDA model can for example be replaced by a GMM to cluster the observed responses. The distribution of the observed ERPs is then described as a mixture of two multivariate Gaussian components with different means and a shared covariance matrix. However, due to the high amount of noise on the recorded ERP signals, it is difficult to obtain an accurate estimate of the GMM parameters.

In this section I will introduce an unsupervised classification method based on the GMM, designed specifically for ERP-based BCIs by Kindermans et al. (2012). It was the first method capable of accurately classifying ERP responses without using a single labelled example. The model tunes its parameters during actual use of the BCI, thereby avoiding the calibration session.

First, I will explain how the constraints imposed by the application are embedded in the expectation maximisation framework to facilitate the estimation of GMM parameters. Afterwards, the complete probabilistic model will be described to classify ERPs without label information.

Exploiting application constraints for EM-based learning

Finding maximum likelihood estimates for the GMM parameters with EM requires the definition of a latent variable. Following the original GMM, the unknown class label $y_{t,i}$ could serve as a latent variable when modelling the distribution of ERP responses $\boldsymbol{x}_{t,i}$. However, the ERP-speller application imposes constraints on the labels $y_{t,i}$ for responses recorded during the same trial t. The constraints can be actively employed to facilitate the estimation of the GMM parameters.

A first constraint imposed by the application is that, during a trial t, there is only one target symbol c_t. Each stimulus that highlights c_t

elicits a target response and each stimulus that does not highlight c_t elicits a non-target response. Consequently, the latent variable $y_{t,i}$ of a response in trial t is linked to the target symbol c_t by means of the stimulus presentation paradigm:

$$y_{t,i}(c_t) = \begin{cases} 1 & \text{if } c_t \in \text{stimulus } i \\ -1 & \text{otherwise} \end{cases}$$

This constraint can be incorporated in the EM framework by using the target symbol as an alternative latent variable to our model. This reduces the number of latent variables to only one per trial and therefore simplifies the calculation of the posterior distribution $p(c_t|\boldsymbol{x}_t, \boldsymbol{\Theta})$ in the E-step. In addition, the desired symbol can be easily inferred from this distribution by selecting the symbol with the highest likelihood.

Secondly, the number of target and non-target stimuli recorded in each trial is determined by the stimulus presentation paradigm through the iteration length n and the frequency of target stimuli r, as explained in Chapter 1. Consequently, the proportion π_k of samples in the target and non-target class is fixed:

$$\pi_+ = \frac{r}{n}$$
$$\pi_- = \frac{n-r}{n}$$

The knowledge of these parameter values forces the GMM to cluster the samples in two groups with the correct size. This in turn provides us the possibility to label the samples in the smaller group as target responses and the ones in the larger group as non-target responses.

A unified probabilistic model

The pseudo-generative model of ERP response signals proposed by Kindermans et al. (2012) is based on three assumptions. First of all, it is assumed that an ERP response $\boldsymbol{x}_{t,i}$ can be projected into a single dimension with a weight vector $\boldsymbol{w}$ in which this projection $\boldsymbol{w}^T\boldsymbol{x}_{t,i}$ is

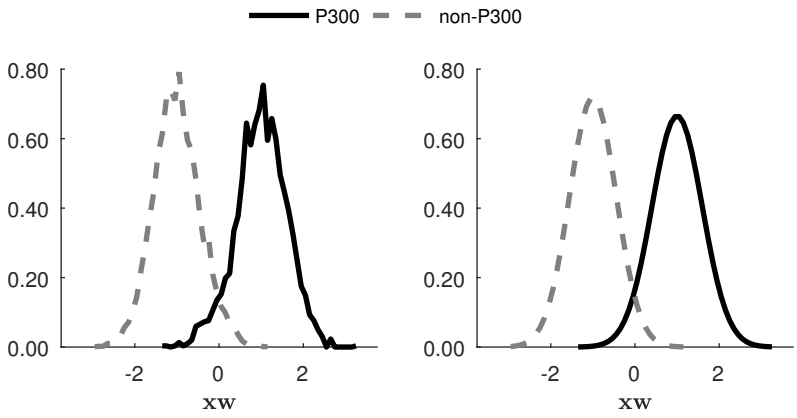

Figure 2.6: The projection of ERP response signals $\boldsymbol{x}$ into one dimension is normally distributed. The left figure shows the histogram of the projection of $\boldsymbol{x}$ on the direction of a weight vector $\boldsymbol{w}$ trained with LDA. The right figure shows two normal distributions fitted to these histograms.

normally distributed[1] with a class-conditional mean $y_{t,i}$ and shared precision β:

$$p(\boldsymbol{x}_{t,i}|c_t, \boldsymbol{w}, \beta) = \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_{t,i}|y_{t,i}(c_t), \beta^{-1})$$

This assumption is less strict than the one that is used in LDA and confirmed empirically by Blankertz et al. (2011). Namely, the normal distribution of the ERP features implies that their projection is also normally distributed. Figure 2.6 illustrates the histogram of a projection of ERP responses obtained with LDA. For each class, the histogram closely follows a normal distribution function, as shown in the right panel.

A second assumption is that all symbols are assumed to occur with the same probability as desired symbol c_t in trial t. With C symbols in the spelling grid we get:

$$p(c_t) = \frac{1}{C}$$

[1]Note that this is not a true distribution on the sample $\boldsymbol{x}$. For this reason the model is *pseudo-generative*.

This prior distribution on the desired symbol can be improved, for example by incorporating information from language models (Kindermans et al., 2014b). In that case, the prior distribution is conditioned on the desired symbols c_{t-1}, c_{t-2}, etc. from previous trials.

Thirdly, less complex models are assumed more likely to accurately describe the ERP data. This is the same assumption as the one used for regularisation of the linear regression model. A normal distribution with zero mean and isotropic covariance matrix is imposed on the weight vector $\boldsymbol{w}$:

$$p(\boldsymbol{w}) = \mathcal{N}(\mathbf{0}, \alpha^{-1}\boldsymbol{I})$$

The complete probabilistic model is summarised as follows:

$$\begin{aligned} p(c_t) &= \frac{1}{C} \\ p(\boldsymbol{x}_{t,i}|c_t, \boldsymbol{w}, \beta) &= \mathcal{N}(\boldsymbol{w}^T\boldsymbol{x}_{t,i}|y_{t,i}(c_t), \beta^{-1}) \\ y_{t,i}(c_t) &= \begin{cases} 1 & \text{if } c_t \in \text{stimulus } i \\ -1 & \text{otherwise} \end{cases} \\ p(\boldsymbol{w}) &= \mathcal{N}(\mathbf{0}, \alpha^{-1}\boldsymbol{I}) \end{aligned}$$

It is a GMM on the projection of the samples. The direct optimisation of the weight vector $\boldsymbol{w}$ avoids the estimation of the covariance matrix and as such makes this method computationally more feasible to be applied online. In the E-step, the current parameter estimates are used to find the posterior distribution of the latent variables:

$$p(c_t|\boldsymbol{X}_t, \boldsymbol{w}, \beta) = \frac{p(c_t)p(\boldsymbol{X}_t|c_t, \boldsymbol{w}, \beta)}{\sum\limits_{c_t} p(c_t)p(\boldsymbol{X}_t|c_t, \boldsymbol{w}, \beta)}$$

In the M-step the expected value of the log-likelihood is maximised with respect to $\boldsymbol{w}$ and β:

$$\boldsymbol{w}, \beta = \arg\max_{\boldsymbol{w},\beta} \sum_{\boldsymbol{c}} p(\boldsymbol{c}|\boldsymbol{X}, \boldsymbol{w}, \beta) \log p(\boldsymbol{X}, \boldsymbol{c}|\boldsymbol{w}, \beta) + \log p(\boldsymbol{w}|\alpha)$$

To obtain the update equations for these parameters, the gradient is computed and set equal to zero. This yields the following solution

(Kindermans et al., 2012):

$$\hat{\beta}^{-1} = \left\langle \sum_{c_t} p(c_t|\boldsymbol{X}, \boldsymbol{w}, \beta) \left(\boldsymbol{w}^T \boldsymbol{x}_{t,i} - y_{t,i}(c_t) \right)^2 \right\rangle_{t,i}$$

$$\hat{\boldsymbol{w}} = \sum_{\boldsymbol{c}} p(\boldsymbol{c}|\boldsymbol{X}, \boldsymbol{w}, \beta) \left(\boldsymbol{X}^T \boldsymbol{X} + \frac{\alpha}{\beta} \boldsymbol{I} \right)^{-1} \boldsymbol{X}^T \boldsymbol{y}(\boldsymbol{c})$$

with $\boldsymbol{y}(\boldsymbol{c})$ the labels that would be given to the stimulus responses when the target symbols are known. The hyper-parameter α is found by direct optimisation of the likelihood:

$$\hat{\alpha} = \frac{D}{\boldsymbol{w}^T \boldsymbol{w}}$$

For details about the derivation of the formulas we refer to the work by Kindermans (2014).

The update equation for the weight vector $\boldsymbol{w}$ is similar to the LSR solution. It is actually a weighted sum of regularised LSR classifiers, one for each possible value of the latent variables. The weight given to each LSR classifier is the probability of this latent variable value, given the observed data and the current estimates of parameters. This should not come as a surprise. The model is built on the same assumption as LSR (normally distributed projections of the data) and maximises the likelihood of the model in the same way as ridge regression.

Once the parameters $\boldsymbol{w}$, β and α are determined, the desired symbol can be inferred for each trial t by applying Bayes' rule:

$$p(c_t|\boldsymbol{X}_t, \boldsymbol{w}, \beta) = \frac{p(c_t) p(\boldsymbol{X}_t|c_t, \boldsymbol{w}, \beta)}{\sum_{c_t} p(c_t) p(\boldsymbol{X}_t|c_t, \boldsymbol{w}, \beta)}$$

The most likely symbol is then selected as target and the corresponding probability can be interpreted as the confidence with which this selection is made. This level of confidence can for example be used in dynamic stopping to determine whether enough data is recorded to accurately infer the desired symbol (Kindermans et al., 2014b).

The performance of this self-learning decoder depends strongly on the random initialisation of its parameters (Kindermans et al.,

2012). To counter this dependency, Kindermans et al. (2012) proposed the following trick. Multiple pairs of classifiers are initialised and updated in parallel. Each pair contains one classifier initialised with a weight vector $\boldsymbol{w}$, randomly drawn form the standard multivariate normal distribution, and another one initialised with $-\boldsymbol{w}$. At the end of each trial, the classifier with the highest data likelihood is considered the best one and subsequently used to predict the target symbol. Furthermore the classifier pairs are updated. Per pair, the weight vector $\boldsymbol{w}$ with the highest data likelihood is selected as the best one. The other classifier in the pair is re-initialised with $-\boldsymbol{w}$. This artificially increases the number of initialisations that is used and as such increases the opportunity of finding a good initialisation.

2.4 Conclusion

In this chapter I introduced machine learning and the related concepts of generalisation, over-fitting, regularisation and cross-validation. Linear regression and linear discriminant analysis were described as supervised machine learning methods to solve regression and classification problems. In addition, the Gaussian mixture model was explained as the unsupervised counterpart of LDA and I demonstrated how the expectation maximisation algorithm is used to train this model. Finally, I showed how the constraints imposed by the application in an ERP-based BCI are incorporated in the EM framework to obtain the method by Kindermans et al. (2012) for decoding ERPs when no labelled calibration data is available. This was the first true calibrationless decoding method for ERP-based BCI. In the next chapter I will evaluate this method for different settings of the ERP application and as such reveal the interaction between the application and calibrationless decoder. For that purpose I will design a tunable stimulus presentation paradigm. The new paradigm will then be used in the subsequent chapters to create a symbiotic ERP-based BCI in which the application will be tuned to empower the machine learning decoder.

3
A tunable stimulus presentation paradigm

In Chapter 1 we introduced the different components of a BCI: (1) the application that is presented to the user, (2) the decoder responsible for converting the user's brain activity to control commands and (3) the user (further on called subject) him/herself. In addition, we explained that, in order to make BCIs usable on a daily basis, these systems are required to be efficient, effective, reliable and easy to use. We illustrated that traditional BCI systems suffer from two major issues. First of all, a trade-off has to be made between effectiveness and efficiency. Second, the calibration of the decoder requires the recording of a labelled dataset which is expensive and/or tedious for the user.

In this and the next two chapters we will focus on ERP-based BCIs that translate the user's intention to a control command. The original and most well-known ERP-BCI is the ERP speller, introduced in Section 1.2.2. It will be used as a case study to evaluate our novel methods. Since their introduction by Farwell and Donchin (1988), a lot of effort has been put into improving this category of BCIs. First of all, various stimulus presentation paradigms have been proposed that obtain different levels of speed and accuracy (Guan et al., 2004; Allison and Pineda, 2006; Townsend et al., 2010; Jin et al., 2011; Townsend et al., 2012). By selecting a specific paradigm, we can make the trade-off between efficiency and effectiveness. Secondly, several methods have been developed to reduce or even avoid the time spent in calibration. For instance, by recycling data from previous sessions or users (Krauledat et al., 2008; Fazli et al., 2009;

Lu et al., 2009; Kindermans et al., 2014b; Wronkiewicz et al., 2015) or by using a weakly calibrated decoder and adapting to the unlabelled data recorded during use (Shenoy et al., 2006; Dähne et al., 2011; Vidaurre et al., 2011b,a). The EM-based decoding method from Kindermans et al. (2012), explained in the previous chapter, was the first truly calibrationless decoding method for ERP-based BCI.

Each improvement that has been proposed optimises either the application or the decoder individually. This stands in stark contrast to the importance of the interaction between these components. The application determines the quantity of the data that is provided to the decoder through the number of stimuli presented to the user. Also, the quality (SNR) of the recorded data is influenced strongly by the relative frequency of target stimuli (Gonsalvez and Polich, 2002; McFarland et al., 2011) as well as the stimulus fashion (Gibert et al., 2008; Salvaris and Sepulveda, 2009; Takano et al., 2009; Kaufmann et al., 2011; Jin et al., 2012). This interaction is even more significant for calibrationless decoders that have to learn from the data recorded during actual use. We will develop the first symbiotic design approach for ERP-based BCIs, in which the different components are co-adapted to improve the overall BCI performance.

In this chapter, we will propose a new stimulus presentation paradigm that provides us flexibility by tuning the number of presented stimuli and the relative frequency of target stimuli. It will allow us to design an experiment that compares the performance of the calibrationless decoder for different application settings. The results will demonstrate the strong interaction between these two entities. In the next chapters, we will use the new paradigm to empower the self-learning decoder.

The next section will explain the design of stimulus presentation paradigms technically and describe how the paradigm influences the decoding performance. Next, we will introduce the new stimulus presentation paradigm and compare it to the basic row-column and checkerboard paradigms in an online experiment with the visual ERP speller. Finally, extensive offline analyses of the results will reveal which factors influence predominantly the performance of the self-learning decoder.

3.1 Difficulties in stimulus presentation paradigm design

The stimulus presentation paradigm determines for each stimulus the group of symbols that is simultaneously highlighted in the grid. The result can be represented by an *allocation matrix* in which each row represents a symbol in the grid and each column a stimulus in the sequence. A cell contains the value '1' when the symbol indicated by the row is highlighted during the stimulus indicated by the column and a value '0' otherwise. Two toy examples for a 3x3 grid containing nine symbols ('A' to 'I') are presented in Figure 3.1.

	S1	S2	S3	S4	S5	S6	S7	S8	S9
A	1	0	0	1	0	1	0	0	0
B	0	1	0	0	1	0	0	1	0
C	0	0	1	0	1	0	1	0	0
D	1	0	0	1	0	0	0	1	0
E	0	1	0	0	0	1	0	1	0
F	1	0	0	1	0	0	0	0	1
G	0	1	0	0	1	0	1	0	0
H	0	0	1	0	0	0	1	0	1
I	0	0	1	0	0	1	0	0	1

(a)

	S1	S2	S3	S4	S5
A	1	0	1	0	0
B	0	1	0	0	1
C	0	0	1	1	0
D	1	0	0	0	1
E	0	1	1	0	0
F	0	0	1	0	1
G	1	1	0	0	0
H	0	0	0	1	1
I	1	0	0	1	0

(b)

Figure 3.1: Example of an allocation matrix in the case of a 3×3 symbol grid for a random paradigm (a) with $r = 3$ and $n = 9$ and (b) with $r = 2$ and $n = 5$. The mean Hamming distance between two rows reduces from 4.5 for $n = 9$ to 2.67 for $n = 5$

The decoder has the task to infer the desired symbol from the classification of stimulus responses and the knowledge of the allocation

matrix. To make this possible, every symbol must be highlighted in a unique set of stimuli. This means that there are no identical rows in the allocation matrix. In that case, the sequence is said to be *decodable*.

The paradigm determines how many stimuli are presented per symbol selection and as such dictates the speed of spelling. As explained in Chapter 1, the sequence of visual stimuli is repeated in several iterations to increase the SNR by averaging the recorded responses. The speed of spelling is regulated by the number of iterations and the iteration length n. The number of times each symbol in the grid is highlighted per iteration will be denoted by r, the number of symbols in the grid by M

The spelling speed can be increased by decreasing either the number of iterations or the iteration length. The optimal number of iterations can be chosen automatically with the dynamic stopping method (Schreuder et al., 2011, 2013; Kindermans et al., 2014b). With this technique, the repeated stimulus sequences are stopped once the classifier is confident enough to accurately select the target symbol.

In contrast, the iteration length n needs to be chosen by the experimenter. A shorter iteration length has several implications for the decoder. To begin with, it reduces the number of responses that are available to accurately infer the desired symbol. Secondly, it makes the target stimulus appear more frequently, which results in a weaker target response (Gonsalvez and Polich, 2002; McFarland et al., 2011). Thirdly, it is more challenging to infer the target symbol because the set of stimuli in which symbols are highlighted will overlap more (the mean Hamming distance between the rows of the allocation matrix is reduced, as illustrated in Figure 3.1). Additionally, more symbols are highlighted simultaneously, which potentially causes *adjacency distraction*. In this phenomenon, the user is distracted by the flash of a symbol next to the target symbol (Fazel-Rezai, 2007). This causes the generation of a target response on a non-target stimulus. If this symbol is subsequently highlighted simultaneously with the true target, the neighbouring symbol might be selected as target. Finally, *double flash errors* occur when the target symbol is highlighted twice in quick succession. This causes the response generated on the second flash to overlap with the first one and have a lower amplitude (Martens et al.,

2009; Gonsalvez and Polich, 2002). Overall, we can conclude that speeding up the spelling process decreases the decoder's accuracy in selecting the target symbol. A trade-off between speed and accuracy is inevitable.

Several stimulus presentation paradigms have been designed as an alternative to the original row-column paradigm (RC) (Guan et al., 2004; Allison and Pineda, 2006; Townsend et al., 2010; Jin et al., 2011; Townsend et al., 2012). As an example, a 6×6 version of the checkerboard paradigm (CB) by Townsend et al. (2010) was discussed in Section 1.3. This paradigm avoids double flashes by preventing symbols to be highlighted in rapid succession. Adjacency distraction on the target selection is reduced by preventing symbols to be highlighted simultaneously with a neighbouring symbol in the grid. By avoiding these major causes of spelling errors, CB is capable of achieving a higher spelling accuracy compared to RC. However, this comes at the cost of a longer iteration length ($n = 18$ compared to $n = 12$ for RC). We question if the iteration length can be reduced while still avoiding the spelling errors.

The decodability requirement imposes a lower limit on the iteration length n. Highlighting each one of the M symbols in a unique combination of r out of n stimuli is only possible as long as there are enough unique combinations available, i.e. when $\binom{n}{r} \geq M$. For the original 6×6 grid of symbols and $r = 2$ the iteration length has a minimal value of $n = 9$. So, in theory, the iteration length of the CB paradigm can be halved. For $r = 3$ the iteration length can be even reduced to a minimum of $n = 8$.

3.2 The switching paradigm

In this section we propose a paradigm in which the parameters can be chosen freely. In this way, the paradigm can cover the entire spectrum of state of the art paradigms. In contrast to CB and other existing paradigms, the most common causes of symbol selection errors are not avoided by setting strict rules but by optimising the highlighting scheme as much as possible. We maximise the number of stimuli between two consecutive highlights of the same symbol as well as

the spread of simultaneously highlighted symbols over the grid. The alleviation of strict rules will allow us to set the iteration length to its theoretical minimum. In this way we speed up the spelling process as much as possible.

3.2.1 Sequence generation algorithm

The paradigm selects pseudo-randomly a decodable sequence of the desired iteration length n and frequency r of target stimuli. Subsequently, this sequence is optimised to reduce the aforementioned potential causes of error. This is achieved by swapping highlighted symbols between the stimuli, hence the name *switching paradigm* (SP). In the next sections, the initialisation and optimisation procedures are discussed.

Pseudo-random initialisation

Any decodable stimulus sequence can be used as initialisation (e.g. a row-column sequence). Algorithm 3.1 proposes a method to construct a decodable sequence with the desired iteration length n. The initialisation algorithm takes the following parameters as input:

- M: the number of symbols in the speller grid
- e: the desired number of iterations in the total sequence.
- n: the iteration length
- r: the number of times each symbol is highlighted per iteration

It selects for each symbol in the grid a unique set of stimuli in which this symbol will be highlighted. The number of symbols highlighted per stimulus is $\frac{r*M}{n}$. If this is not an integer number, stimuli highlighting $\lceil \frac{r*M}{n} \rceil$ as well as $\lfloor \frac{r*M}{n} \rfloor$ symbols are found in the constructed sequence. In the example case where $M = 36$, $r = 2$ and $n = 10$, every iteration contains eight stimuli highlighting seven symbols and two highlighting eight symbols.

Require: $e > 0$,$n > 0$,$r > 0$,$\binom{n}{r} \geq M$

Maximum # symbols highlighted in a stimulus:

1: $Maxd = \lceil \frac{M*r}{n} \rceil$

Minimum # symbols highlighted in a stimulus:

2: $Mind = \lfloor \frac{M*r}{n} \rfloor$

Set of all possible combinations of r out of n stimuli:

3: $C_r^n \leftarrow \{c \in \mathcal{P}(\{s_1, ..., s_n\}) : |c| = r\}$

Repeat the following procedure to generate each of the e iterations:
Initialise set of M randomly chosen combinations

4: $C_{sel} \leftarrow \{c_m \in C_r^n | m = 1...M\}$

Determine for each stimulus s_i the # of allocated symbols

5: $d(s_i) = |\{c | s_i \in c \in C_{sel}\}|$

Remove and add combinations to C_{sel} until all stimuli comply with $Maxd$ and $Mind$

6: **while** $\exists s_i : (d(s_i) > Maxd)$ OR $(d(s_i) < Mind)$ **do**
7: **for** $s_h : d(s_h) \geq Maxd$ **do**
8: **for** $s_l : d(s_l) \leq Mind$ **do**

Select random combination of $r-1$ stimuli $s \notin \{s_l, s_h\}$

9: $R \in C_{r-1}^{n \backslash \{l,h\}}$
10: **if then**$\{s_h, R\} \in C_{sel}$ AND $\{s_l, R\} \notin C_{sel}$
11: $C_{sel} \leftarrow \{C_{sel} \backslash \{s_h, R\}\} \cup \{s_l, R\}$
12: $d(s_i) = |\{c | s_i \in c \in C_{sel}\}|$
13: **break** for loops
14: **end if**
15: **end for**
16: **end for**
17: **end while**

Alg. 3.1: Initialisation: construct a pseudo-random decodable sequence with e iterations of n stimuli in which each of the M symbols are highlighted r times.

Optimisation

The two main causes of spelling errors discussed in section 3.1 are now avoided as much as possible by gradually changing the stimuli in the initialised sequence. First, double flashes are avoided by preventing a symbol to be highlighted twice in rapid succession. An example is shown in Figure 3.2(a) where the double flash of the symbol 'O' between the first and second stimulus is removed by swapping the symbols 'O' and 'I' between the second and third stimulus. In the resulting sequence in Figure 3.2(b) the double flash is expelled.

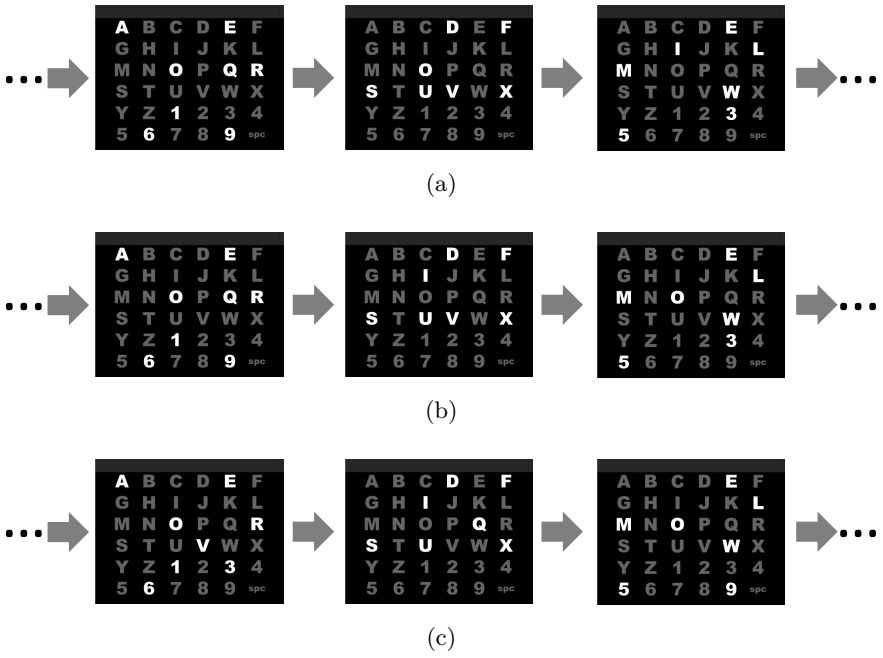

Figure 3.2: Example of a double flash and adjacency distraction being avoided by swapping symbols between stimuli. (a) Part of an initial stimulus sequence with a double flash of the symbol 'O' in the first and second stimulus. (b) Same part of the sequence after swapping the symbols 'O' and 'I' between the second and third stimulus, removing the double flash. (c) Now the neighbouring symbols highlighted simultaneously are removed to reduce potential adjacency distraction errors. This is the result after swapping symbols 'Q' and 'V' between the first and second stimulus and '9' and '3' between the first and third stimulus.

Second, the impact of adjacency distraction on the selection of the target symbol can also be reduced by tuning the highlighting scheme. Symbols highlighted on a grid position closer to the target are more likely to distract the subject. Consequently, adjacency distraction errors can be reduced by preventing symbols to be highlighted simultaneously with neighbours in the grid (Townsend et al., 2010; Frye et al., 2011). Again, this is done by swapping symbols between stimuli. An example is given in Figure 3.2(c). Obviously, swaps are only considered if they do not harm the decodability of the sequence.

The optimisation algorithm works iteratively in two phases. The first phase tries to remove the most severe causes of errors: double flashes and adjacent symbols highlighted simultaneously. The second phase optimises the sequence further: the time between two intensifications of the same symbol and the spread of simultaneously highlighted symbols over the grid are maximised. As each optimisation in one phase creates new opportunities in the other, these phases are executed alternately until no more optimising swaps are found. We allow a maximum of five alternate executions to prevent the algorithm from getting stuck in a loop, constantly swapping the same symbols.

For each iteration in the sequence, every possible pair of stimuli is examined in random order. For each pair of stimuli, every possible swap of highlighted symbols between these stimuli is examined in random order. Two criteria are used to verify if the swap is indeed optimising the sequence. In the first phase these criteria are:

1. For both symbols: does the swap remove double flashes or consecutive intensifications of the symbol with only one stimulus in between?

2. For both stimuli: does the swap reduce the number of direct horizontal, vertical or diagonal neighbouring symbols in the grid that are simultaneously highlighted?

In the second phase these criteria are:

1. For both symbols: does the inter-stimulus interval increase?

2. For both stimuli: does the distance in the grid of the swapped symbol to its closest neighbour increase (i.e. does the spread of highlighted symbols in the grid increase for both stimuli)?

If at least one of these questions has a strictly positive answer and none of the others is strictly negative, the swap is executed. The procedure is repeated until no more optimising swaps are found for the current phase. The algorithm then moves to the next phase. Figure 3.2(c) demonstrates how the obtained sequence of stimuli appears to the user.

3.2.2 Evaluation of the optimisation algorithm

To investigate the performance of the optimisation algorithm, 500 sequences of 10 iterations are generated. With $r = 2$ the iteration length can be reduced to $n = 9$. To give the optimisation algorithm a minimum amount of play we choose $n = 10$. The sequences are initialised with Algorithm 3.1 and compared to the result after optimisation. The number of double flashes seen by the subject in a sequence of 10 iterations is shown in the histograms in Figure 3.3 (the histograms for the basic RC and the 6×6 grid version of the CB paradigm are given for comparison). On average, 2.39 (STD = 1.36) double flashes are noticed in the initial sequence. When optimising the highlighting scheme this reduces by 54 % to only 1.09 (STD = 0.98) double flashes per sequence of 10×10 stimuli. As mentioned before, the CB paradigm omits double flashes completely.

The probability of adjacency distraction is related to the number of times the target symbol is highlighted simultaneously with at least one of its direct (horizontal or vertical) neighbours in the grid. In the initial sequence this happens on average for 49.63 % of the symbol highlights. After optimisation, only 8.61 % of the symbol highlights happen together with a neighbour. Figure 3.3 gives a histogram of the Manhattan distance[1] in the grid from the target to every other element simultaneously highlighted. The optimisation procedure clearly distributes the highlights more evenly over the grid.

In summary, the optimisation algorithm reduces the double flashes and the risk of adjacency distraction. The new paradigm is better

[1] The Manhattan distance between two points A and B in a grid is the number of horizontal and/or vertical steps that have to be taken along the grid lines to reach B from A. Direct horizontal or vertical neighbours have a Manhattan distance one. Diagonal neighbours have a Manhattan distance two.

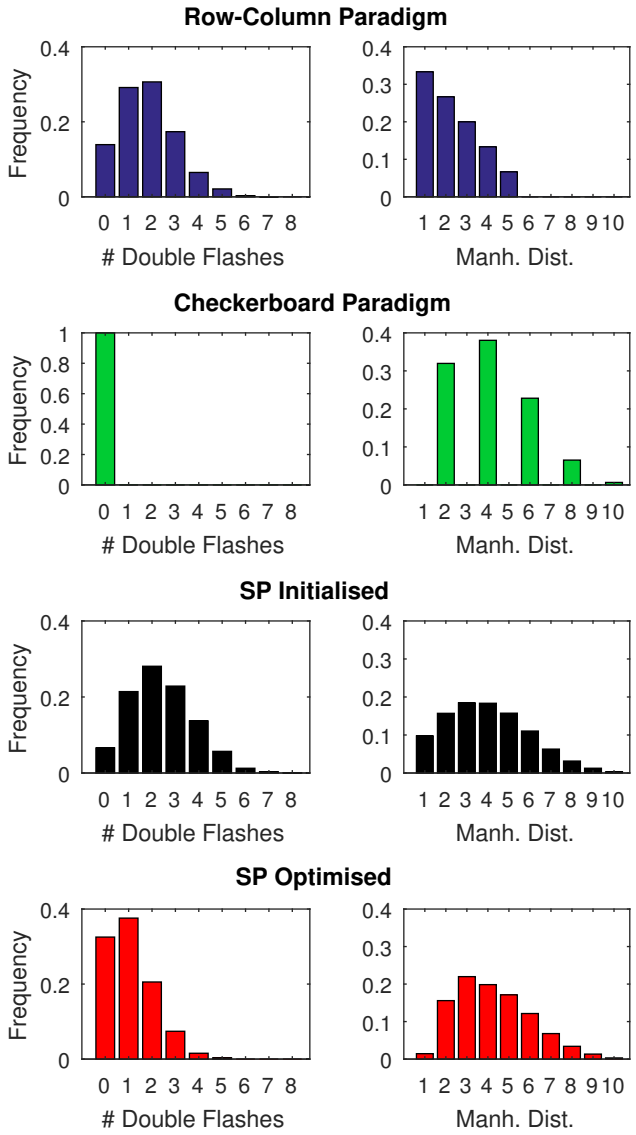

Figure 3.3: Analysis of the sequences of stimuli generated by the row-column paradigm, the checkerboard paradigm, the random initialisation algorithm of the switching paradigm ($n = 10$) and the corresponding sequences after applying the optimisation algorithm. Left: the normalised histogram of the number of double flashes seen in a sequence of 10 iterations. Right: normalised histogram of the Manhattan distance in the grid from a symbol to every other symbol simultaneously highlighted in a sequence of 10 iterations.

than RC. The CB paradigm is still better in terms of the number of double flashes. We have seen that this comes at the cost of a higher iteration length in CB. In the next section we compare the paradigms in a spelling experiment.

3.3 Online evaluation with unsupervised decoding

We compare the spelling performance obtained with the proposed switching paradigm ($n = 10$) to the basic RC and CB paradigm in an online experiment. The unsupervised decoding method by Kindermans et al. (2012), explained in Section 2.3.2, is used to classify the recorded stimulus responses and infer the target symbol. The experiment presented here was the first extensive online evaluation of this unsupervised decoding method in the visual ERP speller. It allows us to examine the influence of different application settings on the unsupervised learning and classification. In the following subsections we present the details of this experiment and discuss the results.

3.3.1 Experimental set-up

Participants

The study was in accordance with the principles embodied in the Declaration of Helsinki and was approved by the Ethical Committee of the Faculty of Psychology and Educational Sciences, Ghent University. Each participant gave informed consent. 24 healthy participants (9 male, 15 female) with an average age of 24.7 years (STD = 6.5) applied for the experiments via an online recruiting system of the faculty. They were given a fee independent of the experiment results. Most of them were undergraduate students of the faculty and as such had some prior knowledge about the concept of recording brain signals with EEG. Two subjects were left-handed. Two had prior experience with the ERP speller (they took part in the pilot tests a couple of months before the actual experiments). All of them had normal or corrected-to-normal visual acuity.

Experiment design

The experiment started with a thorough explanation on how the visual ERP speller works and how it can be used to help people with the locked-in syndrome. In this way we tried to improve the subject's motivation during this three hour experiment. Subjects were seated calmly and comfortably in a chair, facing a 19 inch monitor at a distance of approximately 60 cm. On this screen, the 6×6 grid of symbols was displayed. After the EEG cap was put on, the quality of the EEG signals was verified visually by the researcher. The subjects were shown their own EEG and were asked to blink and chew to become aware of the fact that these actions cause artefacts in the signals and as such can prevent the speller from performing well.

A sentence of 44 characters was spelled three times, making use of the three different paradigms. Subjects were asked to count silently each time their chosen target symbol was highlighted. To spell a symbol, a sequence of 10 iterations of the paradigm under examination was shown. The stimulus and inter-stimulus interval were both set to 100 ms (stimulus onset asynchrony 200 ms). All three spelling sessions took place on the same day, with breaks of five minutes in between.

The order in which the RC, CB and SP paradigms were used was randomised between subjects to average out the effect of fatigue and habituation to the previous paradigm on the speller performance. The concept of the gradually increasing performance of the unsupervised decoder was explained in order not to discourage the subject during the early spelling phase in which most symbols are spelled incorrectly by the self-learning decoder. Figure 3.4 illustrates how the stimuli of the three paradigms appear to the user.

Data acquisition and processing

EEG was recorded at 250 Hz sampling rate with the Easycap (Brain Products, GmbH, Munich, Germany) EEG cap and QuickAmp EEG amplifier. 33 Ag/AgCl active electrodes were used: 31 on positions according to the 10-20 system[2] to capture the EEG, one ground electrode placed on the forehead and one EOG electrode beneath the right

[2] F_{p2}, F_1, F_2, F_5, F_6, F_9, F_{10}, FC_z, FC_3, FC_4, FT_7, FT_8, C_z, C_1, C_2, C_5, C_6, CP_z, CP_3, CP_4, TP_7, TP_8, P_1, P_2, P_5, P_6, P_9, P_{10}, PO_z, O_1, O_2

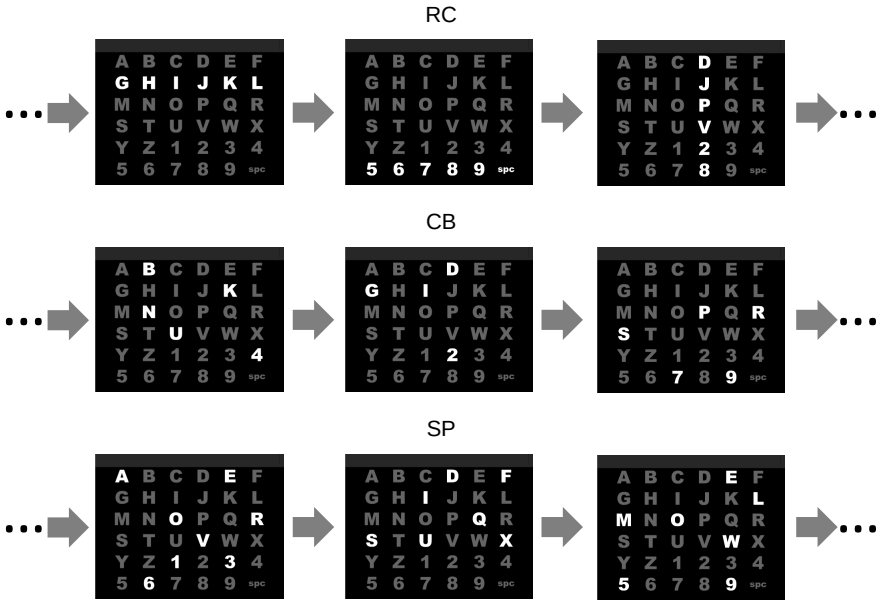

Figure 3.4: Example of stimuli as they appear to the subject: row-column paradigm (RC), checkerboard paradigm (CB) and the novel switching paradigm (SP) with $n = 10$.

eye to capture the exact moment of eye blinking (not used for EOG correction in this study).

The processing of the recorded EEG response signals, the classification of these responses as target or non-target and the resulting selection of the target symbol were done according to the original work by Kindermans et al. (2012). A common average reference filter was applied, followed by a bandpass filter with cutoff frequencies 0.5 Hz and 15 Hz. Each EEG channel was normalised to zero mean and unit variance. Dimensionality reduction retained 10 samples per stimulus response, centered around the expected time step of the P300 wave and uniformly distributed over the range between 140 ms and 460 ms. Finally, a bias term was added as an extra feature to every stimulus response. Tuning the parameters of the decoder was done with the unsupervised method from Kindermans et al. (2012).

Experiments were conducted with our own implementation of the ERP speller in the BCI2000 research platform (Schalk et al., 2004), making use of the BCPy2000 framework (Hill et al., 2008). The appli-

cation was verified extensively before actual use in the experiments.

3.3.2 Results and discussion

We will detail the results from the online study, including a statistical analysis that illustrates the dependency of the decoder performance on the paradigm in use. The scatter plots in Figure 3.5 give a pairwise comparison of the three paradigms for the four performance measures described in Section 1.2.3. Special attention should be given to the correct symbols per minute (CSM) as this represents how fast the user can spell a correct sentence. The result of the extensive statistical analysis is given in the next sections. The details of the analysis can be found in Appendix A. Measurement values will be written as 'mean ± standard deviation', unless stated otherwise.

Spelling accuracy

The top row of scatter plots in Figure 3.5 demonstrates the spelling accuracy obtained with the three paradigms. With the RC paradigm, on average 86.65 % ± 8.69 of the symbols are spelled correctly, whereas SP yields 77.75 % ± 12.79 and the CB paradigm 86.46 % ± 10.90. The basic paradigms are equally accurate but the RC paradigm achieves this accuracy in only $\frac{2}{3}\left(=\frac{n_{RC}}{n_{CB}}\right)$ of the time needed by the CB paradigm to spell a symbol. SP underperforms but needs even less time to spell a symbol. The results are thus overly pessimistic as learning with shorter stimulus sequences is harder due to the lack of recorded data. We will elaborate on this in Section 3.4. A one-way repeated measures ANOVA (Field, 2009) shows that the difference in accuracy is statistically significant, $F(2, 46) = 11.101$, $p < 0.0005$, partial $\eta^2 = 0.326$. Pairwise comparison of the results confirms the conjecture that the accuracy differs between SP and RC ($p = 0.002$) and between SP and CB ($p = 0.001$) but not between RC and CB ($p = 1.000$).

Respelling accuracy

There is less contrast in the respelling accuracy values: 96.40 % ± 7.67 for RC, 95.55 % ± 8.93 for SP and 97.35 % ± 6.20 for CB. Recall that the respelling accuracy defines the quality of the actual text being

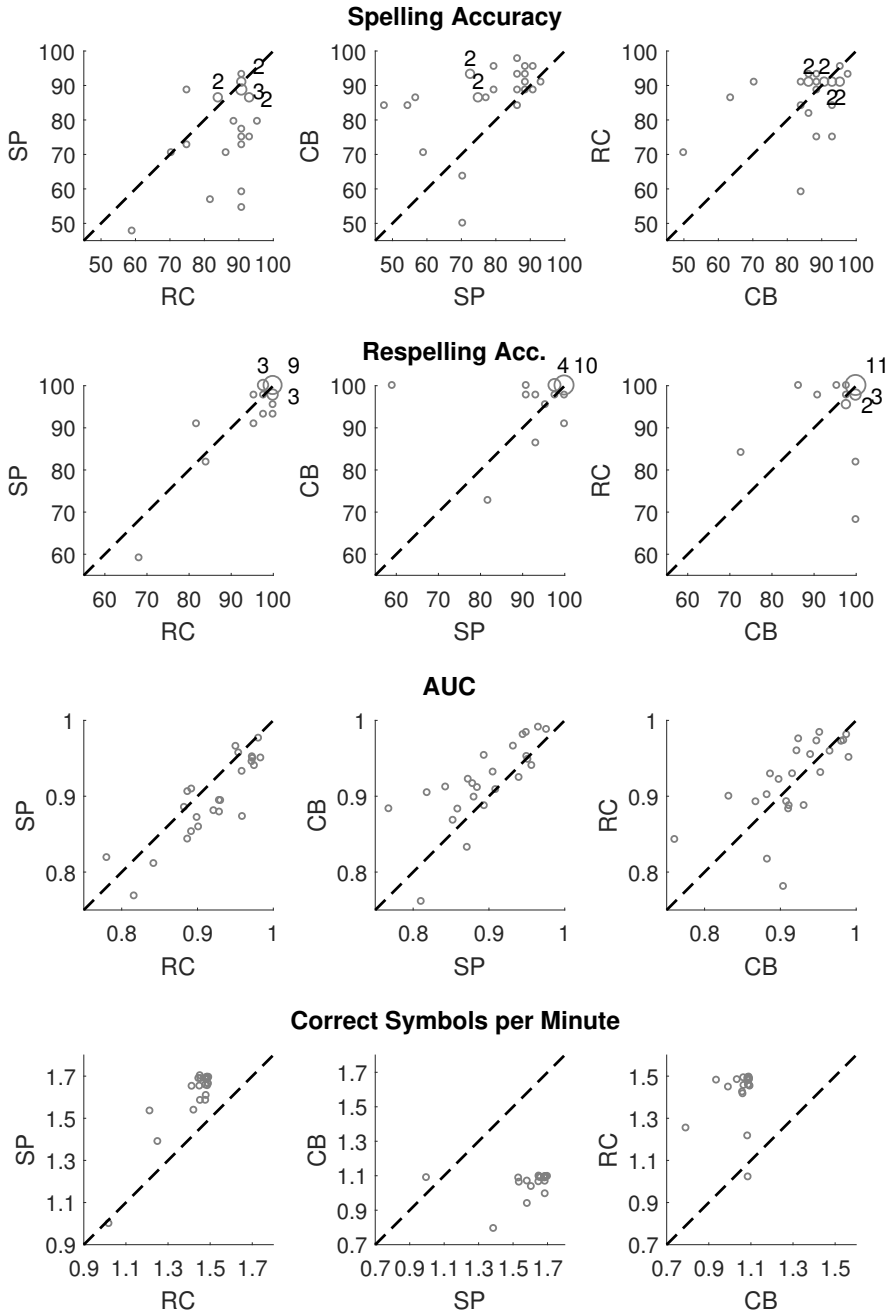

Figure 3.5: Results of the online experiment. Each dot compares the performance for two paradigms for the same subject. The dashed line is the line of equal performance for both paradigms. Overlapping dots are denoted as a bigger dot accompanied by the number of subjects achieving this result.

shown to the user. The fact that SP achieves the same level as the CB paradigm with almost half the iteration length is striking. In Section 3.4 we search for the reason for this remarkable result. A one-way repeated measures ANOVA shows that the difference in respelling accuracy is not statistically significant: $F(2, 46) = 1.156$, $p = 0.324$, partial $\eta^2 = 0.048$. The position of the tested paradigm, in the order of three, is found to have a significant influence on the achieved respelling accuracy. This can be attributed to the increasing fatigue of the subject during the course of the experiment. The experiments with the different paradigms were performed with a small break of 5 minutes in between. Consequently, we expect that subjects were more tired at the start of the second and third experiment, resulting in a lower spelling performance. This influence is found to be the same for all paradigms and as such is averaged as every order was used an equal number of times.

The evolution of the respelling accuracy during the spelling session is shown in Figure 3.6(a). As the self-learning decoder starts with zero knowledge, the first symbols are mostly spelled incorrectly and the accuracy is very low. Once the decoder is provided with enough stimulus response examples to tune its parameters, more correct symbols are being spelled and the decoder adjusts previously spelled symbols, thereby increasing the respelling accuracy. In Figure 3.6(b) the respelling accuracy is shown as a function of the number of stimuli presented to the user. In the beginning of the spelling session the three curves lie closely together, showing that the respelling accuracy in this phase is solely determined by the number of response signals recorded. Therefore, the shorter stimulus sequences are the reason why it takes more symbols for SP to start giving correct output. This in turn explains the significantly lower spelling accuracy described in the previous section. Misspelled symbols in the initial spelling phase are being corrected, bringing the respelling accuracy to the same level as for the basic paradigms by the end of the spelling session. In summary, from the user's point of view, the accuracy increases equally fast in time, but the rate at which symbols can be spelled is higher with our new paradigm.

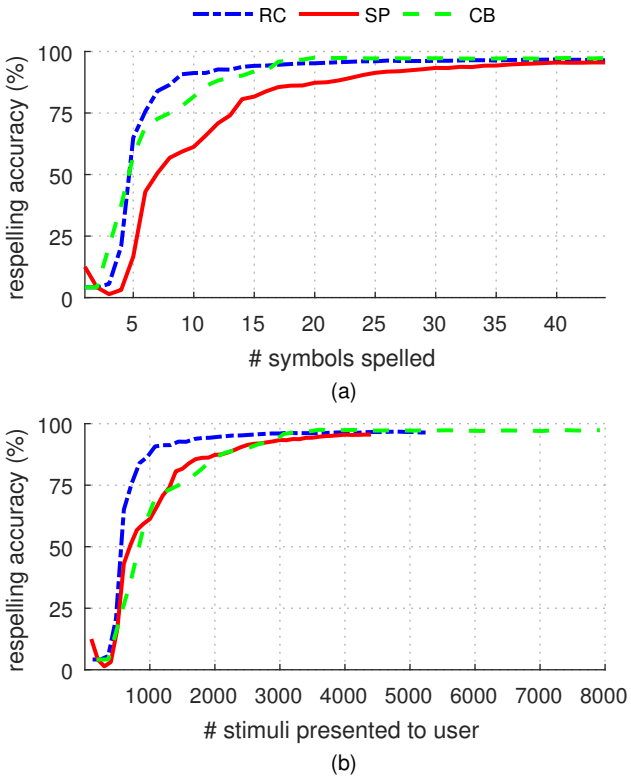

Figure 3.6: Evolution of the respelling accuracy during the spelling session for the three paradigms, averaged over the 24 subjects, as a function of (a) the number of symbols spelled and (b) the number of stimuli presented to the user.

AUC

The evolution of the AUC during the spelling session is shown in Figure 3.7. Figure 3.7(a) demonstrates for each trial the AUC on the total set of data collected up to that trial. In the very beginning of the spelling session, data is still scarce and the decoder has not yet learned enough how to accurately classify the recorded responses. The AUC obtained during this so-called *warm-up period* is low, regardless of which paradigm is used. Once the decoder has collected more data, it corrects its decision on past responses and the AUC rises quickly until saturation. Figure 3.7 shows a slight decrease in AUC after saturation. This is caused by changes in the distribution of the recorded data, e.g. due to the increasing fatigue of the user. As the decoder constantly

has to adapt to these changes, its AUC on past responses slightly decreases.

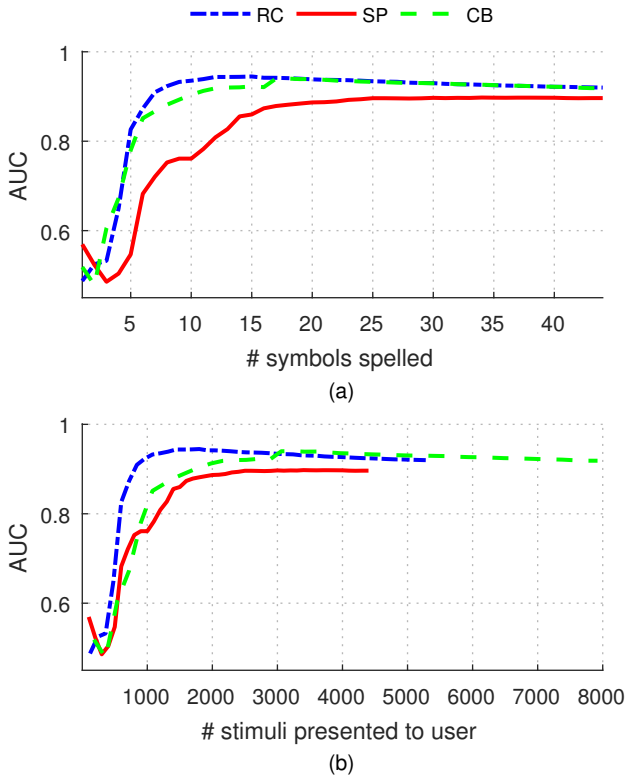

Figure 3.7: Evolution of the AUC during the spelling session for the three paradigms, averaged over the 24 subjects, as a function of (a) the number of symbols spelled and (b) the number of stimuli presented to the user.

The basic paradigms serve the self-learning decoder with more stimulus response examples per spelled symbol. This causes the AUC for these decoders to increase more rapidly compared to SP, as illustrated in Figure 3.7(a). A one-way repeated measures ANOVA is conducted on the AUC obtained at the end of the spelling session. The AUC achieved with SP is significantly lower compared to the basic paradigms, $F(2, 46) = 8.324$, $p = 0.001$, partial $\eta^2 = 0.266$. Pairwise comparison of the results confirms that the AUC differs between SP and RC ($p = 0.001$) and between SP and CB ($p = 0.003$) but not between RC and CB ($p = 1.000$). Recalling the description

of AUC in Section 1.2.3, a lower AUC signifies that a lower fraction of response signals are classified correctly as target or non-target by the decoder. However, if enough stimulus responses are recorded per symbol, the correctly classified ones can give sufficient information to make a correct symbol selection. This explains how the speller with SP can attain the same respelling accuracy as the basic paradigms while the AUC is lower.

To examine the performance of the self-learning decoder in more detail we illustrate the median, minimum and maximum AUC levels obtained over the 24 subjects in Figure 3.8. We notice a large variability in the length of the warm-up period for each paradigm. Some subjects achieve almost perfect classification within the first five symbols while others still achieve an AUC around the 0.50 chance level at 15 symbols for CB and SP. This high variability is caused by inter-subject differences, the random initialisation of the parameters and the EM algorithm that is used to tune these parameters by maximising the likelihood of its own predictions (see Section 2.3.2). With a good initialisation, the predictions are reasonably well from the beginning and the parameter values are improved. When the initialisation is unfortunate, more data is needed to learn how to make correct predictions for the specific subject.

These observations are concordant with the original work on the unsupervised decoding method (Kindermans et al., 2014a). By constantly adapting to the recorded data and correcting its own mistakes, the decoder is capable of achieving a remarkably high AUC level. We stress again that, in contrast to traditional decoding methods, not a single labelled example is used to calibrate the decoder. Nevertheless, the warm-up period is a major issue. It can be demotivating for the subject just like the calibration session in traditional decoders. Furthermore, the high variance in the performance makes the system very unreliable. In fact, the randomly initialised decoder does not have the theoretical guarantee to ever find the correct classification, even when an infinite amount of data is recorded. We will address this problem in Chapter 4.

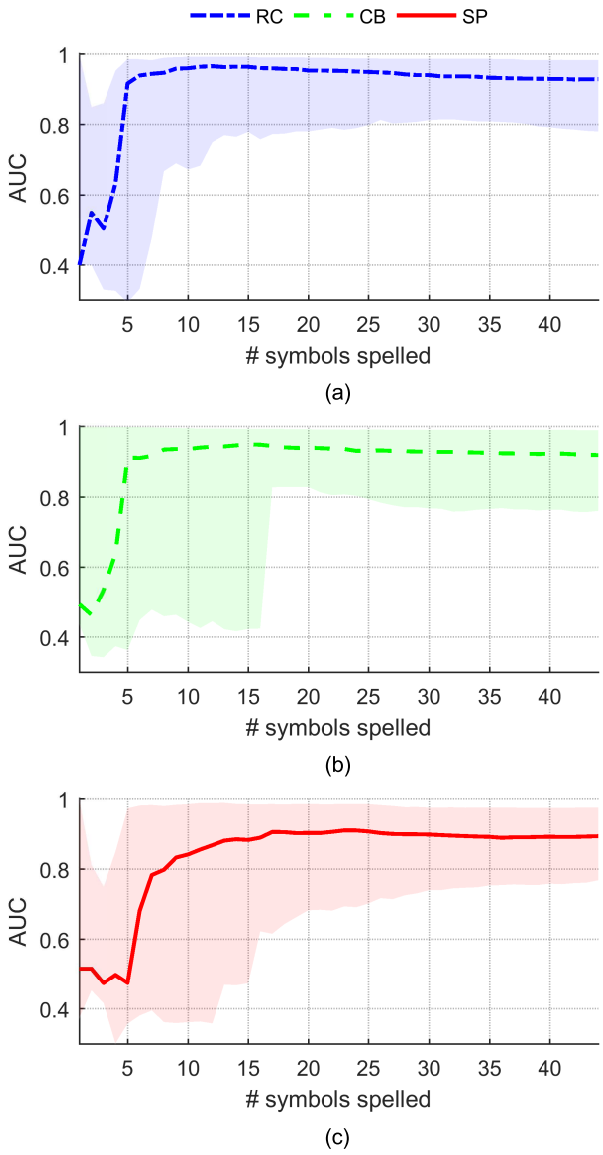

Figure 3.8: Evolution of the AUC obtained during the spelling session for the 24 subjects: median and minimum-maximum range achieved with (a) the row-column paradigm (RC), (b) the checkerboard paradigm (CB), and (c) the switching paradigm (SP).

Correct symbols per minute

As mentioned before, there is a trade-off between the spelling speed and accuracy. In Section 1.2.3, we introduced the correct symbols per minute (CSM) as a measure that includes both performance metrics and as such is more suitable to compare paradigms. It is the ratio of the total number of symbols respelled correctly at the end of the spelling session and the time that was needed to complete that session.

In every scatter plot in the last row of Figure 3.5, all-but-one dots are located at one side of the dashed line. Therefore, no statistical analysis is needed to conclude that SP outperforms on the level of correct symbols spelled per minute. This is obviously the consequence of obtaining the same respelling accuracy using only 56 % of the spelling time used by CB and 83 % of the spelling time used by RC. Figure 3.9 shows the evolution of the number of correctly spelled characters during the spelling session.

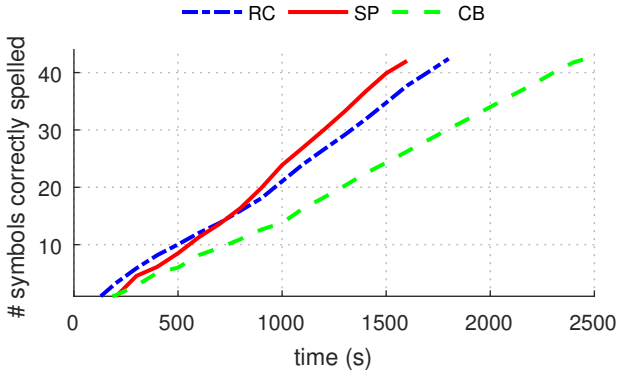

Figure 3.9: Evolution of the number of symbols spelled correctly during the spelling session for the three paradigms, averaged over the 24 subjects.

3.3.3 Summary of online evaluation

To summarise, the experimental results have shown a significantly lower spelling accuracy achieved by SP compared to the basic paradigms. This difference was found to be a consequence of the shorter stimulus sequences used to spell a symbol. A remarkable result is that, in contrast, SP achieved the same level of respelling accuracy

while spelling symbols in almost half the time needed by the CB paradigm. The proposed paradigm thus achieves the same accuracy with a higher speed of spelling. Nevertheless, the difference in the final AUC level achieved is significantly lower for SP. Consequently, the quality of the decoder depends on the paradigm in use, indicating the importance of the application-decoder interaction. This obviously raises questions. First of all we want to know why the AUC is lower for the SP paradigm. Next, as we are striving for a faster speller using fewer iterations, we question if this lower AUC leads to a lower respelling accuracy, compared to the basic paradigms, when the amount of data recorded per symbol is decreased. In the next section we address these questions.

3.4 Examining the application-decoder interaction

For the calibrationless decoder, the responses recorded during the spelling process are vital to learn how to discriminate between target and non-target responses. Consequently, the interaction with the application is even more involved for this self-learning decoder compared to traditional supervised decoding methods. In this section we examine thoroughly why some paradigms result in better learning and classification performance than others.

As discussed before, there are two ways in which the paradigm influences the decoder's capability of learning to detect a target response. First, the paradigm determines the quantity of available responses through the number of stimuli that are presented. Secondly, it also determines the SNR of these response signals, for example through the relative frequency of target stimuli. With simulations on the recorded data, we examine how quantity and quality of the data influence the learning process and which factor is most important. In this way, we hope to get a better understanding of the mechanism of interaction between the application and the self-learning decoder.

Please note that the simulations are included to analyse the effects of the signal quality, isolated from other variables that change across

paradigms. This cannot be done in online experiments due to the physiological processes underlying the EEG. For this reason we would like to stress that the simulations are not a replacement of an online study but merely an analysis tool.

3.4.1 Influence of data quantity and quality on decoder performance

First of all, we counterbalance the influence of the quantity and quality of the recorded data on the performance of the self-learning decoder. For that purpose, the following experiment is simulated. We take the sequence of stimuli shown in the SP experiment and replace the responses on the stimuli with those recorded in the CB experiment. This is demonstrated schematically in Figure 3.10(a). As the CB paradigm has 16 non-target responses recorded per iteration, only half of them are used. In this way, we use the same amount of data recorded per symbol as in the original SP experiment, but the data has the SNR from the CB experiment. The SNR from the CB experiment is expected to be higher due to the lower relative frequency of target stimuli compared to SP. The AUC obtained in the simulation is compared with both original experiments in Figure 3.11(a). The simulation is also done the other way around. Now the data from the SP experiment is used to simulate the CB experiment with iteration length $n = 18$. Because there are not enough non-target responses per symbol recorded in the SP experiment to simulate an experiment with $n = 18$, every second symbol is left out of the simulation and the SP responses recorded during this symbol are used to complement the data for the previous symbol (see Figure 3.10(b)). In Figure 3.11(b) the AUC is compared to a resimulation of the original CB and SP experiment with every second symbol left out.

We can identify three phases in the spelling session, illustrated by the vertical lines in Figure 3.11(a).

Phase 1 : When the total amount of data collected by the decoder is still low, the AUC curve follows the result from the original experiment that used the same amount of data per symbol (the SP experiment in Figure 3.11(a)). This data quantity is the

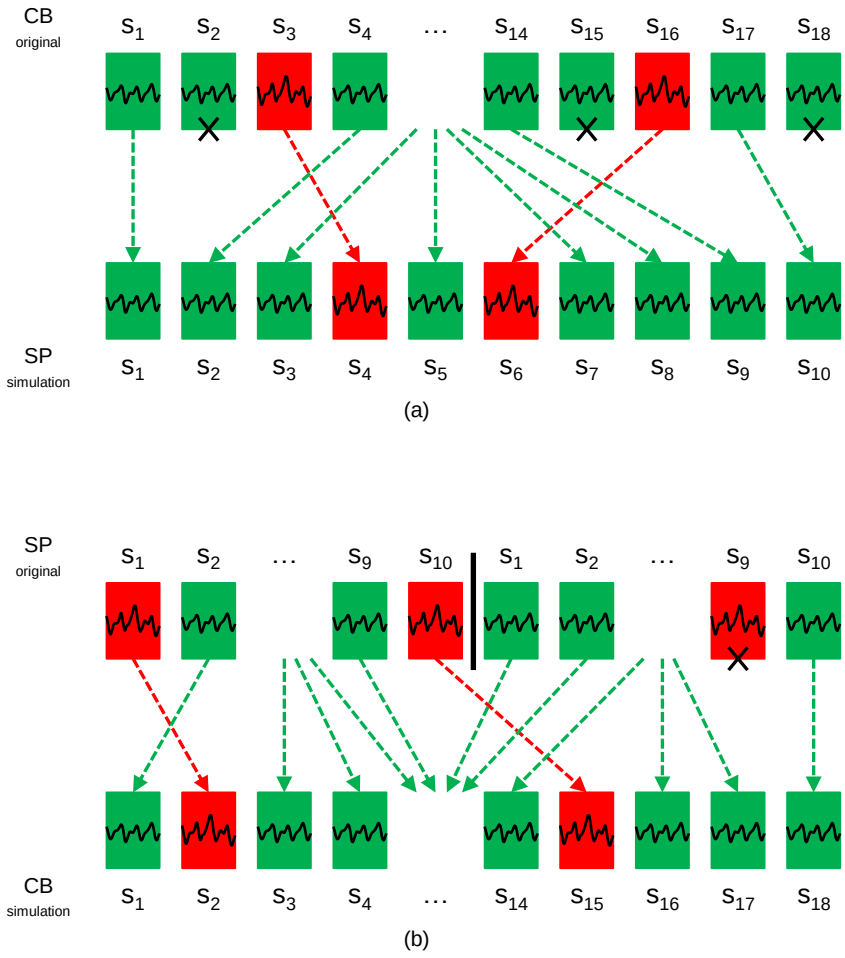

Figure 3.10: Schematic representation of the two simulated experiments. (a) the SP experiment is simulated with the data recorded in the CB experiment. (b) the CB experiment is simulated with the data recorded during the SP experiment. As the iteration length is smaller for SP, every second symbol is left out of the simulation and the SP data recorded during this symbol is used to complement the data for the previous symbol.

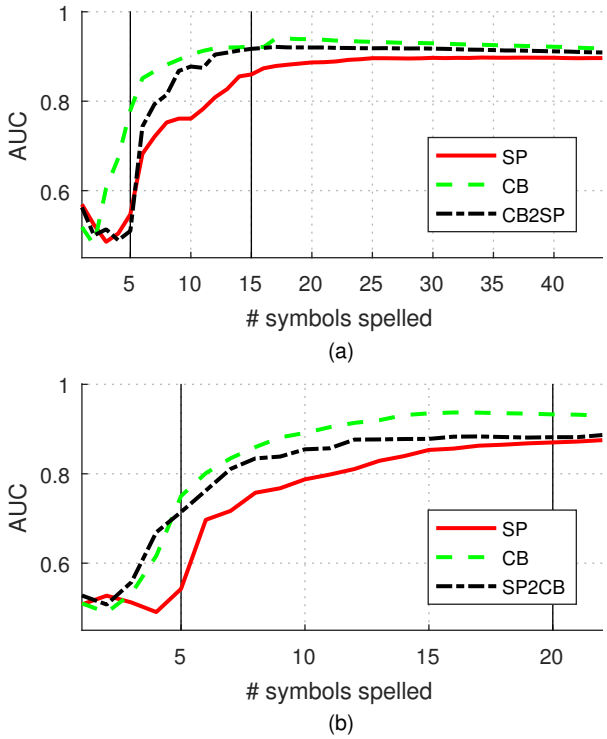

Figure 3.11: The influence of quantity and SNR of recorded data on the performance of the self-learning decoder. The three phases in the spelling session are indicated by the vertical lines. (a) Evolution of the AUC during the simulation of the SP experiment using the data of the CB experiment (CB2SP), compared to the results of the original experiments. (b) Evolution of the AUC during the simulation of the CB experiment using the data of the SP experiment (SP2CB). This AUC is compared with a simulation of the original SP and CB experiment on the same symbols.

determining factor in this phase.

Phase 2 : Around the fifth symbol the decoder has been trained on a larger set of recorded responses. The AUC curve rises quickly, approaching the curve of the experiment that recorded the same SNR (the CB experiment Figure 3.11(a)). At this point the SNR of the data becomes the dominant factor.

Phase 3 : The decoder performance saturates, the three curves con-

verge to the same result.

The self-learning decoder shows different requirements concerning the provided data during different phases of the spelling session. A technique that fully exploits this interaction to improve speller performance will have to adapt to the requirements of the decoder in each phase. The same phases can be found in the second simulation, see Figure 3.11(b).

3.4.2 Influence of data partitioning on decoder performance

The difference in iteration length between SP and CB leads to a different relative frequency of target and non-target signals. The ratio of target and non-target responses recorded per symbol will be denoted further as the *p-ratio*. An iteration of the CB paradigm includes two target stimuli and 16 non-target stimuli, leading to a p-ratio of 1/8. For SP ($n = 10$) the p-ratio is 1/4. To examine the influence of the p-ratio we execute a new simulation where the decoder is provided with an equal amount of data per symbol but in different p-ratios. The SP data is used but we increase the number of iterations to 18. Again, we leave out every second symbol and use this data to complement the data of the previous symbol. The SP paradigm now provides the decoder with 180 stimulus responses per symbol. We compare this to the original CB experiment (leaving every second symbol out), providing the same amount of responses per symbol but with the lower p-ratio as mentioned before. The evolution of the AUC during the two simulated spelling procedures is compared in Figure 3.12. It is clear that a higher p-ratio improves the training of the classifier in Phase 1 where data is scarce. A more balanced partitioning of the data between target and non-target samples thus improves the learning performance of the decoder in this early phase. A paired-samples t-test confirms that at the end of Phase 1 (when the fifth symbol is spelled) the AUC level achieved with the higher p-ratio is significantly higher ($t(23) = 2.247$, $p = 0.035$).

We can conclude that, during the spelling session, there is a shift in the mechanism underlying the paradigm-decoder interaction. When

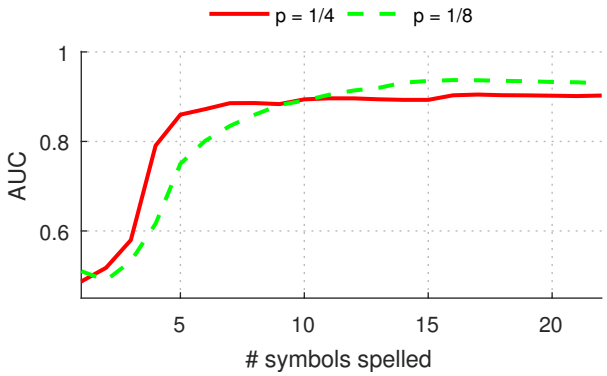

Figure 3.12: Evolution of the AUC during simulation with p-ratio 1/8 and 1/4.

data is scarce, the interaction is determined by the quantity and partitioning of the data provided to the decoder. Once a sufficient amount of data has been collected, the SNR of the data becomes more important.

In our online experiment, as the amount of data recorded per symbol is reasonably high, the SNR is the determining factor of the AUC level that is finally obtained. Therefore, the significantly different AUC obtained with SP stems from a lower SNR in the data.

3.4.3 Influence of application-decoder interaction on speed-accuracy trade-off

The results of the online experiment revealed that the SP paradigm obtained a significantly lower AUC compared to the basic paradigms. In contrast, the respelled symbol accuracy was not significantly different. In this closing section we answer the second question raised: does the lower AUC with SP lead to a lower respelling accuracy when we speed up the spelling process by reducing the number of sequence iterations?

The number of iterations used in our experiment was rather high compared to the conventional number around five iterations but it gives us the opportunity to make a complete speed-accuracy trade-off plot. The recorded data was used to simulate experiments with fewer iterations per symbol. Figure 3.13 illustrates the respelling ac-

curacy, obtained at the end of the spelling session, versus the number of symbols spelled per minute (SM).

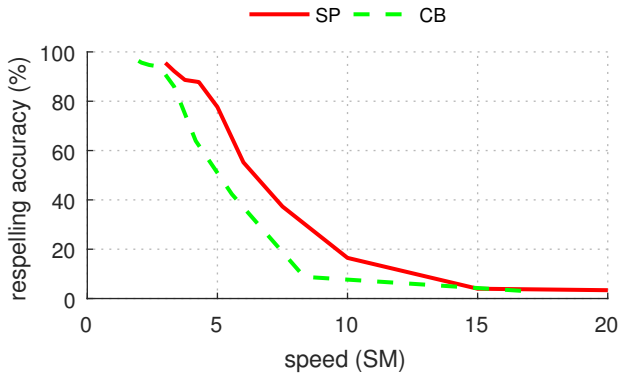

Figure 3.13: Speed-accuracy trade-off: accuracy obtained with SP and the CB paradigm versus the speed of spelling (expressed as symbols per minute) averaged over the 24 subjects.

The conclusions drawn for the course of the spelling session also apply to the end result. When less data is recorded per symbol, the p-ratio of the data is the dominant factor. Consequently the speller with SP performs better. A paired-samples t-test confirms that, for example at a speed of four symbols per minute, SP achieves a significantly higher respelling accuracy ($t(23) = 2.807$, $p = 0.01$)

3.5 Conclusion

In this chapter we showed that the application of an ERP-based BCI can be tuned by means of a new stimulus presentation paradigm. We proposed the switching paradigm for the ERP speller in which the iteration length and relative frequency of target stimuli can be chosen freely. At the same time, the most common causes of spelling errors are avoided by optimising the highlighting scheme. We compared the switching paradigm to the basic row-column and checkerboard paradigms in an online spelling experiment with 24 subjects. The results showed that this new paradigm obtains a higher number of correct symbols spelled per minute.

The online experiment demonstrated the evolution of the perfor-

mance obtained with the EM-based calibrationless decoding method from Kindermans et al. (2012). In the beginning of the spelling procedure, when data is scarce, there is a warm-up period during which the decoder's accuracy is low and highly variable between users. Afterwards, when a larger amount of data has been collected, the classification performance quickly increases and saturates at a higher level. Our offline analysis indicated a change over time in the aspects of the stimulus sequence that are influencing the decoder's performance most. Surprisingly, during the warm-up period, the ratio between the number of target and non-target responses was found to be more important than the signal-to-noise ratio of the data. Only in the second phase, the signal-to-noise ratio becomes the dominant factor. This result shows that the application settings truly affect the performance of the decoder and as such motivates the search for a symbiotic BCI design.

The results obtained with the self-learning decoder are remarkable. At the end of the spelling procedure, more than 95 % of the symbols are spelled correctly. Nevertheless, the warm-up period is a major issue that needs to be solved. It makes the BCI performance very unreliable in the beginning of spelling. Even more, the EM-based decoder does not have the theoretical guarantee to find a correct classification of the ERP responses.

In the following chapters, the switching paradigm will be used to develop a symbiotic ERP-based BCI in which the application is tuned to empower the calibrationless decoding. In the next chapter we will show that the paradigm is especially suitable to meet the requirements of a new self-learning decoding method that is more reliable than EM-based decoding. In the subsequent chapter, we will combine the benefits of both methods to obtain a new adaptive self-learning BCI that is effective, efficient and reliable.

4
Online unsupervised learning with guarantees

Traditional supervised BCI decoders are trained on a set of labelled calibration data, recorded prior to BCI use. As described in Chapter 1, this approach has two main drawbacks. First of all, the calibration session tires the user before he/she can use the BCI, thereby reducing the attentiveness of the subject and degrading BCI performance (Käthner et al., 2014). Secondly, the distribution of the data can change during use due to changes in the background activity or the pattern of the neural control signal (e.g. with increasing fatigue) (Shenoy et al., 2006; Von Bünau et al., 2009). This causes the performance of a supervised decoder to decrease over time.

As described in previous chapters, several methods have been proposed that reduce the need for calibration data by means of transfer learning (Krauledat et al., 2008; Fazli et al., 2009; Lu et al., 2009; Kindermans et al., 2014b; Wronkiewicz et al., 2015), adapting to online recorded data (Shenoy et al., 2006; Dähne et al., 2011; Vidaurre et al., 2011b,a) or completely unsupervised learning (Kindermans et al., 2012). Nevertheless, none of these methods has the theoretical guarantee to obtain a performance that is at least as good as a supervised classifier. This was illustrated in the previous chapter for the EM-based unsupervised decoding method from Kindermans et al. (2012), which showed high variability in performance.

In this chapter we describe the learning from label proportions (LLP) concept from Quadrianto et al. (2009) as a method for reliable classification without labelled data. LLP is a weakly supervised method to estimate the class-conditional mean feature vector in a

classification problem where the available data is unlabelled but provided in groups with a known relative frequency of data points in both classes. The LLP method is illustrated with the following example. Consider the outcome of an election were people can vote for Party A or Party B. We want to examine the difference in voting behaviour between men and women. The dataset is unlabelled as it is not known for individual votes if they were made by a man or a woman. However, the voting results are obtained in groups (namely the separate regions where votes were counted) and the proportion of men and women are known for these groups from demographic data. The following linear system can be set up:

$$\begin{cases} \mu_N = \pi_N \cdot \hat{\mu}_{\male} + (1 - \pi_N) \cdot \hat{\mu}_{\female} \\ \mu_S = \pi_S \cdot \hat{\mu}_{\male} + (1 - \pi_S) \cdot \hat{\mu}_{\female} \end{cases}$$

μ_N and μ_S are the known percentage of votes for the Party A in the northern and southern region respectively. π_N and π_S are the known fraction of men in the population of the northern and southern region. Solving this linear system yields an estimate of the average voting result for men and women separately, $\hat{\mu}_{\male}$ and $\hat{\mu}_{\female}$.

This chapter presents the result from an intensive collaboration with David Hübner and Michael Tangermann (University of Freiburg), Pieter-Jan Kindermans and Klaus-Robert Müller (Technical University of Berlin). We show the applicability of the LLP concept in ERP-based BCI to estimate the mean response feature vectors in the target and non-target class. These estimates are then used in a LSR classifier to obtain a reliable decoder that is guaranteed to converge to the supervised solution. We show how the application can be tuned to meet the requirements imposed by LLP, for which the paradigm proposed in the previous chapter is especially suitable.

In the next section we give the theoretical framework for the learning from label proportions method. Afterwards we describe how the speller application is modified for the decoder. Then, the method is evaluated in an online experiment, conducted by David Hübner, Konstantin Schmid and Michael Tangermann at the University of Freiburg. Finally, an offline resimulation of this experiment will compare LLP to the EM-based decoding method described before. In the

next chapter, we combine the strengths of both methods to obtain a new calibrationless BCI that is both effective and reliable.

4.1 Learning from label proportions

This section explains the learning from label proportions idea as proposed by Quadrianto et al. (2009).

4.1.1 The importance of estimating the mean feature vector

Linear classification methods were introduced in Chapter 2 to solve binary classification problems by finding a projection vector $\boldsymbol{w}$ and assigning a sample $\boldsymbol{x}$ to class C_+ if $\boldsymbol{w}^T\boldsymbol{x} > 0$ and to class C_- otherwise. The optimal weight vector $\boldsymbol{w}^*$ that provides the highest classification accuracy is found by optimising a loss function on a set of N labelled samples. For example, in LSR classification, the sum of squared errors between the projections $\boldsymbol{w}^T\boldsymbol{x}_n$ and their corresponding class label $y_n \in \{+1, -1\}$ is minimised:

$$\begin{aligned}\boldsymbol{w}^* &= \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} (\boldsymbol{w}^T\boldsymbol{x}_n - y_n)^2 \\ &= \arg\min_{\boldsymbol{w}} \sum_{n=1}^{N} \left((\boldsymbol{w}^T\boldsymbol{x}_n)^2 + 1\right) - 2\boldsymbol{w}^T \left(\sum_{n_+} \boldsymbol{x}_{n_+} - \sum_{n_-} \boldsymbol{x}_{n_-}\right)\end{aligned}$$

where n_+ and n_- denote respectively the samples from class C_+ and C_-. In the second line we rewrote the loss function to make the dependency on the class-conditional mean explicit. The first term is independent from any class label. The second term can be reformulated as:

$$2\boldsymbol{w}^T \left(\sum_{n_+} \boldsymbol{x}_{n_+} - \sum_{n_-} \boldsymbol{x}_{n_-}\right) = 2\boldsymbol{w}^T \left(N_+\boldsymbol{\mu}_+ - N_-\boldsymbol{\mu}_-\right)$$

with N_+ and N_- the number of samples in each class. It turns out that the loss function can be determined without labels once the class-

conditional mean feature vector and the relative frequency of samples in each class are known. In fact, this is a property for a subset of loss functions, called *symmetric proper scoring losses*. Besides the square loss, it also includes for example the logistic loss function (Patrini et al., 2014). With these loss functions, the optimal weight vector $\boldsymbol{w}^*$ is defined once the class-conditional means are found.

4.1.2 Estimating the mean feature vector with label proportions

The standard supervised classification methods from Chapter 2 are not applicable when labelled data is unavailable. Suppose however that unlabelled samples are observed in K distinct groups S_k, each having a known relative frequency of samples in both classes. For each group k, the mean feature vector $\boldsymbol{\mu}_k$ can be calculated directly from the observations. Denoting the relative frequency of both classes in group S_k as π_k^+ and π_k^-, the average feature vector in each group can be expressed as a function of the class-conditional means $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$:

$$\begin{aligned}\begin{bmatrix}\boldsymbol{\mu}_1\\ \vdots\\ \boldsymbol{\mu}_K\end{bmatrix} &= \begin{bmatrix}\pi_1^+ & \pi_1^-\\ \vdots & \vdots\\ \pi_K^+ & \pi_K^-\end{bmatrix} \times \begin{bmatrix}\boldsymbol{\mu}_+\\ \boldsymbol{\mu}_-\end{bmatrix}\\ &= \boldsymbol{\Pi} \times \begin{bmatrix}\boldsymbol{\mu}_+\\ \boldsymbol{\mu}_-\end{bmatrix}\end{aligned}$$

Here, the $K \times 2$ mixture matrix $\boldsymbol{\Pi}$ contains the relative frequencies in each group. The linear system solves for the class-wise mean feature vectors. Using the pseudo-inverse matrix $\boldsymbol{\Pi}^{-1} = (\boldsymbol{\Pi}^T\boldsymbol{\Pi})^{-1}\boldsymbol{\Pi}^T$, the solution can be written as follows:

$$\begin{bmatrix}\boldsymbol{\mu}_+\\ \boldsymbol{\mu}_-\end{bmatrix} = \boldsymbol{\Pi}^{-1}\begin{bmatrix}\boldsymbol{\mu}_1\\ \vdots\\ \boldsymbol{\mu}_K\end{bmatrix}$$

Consequently, by learning from the known proportion of labels in each group of observed data, LLP estimates $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$ without labels. It is important to note that this method assumes homogeneity, i.e. that the class-conditional means are the same in each group S_k.

LLP estimates the mean feature vector feature by feature. For example, the d^{th} feature $\mu_{+,d}$ is estimated as a linear combination of the average feature values $\{\mu_{1,d}, \ldots, \mu_{K,d}\}$ in each group. We assume the samples to be independently and identically distributed (IID) with expected value μ_d and variance σ_d^2 for feature d. By the central limit theorem, the average of N feature values $x_{n,d}$ is normally distributed around the true mean μ_d with variance σ_d^2/N for large N:

$$\mu_{k,d} = \frac{x_{1,d} + \cdots + x_{N,d}}{N} \sim \mathcal{N}(\mu_d, \frac{\sigma_d^2}{N})$$

This implies that the estimated group means $\hat{\boldsymbol{\mu}}_k$ converge to their true value $\boldsymbol{\mu}_k$ for $N \to \infty$. Consequently, under the assumption of homogeneity and IID samples, the solution of the linear system is guaranteed to converge to the true class-conditional means $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$.

4.1.3 Comparison to supervised estimation

The error on a parameter estimation is measured by the variance of its estimator. We assume each feature d in the data to be normally distributed[1] with a variance σ_d^2. Furthermore, we denote the elements of the pseudoinverse of the mixture matrix $\mathbf{\Pi}^{-1}$ as follows:

$$\mathbf{\Pi}^{-1} = \mathbf{\Phi} = \begin{bmatrix} \phi_+^1 & \cdots & \phi_+^K \\ \phi_-^1 & \cdots & \phi_-^K \end{bmatrix}$$

[1] This applies specifically to the features of ERP response signals, as shown by Blankertz et al. (2011) and discussed in Section 2.3.2.

With the assumption that the samples are IID, the variance on the LLP estimation of the mean feature $\mu_{+,d}$ is obtained as follows:

$$\begin{aligned}\mathrm{Var}[\hat{\mu}_{+,d}] &= \mathrm{Var}\left[\sum_{k=1}^{K} \phi_+^k \hat{\mu}_{k,d}\right] \\ &= \sum_{k=1}^{K} (\phi_+^k)^2 \mathrm{Var}\left[\hat{\mu}_{k,d}\right] \\ &= \left(\sum_{k=1}^{K} \frac{(\phi_+^k)^2}{N_k}\right) \sigma_d^2\end{aligned}$$

With N_k the number of observed samples in group S_k.

Supervised estimation of the class-conditional mean feature vector with labelled data can be interpreted as a specific case of LLP where one group contains all the samples from C_+ and another group the samples from C_-. The solution for the linear system is very simple in this case:

$$\begin{bmatrix}\boldsymbol{\mu}_+ \\ \boldsymbol{\mu}_-\end{bmatrix} = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix}^{-1} \cdot \begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix}$$

The computation of the average feature vector in each group directly yields the estimated class-conditional means. The variance on this estimation is given by:

$$\mathrm{Var}[\hat{\mu}_{+,d}] = \mathrm{Var}[\hat{\mu}_{1,d}] = \frac{\sigma_d^2}{N_+}$$

Therefore, the variance on the LLP estimation of $\mu_{+,d}$ is equal to the variance on the supervised estimation multiplied by a factor Q:

$$Q = \left(\sum_{k=1}^{K} \frac{N_+}{N_k} (\phi_+^k)^2\right)$$

The variance amplification factor Q depends on the number of groups, the mixture ratios and the proportion of data contained in each group. Table 4.1 lists some examples of mixture matrices and their resulting variance amplification for the specific case of two groups containing an equal portion of the observed data and $N_+ = N_- = N/2$. As illustrated in this table, the more diverse the relative frequencies are

$\mathbf{\Pi}$	$\mathbf{\Phi} = \mathbf{\Pi}^{-1}$	Q
$\begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$	$\begin{bmatrix} 1.125 & -0.125 \\ -0.125 & 1.125 \end{bmatrix}$	1.28
$\begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$	$\begin{bmatrix} 1.75 & -0.75 \\ -0.75 & 1.75 \end{bmatrix}$	3.63
$\begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$	$\begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}$	13
$\begin{bmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{bmatrix}$	$\begin{bmatrix} 5.5 & -4.5 \\ -4.5 & 5.5 \end{bmatrix}$	50.5

Table 4.1: Example of different mixture matrices and the resulting amplification of the LLP estimation error compared to the supervised estimator.

in the different groups, the less the variance is amplified. For a given dataset, the minimal error on the estimation is obtained when each group contains samples from a different class, i.e. the supervised case.

4.2 Symbiotic integration of LLP in the ERP speller

We now apply the LLP concept in the ERP speller to estimate the mean of the response feature vectors in the target and non-target class. This requires a number of changes to the speller interface. First of all, the stimulus responses need to be recorded in separate groups for which the relative frequency of target and non-target responses is known and different. Secondly, the homogeneity assumption requires the mean target and non-target response to be the same in each group of data. In this section we demonstrate how the ERP speller application is modified to meet these requirements.

4.2.1 LLP stimulus presentation paradigm

Chapter 3 described how the ratio of target and non-target responses is determined by two parameters of the stimulus presentation paradigm: the iteration length n and the number of times r that each symbol is highlighted per iteration. The relative frequency is r/n for target responses and $(n-r)/n$ for non-target responses. We proposed a stimulus presentation paradigm in which these parameters can be chosen freely (Verhoeven et al., 2015). In this way, stimulus sequences can be generated with very different relative frequencies. In the current proof of concept we use a first sequence type S_1 with $n = 8$ and $r = 3$, yielding a target frequency of $3/8 = 0.375$ and a second sequence S_2 with $n = 18$ and $r = 2$ resulting in a target frequency of $2/18 = 0.111$. The resulting mixture matrix and its inverse are:

$$\mathbf{\Pi} = \begin{bmatrix} 3/8 & 5/8 \\ 2/18 & 16/18 \end{bmatrix}, \mathbf{\Phi} = \begin{bmatrix} 3.37 & -2.37 \\ -0.42 & 1.42 \end{bmatrix}$$

Note that even more extreme relative frequencies can be obtained with the tunable paradigm. The current choice facilitates the additional modifications to the application as explained further.

The two distinct groups of data are obtained by using the two paradigm settings alternately during the spelling process. However, the faster reoccurring target stimulus in S_1 potentially alters the user's awareness and as such could result in a slightly different mean response in this group of data. To comply with the homogeneity and IID assumption in LLP, the two sequence types are randomly interleaved. The amount of data recorded in each group is equalised by interleaving two sequences of type S_1 with one sequence type S_2. This results in an interleaved sequence of length $n = 34$ that highlights each symbol $k = 8$ times. The variance amplification factor is $Q = 8.17$ for the estimation of the mean target response $\boldsymbol{\mu}_+$ and $Q = 3.21$ for the mean non-target response $\boldsymbol{\mu}_-$. Figure 4.1 schematically presents an interleaved sequence of stimuli and shows how it is used by the LLP method to reconstruct the class-wise mean responses.

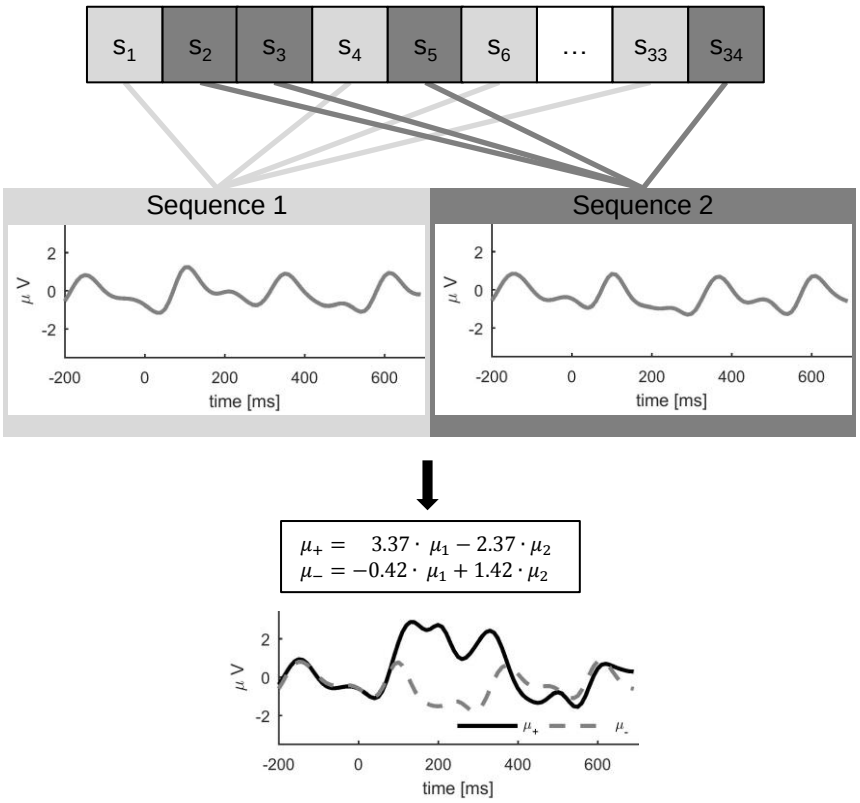

Figure 4.1: Schematic representation of the LLP method to reconstruct the class-conditional ERP responses from the recorded data. Per iteration, 34 stimuli are presented with 16 stimuli belonging to sequence type 1 and 18 to sequence type 2. Next, the stimulus response is averaged per group. Finally, the linear problem is solved to obtain the reconstructed class-conditional mean responses.

4.2.2 Modifications to the spelling interface

The assumption of homogeneity and IID[2] responses requires an extra modification to the spelling interface. The paradigm determines the brightness level of the visual stimuli presented to the user. With M

[2]Even with these modifications, the IID assumption does not hold in practice for the features of the ERP responses due to non-stationarities. Nevertheless, the experimental evaluation in this chapter will demonstrate the performance obtained with the LLP method.

symbols in the grid, the average number of symbols highlighted simultaneously per stimulus is $(M \cdot r/n)$. In a 6×6 speller grid, each stimulus in sequence S_1 highlights 13 or 14 symbols, while stimuli in sequence S_2 highlight just four symbols simultaneously. The resulting difference in brightness is known to influence the response on the visual stimuli (Johannes et al., 1995). For this reason, the 6×6 grid in the original spelling interface is replaced by a 6×7 grid that, besides the regular symbols from the alphabet, contains ten '#' symbols that account for visual blanks. They can never be chosen by the subject as target symbol and are used to increase the brightness level of those stimuli that highlight less symbols. With this adjustment, each stimulus highlights 12 symbols in the grid and the two groups of stimuli become indistinguishable to the user. The new speller interface is illustrated in Figure 4.2.

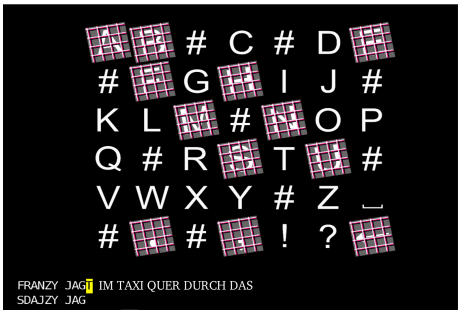

Figure 4.2: The modified spelling interface containing 10 '#' symbols that account for visual blanks. A stimulus highlighting 12 symbols is shown, using the salient highlighting effect designed by Tangermann et al. (2011).

4.3 Online evaluation

An online experiment was conducted at the University of Freiburg to asses the performance of LLP as decoding method in the ERP speller. This section describes the experimental design and discusses the results. Finally, the LLP method is compared to the EM-based unsupervised decoding by Kindermans et al. (2012) in an offline simulation on the recorded dataset.

4.3.1 Experimental set-up

Participants

13 subjects (8 male, 5 female) with an average age of 26 years (STD = 1.5) participated in the study. They were given a fee independently of the experimental result. One subject had prior experience with EEG recording. The study was in accordance with the principles embodied in the declaration of Helsinki and approved by the ethics committee of the University Medical Center Freiburg. Each subject gave written informed consent before taking part in the experiment.

Experiment design

Subjects were seated comfortably in a chair, facing a 24 inch flat screen at a distance of approximately 80 cm that presented the 6×7 grid proposed in Section 4.2. Each subject was asked to spell the following sentence three times:

"FRANZY JAGT IM KOMPLETT VERWAHRLOSTEN TAXI QUER DURCH FREIBURG"

This yields three recording blocks of 63 symbols per subject. The sentence to spell was predefined to allow for accurate evaluation of the spelling performance obtained with the LLP method. The knowledge of the target symbol was not used for any other purpose. Consequently, the stimulus presentation paradigm and the decoder did not use any label information.

First, the position of the current symbol to spell was indicated in the grid during a four seconds cue. Next, two iterations of the interleaved 34 stimuli sequence, proposed in Section 4.2, were presented. To save computation time during the experiment, the interleaved sequence of stimuli was randomly selected from a set of 100 pregenerated sequences. Note that the stimulus presentation paradigm is completely unaware of the target symbol. Every symbol in the grid, except for the uninformative hash symbols, was highlighted an equal number of times in each trial. For the visual stimulation, we used the very salient highlighting effect designed by Tangermann et al. (2011), which included the overlay of the symbols with a coloured grid and a

brief animated rotation. Figure 4.2 illustrates this effect. Each visual stimulus lasted for 100 ms and the ISI was 150 ms, yielding a stimulus onset asynchrony of 250 ms. Therefore, the complete sequence of 68 stimuli took 17 s to be presented. The recorded responses were processed and classified and the predicted symbol was shown to the user for four seconds. Consequently, we obtained a spelling speed of 2.4 symbols per minute. Dynamic stopping was not used in this experiment. A constant rate of recorded responses per trial facilitates the evaluation of how the LLP estimates evolve with an increasing amount of recorded data.

Data acquisition and processing

EEG was recorded at a sampling rate of 1 kHz with the EasyCap EEG cap and the BrainAmp DC (Brain Products) multichannel EEG amplifier. 31 passive Ag/AgCl electrodes were positioned on the scalp according to the extended 10-20 system[3] (Chatrian et al., 1985) and referenced against the nose. The ground location was AFz. Impedances were kept below 20 kΩ. An optical sensor on the screen marked the exact onset time for each stimulus. The data is available on the Zenodo database (DOI: http://doi.org/10.5281/zenodo.192684).

Data processing was conducted during the experiments with the BBCI Toolbox (Blankertz et al., 2010). The recorded EEG was bandpass filtered between 0.5 and 8 Hz with a third order Chebyshev Type II filter and downsampled to 100 Hz. The ERP response signal is taken in the [-200 700] ms interval around the stimulus onset and the average amplitude over the [-200 0] ms interval is subtracted as a baseline reference. The channels Fp1 and Fp2 were not used in further analysis. For the other channels, the average amplitude over six intervals ([50, 120], [121, 200], [201, 280], [281, 380], [381, 530], [531, 700] ms) was computed. This resulted in a total of $29 \cdot 6 = 174$ features per stimulus response.

At the end of each trial, the recorded responses are added to the complete set of observed data and the estimates of the class-conditional mean response and pooled covariance structure are up-

[3] F_{p1}, F_{p2}, F_9, F_7, F_3, F_z, F_4, F_8, F_{10}, FC_5, FC_1, FC_2, FC_6, T_7, C_3, C_z, C_4, T_8, CP_5, CP_1, CP_2, CP_6, P_9, P_7, P_3, P_z, P_4, P_8, P_{10}, O_1, O_2

dated with this extended dataset. Section 4.2 explained how the interleaved sequence of stimuli divides the data into two groups with known relative frequencies and as such allows for the LLP method to estimate the class-conditional mean target and non-target response $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$. The pooled covariance matrix $\boldsymbol{\Sigma}$ is estimated directly from the recorded responses and regularised with Ledoit-Wolf shrinkage (Blankertz et al., 2011; Ledoit and Wolf, 2004; Bartz and Müller, 2014). Using the equivalence between LSR and LDA discussed in Chapter 2, the estimated parameters yield a LSR projection vector $\boldsymbol{w}$:

$$\boldsymbol{w} = \boldsymbol{\Sigma}_{reg}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)$$

Next, the classifier score is determined for each response observed so far. The knowledge of the predefined sentence is then used to report the AUC as the current classification performance. Finally, for each trial and each symbol (except for the visual blanks, which are not considered) the classifier output scores are summed for all stimuli that highlight this specific symbol. The symbol yielding the highest sum was then selected as the predicted target symbol for that trial. Note that the knowledge of the response labels is not used in any way to update this classifier but merely to report classification performance. The classifier was reset at the beginning of each new spelling block of 63 symbols.

4.3.2 Results and discussion

Homogeneity assumption

The LLP method assumes the mean response to be the same in each recorded group. Before discussing the online spelling result with LLP, we check if this assumption holds in our experiment. The top graph in Figure 4.3 illustrates the grand average target and non-target response, as recorded in the Cz electrode, for stimulus sequence type 1 and type 2 separately. It is computed over the three spelling blocks for subject S1. The coloured areas indicate the time intervals over which the signal amplitude is averaged to obtain the six feature values per channel. Overall, the averaged waveforms in the two groups of data are quite similar. Only small differences can be recognised visually.

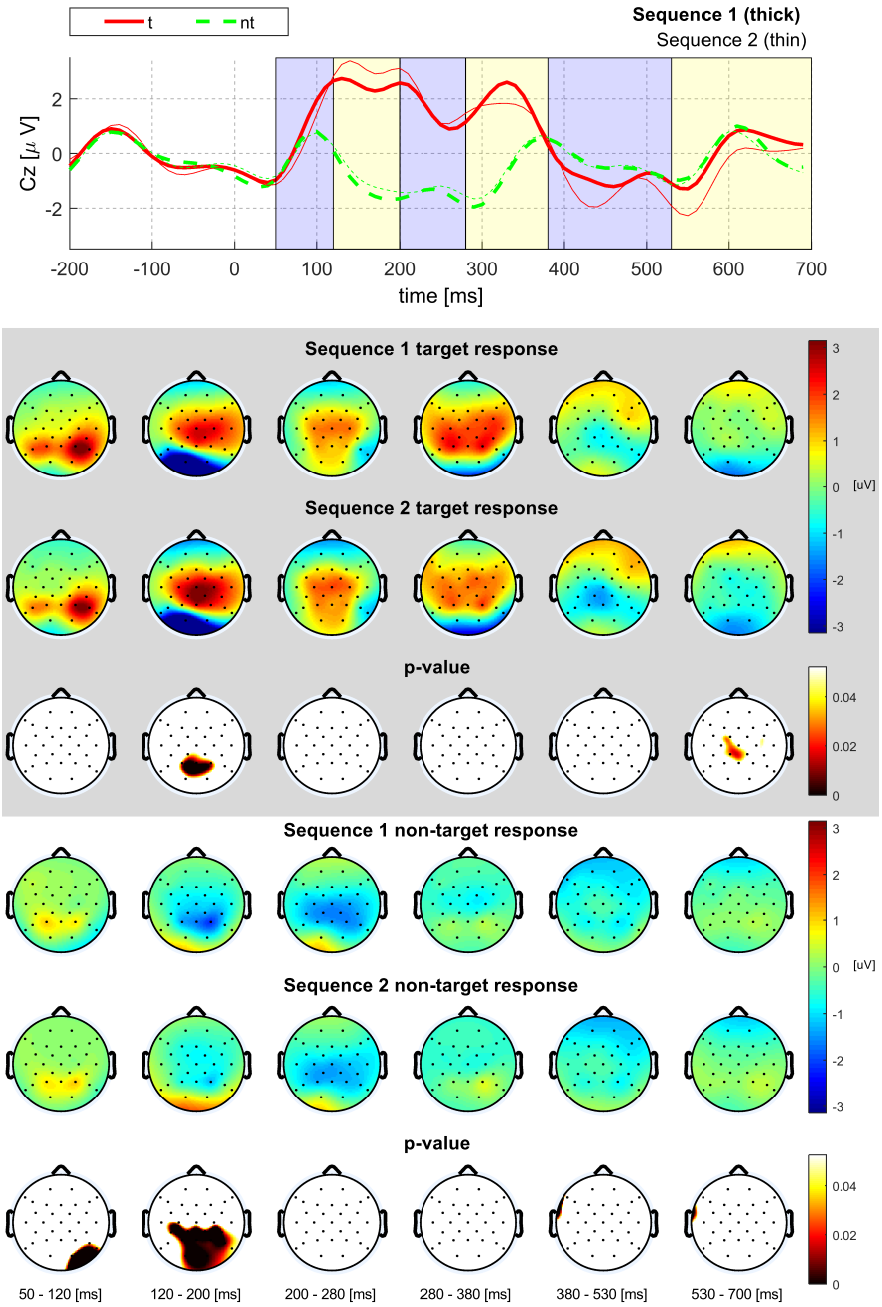

Figure 4.3: Comparison of the class-conditional mean response in the two groups of recorded data for subject S1. The top graph shows the average response on stimuli from both sequences, recorded in the Cz electrode. The rows below give the scalp plots for the mean response feature values and the p-value of a two-sided t-test, testing for significant differences between the feature values recorded in sequence 1 and sequence 2.

Below the graph, we examine the mean responses in scalp maps. Each scalp map corresponds to one of the indicated time intervals. It illustrates the distribution of the recorded electric potential over the scalp, averaged over this interval. The dots in the map correspond to the different EEG channel locations. In this way, a row of six scalp maps can present the complete feature vector. The top three rows below the graph describe the grand average target response. The bottom three rows show the average non-target response. For each class, the top row shows the average response in sequence type 1. The middle row shows the average response in sequence type 2. The bottom row illustrates the p-value of a two-sided t-test for each feature, testing for significant differences between the two sequence types. Visual examination of the maps in rows 1, 2, 4 and 5 again indicates only small differences between the average responses recorded in each sequence type. For most features there is no significant difference between the two groups of data. The fifth row shows some difference in the non-target response during the [120-200] ms interval. Nevertheless, as we will see further, the LLP method was able to estimate these class-conditional mean feature values accurately.

AUC and symbol selection accuracy

Figure 4.4 shows the online evolution of the AUC during the spelling session. Subjects S6 and S10, respectively, obtain the highest and lowest AUC at the end of the spelling block. For this reason, these subjects are chosen as example cases.

The LLP method is compared to a standard supervised LSR classifier. For each subject and each spelling block, the first $N \in [5, 10]$ trials and the corresponding label information are used to train the supervised classifier. This classifier is then applied on the remaining trials in a simulation that mimics the original experiment. In contrast to the LLP method, the supervised classifier does not update its parameters when new data is observed. In Figure 4.4, the supervised AUC is not reported for the first N trials that are used for training. This illustrates what would happen in a real supervised experiment where the user has to spell N symbols in a calibration procedure before actually spelling his/her own desired symbols.

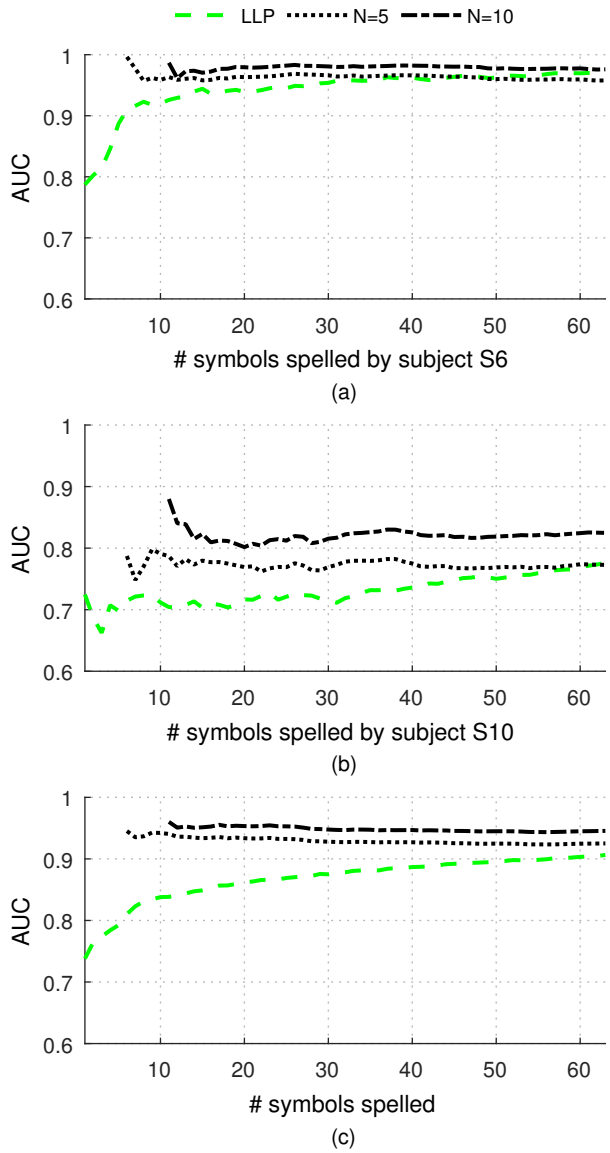

Figure 4.4: Evolution of the AUC during the online spelling experiment with the LLP method. The performance is compared to an offline simulation with a supervised LSR classifier trained on the first $N \in [5, 10]$ trials. (a) AUC for the first block recorded in subject S6. (b) AUC for the first block recorded in subject S10. (c) Grand average over the 13 subjects and their 3 recorded blocks.

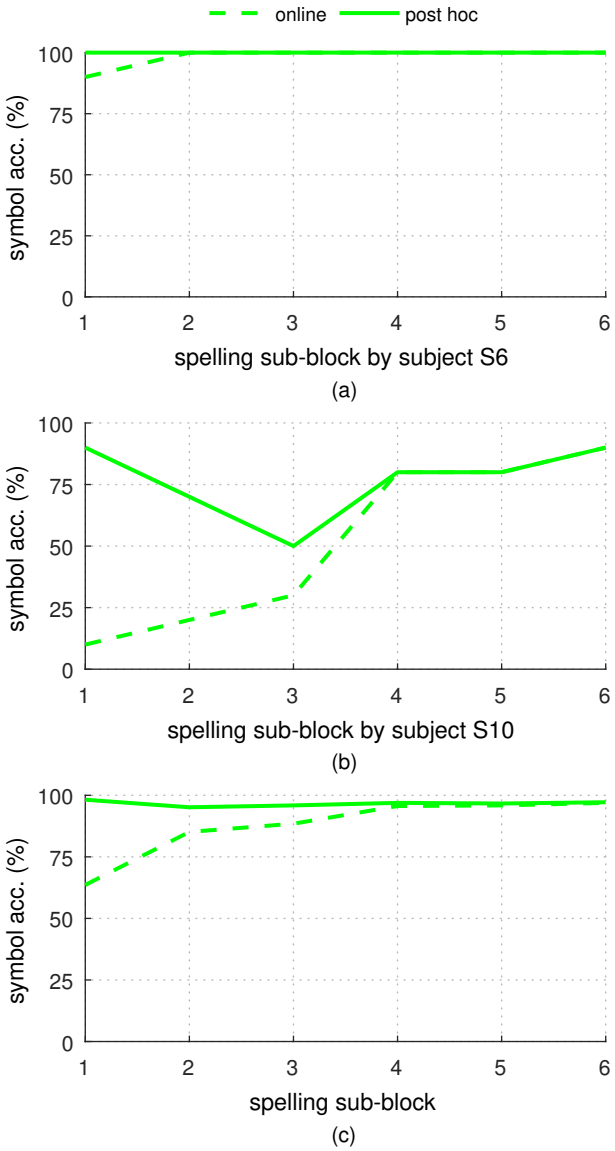

Figure 4.5: Evolution of the symbol accuracy during online spelling with the LLP method and the post-hoc re-evaluation of the finally obtained decoder on all trials. (a) Symbol accuracy per sub-block of 10 symbols spelled by subject S6 in the first spelling block. (b) Same result for subject S10. (c) Grand average over the 13 subjects and their 3 recorded blocks.

The figure shows that the LLP method starts at a relatively high AUC and gradually increases in performance when more data is collected. Figure 4.4(a) and (b) illustrate the diversity of the performance achieved by different subjects. The AUC obtained with the LLP method approaches the performance of a supervised classifier as more data is collected. This confirms that the LLP decoder has the guarantee to converge to the supervised solution. Furthermore, the average AUC obtained by the supervised classifiers shows a slight decrease when more symbols are spelled. As explained before, this may be due to non-stationarity effects in the recorded EEG. In contrast, the LLP method re-estimates its parameters when new data is recorded and as such is capable of adapting to small changes.

In Figure 4.5, the symbol spelling accuracy is reported per sub-block of 10 trials. For example, a reported value of 70 % indicates that in 7 out of those 10 consecutive trials the symbol was selected correctly at the end of the trial. At the end of the spelling session, the decoder is re-evaluated on all recorded trials. The 'post hoc' line in Figure 4.5 reports this respelled accuracy for every sub-block of 10 trials. It does not represent an online evolution of performance, but shows how the trained classifier is capable of correcting the selections it made in the past.

Similar to the AUC, the online symbol selection accuracy obtained with LLP gradually increases with the amount of data collected. Figure 4.5(a) and (b) again illustrate the diversity in the recorded subjects. On average, the post hoc result in Figure 4.5(c) demonstrates a nearly perfect symbol selection. The sentence obtained at the end of the spelling session is very close to the predefined sentence.

Evaluation of the estimated mean response

Figure 4.6 demonstrates the estimation of the class-conditional mean response with LLP. The first spelling block by subject S1 is taken as example case. The top graph in this figure illustrates the average recorded response on target and non-target stimuli. Coloured areas indicate the six time intervals in which the amplitude is averaged to obtain the features for each individual response. The different rows of scalp maps illustrate how the estimation with the LLP method con-

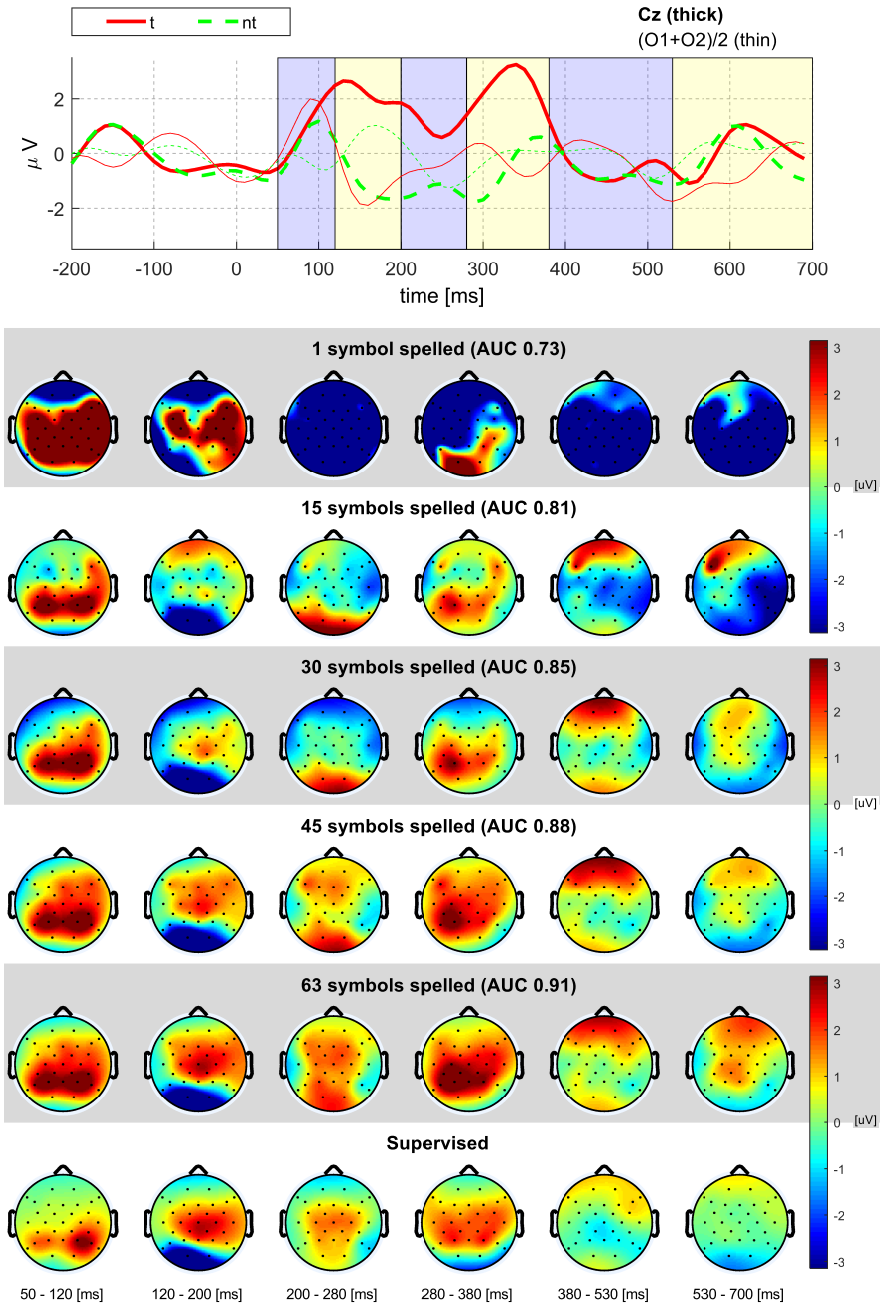

Figure 4.6: Evaluation of the mean response estimation for subject S1. The top graph shows the average target and non-target response. The blue and yellow shaded areas indicate the intervals in which the amplitude is averaged to obtain the six features per channel. The rows below give the scalp plots for the target response as estimated by the LLP method for an increasing amount of available data. The last row illustrates the supervised estimation.

verges to the supervised estimation as more data is collected. The AUC obtained during the spelling experiment with each estimation is indicated in the parentheses. Note that this AUC value also depends on the estimation of the non-target response and the pooled covariance matrix, not shown here. The supervised estimation of the means was obtained with the complete dataset for subject S1 and the corresponding labels.

4.4 Comparing LLP to EM-based decoding

In this final section, the new self-learning decoder with LLP is compared to the EM-based unsupervised decoder from Kindermans et al. (2012). The EM method is applied in an offline resimulation of the experiment described in Section 4.3.1. The stimulus responses are processed sequentially in the order they were observed. The data processing and feature extraction is the same as in the original experiment. At the end of each trial, the EM-based decoder is updated with the extended dataset and its classification performance is assessed in the same way as the LLP method was evaluated online.

The parameters of the EM algorithm are chosen in accordance with the original work by Kindermans et al. (2014b), explained in Section 2.3.2. Five classifier pairs are randomly initialised and updated in parallel. The precision on the prior of $\boldsymbol{w}$ is initialised to $\alpha = 100$ and limited to a maximum value of $\alpha = 200$ to avoid over-fitting.

Figure 4.7 compares the AUC obtained with the EM-method to that of the LLP method. Figure 4.7(a) and Figure 4.7(b) show the result for the first spelling block of respectively subject S6 and subject S10. The thin blue lines in these figures illustrate the AUC obtained by the ten randomly initialised EM decoders individually. Some of them fall back on the same solution. The figure demonstrates that the performance of the EM decoder strongly depends on the initialisation of the parameters. The LLP method does not depend on any random initialisation and therefore shows a robust performance level.

As explained in Chapter 2, the variability in the performance of

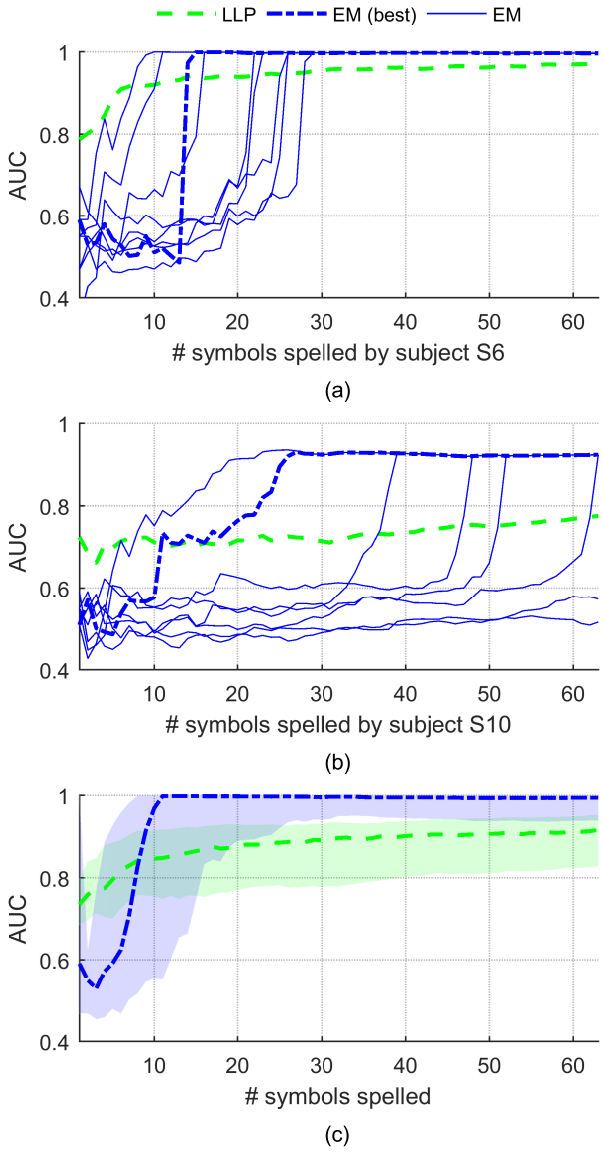

Figure 4.7: Online AUC obtained with the LLP method compared to the EM-based decoder in a resimulation of the online experiment. (a) AUC obtained by subject S6 in the first spelling block. Thin blue solid lines show the AUC obtained by the different EM decoders, randomly initialised and updated in parallel. The thick blue line shows the result obtained by the EM decoder selected at the end of each trial. (b) Result for subject S11 (c) Grand median AUC over all subjects and spelling blocks. The coloured area denotes the 10-90 percentile interval.

the EM decoder is tackled by initialising and updating several EM classifiers in parallel. At the end of each trial, the classifier with the highest data likelihood is selected to report the classifier performance. In addition, in each pair the weight vector of the classifier with the lowest data likelihood is reinitialised to $-\boldsymbol{w}$, with $\boldsymbol{w}$ the weight vector of the one with the highest data likelihood. The thick blue lines in Figure 4.7 show the AUC that is obtained when this trick is used on the five pairs of randomly initialised decoders. The probability of finding a good classification is clearly increased. Nevertheless, the figure shows that the selection of the best classifier based on its data likelihood is not optimal. Besides that, updating multiple classifiers increases the computation time at the end of each trial and as such slows down the spelling procedure.

The LLP and EM method clearly show complementary behaviour during the warm-up period. The LLP method starts with a relatively high AUC and improves slowly, whereas the EM-method achieves initially a lower AUC but improves faster to very high levels.

Figure 4.7(c) shows the grand average result over the 13 subjects and their three spelling blocks obtained with the LLP method and the (multi-decoder) EM method. The shaded area illustrates the 10-90 percentile interval of the result obtained over these subjects. The variance in performance over the different subjects and spelling blocks is significantly lower in LLP compared to EM. For example, at the eighth symbol, the 10-90 percentile interval of the AUC is [0.76, 0.92] for LLP. For the EM decoder this interval is [0.52, 0.99], illustrating that some subjects achieve near perfect scoring while others still perform at chance level. This illustrates once again that, even with the trick of initialising multiple classifiers, the performance obtained with EM is less reliable. In conclusion, the LLP-based decoder is very reliable but learns slower compared to the EM-based decoder.

4.5 Conclusion

In this chapter we described learning from label proportions as a method to reliably estimate the class-conditional mean response without labelled data. We presented the theoretical framework for this

method and showed that the estimated means are guaranteed to converge to the supervised estimate. We demonstrated its applicability in the ERP speller system and used the stimulus presentation paradigm proposed in the previous chapter to tune the speller application to the requirements of this decoder.

An online experiment with 13 subjects showed the reliable and effective spelling performance obtained with the LLP decoder. We compared it to the EM-based and supervised decoders in an offline resimulation of this experiment, which demonstrated the complementary behaviour of the two unsupervised methods. While LLP is guaranteed to converge to the supervised solution, convergence is rather slow. On average, EM obtains a higher performance, but the result is highly variable between subjects and strongly depends on the parameter initialisation. The high reliability of the LLP method comes at the cost of a slower learning performance. We end up with two self-learning decoders with complementary characteristics. This observation will be used in the next chapter where the benefits of EM and LLP will be combined to obtain a decoder that is both effective and reliable at the same time.

5

Improving zero-training BCI by mixing model estimators

In Chapter 1 we stated that a BCI is required to be effective, efficient, reliable and easy to use. To achieve this goal we proposed a new design approach in which the different components of the BCI are co-adapted to each other. The symbiotic design for ERP-BCI started in Chapter 3, where we proposed a tunable stimulus presentation paradigm. The paradigm was used to examine the interaction between the application and a self-learning decoder, which tunes its parameters during actual use of the BCI. We used the first truly calibrationless decoder for ERP-BCI from Kindermans et al. (2012). The online experiment demonstrated the remarkable performance obtained with this unsupervised classifier. Nevertheless, due to the random initialisation of its parameters and their tuning with the EM algorithm, this decoder does not have the theoretical guarantee to find a correct classification of stimulus responses. For this reason, the EM-based self-learning decoder is not reliable.

We used our flexible paradigm in Chapter 4 to apply learning from label proportions in ERP-BCI. This method is capable of estimating the decoder's parameters reliably without labelled calibration data. In contrast to the EM-based decoder, the LLP-based classifier is guaranteed to converge to the supervised solution when more data is collected (Hübner et al., 2017). Nevertheless, the resulting high reliability comes at the cost of a slower learning process compared to EM. The two self-learning decoders clearly show complementary strengths and weaknesses.

In this chapter, I will propose a method to optimally combine

the LLP and EM decoding methods in a theoretical way. By letting each method's strengths compensate for the weaknesses of the other, we intend to obtain a self-learning decoder that is both reliable and effective. The proposed method is inspired by the shrinkage approach for regularisation of supervised models (Höhne et al., 2016) and the mixing of parametric and non-parametric statistical estimators (Olkin and Spiegelman, 1987).

In the next section, we focus on the estimation of the mean ERP response and compare the unsupervised LLP and EM methods theoretically. Afterwards we propose the new estimator as a mixture of the existing estimators and present an analytical formula to compute the optimal mixing coefficient. Then, the mixture method is compared to LLP and EM in an extensive offline simulation of an experiment with the visual ERP speller. Finally, the comparison is augmented with a true online experiment, conducted at the University of Freiburg in collaboration with David Hübner and Michael Tangermann.

5.1 Estimation of the mean ERP response

In Chapter 2 we introduced least squares regression (LSR) as a method for classification. The LSR classifier assumes that there exists a one-dimensional projection of the features that is normally distributed with a class-conditional mean and shared variance. As the features derived from EEG closely follow the normality assumption (Blankertz et al., 2011), this simple technique has been widely applied and was shown to be competitive with more complex methods for the classification of neural control signals (Lotte et al., 2007; Müller et al., 2008; Blankertz et al., 2011; Kindermans et al., 2011). We explained that the optimal LSR projection vector $\boldsymbol{w}$ is obtained with the following formula:

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}(\boldsymbol{X}^T\boldsymbol{t})$$

In addition, we demonstrated that the linear discriminant analysis (LDA) classifier is a specific case of LSR that is obtained when the data is centred and the class labels rescaled to $y \in \{N/N_+, -N/N_-\}$, with N_+ and N_- the number of samples in each class. Substituting

these assumptions in the formula for the LSR weight vector yields the LDA solution:

$$\begin{aligned}\boldsymbol{w}^* &= \left(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\left(\frac{N}{N_+}\sum_{i_+=1}^{N_+}\boldsymbol{x}_{i_+} - \frac{N}{N_-}\sum_{i_-=1}^{N_-}\boldsymbol{x}_{i_-}\right)\\ &= \boldsymbol{\Sigma}_{reg}^{-1}\left(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-\right)\end{aligned}$$

where $\boldsymbol{x}_{i_+}$ and $\boldsymbol{x}_{i_-}$ denote the samples in each class, $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$ are the class-conditional mean feature vectors and $\boldsymbol{\Sigma}_{reg}$ is the shrunk pooled covariance matrix of the data.

Training this classifier reduces to estimating the class-conditional mean feature vectors and covariance structure. In supervised LSR these parameters are set to sample estimates on a labelled dataset, e.g. recorded during a calibration session. As the formula above shows, the labels are required to separate the data and compute the class-conditional means. We have described two alternative methods for ERP-BCI that estimate the mean target response $\boldsymbol{\mu}_+$ and non-target response $\boldsymbol{\mu}_-$ without label information. In this way, the calibration session can be avoided for ERP-BCI.

First of all, we detailed the unsupervised decoding method by Kindermans et al. (2012, 2014b) in Chapter 2. It uses a pseudo-generative model to describe the ERP application and applies the EM algorithm to find a maximum likelihood estimate (MLE) for the weight vector $\boldsymbol{w}$. The vector is initialised randomly with a multivariate normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and updated during the subsequent expectation and maximisation steps at the end of each trial. The update equation for $\boldsymbol{w}$ at the end of a trial was presented in Chapter 2 and is repeated here:

$$\hat{\boldsymbol{w}} = \sum_{\boldsymbol{c}} p(\boldsymbol{c}|\boldsymbol{X}, \boldsymbol{w}, \beta)\left(\boldsymbol{X}^T\boldsymbol{X} + \frac{\alpha}{\beta}\boldsymbol{I}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}(\boldsymbol{c})$$

This update equation was interpreted as a weighted sum of regularised LSR classifiers, one for each possible assignation of the vector $\boldsymbol{c}$ that contains the predicted target symbols for all observed trials. The weight of each classifier is the likelihood of $\boldsymbol{c}$, given the observed data and the current estimate for $\boldsymbol{w}$.

With centred data and rescaled labels, this equation can again be rewritten to make the estimates of the class-conditional means clearly

visible. The complete derivation is given in Appendix B. The resulting update equations for $\boldsymbol{\mu}_+$, $\boldsymbol{\mu}_-$ and $\boldsymbol{w}$ at the end of a trial are:

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_+ &= \frac{1}{N_+} \sum_{t,i} \left(\sum_{c_t \in i_t} p(c_t \mid \boldsymbol{X}_t, \boldsymbol{w}, \beta) \right) \boldsymbol{x}_{t,i} \\
\hat{\boldsymbol{\mu}}_- &= \frac{1}{N_-} \sum_{t,i} \left(\sum_{c_t \notin i_t} p(c_t \mid \boldsymbol{X}_t, \boldsymbol{w}, \beta) \right) \boldsymbol{x}_{t,i} \\
\hat{\boldsymbol{\Sigma}} &= \frac{\boldsymbol{X}^T \boldsymbol{X}}{N} \\
\hat{\boldsymbol{w}} &= (\hat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{I})^{-1} (\hat{\boldsymbol{\mu}}_+ - \hat{\boldsymbol{\mu}}_-)
\end{aligned}$$

where λ replaces the ratio α/β as regularisation constant and $\hat{\boldsymbol{\Sigma}}$ is the estimate of the pooled covariance matrix. The index t iterates over all recorded trials and i over the stimuli recorded in that trial. The means are estimated as a weighted sum of the responses $\boldsymbol{x}_{t,i}$ recorded in all trials observed so far. The weight given to a response in the estimate of $\hat{\boldsymbol{\mu}}_+$ is the probability that this response is target according to the current classifier. This procedure of maximising the expected data likelihood makes the classifier performance highly dependent on the random initialisation of the parameters.

In addition, the LLP method was introduced in Chapter 4. It estimates the class-wise mean ERP responses by computing the sample mean in two separate groups of data with different target/non-target ratios and solving a linear system. The formulas obtained in Chapter 4 are repeated here:

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_+ &= \phi_+^1 \frac{1}{N_1} \sum_{t,i \in S_1} \boldsymbol{x}_{t,i} + \phi_+^2 \frac{1}{N_2} \sum_{t,i \in S_2} \boldsymbol{x}_{t,i} \\
\hat{\boldsymbol{\mu}}_- &= \phi_-^1 \frac{1}{N_1} \sum_{t,i \in S_1} \boldsymbol{x}_{t,i} + \phi_-^2 \frac{1}{N_2} \sum_{t,i \in S_2} \boldsymbol{x}_{t,i} \\
\hat{\boldsymbol{\Sigma}} &= \frac{\boldsymbol{X}^T \boldsymbol{X}}{N} \\
\hat{\boldsymbol{w}} &= (\hat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{I})^{-1} (\hat{\boldsymbol{\mu}}_+ - \hat{\boldsymbol{\mu}}_-) \,.
\end{aligned}$$

In contrast to the EM-method, the class-wise mean estimates are weighted sums with fixed coefficients that do not depend on any ran-

dom initialisation of parameters.

Both unsupervised decoding methods use the same update equation for the classifier weight vector but have a different approach in estimating the class-conditional mean ERP responses. In the previous chapter it was shown that this results in complementary classification performance during an experiment with the visual ERP speller. While EM has the potential to learn very fast, it is less robust than LLP. On the other hand, the reliable LLP decoder learns slower and is therefore less effective compared to a well-initialised EM decoder.

5.2 Mixing estimations of the mean response

We propose a method to combine two unsupervised estimators for the class-conditional means in a theoretical way. The new estimator is proposed as a mixture of estimators:

$$\hat{\boldsymbol{\mu}}(\gamma) = (1-\gamma)\hat{\boldsymbol{\mu}}_A + \gamma\hat{\boldsymbol{\mu}}_B$$

were $\hat{\boldsymbol{\mu}}$ denotes the new estimator of the mean target or non-target response, $\hat{\boldsymbol{\mu}}_A$ and $\hat{\boldsymbol{\mu}}_B$ denote existing estimators and γ is the mixing coefficient. This method is generally applicable to any mixture of two mean estimation methods. The method will be evaluated in Section 5.3 for the mixture of the EM estimator $\hat{\boldsymbol{\mu}}_A = \hat{\boldsymbol{\mu}}_{EM}$ and the LLP estimator $\hat{\boldsymbol{\mu}}_B = \hat{\boldsymbol{\mu}}_{LLP}$. Our final goal is to obtain a new estimator for the class-conditional mean response that is as effective as a well-initialised EM decoder and as reliable as a LLP decoder.

5.2.1 Optimal mixing coefficient

Inspired by the concept of mean shrinkage for supervised classification (Höhne et al., 2016), the optimal mixing coefficient γ^* is obtained as the value that minimises the expected mean squared error between the estimator value $\hat{\boldsymbol{\mu}}$ and the unknown true parameter value $\boldsymbol{\mu}$:

$$\gamma^* = \arg\min_{\gamma} E\left[\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}(\gamma)\|^2\right]$$

The mathematical derivation can be found in Appendix B. The result is:

$$\gamma^* = \frac{1}{2}\left(\frac{\sum_d Var\left[\hat{\mu}_{A,d}\right] - \sum_d Var\left[\hat{\mu}_{B,d}\right]}{\|\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B\|^2} + 1\right) \tag{5.1}$$

Here, $Var\left[\hat{\mu}_{A,d}\right]$ denotes the variance on the estimation of the d$^{\text{th}}$ component of μ by estimator A. Because the LLP method has closed-form expressions for the mean response estimation, the variance on this estimator can be extracted directly from the variance on the data. In contrast, there is no closed-form expression for the maximum likelihood estimation calculated with EM. Nevertheless, under the regularity conditions of the likelihood function $L\left(\boldsymbol{\mu} \mid \boldsymbol{X}\right)$ (DuMouchel, 1973), which are met, the MLE approaches a normal distribution when more data is obtained:

$$\hat{\boldsymbol{\mu}}_{EM} \sim \mathcal{N}\left(\boldsymbol{\mu}, \{\boldsymbol{I}(\boldsymbol{\mu})\}^{-1}\right)$$

The variance can be derived from the Fisher information matrix $\boldsymbol{I}(\boldsymbol{\mu})$ of the parameter $\boldsymbol{\mu}$. Details about the derivation of these estimator variances can be found in Appendix B. Once the variance on both estimators is calculated, the optimal value of the mixing coefficient is obtained with Equation 5.1.

5.2.2 Mean shrinkage

It is expected that the mixture of the EM estimator with any constant vector will improve the robustness of the decoder. This could raise questions about the true added value of LLP in the mixed estimator. To demonstrate the added value of the information embedded in the LLP estimator, we will compare the mixture of EM and LLP with a mixture of EM and a random but constant vector $\boldsymbol{m}$. As the constant vector has no variance, the mixing coefficient will be computed using the formula proposed by James and Stein (1961):

$$\gamma = \frac{\sum_d Var\left[\hat{\mu}_{EM,d}\right]}{\|\hat{\boldsymbol{\mu}}_{EM} - \boldsymbol{m}\|^2}$$

where, in contrast to the original method by James and Stein (1961), $Var\left[\hat{\mu}_{EM,d}\right]$ is an approximation of the expected variance on the estimator instead of a sample statistic based on calibration data (see

Appendix B). The vector $\boldsymbol{m}$ will be drawn randomly from a multivariate normal distribution with zero mean and identity covariance matrix. This comparison will illustrate the complementarity of the information in the LLP and EM estimators.

5.3 Offline evaluation

In this section, we evaluate the mixing method with a simulation of an online spelling experiment. Its performance is compared to the EM and LLP methods in terms of AUC and symbol selection accuracy. First we will explain how the simulation of the spelling procedure mimics the online experiment as closely as possible. Afterwards, we will discuss the simulation results.

5.3.1 Experimental setup

Data collection

We resimulate the online experiment presented in Chapter 4. The preprocessing of the recorded data was described in Section 4.3.1 and repeated here for the reader's convenience. The data is band-pass filtered between 0.5 and 8 Hz with a third order Chebyshev Type II filter and downsampled to 100 Hz. The ERP response signal is taken in the [-200 700] ms interval around the stimulus event and the average amplitude in the [-200 0] ms interval is subtracted as a baseline reference. In contrast to the original online experiment, the response EEG is averaged in each consecutive interval of 100 ms, to obtain 9 features per channel or a total of 279 features per ERP response.

Simulation of the online spelling procedure

To simulate an online experiment, the stimulus responses are processed sequentially. At the end of each trial, the newly observed set of 68 ERP signals is added to the total set of collected data. This extended dataset is centred to zero mean and whitened (features are decorrelated and have unit variance) to avoid high-variant features to

dominate the computation of the mixing coefficient γ. Next, the classifier parameters are updated on this dataset and all the observed data points are classified. The classifier output on the responses from the new trial are aggregated to predict the target symbol. At that point the simulation of the online experiment is paused and the knowledge of the symbols in the predefined sentence is used to evaluate the current classifier. This label information is only used for evaluation, not for training. Consequently, the simulated spelling experiment is always oblivious to label information.

The experiment is simulated for a LSR classifier with means estimated by the LLP method, the EM method and the proposed mixing of these two methods. As explained in Chapter 2, the high variability in the performance obtained with EM is usually tackled by initialising and updating several EM decoders simultaneously (Kindermans et al., 2014b). In the simulation presented here we use a pure EM decoder with a single initialisation. The experiment is simulated 10 times to illustrate the performance achieved with different parameter initialisations. In each run, the EM and mixed decoder are initialised with the same randomly chosen parameter values.

Performance measures

Three measures are used to asses the performance of the decoder during the simulation of the online experiment and in our subsequent offline analysis.

First of all, with AUC we measure the performance in classifying single ERP responses. The classifier is updated at the end of each trial and as such the classification of responses recorded in previous trials can change. For this reason, the current quality of the classifier is measured by computing the AUC on the total set of stimulus responses that have been collected up to this point.

Second, as the final goal of the ERP speller is to spell symbols with high accuracy, we also measure the percentage of symbols that is spelled correctly. Although the continuously updated classifier is able to correct the selection of previously spelled symbols, the user is most concerned with the symbol that is selected at the end of the new trial. For this reason, the symbol selection accuracy is reported as follows.

In each consecutive sub-block of 10 trials we determine the percentage of symbols that are spelled correctly at the end of their corresponding trial. Consequently, a symbol accuracy of 70 % indicates that in the sub-block of interest, 7 out of 10 symbols were selected correctly.

To compare the result obtained in the 10 runs with EM and the mixing method, a Wilcoxon signed-rank test is used. This is a non-parametric test to compare paired samples when the population cannot be assumed to be normally distributed, as assessed by the Lilliefors test. The difference in AUC or symbol accuracy is considered significant if the p-value resulting from this test is smaller than 0.05.

5.3.2 Results and discussion

Online evolution of the AUC

Figure 5.1 shows the evolution of the AUC obtained with LLP, EM and the mixing method during the spelling of the 63 symbols sentence. The number of spelled symbols is shown on the horizontal axis to indicate the amount of recorded data. Subjects S8 and S9 are the two subjects that needed respectively the most and least data before the EM method achieved a higher AUC than LLP. For this reason, these subjects were chosen as specific cases for illustration purposes. Figure 5.1(a) shows the AUC for subject S8 during the first spelling block. The result for EM and the mixture method (MIX) is the median over 10 different runs with randomly initialised classifier weights. The coloured area between the 10 % and 90 % percentile illustrates the variation on the result obtained with random parameter initialisation. The p-value resulting from a Wilcoxon signed-rank test comparing EM with MIX is also given. The point were there is no more variation in one of the methods is marked with an $\times$. The statistical test is not applicable beyond this point. The thin black line shows the threshold for statistical significance ($p = 0.05$). Figure 5.1(b) shows the corresponding result for subject S9. Figure 5.1(c) shows the grand average over all subjects, the three blocks recorded per subject and the 10 different runs.

The figure confirms once again the complementary behaviour of the EM and LLP methods. The difference in classification perfor-

mance stems from the different mean estimation methods as the pooled covariance is the same for all classifiers. It is clear from Figure 5.1(c) that, on average, the LLP method gives a better estimate for lower amounts of recorded data but improves only very slowly when more data is collected. In contrast, EM gives better estimates than LLP for higher amounts of data. This is in accordance with the results demonstrated in the previous chapter and discussed by Hübner et al. (2017). The blue shaded area in Figure 5.1(a) and Figure 5.1(b) shows the variation in EM performance caused by different parameter initialisations. This variation only decreases as more data is collected. In contrast, the LLP method does not depend on any random initialisation and does not show this kind of variation in classification performance.

The average AUC achieved with the mixing method is higher compared to EM and LLP. The dotted line in Figure 5.1(c) shows that the difference with the EM method is statistically significant, as confirmed by the p-value of the Wilcoxon signed-rank test. The p-value in this graph suddenly drops to zero when the result for the majority of the subjects saturates. The saturation level is on average 0.97 for EM and MIX.

The mixing method does more than a simple mutual compensation of the pros and cons found in the two standard methods. On average, the AUC surpasses the best method from the third symbol until saturation. This is a first sign that the information inherent to the two standard estimators is complementary. The mixing method combines this complementary information in a new estimator that is better than any of the estimators it is made from.

In addition to the improved AUC, Figure 5.1(a) and Figure 5.1(b) demonstrate another advantage of the mixing method. The variation in performance is highly reduced compared to EM. Mixing the EM estimator with LLP makes the classifier output far less dependent on the random parameter initialisation compared to the standard EM method. In previous work with the EM decoder, this dependency was tackled by training several classifiers in parallel. The necessary number of decoders is determined empirically and depends on the subject. In Chapter 4 we explained that the selection of the best speller with data likelihood is suboptimal and that the computation

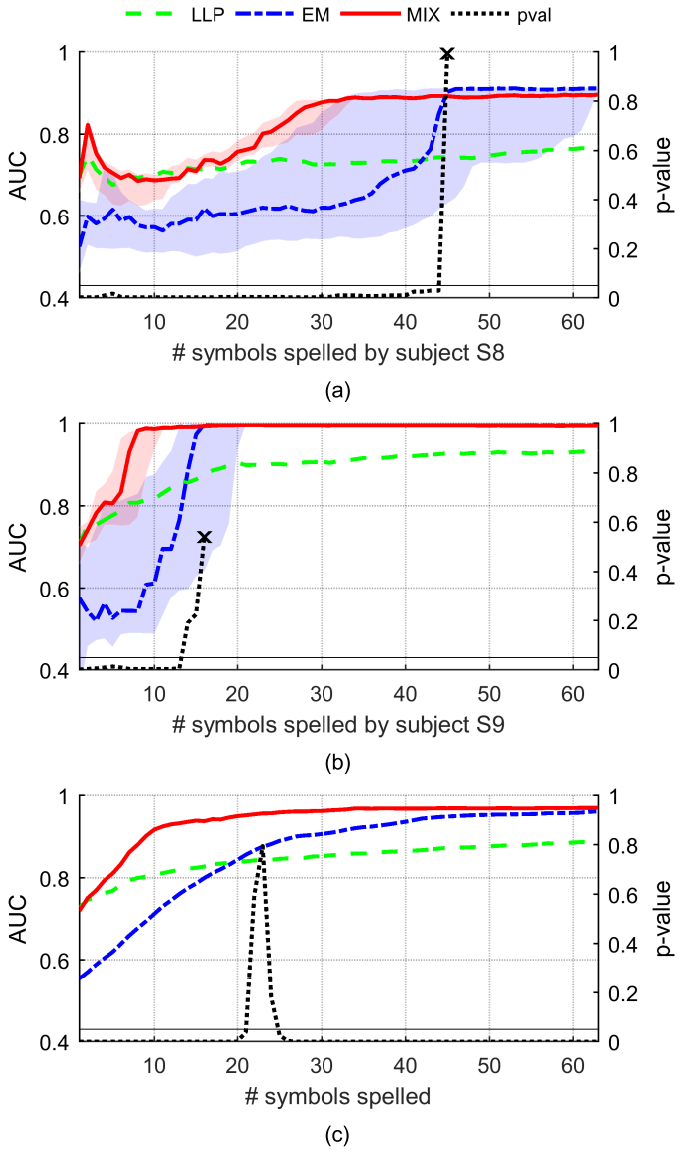

Figure 5.1: AUC obtained during online spelling with the three mean estimation methods. (a) AUC for the first block recorded in subject S8. The result shown for EM and MIX is the median over 10 runs with randomly initialised parameters. The area between the 10 % and 90 % percentile is shaded. Per trial, the p-value of a Wilcoxon signed-rank test compares the results of EM and MIX. (b) Result for the first block recorded in subject S9. (c) Grand average over the 13 subjects and their 3 recorded blocks.

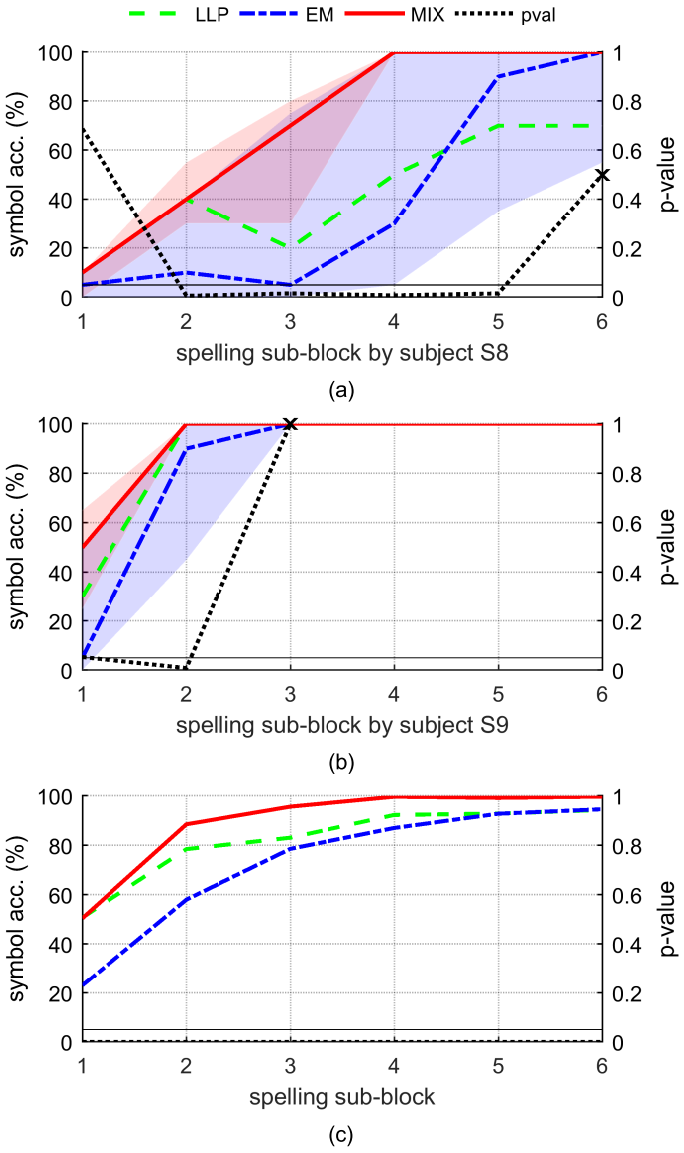

Figure 5.2: Evolution of the symbol accuracy per sub-block of 10 trials. (a) Symbol accuracy for the first block recorded in subject S8. The result shown for EM and MIX is the median over 10 runs with different parameter initialisations. The area between the 10 % and 90 % percentile is shaded. Per trial, the p-value of a Wilcoxon signed-rank test is given, comparing the results of EM and MIX. (b) Same result for the first block recorded in subject S9. (c) Grand average over the 13 subjects and their 3 recorded blocks.

time between the presentation of the last visual stimulus and the feedback of the selected symbol is prolonged. With the mixing method this is no longer necessary, which makes online application of our novel self-learning decoder more efficient.

Online evolution of the symbol selection accuracy

Figure 5.2 displays the evolution in symbol selection accuracy, measured per sub-block of 10 trials. Again, the mixing method proves to be superior to the two standard methods, on average as well as for the two specific subjects illustrated. By mixing estimators, an average symbol accuracy of 89.7 % is achieved in the second sub-block compared to 79.2 % and 57.5 % for LLP and EM respectively. The result is even more remarkable for subject S8. The two standard methods give a very low symbol accuracy in the third sub-block (LLP: 20 %, EM: 5 %) while the mixing method classifies 70 % of the symbols correctly. The lower variance on the results again illustrates the reduced dependency on parameter initialisations compared to EM.

Evaluation of the estimated mean ERP

The actual mean estimation performance is shown in Figure 5.3 for the specific example of subject S1. The true and estimated mean target response after the spelling of the 27th symbol are presented in scalp maps for the different estimation methods. The maps in the top row demonstrate the supervised estimate, computed with label information. The LLP estimate shows some resemblance with the true mean but can clearly still improve. The EM estimate again shows to be dependent on the initialisation. The estimate that resulted in the lowest and highest AUC is given. Although LLP and the badly initialised EM estimate are still far from the true mean response, the mixing of these estimators is very close to the true mean as can be seen in the fifth row scalp plots of Figure 5.3. This illustrates once more the remarkable performance of our method.

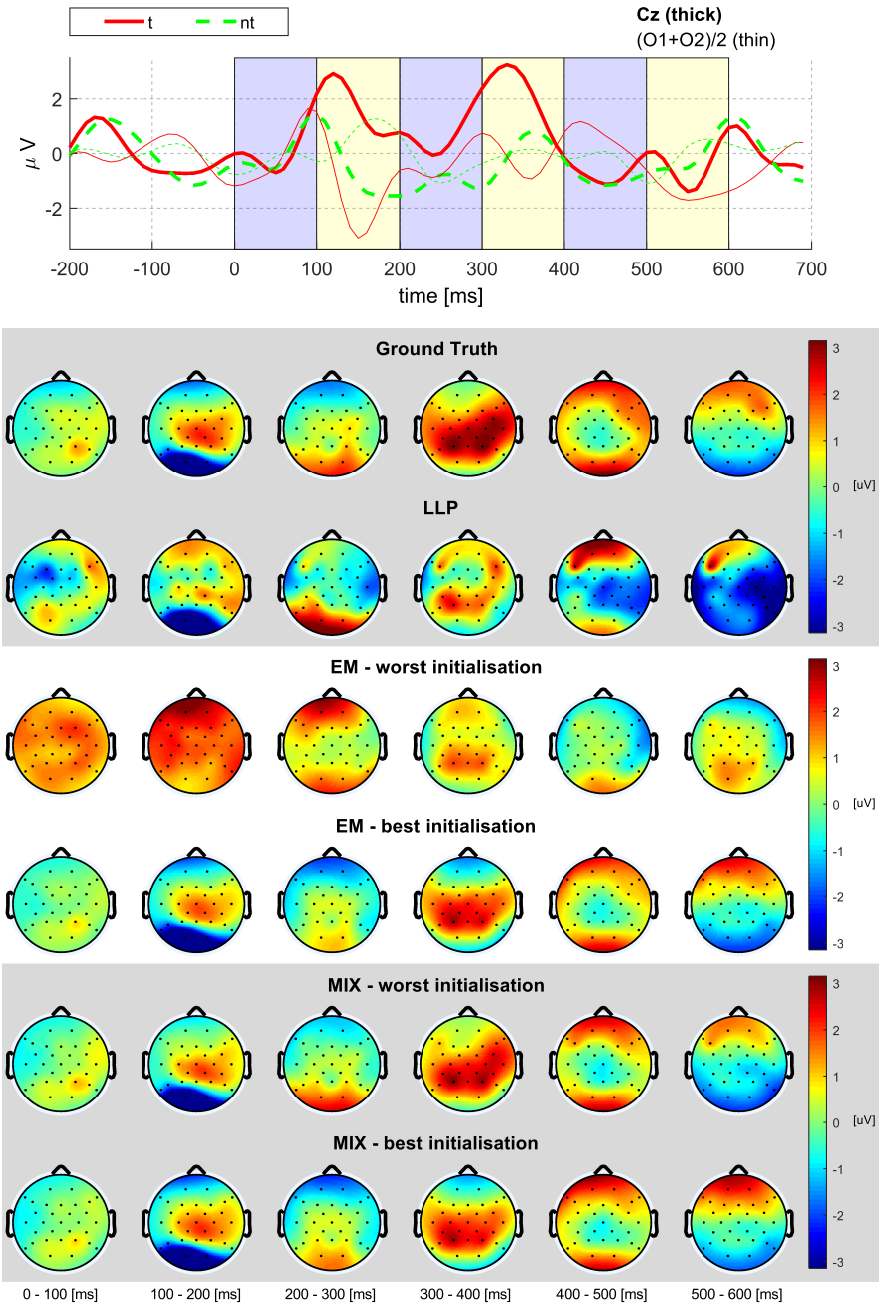

Figure 5.3: Evaluation of the mean response estimation for subject S1 after 27 spelling trials. The top graph shows the averaged target and non-target response as measured in the Cz electrode and the occipital electrodes. The blue and yellow shaded areas indicate the intervals in which the amplitude is averaged to obtain the scalp maps below. The first row illustrates the ground truth. Other rows show the estimation of this target response with the unsupervised LLP, EM and mixing method.

Interpretation of the optimal mixing coefficient

Figure 5.4 shows the evolution of the optimal mixing coefficient for both the mean target and non-target response estimation as a function of the number of symbols spelled. Recall that a γ value of 1 corresponds to pure LLP estimation while a value of 0 corresponds to pure EM estimation.

As the two estimators behave differently when more data is collected, we expect the mixing coefficient γ to change drastically during the spelling procedure. As shown in Figure 5.4, the value of γ drops quickly as more data is recorded. This confirms once more that mixing with the LLP estimator is especially effective for very low amounts of data, where the EM estimator performs badly. The figures show nearly no variance over different parameter initialisations. This low variation is also observed in the evolution of the AUC and spelling accuracy discussed in the previous section.

One could expect the value of γ to converge to zero as a maximum likelihood estimate is known to have the least variance over all unbiased estimators when an unlimited amount of data is available. However, formula 5.1 shows that the value to which γ converges is determined by how fast the difference in variance decreases compared to how fast the norm of the estimator difference decreases. γ only converges to zero when this norm decreases faster. It is clear from Figure 5.4 that this is not the case. However, EM only optimises a lower bound on the likelihood and does not optimise for maximum likelihood directly. The fact that EM has the least variance only ensures that the convergence value will be lower than 0.5. Consequently it is hard to interpret the evolution of the γ value as more data is collected.

In Figure 5.5, the result obtained with a fixed value $\gamma \in [0, 0.1, 0.2, 0.5, 0.7, 1]$ is shown. Again, $\gamma = 0$ coincides with the EM method and $\gamma = 1$ with the LLP method. The figure shows that the proposed formula indeed finds the best value for the mixing coefficient. A value that is too low shows lower performance at the beginning of the spelling procedure. On the contrary, a value that is higher yields a lower AUC later in the spelling procedure. For the individual subjects shown, the optimal mixing coefficient is slightly over- or un-

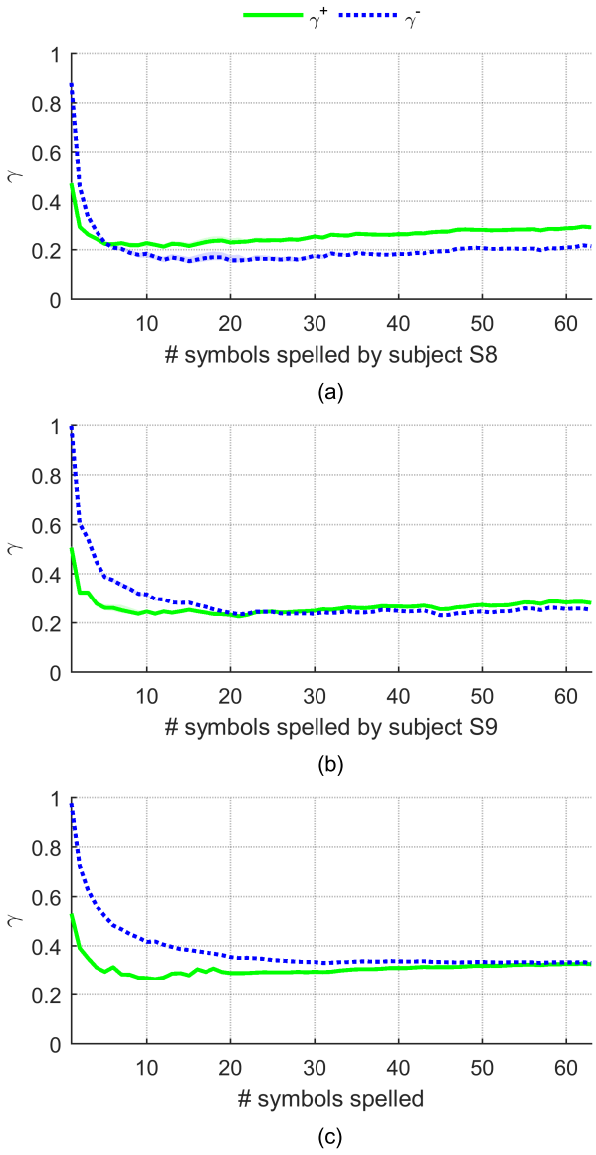

Figure 5.4: Evolution of the optimal mixing coefficient for the estimation of the mean target and non-target response. A γ value of 1 corresponds to pure LLP estimation while a value of 0 corresponds to pure EM estimation (a) Mixing coefficient for the first block recorded in subject S8. The result shown is the median over 10 runs with randomly initialised parameters. The area between the 10 % and 90 % percentile is shaded but not visible as there is almost no variation. (b) Result for subject S9. (c) Grand average over all subjects and spelling blocks.

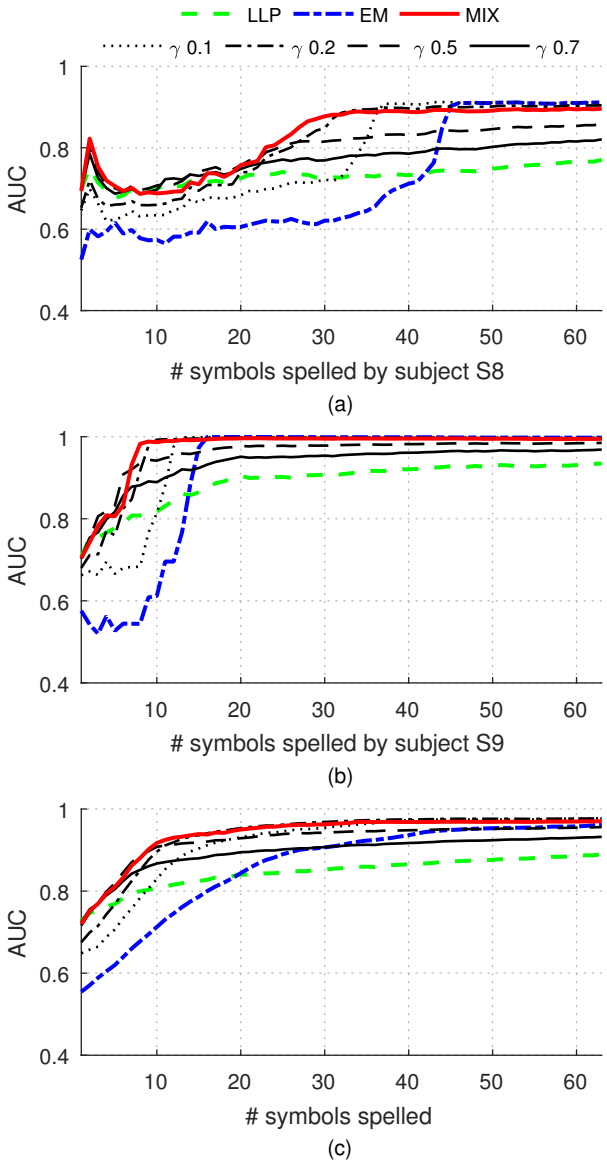

Figure 5.5: Evolution of the AUC during onling spelling with the mixing method. The result is shown for the mixing coefficient $\gamma \in [0(EM), 0.1, 0.2, 0.5, 0.7, 1(LLP)]$ and the mixing coefficient as determined by the proposed formula. (a) The result for subject S8, median over 10 runs. (b) Same result for subject S9. (c) Grand average over 13 subjects and 3 recorded blocks.

derestimated at the beginning of the spelling procedure. A potential cause for this small error is the estimation of the variance on the EM estimator, see Appendix B.

The added value of mixing with LLP

Figure 5.6 compares the evolution of the AUC for the mixture of the EM method with LLP and the mixture of EM with a constant vector $\boldsymbol{m}$ (FIX), sampled from the standard multivariate normal distribution. The p-value resulting from a Wilcoxon signed-rank test comparing MIX with FIX is also given. The point were there is no more variation in one of the methods is marked with an $\times$. The statistical test is not applicable beyond this point. The thin black line shows the threshold for statistical significance ($p = 0.05$). This figure is used to illustrate the added value of the LLP estimate in the mixing method.

Mixing the EM estimator with a random vector results in a slight improvement for some subjects but this is significantly lower than mixing with the LLP method as illustrated by the p-value reported in Figure 5.6. This shows that the LLP and EM estimators are indeed complementary in the information they provide about the class-wise means.

Comparison with supervised LSR classification

We also perform a simulation with a supervised classifier. This classifier is trained on $N \in [5, 10, 20, 30]$ trials of data. For each subject and each of the three spelling blocks per subject, the first N trials and their corresponding label information are used to train the supervised classifier. This classifier is then applied to the remaining trials of the spelling block. The supervised classifiers are compared to the unsupervised mixing method in terms of AUC (Figure 5.7) and symbol selection accuracy (Figure 5.8). The supervised classification performance is not reported for the first N trials that are used for training. This represents the effect of a true calibration procedure in which the user is required to follow the spelling procedure without the capability of actually spelling the symbols he/she desires.

The AUC of a supervised LSR classifier is compared to the mixture method in Figure 5.7. From the figure it is clear that we have included

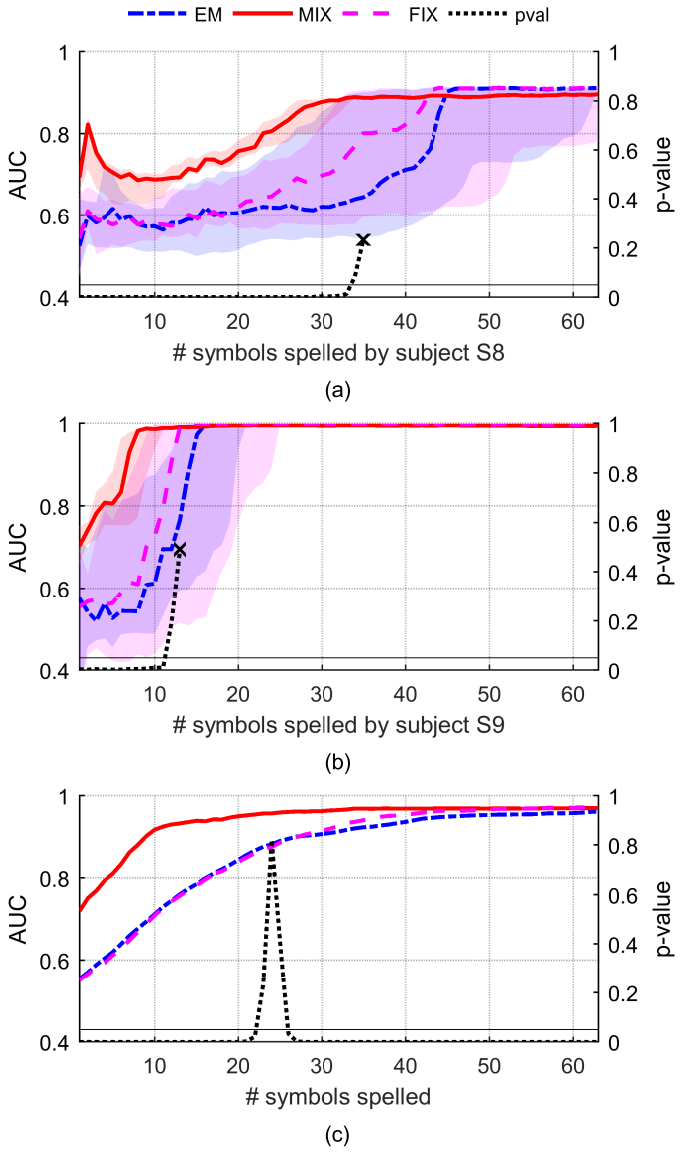

Figure 5.6: AUC during online spelling with the EM method and mixture of EM with LLP (MIX) and a random constant vector (FIX). (a) AUC for the first block recorded in subject S8. The result shown is the median over 10 runs with randomly initialised parameters. The area between the 10 % and 90 % percentile is shaded. Per trial, the p-value of a Wilcoxon signed-rank test is given, comparing the results of MIX and FIX. (b) Result for subject S9. (c) Grand average over all subjects and spelling blocks.

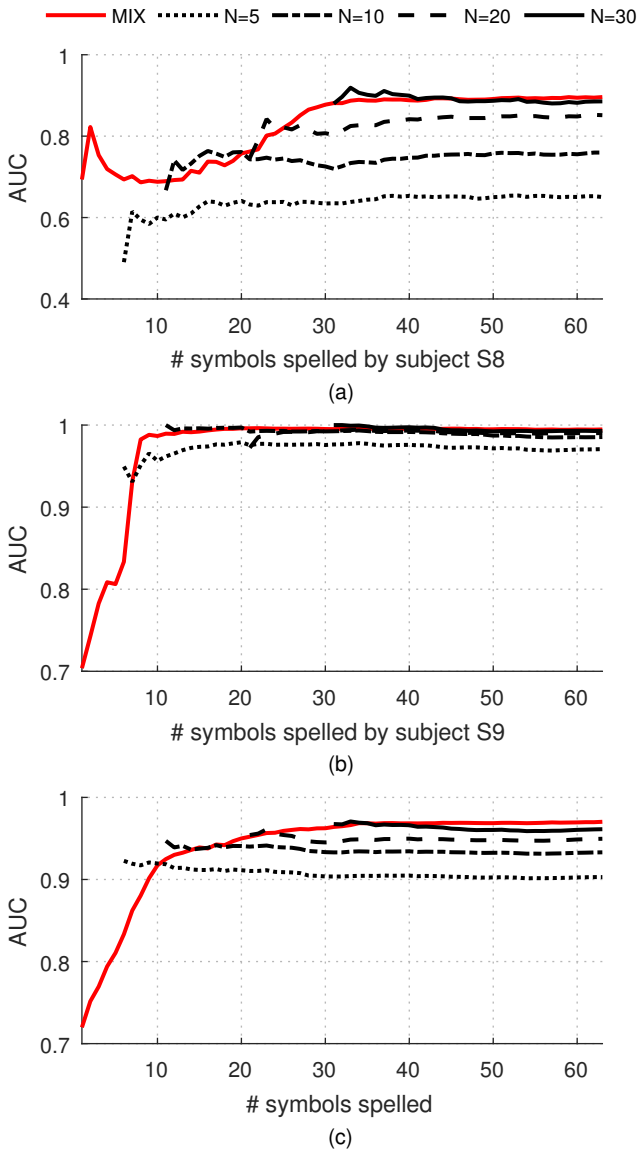

Figure 5.7: Evolution of the AUC during simulated online spelling with the mixing method and a supervised LSR classifier trained on the first N trials of data, $N = [5, 10, 20, 30]$. (a) AUC for the first block recorded in subject S8. The result shown for MIX is the median over 10 runs. (b) Same result for the first block recorded in subject S9. (c) Grand average over the 13 subjects and their 3 recorded blocks. The scale on the vertical axis is altered for better visual inspection.

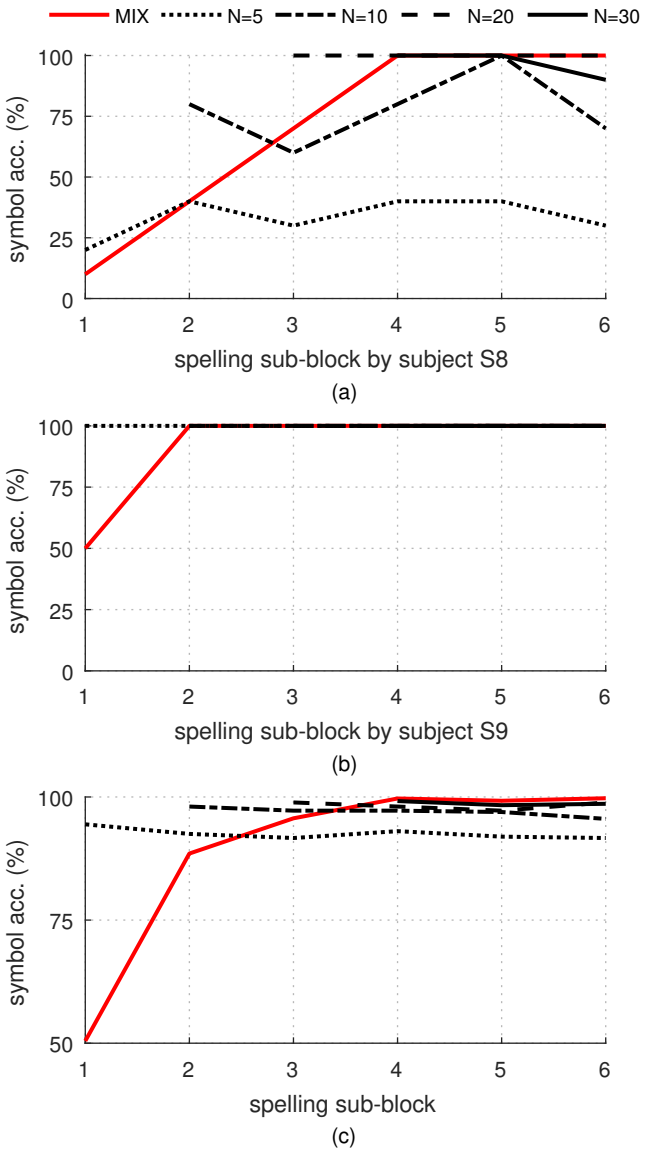

Figure 5.8: Evolution of the symbol accuracy during simulated online spelling with the mixing method and a supervised LSR classifier trained on the first N trials of data, $N = [5, 10, 20, 30]$. (a) symbol accuracy for the first block recorded in subject S8. The result shown for MIX is the median over 10 runs. (b) Same result for the first block recorded in subject S9. (c) Grand average over the 13 subjects and their 3 recorded blocks. The scale on the vertical axis is altered for better visual inspection. Note that for N=5 the supervised symbol accuracy is tested on the last 5 symbols in the first sub-block.

good, as well as bad subjects. For subject S9, training the classifier on 10 symbol spelling trials suffices to obtain an accuracy similar to the one of the unsupervised classifier. For the low-performing subject S8, data worth 30 symbols is required to make the supervised classifier perform as well as the unsupervised classifier. Remark again that the unsupervised mixing method achieves these results without using any label information and that using the unsupervised approaches, it is possible to productively use the BCI, which is not the case during a calibration session.

On average, the AUC curve obtained with the unsupervised mixing method is close to a convex hull around the curves produced by the supervised classifiers. This means that, with our mixing method, the online observed unlabelled data is almost as valuable as labelled calibration data. Furthermore, a slight decrease can be noticed in the AUC of the supervised classifiers as more symbols are spelled. This may be due to non-stationarity effects in the recorded EEG, e.g. when background activity changes or when the elicited ERP response changes over time (Shenoy et al., 2006; Von Bünau et al., 2009). The unsupervised classifier that is continuously updated with newly recorded data adapts to these changes. On the contrary, the supervised classifier is trained on a calibration set of data and does not adapt during the experiment. The same conclusions can be drawn from the symbol selection accuracy reported in Figure 5.8.

5.3.3 Summary

We have presented the results of a simulation that mimics an online spelling procedure as close as possible. While the actual online experiment can only be performed with one type of classification approach, the simulation allowed us to compare three approaches on the same data. Having used data of 13 subjects, we were able to estimate mean performances reliably and thus compare our novel classification approach with the existing ones. We found that the proposed theoretical method of mixing the LLP and EM estimation of the class-conditional mean ERP response yields an improved estimator that adopts the benefits of both methods. A classifier using this new estimator was found to decode more effectively in terms of AUC and

symbol selection. Besides that, the new classifier shows less variance on the results and as such is also more reliable compared to EM.

5.4 Online evaluation

In this final section, the mixing method is validated in an online setting. This experiment was conducted at the University of Freiburg in collaboration with David Hübner and Michael Tangermann. The goal of this study is to prove that our novel method is truly applicable in a practical BCI spelling system. It is compared to the original LLP and EM method for decoding stimulus responses.

As described in Chapter 2, the original EM-based decoding method as proposed by (Kindermans et al., 2012, 2014b) uses the following two techniques to improve decoding performance. First of all, the shrinkage coefficient of the pooled covariance is not calculated with the analytical formula by Ledoit and Wolf (2004) but estimated during the EM iterations. Second, the original EM method uses five pairs of classifiers that are updated in parallel (Kindermans et al., 2014b). The classifier with the highest data likelihood selects the target symbol. These modifications are applied in the current experiment to compare the mixing method to the original EM method. For the mixing method, only one randomly initialised classifier is used as the simulations showed that the new method does not need this trick for reliable decoding.

5.4.1 Experimental setup

The experiment is very similar to the one described in Chapter 4. Six healthy subjects (3 male, 3 female), aged between 22 and 31, were asked to spell the following sentence of 35 symbols: "FRANZY JAGT IM TAXI DURCH FREIBURG ". The spelling procedure was repeated for the three classification methods: LLP, EM and the mixing method. To reduce order effects, the 6 subjects used the three classifiers in different orders. Each possible order was used by one subject. The spelling procedure, EEG recording and feature extraction were the same as explained in Chapter 4. The EEG study was approved by the Ethics

Committee of the University Medical Center Freiburg. Subjects gave written informed consent prior to the beginning of the experiment.

5.4.2 Results and discussion

Figure 5.9 shows the evolution of the three unsupervised decoders during the spelling procedure. As each decoding method is used only once per subject, there is no variance over different initialisations to be reported as was done in the offline analysis. The results confirm the conclusions from the offline simulated experiment. The mixing method outperforms the LLP and EM method both in terms of AUC and symbol selection accuracy.

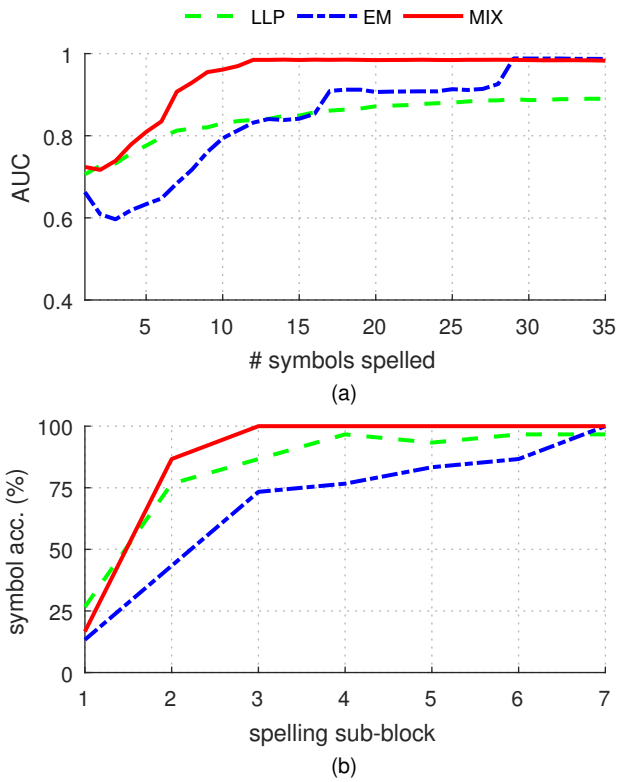

Figure 5.9: Performance of the three unsupervised classification methods during an online spelling procedure. (a) AUC. (b) symbol accuracy per sub-block of five trials. Results are averaged over the 6 subjects.

There is one important performance metric that we have not discussed for our new self-learning decoder: the speed of spelling. As the paradigm is adapted to the requirements of the LLP-based decoder, it is interesting to examine how this affects the speed of spelling. In Chapter 3, the spelling speed was assessed for the original RC, CB and SP ($n = 10$) paradigms as the number of symbols correctly re-spelled per minute (CSM). In the current experiment, a CSM of 2.4 symbols per minute is obtained on average. This result is not directly comparable to the CSM obtained in Chapter 3 due to the difference in experimental setup, stimulus onset asynchrony and subjects between both experiments. Nevertheless, it should be noticed that the interleaved paradigm potentially influences the speed that can be obtained. First of all, the interleaved paradigm has an iteration length $n = 34$ with $r = 8$ target stimuli per iteration, resulting in a target/non-target ratio of 0.235. This is higher than the 0.2 target/non-target ratio obtained with SP. Hence, with the interleaved paradigm, less stimuli need to be presented in order to record the same number of target responses compared to the original SP, RC and CB paradigms. However, when a dynamic stopping technique is employed to speed up the spelling, the interleaved sequence can only be stopped after every iteration of 34 stimuli. In contrast, the original SP paradigm can be stopped at every 10 stimuli. This limits the maximum speed of spelling that can be obtained with the interleaved paradigm. The stopping resolution can be increased by interleaving sequences of shorter lengths, e.g. a sequence with parameters $(n = 12, r = 2)$ and one with $(n = 8, r = 3)$. In that case, there is less contrast in the proportions of target stimuli between the two sequence types, which in turn potentially reduces the accuracy of the LLP-based decoder. Future work will determine the impact of these alternative paradigm settings on the accuracy and spelling speed that can be obtained with our mixture method.

5.5 Conclusion

In this chapter we proposed a method to combine the benefits of different calibrationless decoding methods by mixing their estimations

of the class-conditional means in a theoretical way. An analytical formula was determined to calculate the optimal mixing coefficient. The method was applied to the unsupervised classification methods based on EM and LLP for ERP-based BCI.

We have conducted a simulation of an online experiment with 13 subjects to compare the basic LLP- and EM-based decoding to their mixture when applied in a visual ERP speller. The mixing method outperforms both basic methods in terms of AUC and symbol selection accuracy. Furthermore, the results obtained with this novel decoder vary less compared to EM and as such make the spelling performance more reliable. Finally, we have performed an online experiment on six subjects which confirmed these results.

This concludes our symbiotic design of the ERP-speller. The unsupervised decoder learns from scratch and tunes its parameters to the recorded brain activity, thereby avoiding tedious calibration sessions. It combines the effectiveness of the EM-based decoder with the reliability of the LLP-based decoding method. For that purpose, the application was tuned to the requirements of the decoder by means of our flexible paradigm. The obtained BCI system is effective, efficient, reliable and easy to use.

6

Automated diagnosis of temporal lobe epilepsy

In previous chapters we successfully applied machine learning to decode information from recorded brain activity in ERP-based BCI. By classifying the response on external stimuli as target or non-target, we were able to infer the user's intention. We faced several challenges in this classification task. The data that was available to tune the classifier was scarce, noisy and had a high number of features.

In this chapter, we apply machine learning in a completely different and relatively new type of BCI: an automated diagnosis system. In collaboration with the University and Hospital of Geneva we develop a tool that diagnoses subjects with temporal lobe epilepsy (TLE) in a data-driven way. In addition, it determines if the epileptogenic zone is located in the left or right hemisphere. While this is a completely different application of decoding brain activity, we face the same challenges as before: the available data is scarce, noisy and has a high dimensionality.

Mesial temporal lobe epilepsy (TLE) is the most common type of epilepsy in adults that cannot be treated with anti-epileptic drugs. These patients are candidates for surgery. In order to localise the epileptogenic zone, EEG is recorded to identify the origin of pathological activity such as seizures or interictal epileptiform discharges (IED). However, in some patients, epileptic phenomena are infrequent or completely absent in the recorded EEG. Furthermore, the long-term EEG monitoring is very expensive and stressful for the patient. Patients that are candidates for epilepsy surgery could benefit greatly from a system that diagnoses and lateralises TLE from short scalp

EEG recordings in the absence of visible pathological activity.

Epilepsy is increasingly recognised as a network disease (Laufs, 2012). The functional relationships between the activity in different brain regions could help to better understand epileptic networks. Directed functional connectivity estimates the information transfer between brain regions and the directionality of it. Several studies have shown that directed functional connectivity measures, based on intracranial EEG, can help to identify the epileptogenic zone (Wilke et al., 2009; van Mierlo et al., 2013, 2014). Furthermore, directed functional connectivity applied to brain sources estimated from high-density scalp EEG has revealed interictal network patterns concordant with cognitive deficits in TLE (Coito et al., 2015) and significant connectivity differences in TLE compared to healthy controls in the absence of interictal spikes (Coito et al., 2016).

Machine learning algorithms have been used for automatic detection and localisation of the epileptogenic zone in TLE using a multitude of imaging modalities (Focke et al., 2012; Kerr et al., 2013; Cantor-Rivera et al., 2015; Chiang et al., 2015; Yang et al., 2015; Kamiya et al., 2016). However, no study has attempted to automatically diagnose and lateralise TLE using scalp EEG.

In this chapter, we present a diagnostic and lateralisation classification system for TLE in the absence of visible epileptic activity. For that purpose we use EEG-derived directed functional connectivity values. The next section describes how brain activity was recorded and how the functional connectivity values were computed. We explain the random forest technique for classification and how it is used to select the subset of connectivity values that contains relevant information. Finally, we present the results of our classification system and compare them to previous studies using other imaging modalities.

6.1 Materials and methods

6.1.1 Participants

Our database included 20 LTLE patients, 20 RTLE patients and 35 healthy subjects. Patients were retrospectively selected from the high-

density EEG database of the University Hospital of Geneva, University Hospital of Bern and Paracelsus Medical University in Salzburg according to the following inclusion criteria: drug-resistant TLE, unilateral anteromedial localisation of the epileptogenic zone confirmed by good surgical outcome (Engel's class I or II), intracranial EEG or concordant presurgical evaluation methods and the existence of at least a 10-15 minutes resting-state eyes-closed high-density EEG recording (256 channels). All patients had interictal activity on long-term EEG concordant with the diagnosis of unilateral TLE. Most of them had extensive presurgical evaluation including ictal video-EEG, PET, SPECT and electric source imaging. The patients' dataset used in this study was the same as reported in previous work by Coito et al. (2016). The clinical details can be found in Appendix C.

All patients were evaluated in the epilepsy units of the respective hospitals. The three local ethical committees approved this study. Written informed consent was obtained from all participants.

6.1.2 Computation of directed functional connectivity

EEG recording and preprocessing

Subjects underwent a resting-state eyes-closed recording using an EEG system (Electrical Geodesics system) with 256 electrodes. The facial electrodes as well as the electrodes on the neck were removed since those usually contain artefacts from facial muscle movements and lower impedances. A total of 204 electrodes were used for further analysis. The signals were filtered offline between 1 and 100 Hz and then downsampled to 250 Hz. The analysed signals and topographies were visually inspected and bad channels were interpolated using the 3D splines method, as implemented in the freely available Cartool software (Brunet et al., 2011). Sixty epochs of 1 second during wakefulness, free of artefacts and IEDs, were selected per subject.

Electrical source imaging and selection of regions of interest

The activity of brain sources during the selected EEG epochs was obtained using electrical source imaging as explained in Chapter 1. The forward model was constructed based on a simplified realistic head model using each individual's T1-weighted MRI with consideration of skull thickness (locally spherical model with anatomical constraints, LSMAC (Brunet et al., 2011; Birot et al., 2014)). Around 5000 solution points were equally distributed in the grey matter. A linear distributed inverse solution with biophysical constraints was used to calculate 3D current source density (local auto-regressive averages, LAURA (de Peralta Menendez et al., 2004)).

The grey matter was parcelled into 82 regions of interest (ROI) based on the automated anatomical labelling digital atlas after normalisation to the MNI space using the SPM8 software (Tzourio-Mazoyer et al., 2002). The solution point closest to the centroid of each ROI was considered representative for the source activity in this ROI. This reduced the dimensionality of the solution space. In order to take the time-varying three-dimensional orientation of the source dipoles into account, as well as to obtain a scalar time-series from the 3D dipole time-series, these were projected onto the predominant dipole direction of each ROI over all epochs (Coito et al., 2015; Plomp et al., 2015). This procedure resulted in 82 time-series representing the activity of each individual ROI during the 60 selected epochs.

Directed functional connectivity

Directed functional connectivity is commonly assessed using the concept of Granger-causality. A signal is said to Granger-cause another signal if the knowledge of the past of the former reduces the prediction error of the present of the latter (Granger, 1969). One of the multivariate approaches to estimate brain connectivity in the frequency domain using the concept of Granger-causality is partial directed coherence (PDC) (Baccalá and Sameshima, 2014). PDC estimates the directional and direct interactions between all signals in a multivariate process. It is computed using multivariate autoregressive models of an appropriate order, which simultaneously model multiple time-series,

in this case the source signals obtained in the 82 ROIs.

We used a multivariate autoregressive model of order 10, corresponding to 40 ms of the signal, in concordance with previous studies with similar epoch length and sampling frequency (Astolfi et al., 2008; Coito et al., 2016). The model coefficients were computed using the Nutall-Strand algorithm (Marple, 1987; Schlögl, 2006). We computed the squared PDC normalised with respect to the inflows and then scaled the results by weighting with the normalised spectral power of the source region (weighted PDC, wPDC) (Astolfi et al., 2006; Plomp et al., 2014). To obtain the spectral power we computed the fast Fourier transform for each electrode, applied source imaging to the real and imaginary part of the Fourier transform separately and then combined them to avoid frequency doubling (Frei et al., 2001; Koenig and Pascual-Marqui, 2009; Coito et al., 2015). The mean spectral power was obtained for each patient and scaled (0-1, in the same way as PDC) across ROIs and frequencies (1-40 Hz). In this way, we used the spectral power of the signal to weigh the connectivity matrices (Plomp et al., 2014). Given the 20 mm spatial accuracy of electrical source imaging for localising interictal epileptic activity, the outflows seen in the amygdala, hippocampus and parahippocampal gyrus should not be considered strictly independent but rather globally as medial temporal lobe activity (Mégevand et al., 2014).

For each subject, we obtained a 3D connectivity matrix (82 regions x 82 regions x frequency), which represents the outflow from one region to another for each frequency. For further analysis, we reduced the connectivity matrix to 3 frequency bands: theta (4-8 Hz), alpha (8-12 Hz) and beta (12-30 Hz), by calculating the mean connectivity value in each band.

6.1.3 Feature selection and classification

Random forests for classification

Random forests (RF) (Breiman, 2001) is a machine learning technique in which an ensemble of elementary classifiers is trained and its outputs aggregated to classify a new input sample. In RF, the ensemble is composed of many classification or regression trees (Loh, 2011), each

trained on a different bootstrap subset of the available samples. When a new input is to be classified, each tree in the ensemble makes the classification and the sample is assigned to the class that was chosen by the majority of the trees.

The advantage of using RF as classification technique for computer-aided diagnosis of neurological diseases is manifold. First, it is known to manage classification problems with a low number of recorded input samples and a high number of feature values per input. As such it has been shown to outperform other classification techniques, such as SVMs and logistic regression, for automated diagnosis (Khalilia et al., 2011; Ozcift and Gulten, 2011). Secondly, the samples that do not appear in the bootstrap subset (called *out-of-bag samples*) can be used to test the trees on unseen inputs without the need for extra samples in a separate test set. This avoids the high cost of recording extra subjects to test the system. Finally, the decision trees and their aggregation by voting make the internal classification mechanism transparent and easy to understand, which is important for integration of automated diagnosis systems in clinical practice.

RF lends itself as an ideal technique for the selection of relevant features. The performance of the forest on the out-of-bag subjects can be used to compute an importance value to each feature, which incorporates the interaction between features. Importance values are used in this study for the selection and interpretation of relevant features, as explained further.

A downside of RF is that its performance is known to suffer from class-imbalance in the dataset (Chen et al., 2004). This is the case in our dataset where we have 40 TLE patients compared to 35 healthy controls. We try to compensate for this limitation by using a slightly adapted version of RF: balanced random forests. This classifier differs from standard RF in the way that subsets containing an equal number of subjects from both classes are used to train the decision trees. The scikit-learn library (Pedregosa et al., 2011) was used to implement the balanced RF classifier. Every forest contained 1000 trees. The size of the random set of features from which splits were chosen was log2(M), where M is the total number of features per subject.

All performance measures reported in this work are calculated in a leave-one-out cross validation (LOOCV). In this procedure, each sub-

ject is left out of the dataset once, while the others are used for feature selection and classifier training. The classifier system is then tested on the left out subject. In this way, the relationship between training and validation data resembles the relationship between training and test data in a true clinical setting (Saeb et al., 2017). The evaluation illustrates the average performance on a new subject, unseen by the system.

The system has three output classes: healthy subject, LTLE and RTLE. Building a three-class classifier with RF is possible but far more complex than building multiple two-class classifiers and combining their results. Moreover, the natural clinical process requires a system in which the subject is first diagnosed with TLE and then, if applicable, the TLE is lateralised. Therefore, we build two separate classifiers, one for diagnosis (TLE vs. healthy subjects) and one for lateralisation (LTLE vs. RTLE). The two classifiers are applied sequentially to obtain the final prediction.

Selection of relevant features

The calculation of the connectivity between every pair of regions in the three frequency bands results in 20.172 features for each individual subject. An optimal subset of these features needs to be selected in order to avoid creating false decision rules when training the classifier on the example data. As an example, consider the case where a certain connection is slightly stronger for RTLE compared to LTLE patients in the majority of our patients, but not for the whole population of TLE patients. A classifier taking this contingency as a general rule for lateralisation can perform poorly on new subjects. This issue of over-fitting to example data was explained in Chapter 2. It becomes more likely with decreasing number of subjects and increasing number of features per subject (Guyon and Elisseeff, 2003; Mwangi et al., 2014). To avoid over-fitting, we allow a maximum of one feature per ten subjects, resulting in a maximum of seven features for diagnosis and four features for lateralisation. This is more a rule of thumb than an optimised decision.

First, the 82 regions are reduced to a set of 14 regions that have shown differences between groups in a previous study (Coito et al.,

2016) or are known to be involved in TLE: left and right hippocampus (Hipp), amygdala (Amyg), parahippocampus (PHipp), anterior cingulate cortex (ACC), posterior cingulate cortex (PCC), olfactory cortex and medial temporal pole (TPMid). This leaves us with 588 features that are used to build a first RF classifier. Next, the feature selection method specifically designed for RF by Genuer et al. (2010) is used to further reduce the number of features. This method selects features based on a measure of their importance for accurate classification. The importance of a feature f in the classifier is calculated as the decrease in classification performance when the values of f are randomly permuted in the dataset. As random permutation breaks the link between the feature f and the class labels, this permutation importance reflects how much classification power is lost when this feature is taken out of the design of the system.

In contrast to the original feature selection method by Genuer et al. (2010), we use the AUC instead of classification accuracy for the evaluation of the classifiers and the computation of feature importance values. As explained in Section 1.2.3, the AUC uses the assigned probability that a subject belongs to a certain class rather than the assigned class itself and as such is a more complete evaluation metric for binary classifiers compared to the classification accuracy (Huang and Ling, 2005).

Following the feature selection method proposed by Genuer et al. (2010), features with an importance value close to zero are considered irrelevant and thus removed from the set. Further reduction is obtained by removing redundant information. For that purpose, the minimal subset of features that contains the maximum amount of discriminant information is selected. The method considers the interaction between features during this selection, which is important as the relevance of an outflow may depend on which other outflows are considered as features. For interpretation of the feature selection result, we calculate the actual interaction effect of a feature f_1 on another feature f_2 as the decrease in permutation importance of f_2 when f_1 is removed from the design (again by permuting its values). A positive interaction indicates that the discriminative information in f_2 is more relevant when f_1 is included in the design. Higher order interactions (e.g. between three features) can also have an impact. However, with

increasing order, more data is required to obtain a reasonably accurate measure of interaction. The first order interaction is computed here to illustrate the impact of feature interaction in general.

In each iteration of the LOOCV procedure, feature selection is done for the diagnosis and lateralisation classifier individually. The classifiers are then trained on the selected set of features and the classification result is reported for the subject that is left out.

6.2 Results and discussion

6.2.1 EEG-based connectivity measures for diagnosis and lateralisation of TLE

Table 6.1 shows the different performance measures obtained with the diagnosis and lateralisation classifiers. The positive class denotes TLE in the case of diagnosis and LTLE in the lateralisation case. The accuracy is the percentage of all subjects classified correctly. The sensitivity is the fraction of positives identified as such while the specificity is the fraction of negatives classified correctly. Predictive values are defined to be complementary. The positive predictive value is the percentage of subjects classified as positives that are truly positive. It illustrates how confident we can be about the classifier's output. AUC was defined before in Section 1.2.3. The diagnosis classifier achieves an accuracy of 90.7 %, sensitivity of 95 %, specificity of 85.7 % and AUC of 0.89. For lateralisation, the AUC is 0.911 and all other performance measures 90 %.

Putting the two classifiers in sequence, Table 6.2 shows the confusion matrix of this three-class classifier system in LOOCV. It illustrates how the subjects from a certain class are assigned to the three classes by our system. Overall, our system classifies 85.3 % of the subjects correctly. The accuracy in classifying LTLE, RTLE and healthy controls was 80 %, 90 % and 85.7 % respectively.

This is the first study showing that functional connectivity using EEG without visible scalp pathological activity can be used for automated diagnosis and lateralisation of TLE with high accuracy. This

Performance measure	Diagnosis	Lateralisation
Accuracy (%)	90.7	90.0
Sensitivity (%)	95.0	90.0
Specificity (%)	85.7	90.0
Positive Predictive Value (%)	88.4	90.0
Negative Predictive Value (%)	93.8	90.0
AUC	0.890	0.911

Table 6.1: Performance of the diagnosis and lateralisation classifiers separately.

		Predicted		
		LTLE	RTLE	Control
Actual	LTLE	**16**	2	2
	RTLE	2	**18**	0
	Control	0	5	**30**

Table 6.2: Confusion matrix for the three-class classification system. Each row illustrates how the subjects from the corresponding class are classified by our two-step system. First, subjects are diagnosed as TLE or healthy control by the diagnosis classifier. Next, those classified as TLE are subsequently lateralised as LTLE or RTLE.

could support TLE diagnosis in patients who do not show IEDs during routine scalp EEG recording. Furthermore, it could constitute a powerful lateralising clinical aid in patients who are candidates for epilepsy surgery, especially in difficult cases where the currently used presurgical evaluation methods are not concordant.

Previous studies have used structural MRI, diffusion tensor imaging (DTI), functional MRI (fMRI) or PET for automated diagnosis and lateralisation of TLE. In Table 6.3 we summarise the techniques, selected features and main findings of these studies. In TLE patients with hippocampus sclerosis (HS), SVMs were applied to T1-weighted images and DTI (Focke et al., 2012). They achieved an accuracy of 100 % for lateralisation and an accuracy of 93 % with a three-way SVM classifier (LTLE vs. RTLE vs. Controls). Excluding the contribution of the hippocampus yielded a lateralisation accuracy of 92 %, comparable to our results, but a lower diagnostic accuracy of 76 %. It is noteworthy that the study solely included TLE patients with HS. However, HS is present only in 65 % of surgical TLE population (Babb et al., 1984). In our study, patients with other types of lesion or even without detectable lesions were included, extending the use of our classifier to a more general population of TLE in which diagnosis can be more difficult. In a FDG-PET study, interictal metabolic changes as input of the classifier (a multilayer perceptron), led to an accuracy of 76 % to simultaneously diagnose and lateralise TLE (Kerr et al., 2013). A SVM applied to graph theory measures obtained from DTI images, achieved an accuracy of 86.4 % and an AUC of 0.91 for lateralisation, but no results were reported for diagnosis (Kamiya et al., 2016). Using features from the T1-weighted MR images, a diagnostic accuracy of 88.9 % was achieved with a linear SVM but no lateralisation result was reported (Cantor-Rivera et al., 2015). Using resting-state fMRI and functional connectivity graph measures, a lateralisation classifier achieving 95.8 % was built on a rather small set of subjects (14 LTLE and 10 RTLE patients) (Chiang et al., 2015). In another study, fMRI-based functional connectivity values and network metrics were used to lateralise TLE on a small cohort of patients (7 LTLE and 5 RLTE) (Yang et al., 2015). A linear SVM for lateralisation gave an accuracy of 83.3 %.

In this work, we obtained comparable or higher accuracies, sensi-

	Subjects			Imaging	Features	Classifier	Diagnosis			Lateralisation		
	LTLE	RTLE	Contr.				Acc.	Sens.	Spec.	Acc.	LTLE	RTLE
Focke et al. (2012)	20	18	22	MRI	T1 and DTI image voxels	SVM (lin.)	93.2	89.5	100	100	100	100
					with hippocampi masked out		76.3	67.6	90.9	92.1	95.0	88.9
Kerr et al. (2013)	39	34	32	interictal PET	ROI metabolic changes	MLP	82.9	87.7	71.9	89	n/a	n/a
Kamiya et al. (2016)	15	29	14	MRI	network metrics from DTI structural connectomes	SVM (rbf)	n/a	n/a	n/a	86.4	89.7	80.0
Cantor-Rivera et al. (2015)	9	8	19	MRI	ROI mean intensity and lateral asymmetry	SVM (lin.)	88.9	82.4	94.7	n/a	n/a	n/a
Chiang et al. (2015)	14	10	n/a	rs-fMRI	functional connectivity network metrics	QDA	n/a	n/a	n/a	95.8	92.9	100
Yang et al. (2015)	7	5	n/a	rs-fMRI	functional connectivity values and network metrics	SVM (lin.)	n/a	n/a	n/a	83.3	86.0	80.0
current study	20	20	35	rs-EEG	functional connectivity values	RF	90.7	95	85.7	90.0	90.0	90.0

Table 6.3: Comparison between several studies for automated diagnosis and lateralisation of temporal lobe epilepsy. The data for the current study are given in the last row for comparison

tivities and specificities than those reported in these previous studies which used other imaging tools. Moreover, our results were obtained using only one minute of artefact-free EEG, extracted from a 10- to 15 minutes recording, which is less time-consuming than other imaging modalities. Due to the low cost and wide availability of EEG compared with other modalities, EEG-based measures could be widely implemented for diagnosis and lateralisation.

6.2.2 Main features for diagnosing and lateralising TLE

The selected set of features slightly differs between LOOCV iterations due to the intrinsic randomness of the procedure and the RF technique. Table 6.4 and Table 6.5 show the most frequently occurring subset of features, ranked according to their average importance value. For each feature, the p-value of a non-parametric Mann-Whitney-Wilcoxon test is given, testing the null hypothesis that values are equally distributed in the two competing classes. The feature with highest importance for diagnosis is the outflow from the right to the left hippocampus. For lateralisation, the outflow from the right anterior cingulate cortex to the right hippocampus has the highest importance.

For the feature selection, although we preselect 14 regions based on the previous study by Coito et al. (2016), the selection of connections between these regions is done automatically, in a data-driven way and independently from prior clinical knowledge. This allows us to identify new potential biomarkers for diagnosis and lateralisation of TLE. We remark that the pre-selection of regions also has the disadvantage of missing potentially important regions for diagnosis and lateralisation. The feature selection and classification system can be designed with randomly selected regions in order to search for potential biomarkers. This is however beyond the scope of the current study.

The results show that the outflow from the hippocampus and anterior cingulate cortex are the best predictive features to automatically diagnose and lateralise TLE. This is in accordance with previous work on the connectivity pattern differences between LTLE, RTLE and healthy controls (Coito et al., 2016). Indeed, the importance of the

Feature	Importance ($\cdot 10^{-2}$)	p-value
θ Hipp-R $\rightarrow$ Hipp-L	5.29	0.276
α Hipp-L $\rightarrow$ ACC-R	5.23	0.004
β PCC-L $\rightarrow$ Amyg-R	5.07	0.005
α Hipp-L $\rightarrow$ TPMid-R	2.52	0.006
θ Hipp-R $\rightarrow$ Amyg-R	2.37	0.326
β ACC-R $\rightarrow$ TPMid-L	1.33	0.012

Table 6.4: Feature selection result for diagnosis. Selected features are sorted from the most to the least important for classification.

Feature	Importance ($\cdot 10^{-2}$)	p-value
α ACC-R $\rightarrow$ Hipp-R	9.28	0.068
θ Hipp-R $\rightarrow$ Hipp-L	7.58	0.394
θ TPMid-R $\rightarrow$ Amyg-R	7.08	0.091

Table 6.5: Feature selection result for lateralisation. Selected features are sorted from the most to the least important for classification.

hippocampus and anterior cingulate cortex in TLE has been widely recognised. The hippocampus has a pivotal role in the generation of interictal and ictal activity in the majority of TLE cases. Accordingly, many studies have reported reduced functional connectivity between both hippocampi, hippocampus and amygdala, or hippocampus and other regions of the default-mode network, namely the anterior and posterior cingulate cortex (Laufs et al., 2007; Liao et al., 2010; Pereira et al., 2010; Zhang et al., 2010; Pittau et al., 2012; Coito et al., 2016). From a methodological perspective, there is converging evidence from intracranial and scalp EEG recordings that medial temporal lobe activity can be recorded with scalp EEG (Koessler et al., 2015; Nahum et al., 2011). Anterior cingulate cortex functional connectivity has also been shown to be decreased in TLE patients compared to healthy controls (Stretton et al., 2014; Coito et al., 2016) and could be related to frequent mood disorders in TLE, since the anterior cingulate cortex is a key node in the emotional processing network (Bush et al., 2000).

6.2.3 Importance of feature interaction

Figure 6.1 shows the interaction between the selected features. For diagnosis, the interaction between the outflow from the right to the left hippocampus and the outflow from the left PCC to the right amygdala were the most important. For lateralisation, the interaction between the outflow from the right to the left hippocampus and the outflow from the right ACC to the right hippocampus were most important.

Previous work used statistical tests to find features that had significantly different values in subjects with LTLE, RTLE and healthy subjects (Coito et al., 2016). However, these statistical analyses consider features individually, while the relevance of a connectivity value for classification depends also on which other connectivity values are considered as features. The outflow from the right hippocampus to the left hippocampus and right amygdala were not significantly different for TLE compared with healthy controls, while they were among the most important features for classification. As shown in Figure 6.1, these two connectivity values strongly interact with other features in the selection. Although no significant differences in region-to-region directed functional connectivity were found between LTLE and RTLE,

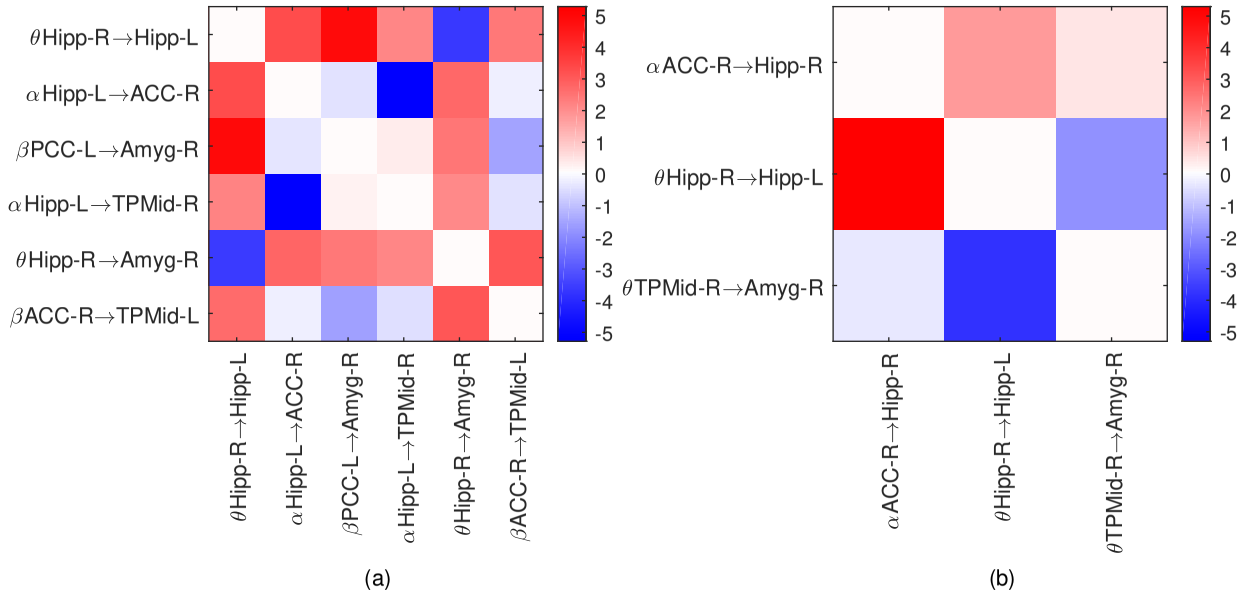

Figure 6.1: Feature interaction effect - Size of the interaction effect between features for diagnosis (a) and lateralisation (b). The colour on the intersection of row f_r and column f_c indicates the interaction effect of feature f_r on f_c measured as the decrease in feature importance of f_c when f_r is left out of the design. Blue means a negative interaction, the discriminative information in f_c is less relevant when f_r is included in the design. Red means a positive interaction, the discriminative information in f_c becomes more relevant when f_r is included in the design.

as also reported previously by (Coito et al., 2016), the combination of these non-significant features seems to be sufficient for a good classification. Therefore, we show that classification algorithms that take into account the interaction between features can outperform significance tests between groups, which also allows us to find new biomarkers for diagnosis and lateralisation of TLE.

6.3 Conclusion

In this chapter we built an automated diagnosis and lateralisation system for temporal lobe epilepsy using EEG-derived directed functional connectivity and random forests classifiers. The high accuracy achieved in this study for the automatic diagnosis of TLE based on functional connectivity measures using EEG periods without pathological activity shows that this approach could constitute a valuable bedside aid for clinicians. Our classification results were comparable to or better than earlier reported results using other imaging modalities. We showed that the outflows from the hippocampus and anterior cingulate cortex are crucial features for the classifiers, in line with previous work showing the importance of these regions in TLE. The interaction between connectivity values are important for classification accuracy, even when connectivity values considered region by region might not be significantly different between groups.

The automated diagnosis of TLE based on EEG periods without IEDs has several important advantages: (1) resting-state EEG can be recorded in less than one hour, overcoming long-term monitoring and its related costs, (2) no IEDs or ictal activity are required, enabling the use of this method in patients with low seizure and/or IEDs frequency, (3) the features that result in the best classification provide insight into the differences between the groups (controls, LTLE and RTLE) and thus the mechanism of action of TLE.

7 Conclusions and future perspectives

Since the inception of brain-computer interfaces in 1973 by Jacques J. Vidal, the growing BCI community has made a remarkable effort to integrate these systems in the daily life of those who could benefit from them. All these endeavours have significantly pushed the field forward but several issues remain, keeping BCI systems from being applied widely. The underlying cause is that state of the art methods still regard the different components of the BCI as separate entities. Research is focussed on the improvement of either the application, decoder or user and not on the system as a whole. This approach limits the potential improvements that can be obtained.

In this thesis, I have developed a symbiotic design approach for BCIs based on event-related potentials. The different components are co-adapted to each other and as such obtain an overall improvement in accuracy, efficiency and reliability of the BCI. For the translation of brain activity to useful information I used machine learning techniques. The benefits and challenges of using machine learning for the decoding of brain activity were also shown for a relatively new category of BCIs: computer-aided diagnosis systems. I have developed an automated, data-driven, system for accurate diagnosis and lateralisation of temporal lobe epilepsy based on measurements of the functional connectivity between different regions in the brain.

In this final chapter I outline the main conclusions resulting from this work. Afterwards, I explain how these findings can contribute to the future of BCI and can have an impact outside the BCI domain.

7.1 Research conclusions

7.1.1 A tunable paradigm reveals the interaction between application and decoder

A whole spectrum of stimulus presentation paradigms has been developed for ERP-BCI by the BCI community. Paradigms that lead to faster spelling inevitably lead to a lower spelling accuracy and vice versa. For this reason, a trade-off between speed and accuracy has to be made when choosing the paradigm. The most suitable paradigm depends on the subject's preferences and the requirements of the decoder. Nevertheless, existing paradigms do not take this interaction into account.

In order to make the paradigm tunable to the decoder and user, we proposed a new stimulus presentation modality for which the iteration length and relative frequency of target stimuli can be chosen freely. An online experiment with 24 subjects compared this paradigm to the basic paradigms in a self-learning ERP speller. The results showed that, with the new paradigm, a higher number of correct symbols can be spelled per minute.

Even more important was the offline analysis of these results, which revealed the underlying mechanism of interaction between the speller application and the self-learning decoder. In the beginning of the spelling, when data is scarce, the decoding accuracy is low and highly dependent on the random initialisation of the decoder's parameters. This makes the system very unreliable. We found that, during this period, the balance between the number of recorded target and non-target stimuli is more important for classification performance than signal quality. Only when more data is recorded, the classification performance quickly increases and the SNR of the data becomes the most important aspect.

While these findings are interesting and can guide future paradigm development, the importance of this contribution is that it enabled us to develop the LLP-based self-learning BCI. The tunable paradigm clears the path for an integrated design where the application is adapted to the specific requirements of this decoder.

7.1.2 Zero-training BCI with quality guarantees

Traditional BCI systems are calibrated with labelled data, recorded during a separate calibration session. As this is tiresome for the user, the BCI community has developed several methods to reduce or even avoid these tedious sessions. For example, transfer learning methods recycle data from previous sessions or users. Alternatively, adaptive methods use a weakly calibrated decoder to classify ERP responses online and use their own predictions to tune their parameters. Recently, Kindermans et al. (2012) proposed the first truly self-learning decoder. It starts with random parameter values and uses the expectation maximisation algorithm to tune them during actual use of the BCI. None of these methods has the theoretical guarantee to find a good classification of stimulus responses. This makes the BCI with reduced calibration unreliable.

We demonstrated the applicability of the learning from label proportions concept to ERP-BCI. This method is capable of finding an estimate of the class-conditional mean response feature vector that is theoretically guaranteed to make the classifier converge to a traditional supervised classifier. The method requires the data to be observed in separate groups for which the relative frequency of responses in each class is known. For that purpose, we used our tunable paradigm.

We evaluated the spelling performance of the LLP method in an online spelling experiment with 13 subjects. The results showed that the LLP decoder successfully infers the user's intention and converges to the supervised performance when more data is collected. An offline resimulation of the experiment compared the results to the EM-based unsupervised decoding method. LLP is stable and reliable but does not obtain the same level of performance as a well-initialised EM decoder. On the other hand, the EM-based decoder works well empirically but the quality of the decoder demonstrates high variability, since it depends strongly on the initialisation of its parameters. To combine the strengths of both approaches, our third contribution is the analytical mixing of LLP and EM.

7.1.3 Mixing model estimators makes zero-training reliable and effective

To obtain optimal performance we presented a method to combine the aforementioned two approaches for unsupervised classification of ERP responses. The LLP-based self-learning decoder demonstrated robust classification performance. In contrast, the EM-based decoding for ERP-BCI does not have this guarantee. However, while this decoder has high variance due to random initialisation of its parameters, it obtains a higher accuracy faster than LLP when the initialisation is fortunate.

We demonstrated that both the EM and LLP-based unsupervised decoders have a structure similar to LSR classifiers. The methods differ in their estimation of the class-conditional mean, which leads to their complementary learning behaviour during the spelling session. We proposed a method to mix these two different estimators and determined an analytical formula to calculate the optimal mixing coefficient.

We compared the mixing method to the aforementioned methods in a resimulation of an online experiment with 13 subjects. Our experiments indicated that the analytical mixing makes the decoder as robust as a pure LLP system, but thanks to the inclusion of the EM component, it can achieve optimal performance. This paves the way for a new generation of ERP spellers that can be used reliably without calibration, but are also guaranteed to work well without frequent supervised fine-tuning. Furthermore, since LLP gives us guarantees and the mixing model can be seen as a regularisation towards the LLP solution, this gives us the possibility to create new BCIs where the online data, without explicit knowledge of the user's intention, is as valuable as labelled calibration data (Verhoeven et al., 2017).

7.1.4 Feature interaction indicates diagnosis in TLE

Computer-aided diagnosis systems for brain diseases are a relatively new category of BCIs. They are especially suitable in cases where it is challenging to record pathological brain activity that is visible to

the human eye. We developed a classification system for the diagnosis and lateralisation of temporal lobe epilepsy.

The systems classifies subjects based on a measure of the connectivity between different brain regions. The most informative connectivities are selected automatically from the database. Some of the selected features did not show statistically significant differences between groups. This demonstrates the added value of machine learning in computer-aided diagnosis. It goes beyond standard statistics and is capable of incorporating the interaction between features to accurately discriminate between patients and healthy subjects.

7.2 Future directions

In this dissertation we focussed on two specific case studies: BCIs based on event-related potentials and computer-aided diagnosis of temporal lobe epilepsy. Nevertheless, the approach developed in this dissertation has the potential to impact the development of other types of BCI systems and can even be applied outside the BCI domain. In this final section, I will explain how the findings in the thesis can contribute to the future of BCI, computer-aided diagnosis systems and the application of self-learning classifiers in general.

7.2.1 The need for a symbiotic BCI design

Our symbiotic design of an ERP-BCI improved the performance of this system greatly. In our final ERP speller, the three components are co-adapted to each other. First, the self-learning decoder adapts its parameters to the specific characteristics of the user's brain responses during actual use of the BCI. We have improved this dynamic learning behaviour in this dissertation. Second, the user is adaptive by nature. He/she receives feedback from the speller and as such automatically learns to use the system more effectively. Finally, we have made the application of ERP-BCI tunable with our flexible stimulus presentation paradigm.

In future work, the synergy between the three sub-systems of the ERP-BCI will be improved by searching for a dynamic optimisation

of the application. A first direction is the application of the dynamic stopping technique to dynamically optimise the number of sequence iterations that is presented to the user. In addition, the iteration length should be tuned to the changing requirements of the self-learning decoder during the learning process and the (changing) preferences of the user.

This dissertation focussed mainly on the technical improvement of the BCI. Improving the usability was focussed on the important challenge of eliminating the calibration without decreasing the accuracy or reliability of the system. Nevertheless, the design of assistive technology should be user-oriented. Future work needs to focus on the user-paradigm interaction to improve user experience and should include experiments measuring the visual and mental fatigue of the user.

The symbiotic design approach is not limited to ERP-BCI. The exceptional results obtained in this dissertation should serve as a source of inspiration for the synergistic design of other BCI systems, for instance those based on motor imagery. We strongly believe that the co-adaptation can truly empower the performance for any type of BCI and will enable the BCI community to a make a large leap towards the integration of BCI systems in our daily life.

7.2.2 Combining self-learning classification models

We have shown that it is possible to combine the strengths of different self-learning methods by combining them in a theoretical way. In this work we focussed on improving the estimation of the class-conditional mean feature vector for ERP-BCI. With this approach we created a new generation of BCI systems in which the unlabelled data, recorded during use, is almost as valuable as the data recorded during a separate calibration session. This approach can be extended in several future directions within as well as outside the BCI domain. For example, the method can be extended to more than two self-learning decoding methods and validated on other decoders than the LLP and EM-based decoders used in this dissertation.

An important future direction is the integration of transfer learn-

ing. Transfer learning was incorporated in the EM-based calibrationless decoder from Kindermans et al. (2012). It was shown that the general model (a weighted average of the weight vector obtained for previous subjects) can regularise the model for a new subject and greatly improve the performance of the self-learning decoder. Future work will demonstrate how this alternative method for improving self-learning decoders relates to our mixing of different unsupervised decoders. The incorporation of transfer learning in our symbiotic design has the potential to improve the performance of ERP-BCI even further.

The applicability of our mixing method is not only limited to the domain of BCI. Self-learning, adaptive classifiers gain a lot of interest in almost any domain were large amounts of data are collected. To keep the classification accuracy at an acceptable level, classifiers are required to adapt to the changes in the constant stream of new data that is obtained. Our method is capable of mixing any two self-learning classifiers for which the estimation error can be calculated. The method is especially suitable in domains were data is very noisy and complex, for example financial data.

7.2.3 EEG-based automatic diagnosis of neurological diseases

The automatic diagnosis system for temporal lobe epilepsy, based on functional connectivity measures using scalp EEG, achieved a high accuracy. This demonstrates that our system could constitute valuable bedside aid for clinicians. Due to the low cost and wide availability of EEG compared with other modalities, the classifier can be widely implemented in the clinical environment. In a first phase, the computer-aided diagnosis can be used in parallel with the traditional presurgical methods. In this way, more data can be collected rapidly and used for further fine-tuning. In our current approach we used a pre-selection of brain regions that are known to have indicated significant differences between groups in previous work, or are known to be related to the disorder. With a larger database, the feature selection and classification system can be designed with randomly selected regions in order to search for new potential biomarkers.

Further studies are warranted to determine whether our classification system is efficient in patients with equivocal lateralisation or apparent bilateral TLE, who benefit from subsequent validation with invasive EEG. Our approach may also be used to build classification systems for other types of epilepsy and may be even used to differentiate between types of epilepsy. Furthermore, comparative studies will assess the performance of the classification systems using other EEG network measures such as undirected connectivity or a combination with the currently used directed connectivity measures.

Future work will examine whether functional network measures can be used to predict the responsiveness of a patient to a specific category of anti-epileptic drugs. In this way, the long and inconvenient process of trying several types of medication can potentially be significantly reduced. The transparency of our approach can be employed to search for sub-groups of patients that are not completely covered by the current range of medication. This could be interesting for the pharmaceutical industry to allocate resources more efficiently.

The applicability of computer-aided diagnosis based on functional connectivity goes beyond the field of epilepsy. Network measures in the brain have gained a lot of interest in the identification of other neurological disorders such as Alzheimer's disease, depression etc. Computer-aided diagnosis has a lot of potential to become an assisting tool for clinicians. In the future, it may even become indispensable for the early detection and subsequent treatment of these disorders.

A

Statistical analysis of experimental results with the switching paradigm

This appendix describes the complete statistical analysis that is used to describe the results reported in Chapter 3. A one-way repeated measures ANOVA is used for comparison of the online experimental results for the three paradigms. A paired-samples t-test is used to compare two given conditions in the offline analysis.

As assessed by inspection of a box plot, values greater than three box-lengths from the edge of the box are labelled as extreme outliers. For the repeated measures ANOVA, the assumption of sphericity is investigated with Mauchly's test (M). The assumption of normality is investigated with Shapiro-Wilk's test (SW). If the normality assumption is violated, the data is transformed with one of the following formulas. For strongly negatively skewed data:

$$y = log_{10}(x_{max} - x)$$

For extremely negatively skewed data:

$$y = 1/(x_{max} - x)$$

with x_{max} the maximum value of the dependent variable.

Online results

Spelling accuracy

There are two extreme outliers: the RC result for subject S05 and the CB-result for S13. The data is not normally distributed. Therefore, the strongly negatively skewed data is transformed. The resulting data is normally distributed for the SP and CB paradigm but not for the RC paradigm (SW: $p = 0.019 < 0.05$). As the repeated measures ANOVA is robust to violations of normality this is sufficient. With this transformation there are no extreme outliers any more. Mauchly's test of sphericity indicates that the assumption of sphericity is not violated, $\chi^2(2) = 0.345, p = 0.841$.

The online spelling accuracy is statistically significantly different for the paradigms, $F(2, 46) = 11.101, p < 0.0005$, partial $\eta^2 = 0.326$. Pairwise comparison of the results confirms the conjecture that the accuracy differs between SP and RC ($p = 0.002$) and between SP and CB ($p = 0.001$) but not between RC and CB. The same conclusions are drawn from analysis on the untransformed accuracy.

Respelling accuracy

There are several extreme outliers in the data and the data is not normally distributed. Therefore, the extremely negatively skewed data is transformed. With this transformation, there are no outliers any more. The assumption of sphericity is not violated (M: $\chi^2(2) = 0.367, p = 0.832$). No statistically significant difference in respelling accuracy is found, $F(2, 46) = 1.156, p = 0.324$, partial $\eta^2 = 0.048$.

For completeness, we determine the influence of the order in which the paradigms are used during the experiment. There is no statistically significant interaction between the order in which the paradigms are used during the test and the paradigm itself on its resulting respelling accuracy, $F(10, 36) = 1.961$, $p = 0.068$, partial $\eta^2 = 0.353$. The main effect of the order however showed that there is a statistically significant difference in respelled accuracy between the different

orders $F(5, 18) = 3.423$, $p = 0.024$, partial $\eta^2 = 0.487$. This means that the position of the tested paradigm, in the order of three, has an influence on the achieved respelling accuracy but that this influence is the same for all paradigms.

AUC

There is an extreme outlier for subject S13 on the CB paradigm and the data is not normally distributed. Therefore, the extremely negatively skewed data is transformed and the outlier is removed by replacing it with the closest value found for other subjects. The assumption of sphericity is not violated (M: $\chi^2(2) = 5.248, p = 0.073$). AUC was statistically significantly different for the different paradigms, $F(2, 46) = 8.324$, $p = 0.001$, partial $\eta^2 = 0.266$. Pairwise comparison of the results confirms that the AUC differs between SP and RC ($p = 0.001$) and between SP and CB ($p = 0.003$) but not between RC and CB ($p = 1.000$). The same conclusions are drawn from analysis on the untransformed AUC.

Offline analysis

p-ratio at symbol 5

The data is not normally distributed (SW: $p = 0.028$). The extremely negatively skewed data is transformed. There are no extreme outliers. The difference in accuracy is statistically significant, $t(23) = 2.807$, $p = 0.010$

SP and CB accuracy at 4 SM.

The strongly negatively skewed data is transformed to meet the normality assumption ($p = 0.170$). There are no extreme outliers. The difference in AUC level is statistically significant, $t(23) = 2.247$, $p = 0.035$

B

Equations for the mixing of model estimators

Reformulation of the EM-based decoder

For EM-based decoding, the weight vector $\boldsymbol{w}$ is updated in the maximisation step with the following update equation:

$$\hat{\boldsymbol{w}} = \sum_{\boldsymbol{c}} p(\boldsymbol{c}|\boldsymbol{X}, \boldsymbol{w}, \beta) \left(\boldsymbol{X}^T \boldsymbol{X} + \frac{\alpha}{\beta} \boldsymbol{I} \right)^{-1} \boldsymbol{X}^T \boldsymbol{y}(\boldsymbol{c})$$

$\boldsymbol{c}$ denotes a vector that assign a target symbol to each recorded trial, $\boldsymbol{X}$ is the design matrix containing all recorded responses from all trials in its rows. The function $\boldsymbol{y}(\boldsymbol{c})$ assigns a label to each recorded response, based on the desired symbol in each trial and the constraints imposed by the stimulus presentation paradigm. The sum over all possible assignations of the vector $\boldsymbol{c}$ is expanded to a sum over each element:

$$\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^T \boldsymbol{X} + \frac{\alpha}{\beta} \boldsymbol{I} \right)^{-1} \sum_{c_1} \sum_{c_2} \dots \sum_{c_t} \left[\left(\prod_t p(c_t|\boldsymbol{X}, \boldsymbol{w}, \beta) \right) \sum_t \boldsymbol{X}_t^T \boldsymbol{y}_t(\boldsymbol{c}) \right]$$

where $\boldsymbol{X}_t$ and $\boldsymbol{y}_t(\boldsymbol{c})$ denote respectively the responses and labels from trial t. Within a single trial t, the labels of the responses are only related to the desired symbol c_t in that trial. Therefore, the expression

can be simplified:

$$\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^T\boldsymbol{X} + \frac{\alpha}{\beta}\boldsymbol{I}\right)^{-1} \sum_t \sum_{c_t} p(c_t|\boldsymbol{X}_t, \boldsymbol{w}, \beta)\boldsymbol{X}_t^T\boldsymbol{y}_t(c_t)$$

With the rescaled labels $y \in \{N/N_+, -N/N_-\}$ this yields:

$$\hat{\boldsymbol{w}} = \left(\boldsymbol{X}^T\boldsymbol{X} + \frac{\alpha}{\beta}\boldsymbol{I}\right)^{-1} \left[\frac{N}{N_+} \sum_t \sum_{c_t} \sum_{(t,i)|c_t \in i_t} p(c_t|\boldsymbol{X}_t, \boldsymbol{w}, \beta)\boldsymbol{x}_{t,i} - \frac{N}{N_-} \sum_t \sum_{c_t} \sum_{(t,i)|c_t \notin i_t} p(c_t|\boldsymbol{X}_t, \boldsymbol{w}, \beta)\boldsymbol{x}_{t,i}\right]$$

where $c_t \in i_t$ denotes that the symbol c_t is highlighted during the i^{th} stimulus in trial t. Consider the first term between brackets. The sum over c_t is in fact a sum over the rows of the allocation matrix for trial t, generated by the stimulus paradigm. For each symbol in the grid, the inner sum iterates the stimuli that highlighted this symbol. Consequently, the combination of these two sums is a sum over all cells in the allocation matrix that contain value 1. The same result can be obtained by a summation over all stimuli and an inner sum over all symbols highlighted in that stimulus. We obtain the following equation:

$$\hat{\boldsymbol{w}} = \left(\frac{1}{N}\boldsymbol{X}^T\boldsymbol{X} + \frac{\alpha}{N\beta}\boldsymbol{I}\right)^{-1} \left[\frac{1}{N_+} \sum_t \sum_i \sum_{c_t \in i_t} p(c_t|\boldsymbol{X}_t, \boldsymbol{w}, \beta)\boldsymbol{x}_{t,i} - \frac{1}{N_-} \sum_t \sum_i \sum_{c_t \notin i_t} p(c_t|\boldsymbol{X}_t, \boldsymbol{w}, \beta)\boldsymbol{x}_{t,i}\right]$$

Both terms are a weighted sum over all stimulus responses in all trials. In the first term, the weight is the probability that the response is target, obtained by a summation over all the symbols that it highlights. In the second term, the weight is the probability that the response is non-target. Consequently, these two terms represent the estimation of the class-conditional mean responses by this method. This yields the following reformulation of the EM-based update equation for the

weight vector.

$$\hat{\boldsymbol{\mu}}_+ = \frac{1}{N_+} \sum_{t,i} \left(\sum_{c_t \in i_t} p(c_t \mid \boldsymbol{X}_t, \boldsymbol{w}, \beta) \right) \boldsymbol{x}_{t,i}$$
$$\hat{\boldsymbol{\mu}}_- = \frac{1}{N_-} \sum_{t,i} \left(\sum_{c_t \notin i_t} p(c_t \mid \boldsymbol{X}_t, \boldsymbol{w}, \beta) \right) \boldsymbol{x}_{t,i}$$
$$\hat{\boldsymbol{\Sigma}} = \frac{\boldsymbol{X}^T \boldsymbol{X}}{N}$$
$$\hat{\boldsymbol{w}} = (\hat{\boldsymbol{\Sigma}} + \lambda \boldsymbol{I})^{-1} (\hat{\boldsymbol{\mu}}_+ - \hat{\boldsymbol{\mu}}_-)$$

Where the regularisation constant is denoted as λ

Derivation of the formula for the optimal mixing coefficient

The estimate of a class-wise (target or non-target) mean feature vector $\hat{\boldsymbol{\mu}}$ is defined as the mixture of two independent estimates $\hat{\boldsymbol{\mu}}_A$ and $\hat{\boldsymbol{\mu}}_B$ with mixing coefficient γ:

$$\hat{\boldsymbol{\mu}}(\gamma) = (1 - \gamma)\hat{\boldsymbol{\mu}}_A + \gamma \hat{\boldsymbol{\mu}}_B$$

$\hat{\boldsymbol{\mu}}_A$ could for example be the maximum likelihood estimate and $\hat{\boldsymbol{\mu}}_B$ the label proportions result. The optimal γ^* is the value that brings $\hat{\boldsymbol{\mu}}$ as close as possible to the real class-wise mean $\boldsymbol{\mu}$. Unfortunately $\boldsymbol{\mu}$ is unknown. The expected value of the squared error between $\hat{\boldsymbol{\mu}}$ and $\boldsymbol{\mu}$ is minimised:

$$\gamma^* = \arg\min_{\gamma} E\left[\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}(\gamma)\|^2 \right]$$

For the label proportions method it was shown that the estimation of the class-wise means are guaranteed to converge to the true means $E[\hat{\boldsymbol{\mu}}_{LLP}] = \boldsymbol{\mu}$ (Hübner et al., 2017). Although the EM method does not find the true MLE but rather optimises a lower bound on it, we use $E[\hat{\boldsymbol{\mu}}_{EM}] = \boldsymbol{\mu}$ as a heuristic. Using $E[\hat{\boldsymbol{\mu}}_A] = E[\hat{\boldsymbol{\mu}}_B] = \boldsymbol{\mu}$ and the

linearity of the expectation operator we get:

$$
\begin{aligned}
& E\left[\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}(\gamma)\|^2\right] \\
=\ & E\left[\|(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)+\gamma(\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B)\|^2\right] \\
=\ & E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& +\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T(\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B)\right]+\gamma E\left[(\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B)^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& +\gamma^2 E\left[(\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B)^T(\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B)\right] \\
=\ & E\left[\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A\|^2\right] \\
& +\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T\hat{\boldsymbol{\mu}}_A+\hat{\boldsymbol{\mu}}_A^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& -\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T\hat{\boldsymbol{\mu}}_B+\hat{\boldsymbol{\mu}}_B^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& +\gamma^2 E\left[\|\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B\|^2\right] \\
=\ & E\left[\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A\|^2\right] \\
& +\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T\hat{\boldsymbol{\mu}}_A+\hat{\boldsymbol{\mu}}_A^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& -\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T\right]\boldsymbol{\mu}-\gamma\boldsymbol{\mu}^T E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& -\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T\hat{\boldsymbol{\mu}}_B+\hat{\boldsymbol{\mu}}_B^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& +\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T\right]\boldsymbol{\mu}+\gamma\boldsymbol{\mu}^T E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& +\gamma^2 E\left[\|\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B\|^2\right] \\
=\ & E\left[\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A\|^2\right] \\
& +\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T(\hat{\boldsymbol{\mu}}_A-\boldsymbol{\mu})+(\hat{\boldsymbol{\mu}}_A-\boldsymbol{\mu})^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& -\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T(\hat{\boldsymbol{\mu}}_B-\boldsymbol{\mu})+(\hat{\boldsymbol{\mu}}_B-\boldsymbol{\mu})^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right] \\
& +\gamma^2 E\left[\|\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B\|^2\right] \\
=\ & E\left[\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A\|^2\right] \\
& -2\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)\right]+2\gamma E\left[(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A)^T(\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_B)\right] \\
& +\gamma^2 E\left[\|\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B\|^2\right] \\
=\ & E\left[\|\boldsymbol{\mu}-\hat{\boldsymbol{\mu}}_A\|^2\right] \\
& -2\gamma\textstyle\sum_d Var\left[\hat{\mu}_{A,d}\right]+2\gamma\sum_d Cov\left[\hat{\mu}_{A,d},\hat{\mu}_{B,d}\right] \\
& +\gamma^2 E\left[\|\hat{\boldsymbol{\mu}}_A-\hat{\boldsymbol{\mu}}_B\|^2\right]
\end{aligned}
$$

Here, $Var\left[\hat{\mu}_{A,d}\right]$ denotes the variance on the estimation of the d$^{\text{th}}$ component of $\boldsymbol{\mu}$. Likewise $Cov\left[\hat{\mu}_{A,d}, \hat{\mu}_{B,d}\right]$ denotes the covariance between the two different estimates of the d$^{\text{th}}$ component of $\boldsymbol{\mu}$.

The optimal value γ^* can be found by setting the derivative of this expected value to zero and solving for γ:

$$\gamma^* = \frac{\sum_d Var\left[\hat{\mu}_{A,d}\right] - \sum_d Cov\left[\hat{\mu}_{A,d}, \hat{\mu}_{B,d}\right]}{\|\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B\|^2}$$

As the solution is optimal, it should give the same result when $\hat{\boldsymbol{\mu}}_A$ and $\hat{\boldsymbol{\mu}}_B$ switch positions in the original formula for $\hat{\boldsymbol{\mu}}$:

$$\begin{aligned} \hat{\boldsymbol{\mu}}(\eta) &= \eta\hat{\boldsymbol{\mu}}_A + (1-\eta)\hat{\boldsymbol{\mu}}_B \\ \eta^* &= \frac{\sum_d Var\left[\hat{\mu}_{B,d}\right] - \sum_d Cov\left[\hat{\mu}_{B,d}, \hat{\mu}_{A,d}\right]}{\|\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B\|^2} \\ &= (1-\gamma^*) \end{aligned}$$

Plugging in the expression for $\gamma*$ yields:

$$\begin{aligned} &\textstyle\sum_d Cov\left[\hat{\mu}_{A,d}, \hat{\mu}_{B,d}\right] = \\ &\frac{1}{2}\left(\textstyle\sum_d Var\left[\hat{\mu}_{A,d}\right] + \sum_d Var\left[\hat{\mu}_{B,d}\right] - \|\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B\|^2\right) \end{aligned}$$

Replacing the covariance in the original formula for γ^* results in a symmetric expression:

$$\gamma^* = \frac{1}{2}\left(\frac{\sum_d Var\left[\hat{\mu}_{A,d}\right] - \sum_d Var\left[\hat{\mu}_{B,d}\right]}{\|\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B\|^2} + 1\right)$$

For practical purposes the value of γ^* is limited to the interval $[0, 1]$ as it is also done in covariance shrinkage (Blankertz et al., 2011).

Derivation of the estimator variances

For the label proportions method we have a closed form expression for the estimation:

$$\begin{bmatrix} \hat{\boldsymbol{\mu}}^{+}_{LLP} \\ \hat{\boldsymbol{\mu}}^{-}_{LLP} \end{bmatrix} = \begin{bmatrix} \phi^1_+ \ \phi^2_+ \\ \phi^1_- \ \phi^2_- \end{bmatrix} \cdot \begin{bmatrix} \hat{\boldsymbol{\mu}}_1 \\ \hat{\boldsymbol{\mu}}_2 \end{bmatrix}$$

With $\hat{\boldsymbol{\mu}}_1$ and $\hat{\boldsymbol{\mu}}_2$ the average response in the two groups of stimuli S_1 and S_2. The d^{th} component of the estimator $\hat{\boldsymbol{\mu}}^{+}_{LLP}$ then is:

$$\begin{aligned} \hat{\mu}^{+}_{LLP,d} &= \phi^1_+ \hat{\mu}_{1,d} + \phi^2_+ \hat{\mu}_{2,d} \\ &= \phi^1_+ \frac{1}{N_1} \sum_{i \in S_1} x_{i,d} + \phi^2_+ \frac{1}{N_2} \sum_{i \in S_2} x_{i,d} \end{aligned}$$

With N_1 and N_2 the number of stimuli in the two interleaved sequences. Now, making use of the assumption that the responses on the stimuli are independently and identically distributed we obtain the variance of this component:

$$\begin{aligned} Var\left[\hat{\mu}^{+}_{LLP,d}\right] &= \left(\frac{\phi^1_+}{N_1}\right)^2 \sum_{i \in S_1} Var\left[x_{i,d}\right] \\ &\quad + \left(\frac{\phi^2_+}{N_2}\right)^2 \sum_{i \in S_2} Var\left[x_{i,d}\right] \\ &= \frac{\left(\phi^1_+\right)^2}{N_1} \sigma^2_{1,d} + \frac{\left(\phi^2_+\right)^2}{N_2} \sigma^2_{2,d} \end{aligned}$$

With $\sigma^2_{1,d}$ and $\sigma^2_{2,d}$ the sample variance of the d^{th} feature in the responses recorded on stimuli of sequence type 1 and type 2 respectively. Likewise we get:

$$Var\left[\hat{\mu}^{-}_{LLP,d}\right] = \frac{\left(\phi^1_-\right)^2}{N_1} \sigma^2_{1,d} + \frac{\left(\phi^2_-\right)^2}{N_2} \sigma^2_{2,d}$$

Unfortunately we do not have a closed form expression for the maximum likelihood estimator. This is also the reason why we need the expectation maximisation algorithm. Consequently, it is a lot

harder to find the statistical properties of this estimator. However, under certain regularity conditions of the likelihood function $L(\boldsymbol{\mu} \mid \boldsymbol{X})$ outlined in (DuMouchel, 1973), which are met, the asymptotic behaviour of the MLE is known (Lehmann, 1999). For large samples, the MLE approaches a normal distribution:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{MLE} &\sim \mathcal{N}\left(\boldsymbol{\mu}, \{\boldsymbol{I}(\boldsymbol{\mu})\}^{-1}\right) \\ \boldsymbol{I}(\boldsymbol{\mu}) &= -E\left[\frac{\partial^2}{\partial \boldsymbol{\mu} \partial \boldsymbol{\mu}^T} \ln L\right] \end{aligned}$$

With $\boldsymbol{I}$ the Fisher information matrix. When data $\boldsymbol{X}$ is observed and the MLE is optimised using the EM algorithm we can obtain its variance as follows:

$$\sum\nolimits_d Var\left[\hat{\mu}_{EM,d}\right] = -\mathrm{Tr}\left[\left(\frac{\partial^2}{\partial \boldsymbol{\mu} \partial \boldsymbol{\mu}^T} \ln L \Big|_{\boldsymbol{\mu}=\hat{\boldsymbol{\mu}}_{EM}}\right)^{-1}\right]$$

This is however only exact when the amount of data recorded is unlimited.

C

Clinical details of the temporal lobe epilepsy study

The patients dataset used in this study is the same as reported by Coito et al. (2016). The clinical details can be found in Table C.1 and Table C.2.

Patient	Gender	Age (y)	Age of epilepsy onset (y)	Focus side	Structural epileptogenic lesion on MRI	Surgery Outcome (Engel's class)	Intracranial EEG
P1	F	15	9	left	Tumor	I	No
P2	M	33	15	left	Normal	II	Yes
P3	F	47	3	left	HS	I	No
P4	F	15	9 months	left	HS	I	Yes
P5	F	41	3	left	HS	n/a	No
P6	F	25	20	left	Normal	n/a	No
P7	M	18	13	left	Normal	I	Yes
P8	M	35	27	left	HS	I	No
P9	M	16	1	left	Amyg-Hipp dysplasia	I	No
P10	M	15	11	left	HS + left TA	I	No
P11	F	15	3	left	left TA	I	No
P12	F	40	12	left	HS	I	No
P13	F	20	5	left	HS	I	No
P14	M	31	29	left	Normal	I	Yes
P15	M	45	17	left	HS	I	No
P16	F	29	26	left	Amyg-Hipp DNET	I	No
P17	M	53	2	left	HS	I	No
P18	F	37	28	left	Normal	I	No
P19	F	25	20	left	Cortical dysplasia	n/a	Yes
P20	M	18	8	left	HS	I	No

Table C.1: Patients clinical details: LTLE patients. HS - hippocampus sclerosis; TA - temporal atrophy

Patient	Gender	Age (y)	Age of epilepsy onset (y)	Focus side	Structural epileptogenic lesion on MRI	Surgery Outcome (Engel's class)	Intracranial EEG
P21	F	27	27	right	Amyg DNET	I	No
P22	F	53	25	right	Hypertrophy right Amygdala	n/a	No
P23	F	36	6 months	right	HS	I	No
P24	M	37	20	right	HS	I	No
P25	F	50	31	right	HS + right TA	II	No
P26	F	44	20	right	HS	II	No
P27	M	34	7	right	HS	II	No
P28	F	55	4	right	HS	I	No
P29	M	35	22	right	normal	n/a	Yes
P30	M	18	12	right	HS	I	No
P31	F	37	25	right	HS	I	No
P32	M	16	11	right	HS	I	Yes
P33	F	30	25	right	Dysplasia amygdala	I	No
P34	M	27	5	right	HS	I	No
P35	M	27	18	right	Dysplasia hippocampus + gyrus lingualis	I	Yes
P36	F	57	15	right	HS	I	Yes
P37	M	24	17	right	HS	I	No
P38	F	30	27	right	HS	n/a	No
P39	M	50	11	right	HS	I	Yes
P40	M	16	11	right	Amygdala dysplasia	II	No

Table C.2: Patients clinical details: RTLE patients. HS - hippocampus sclerosis; TA - temporal atrophy

Bibliography

Ahn, M., Cho, H., and Jun, S. C. (2011). Calibration time reduction through source imaging in brain-computer interface (BCI). In *International Conference on Human-Computer Interaction*, pages 269–273. Springer.

Ahn, M., Lee, M., Choi, J., and Jun, S. C. (2014). A review of brain-computer interface games and an opinion survey from researchers, developers and users. *Sensors*, 14(8):14601–14633.

Allison, B. Z. and Pineda, J. A. (2006). Effects of SOA and flash pattern manipulations on ERPs, performance, and preference: implications for a BCI system. *International journal of psychophysiology*, 59(2):127–140.

Astolfi, L., Cincotti, F., Mattia, D., Fallani, F. D. V., Tocci, A., Colosimo, A., Salinari, S., Marciani, M. G., Hesse, W., Witte, H., et al. (2008). Tracking the time-varying cortical connectivity patterns by adaptive multivariate estimators. *IEEE Transactions on Biomedical Engineering*, 55(3):902–913.

Astolfi, L., Cincotti, F., Mattia, D., Marciani, M. G., Baccala, L. A., Fallani, F. D. V., Salinari, S., Ursino, M., Zavaglia, M., and Babiloni, F. (2006). Assessing cortical functional connectivity by partial directed coherence: simulations and application to real data. *IEEE Transactions on Biomedical Engineering*, 53(9):1802–1812.

Babb, T. L., Brown, W. J., Pretorius, J., Davenport, C., Lieb, J. P., and Crandall, P. H. (1984). Temporal lobe volumetric cell densities in temporal lobe epilepsy. *Epilepsia*, 25(6):729–740.

Baccalá, L. A. and Sameshima, K. (2014). Partial directed coherence. In *Methods in Brain Connectivity Inference through Multivariate Time Series Analysis*, pages 57–73. CRC Press.

Bartz, D. and Müller, K.-R. (2014). Covariance shrinkage for autocorrelated data. In *Advances in neural information processing systems*, pages 1592–1600.

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

Billinger, M., Daly, I., Kaiser, V., Jin, J., Allison, B. Z., Müller-Putz, G. R., and Brunner, C. (2012). Is it significant? guidelines for reporting BCI performance. In *Towards Practical Brain-Computer Interfaces*, pages 333–354. Springer.

Birbaumer, N. (2006). Breaking the silence: brain-computer interfaces (BCI) for communication and motor control. *Psychophysiology*, 43(6):517–532.

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

Birot, G., Spinelli, L., Vulliémoz, S., Mégevand, P., Brunet, D., Seeck, M., and Michel, C. M. (2014). Head model and electrical source imaging: a study of 38 epileptic patients. *NeuroImage: Clinical*, 5:77–83.

Bishop, C. (2007). Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn.

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.

Blankertz, B., Tangermann, M., Vidaurre, C., Fazli, S., Sannelli, C., Haufe, S., Maeder, C., Ramsey, L. E., Sturm, I., Curio, G., et al. (2010). The Berlin brain-computer interface: non-medical uses of BCI technology. *Frontiers in neuroscience*, 4:198.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brinkmann, B. H., Wagenaar, J., Abbot, D., Adkins, P., Bosshard, S. C., Chen, M., Tieng, Q. M., He, J., Muñoz-Almaraz, F., Botella-Rocamora, P., et al. (2016). Crowdsourcing reproducible seizure forecasting in human and canine epilepsy. *Brain*, 139(6):1713–1722.

Brunet, D., Murray, M. M., and Michel, C. M. (2011). Spatiotemporal analysis of multichannel EEG: CARTOOL. *Computational intelligence and neuroscience*, 2011:2.

Bush, G., Luu, P., and Posner, M. I. (2000). Cognitive and emotional influences in anterior cingulate cortex. *Trends in cognitive sciences*, 4(6):215–222.

Cantor-Rivera, D., Khan, A. R., Goubran, M., Mirsattari, S. M., and Peters, T. M. (2015). Detection of temporal lobe epilepsy using support vector machines in multi-parametric quantitative MR imaging. *Computerized Medical Imaging and Graphics*, 41:14–28.

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

Cecotti, H. and Graser, A. (2011). Convolutional neural networks for P300 detection with application to brain-computer interfaces. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):433–445.

Chapin, J. K., Moxon, K. A., Markowitz, R. S., and Nicolelis, M. A. (1999). Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature neuroscience*, 2(7):664–670.

Chatrian, G., Lettich, E., and Nelson, P. (1985). Ten percent electrode system for topographic studies of spontaneous and evoked EEG activities. *American Journal of EEG technology*, 25(2):83–92.

Chen, C., Liaw, A., and Breiman, L. (2004). Using random forest to learn imbalanced data. *University of California, Berkeley*, 110.

Cheng, M., Gao, X., Gao, S., and Xu, D. (2002). Design and implementation of a brain-computer interface with high transfer rates. *IEEE transactions on biomedical engineering*, 49(10):1181–1186.

Chiang, S., Levin, H. S., and Haneef, Z. (2015). Computer-automated focus lateralization of temporal lobe epilepsy using fMRI. *Journal of Magnetic Resonance Imaging*, 41(6):1689–1694.

Choi, K. and Cichocki, A. (2008). Control of a wheelchair by motor imagery in real time. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 330–337. Springer.

Clark, M. C., Hall, L. O., Goldgof, D. B., Velthuizen, R., Murtagh, F. R., and Silbiger, M. S. (1998). Automatic tumor segmentation using knowledge-based techniques. *IEEE transactions on medical imaging*, 17(2):187–201.

Clemens, B., Bánk, J., Piros, P., Bessenyei, M., Vető, S., Tóth, M., and Kondákor, I. (2008). Three-dimensional localization of abnormal EEG activity in migraine. *Brain topography*, 21(1):36–42.

Coito, A., Genetti, M., Pittau, F., Iannotti, G. R., Thomschewski, A., Höller, Y., Trinka, E., Wiest, R., Seeck, M., Michel, C. M., et al. (2016). Altered directed functional connectivity in temporal lobe epilepsy in the absence of interictal spikes: A high density EEG study. *Epilepsia*.

Coito, A., Plomp, G., Genetti, M., Abela, E., Wiest, R., Seeck, M., Michel, C. M., and Vulliemoz, S. (2015). Dynamic directed interictal connectivity in left and right temporal lobe epilepsy. *Epilepsia*, 56(2):207–217.

Cook, M. J., O'Brien, T. J., Berkovic, S. F., Murphy, M., Morokoff, A., Fabinyi, G., D'Souza, W., Yerra, R., Archer, J., Litewka, L., et al. (2013). Prediction of seizure likelihood with a long-term, implanted seizure advisory system in patients with drug-resistant epilepsy: a first-in-man study. *The Lancet Neurology*, 12(6):563–571.

Crosby, P. A., Daly, C. N., Money, D. K., Patrick, J. F., Seligman, P. M., and Kuzma, J. A. (1985). Cochlear implant system for an auditory prosthesis. US Patent 4,532,930.

Dähne, S., Höhne, J., and Tangermann, M. (2011). *Adaptive classification improves control performance in ERP-based BCIs*. na.

Davidson, R. J. and Hugdahl, K. (1996). *Brain asymmetry*. Mit Press.

de Peralta Menendez, R. G., Murray, M. M., Michel, C. M., Martuzzi, R., and Andino, S. L. G. (2004). Electrical neuroimaging based on biophysical constraints. *Neuroimage*, 21(2):527–539.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Devlaminck, D., Wyns, B., Grosse-Wentrup, M., Otte, G., and Santens, P. (2011). Multisubject learning for common spatial patterns in motor-imagery BCI. *Computational intelligence and neuroscience*, 2011:8.

Dierks, T., Ihl, R., Frölich, L., and Maurer, K. (1993). Dementia of the Alzheimer type: effects on the spontaneous EEG described by dipole sources. *Psychiatry Research: Neuroimaging*, 50(3):151–162.

Donchin, E. (1981). Surprise!... surprise? *Psychophysiology*, 18(5):493–513.

Dornhege, G., Milan, J. d. R., Hinterberger, T., McFarland, D. J., and Müller, K.-R. (2007). *Toward brain-computer interfacing*. MIT press.

DuMouchel, W. H. (1973). On the asymptotic normality of the maximum-likelihood estimate when sampling from a stable distribution. *The Annals of Statistics*, pages 948–957.

Fabiani, M., Gratton, G., Karis, D., and Donchin, E. (1987). Definition, identification, and reliability of measurement of the P300 component of the event-related brain potential. *Advances in psychophysiology*, 2(S 1):78.

Farwell, L. A. and Donchin, E. (1988). Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalography and clinical Neurophysiology*, 70(6):510–523.

Fazel-Rezai, R. (2007). Human error in P300 speller paradigm for brain-computer interface. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 2516–2519. IEEE.

Fazli, S., Popescu, F., Danóczy, M., Blankertz, B., Müller, K.-R., and Grozea, C. (2009). Subject-independent mental state classification in single trials. *Neural networks*, 22(9):1305–1312.

Field, A. (2009). *Discovering statistics using SPSS*. Sage publications.

Fisher, R. S., Boas, W. v. E., Blume, W., Elger, C., Genton, P., Lee, P., and Engel, J. (2005). Epileptic seizures and epilepsy: definitions proposed by the international league against epilepsy (ILAE) and the international bureau for epilepsy (IBE). *Epilepsia*, 46(4):470–472.

Focke, N. K., Yogarajah, M., Symms, M. R., Gruber, O., Paulus, W., and Duncan, J. S. (2012). Automated MR image classification in temporal lobe epilepsy. *Neuroimage*, 59(1):356–362.

Frei, E., Gamma, A., Pascual-Marqui, R., Lehmann, D., Hell, D., and Vollenweider, F. X. (2001). Localization of MDMA-induced brain activity in healthy volunteers using low resolution brain electromagnetic tomography (LORETA). *Human brain mapping*, 14(3):152–165.

French, J. A. (2007). Refractory epilepsy: clinical overview. *Epilepsia*, 48(s1):3–7.

Frye, G., Hauser, C., Townsend, G., and Sellers, E. (2011). Suppressing flashes of items surrounding targets during calibration of a P300-based brain-computer interface improves performance. *Journal of neural engineering*, 8(2):025024.

Furdea, A., Halder, S., Krusienski, D., Bross, D., Nijboer, F., Birbaumer, N., and Kübler, A. (2009). An auditory oddball (P300) spelling system for brain-computer interfaces. *Psychophysiology*, 46(3):617–625.

Garcia-Molina, G., Tsoneva, T., and Nijholt, A. (2013). Emotional brain-computer interfaces. *International journal of autonomous and adaptive communications systems*, 6(1):9–25.

Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236.

Gibert, G., Attina, V., Mattout, J., Maby, E., and Bertrand, O. (2008). *Size enhancement coupled with intensification of symbols improves P300 speller accuracy*. Citeseer.

Gonsalvez, C. J. and Polich, J. (2002). P300 amplitude is determined by target-to-target interval. *Psychophysiology*, 39(3):388–396.

Górriz, J., Ramírez, J., Lassl, A., Salas-Gonzalez, D., Lang, E., Puntonet, C., Álvarez, I., López, M., and Gómez-Río, M. (2008). Automatic computer-aided diagnosis tool using component-based SVM. In *Nuclear Science Symposium Conference Record, 2008. NSS'08. IEEE*, pages 4392–4395. IEEE.

Gotman, J. (1982). Automatic recognition of epileptic seizures in the EEG. *Electroencephalography and clinical Neurophysiology*, 54(5):530–540.

Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438.

Guan, C., Thulasidas, M., and Wu, J. (2004). High performance P300 speller for brain-computer interface. In *Biomedical Circuits and Systems, 2004 IEEE International Workshop on*, pages S3–5. IEEE.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Herrmann, M. J., Ehlis, A.-C., Muehlberger, A., and Fallgatter, A. J. (2005). Source localization of early stages of face processing. *Brain topography*, 18(2):77–85.

Hill, N., Schreiner, T., Puzicha, C., and Farquhar, J. (2008). BCPy2000. In *Workshop" Machine Learning Open-Source Software" at NIPS 2008*.

Höhne, J., Bartz, D., Hebart, M. N., Müller, K.-R., and Blankertz, B. (2016). Analyzing neuroimaging data with subclasses: a shrinkage approach. *NeuroImage*, 124:740–751.

Huang, J. and Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310.

Hübner, D., Verhoeven, T., Schmid, K., Müller, K.-R., Tangermann, M., and Kindermans, P.-J. (2017). Learning from label proportions in brain-computer interfaces: online unsupervised learning with guarantees. *PloS one*, 12(4):e0175856.

James, W. and Stein, C. (1961). Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379.

Jatoi, M. A., Kamel, N., Malik, A. S., Faye, I., and Begum, T. (2014). A survey of methods used for source localization using EEG signals. *Biomedical Signal Processing and Control*, 11:42–52.

Jayaram, V., Alamgir, M., Altun, Y., Scholkopf, B., and Grosse-Wentrup, M. (2016). Transfer learning in brain-computer interfaces. *IEEE Computational Intelligence Magazine*, 11(1):20–31.

Jin, J., Allison, B. Z., Sellers, E. W., Brunner, C., Horki, P., Wang, X., and Neuper, C. (2011). Optimized stimulus presentation patterns for an event-related potential EEG-based brain-computer interface. *Medical & biological engineering & computing*, 49(2):181–191.

Jin, J., Allison, B. Z., Wang, X., and Neuper, C. (2012). A combined brain–computer interface based on P300 potentials and motion-onset visual evoked potentials. *Journal of neuroscience methods*, 205(2):265–276.

Johannes, S., Münte, T., Heinze, H., and Mangun, G. R. (1995). Luminance and spatial attention effects on early visual processing. *Cognitive Brain Research*, 2(3):189–205.

Kamiya, K., Amemiya, S., Suzuki, Y., Kunii, N., Kawai, K., KUNIMATSU, A., SAITO, N., Shigeki, A., and OHTOMO, K. (2016). Machine learning of DTI structural brain connectomes for lateralization of temporal lobe epilepsy. *Magnetic Resonance in Medical Sciences*, 15(1):121–129.

Käthner, I., Wriessnegger, S. C., Müller-Putz, G. R., Kübler, A., and Halder, S. (2014). Effects of mental workload and fatigue on the P300, alpha and theta band power during operation of an ERP (P300) brain-computer interface. *Biological psychology*, 102:118–129.

Kaufmann, T., Schulz, S., Grünzinger, C., and Kübler, A. (2011). Flashing characters with famous faces improves ERP-based brain–computer interface performance. *Journal of neural engineering*, 8(5):056016.

Kerr, W. T., Nguyen, S. T., Cho, A. Y., Lau, E. P., Silverman, D. H., Douglas, P. K., Reddy, N. M., Anderson, A., Bramen, J., Salamon, N., et al. (2013). Computer-aided diagnosis and localization of lateralized temporal lobe epilepsy using interictal fdg-pet. *Frontiers in neurology*, 4.

Khalilia, M., Chakraborty, S., and Popescu, M. (2011). Predicting

disease risks from highly imbalanced data using random forest. *BMC medical informatics and decision making*, 11(1):51.

Kindermans, P.-J. (2014). *A Bayesian machine learning framework for true zero-training brain-computer interfaces*. PhD thesis, Ghent University.

Kindermans, P.-J., Schreuder, M., Schrauwen, B., Müller, K.-R., and Tangermann, M. (2014a). True zero-training brain-computer interfacing – an online study. *PloS one*, 9(7):e102504.

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

Kindermans, P.-J., Verschore, H., and Schrauwen, B. (2013). A unified probabilistic approach to improve spelling in an event-related potential-based brain-computer interface. *IEEE Transactions on Biomedical Engineering*, 60(10):2696–2705.

Kindermans, P.-J., Verstraeten, D., Buteneers, P., and Schrauwen, B. (2011). How do you like your P300 speller: adaptive, accurate and simple? In *5th International Brain-Computer Interface Conference (BCI-2011)*.

Kindermans, P.-J., Verstraeten, D., and Schrauwen, B. (2012). A bayesian model for exploiting application constraints to enable unsupervised training of a P300-based BCI. *PloS one*, 7(4):e33758.

Kleih, S., Nijboer, F., Halder, S., and Kübler, A. (2010). Motivation modulates the P300 amplitude during brain-computer interface use. *Clinical Neurophysiology*, 121(7):1023–1031.

Klem, G. H., Lüders, H. O., Jasper, H., Elger, C., et al. (1999). The ten-twenty electrode system of the international federation. *Electroencephalogr Clin Neurophysiol*, 52(3):3–6.

Koch, H., Christensen, J. A., Frandsen, R., Arvastson, L., Christensen, S. R., Sorensen, H. B., and Jennum, P. (2013). Classification of iRBD and Parkinson's patients using a general data-driven sleep staging model built on EEG. In *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, pages 4275–4278. IEEE.

Koenig, T. and Pascual-Marqui, R. (2009). Multichannel frequency and time-frequency analysis. *Electrical Neuroimaging*, pages 145–168.

Koessler, L., Cecchin, T., Colnat-Coulbois, S., Vignal, J.-P., Jonas, J., Vespignani, H., Ramantani, G., and Maillard, L. G. (2015). Catching the invisible: mesial temporal source contribution to simultaneous EEG and SEEG recordings. *Brain topography*, 28(1):5–20.

Krauledat, M., Tangermann, M., Blankertz, B., and Müller, K.-R. (2008). Towards zero training for brain-computer interfacing. *PloS one*, 3(8):e2967.

Krusienski, D. J., Grosse-Wentrup, M., Galán, F., Coyle, D., Miller, K. J., Forney, E., and Anderson, C. W. (2011). Critical issues in state-of-the-art brain-computer interface signal processing. *Journal of neural engineering*, 8(2):025002.

Krusienski, D. J., Sellers, E. W., Cabestaing, F., Bayoudh, S., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R. (2006). A comparison of classification techniques for the P300 speller. *Journal of neural engineering*, 3(4):299.

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

Laufs, H. (2012). Functional imaging of seizures and epilepsy: evolution from zones to networks. *Current opinion in neurology*, 25(2):194–200.

Laufs, H., Hamandi, K., Salek-Haddadi, A., Kleinschmidt, A. K., Duncan, J. S., and Lemieux, L. (2007). Temporal lobe interictal epileptic discharges affect cerebral activity in "default mode" brain regions. *Human brain mapping*, 28(10):1023–1032.

Laureys, S., Pellas, F., Van Eeckhout, P., Ghorbel, S., Schnakers, C., Perrin, F., Berre, J., Faymonville, M.-E., Pantke, K.-H., Damas, F., et al. (2005). The locked-in syndrome: what is it like to be conscious but paralyzed and voiceless? *Progress in brain research*, 150:495–611.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411.

Lehmann, E. L. (1999). *Elements of large-sample theory*. Springer Science & Business Media.

Lemm, S., Blankertz, B., Dickhaus, T., and Müller, K.-R. (2011). Introduction to machine learning for brain imaging. *Neuroimage*, 56(2):387–399.

Li, J., Liang, J., Zhao, Q., Li, J., Hong, K., and Zhang, L. (2013). Design of assistive wheelchair system directly steered by human thoughts. *International journal of neural systems*, 23(03):1350013.

Li, Y., Guan, C., Li, H., and Chin, Z. (2008). A self-training semi-supervised SVM algorithm and its application in an EEG-based brain-computer interface speller system. *Pattern Recognition Letters*, 29(9):1285–1294.

Liang, S.-F., Shaw, F.-Z., Young, C.-P., Chang, D.-W., and Liao, Y.-C. (2010). A closed-loop brain computer interface for real-time seizure detection and control. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 4950–4953. IEEE.

Liao, W., Zhang, Z., Pan, Z., Mantini, D., Ding, J., Duan, X., Luo, C., Lu, G., and Chen, H. (2010). Altered functional connectiv-

ity and small-world in mesial temporal lobe epilepsy. *PloS one*, 5(1):e8525.

Ling, C. X., Huang, J., and Zhang, H. (2003). AUC: a better measure than accuracy in comparing learning algorithms. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 329–341. Springer.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

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(2):R1.

Lotte, F. and Guan, C. (2011). Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms. *IEEE Transactions on biomedical Engineering*, 58(2):355–362.

Lu, S., Guan, C., and Zhang, H. (2009). Unsupervised brain-computer interface based on intersubject information and online adaptation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(2):135–145.

Lynall, M.-E., Bassett, D. S., Kerwin, R., McKenna, P. J., Kitzbichler, M., Muller, U., and Bullmore, E. (2010). Functional connectivity and brain networks in schizophrenia. *Journal of Neuroscience*, 30(28):9477–9487.

Mak, J., Arbel, Y., Minett, J. W., McCane, L. M., Yuksel, B., Ryan, D., Thompson, D., Bianchi, L., and Erdogmus, D. (2011). Optimizing the P300-based brain–computer interface: current status, limitations and future directions. *Journal of neural engineering*, 8(2):025003.

Marple, S. L. (1987). *Digital spectral analysis: with applications*, volume 5. Prentice-Hall Englewood Cliffs, NJ.

Martens, S., Hill, N., Farquhar, J., and Schölkopf, B. (2009). Overlap and refractory effects in a brain-computer interface speller based on the visual P300 event-related potential. *Journal of neural engineering*, 6(2):026003.

McFarland, D. J., Sarnacki, W. A., Townsend, G., Vaughan, T., and Wolpaw, J. R. (2011). The P300-based brain-computer interface (BCI): effects of stimulus rate. *Clinical neurophysiology*, 122(4):731–737.

McFarland, D. J., Sarnacki, W. A., and Wolpaw, J. R. (2003). Brain-computer interface (BCI) operation: optimizing information transfer rates. *Biological psychology*, 63(3):237–251.

McLachlan, G. and Krishnan, T. (2007). *The EM algorithm and extensions*, volume 382. John Wiley & Sons.

Mégevand, P., Spinelli, L., Genetti, M., Brodbeck, V., Momjian, S., Schaller, K., Michel, C. M., Vulliemoz, S., and Seeck, M. (2014). Electric source imaging of interictal activity accurately localises the seizure onset zone. *Journal of Neurology, Neurosurgery & Psychiatry*, 85(1):38–43.

Michel, C. M., Murray, M. M., Lantz, G., Gonzalez, S., Spinelli, L., and de Peralta, R. G. (2004). EEG source imaging. *Clinical neurophysiology*, 115(10):2195–2222.

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

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

Müller, K.-R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., and Blankertz, B. (2008). Machine learning for real-time single-trial EEG-analysis: from brain-computer interfacing to mental state monitoring. *Journal of neuroscience methods*, 167(1):82–90.

Müller-Putz, G. R., Scherer, R., Brauneis, C., and Pfurtscheller, G. (2005). Steady-state visual evoked potential (SSVEP)-based communication: impact of harmonic frequency components. *Journal of neural engineering*, 2(4):123.

Mwangi, B., Tian, T. S., and Soares, J. C. (2014). A review of feature reduction techniques in neuroimaging. *Neuroinformatics*, 12(2):229–244.

Nahum, L., Gabriel, D., Spinelli, L., Momjian, S., Seeck, M., Michel, C. M., and Schnider, A. (2011). Rapid consolidation and the human hippocampus: Intracranial recordings confirm surface EEG. *Hippocampus*, 21(7):689–693.

Olkin, I. and Spiegelman, C. H. (1987). A semiparametric approach to density estimation. *Journal of the American Statistical Association*, 82(399):858–865.

Ozcift, A. and Gulten, A. (2011). Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms. *Computer methods and programs in biomedicine*, 104(3):443–451.

Panicker, R. C., Puthusserypady, S., and Sun, Y. (2010). Adaptation in P300 brain-computer interfaces: a two-classifier cotraining approach. *IEEE Transactions on Biomedical Engineering*, 57(12):2927–2935.

Patrini, G., Nock, R., Caetano, T., and Rivera, P. (2014). (almost) no label no cry. In *Advances in Neural Information Processing Systems*, pages 190–198.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Pereira, F. R., Alessio, A., Sercheli, M. S., Pedro, T., Bilevicius, E., Rondina, J. M., Ozelo, H. F., Castellano, G., Covolan, R. J.,

Damasceno, B. P., et al. (2010). Asymmetrical hippocampal connectivity in mesial temporal lobe epilepsy: evidence from resting state fMRI. *BMC neuroscience*, 11(1):66.

Pfefferbaum, A., Wenegrat, B. G., Ford, J. M., Roth, W. T., and Kopell, B. S. (1984). Clinical application of the P3 component of event-related potentials. II. dementia, depression and schizophrenia. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 59(2):104–124.

Pfurtscheller, G. and Da Silva, F. L. (1999). Event-related EEG/MEG synchronization and desynchronization: basic principles. *Clinical neurophysiology*, 110(11):1842–1857.

Pfurtscheller, G., Müller-Putz, G. R., Scherer, R., and Neuper, C. (2008). Rehabilitation with brain-computer interface systems. *Computer*, 41(10).

Pichiorri, F., Morone, G., Petti, M., Toppi, J., Pisotta, I., Molinari, M., Paolucci, S., Inghilleri, M., Astolfi, L., Cincotti, F., et al. (2015). Brain-computer interface boosts motor imagery practice during stroke recovery. *Annals of neurology*, 77(5):851–865.

Pittau, F., Grova, C., Moeller, F., Dubeau, F., and Gotman, J. (2012). Patterns of altered functional connectivity in mesial temporal lobe epilepsy. *Epilepsia*, 53(6):1013–1023.

Plomp, G., Hervais-Adelman, A., Astolfi, L., and Michel, C. M. (2015). Early recurrence and ongoing parietal driving during elementary visual processing. *Scientific reports*, 5.

Plomp, G., Quairiaux, C., Michel, C. M., and Astolfi, L. (2014). The physiological plausibility of time-varying granger-causal modeling: normalization and weighting by spectral power. *NeuroImage*, 97:206–216.

Polikoff, J. B., Bunnell, H. T., and Borkowski Jr, W. J. (1995). Toward a P300-based computer interface. In *RESNA'95 Annual Conference and RESNAPRESS and Arlington Va*, pages 178–180.

Quadrianto, N., Smola, A. J., Caetano, T. S., and Le, Q. V. (2009). Estimating labels from label proportions. *Journal of Machine Learning Research*, 10(Oct):2349–2374.

Rakotomamonjy, A. and Guigue, V. (2008). BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller. *IEEE transactions on biomedical engineering*, 55(3):1147–1154.

Rowland, L. P. and Shneider, N. A. (2001). Amyotrophic lateral sclerosis. *New England Journal of Medicine*, 344(22):1688–1700.

Saeb, S., Lonini, L., Jayaraman, A., Mohr, D. C., and Kording, K. P. (2017). The need to approximate the use-case in clinical machine learning. *GigaScience*.

Salvaris, M. and Sepulveda, F. (2009). Visual modifications on the P300 speller BCI paradigm. *Journal of neural engineering*, 6(4):046011.

Samek, W., Vidaurre, C., Müller, K.-R., and Kawanabe, M. (2012). Stationary common spatial patterns for brain-computer interfacing. *Journal of neural engineering*, 9(2):026013.

Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., and Wolpaw, J. R. (2004). BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on biomedical engineering*, 51(6):1034–1043.

Schlögl, A. (2006). A comparison of multivariate autoregressive estimators. *Signal processing*, 86(9):2426–2429.

Schreuder, M., Höhne, J., Blankertz, B., Haufe, S., Dickhaus, T., and Tangermann, M. (2013). Optimizing event-related potential based brain-computer interfaces: a systematic evaluation of dynamic stopping methods. *Journal of neural engineering*, 10(3):036025.

Schreuder, M., Höhne, J., Treder, M., Blankertz, B., and Tangermann, M. (2011). Performance optimization of ERP-based BCIs using

dynamic stopping. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 4580–4583. IEEE.

Sellers, E. W., Krusienski, D. J., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R. (2006). A P300 event-related potential brain-computer interface (BCI): the effects of matrix size and inter stimulus interval on performance. *Biological psychology*, 73(3):242–252.

Sharanreddy, M. and Kulkarni, P. (2013). Automated EEG signal analysis for identification of epilepsy seizures and brain tumour. *Journal of medical engineering & technology*, 37(8):511–519.

Shenoy, P., Krauledat, M., Blankertz, B., Rao, R. P., and Müller, K.-R. (2006). Towards adaptive classification for BCI. *Journal of neural engineering*, 3(1):R13.

Staljanssens, W., Strobbe, G., Van Holen, R., Birot, G., Gschwind, M., Seeck, M., Vandenberghe, S., Vulliémoz, S., and van Mierlo, P. (2016). Seizure onset zone localization from ictal high-density EEG in refractory focal epilepsy. *Brain Topography*, pages 1–15.

Stretton, J., Pope, R., Winston, G., Sidhu, M., Symms, M., Duncan, J., Koepp, M., Thompson, P., and Foong, J. (2014). Temporal lobe epilepsy and affective disorders: the role of the subgenual anterior cingulate cortex. *Journal of Neurology, Neurosurgery & Psychiatry*, pages jnnp–2013.

Supekar, K., Menon, V., Rubin, D., Musen, M., and Greicius, M. D. (2008). Network analysis of intrinsic functional brain connectivity in Alzheimer's disease. *PLoS Comput Biol*, 4(6):e1000100.

Sutton, S., Braren, M., Zubin, J., and John, E. (1965). Evoked-potential correlates of stimulus uncertainty. *Science*, 150(3700):1187–1188.

Takano, K., Komatsu, T., Hata, N., Nakajima, Y., and Kansaku, K. (2009). Visual stimuli for the P300 brain-computer interface: a comparison of white/gray and green/blue flicker matrices. *Clinical neurophysiology*, 120(8):1562–1566.

Tangermann, M., Schreuder, M., Dähne, S., Höhne, J., Regler, S., Ramsay, A., Quek, M., Williamson, J., and Murray-Smith, R. (2011). Optimized stimulation events for a visual ERP BCI. *Int. J. Bioelectromagn*, 13(3):119–120.

Taylor, D. M., Tillery, S. I. H., and Schwartz, A. B. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science*, 296(5574):1829–1832.

Thompson, D. E., Quitadamo, L. R., Mainardi, L., Gao, S., Kindermans, P.-J., Simeral, J. D., Fazel-Rezai, R., Matteucci, M., Falk, T. H., Bianchi, L., et al. (2014). Performance measurement for brain-computer or brain-machine interfaces: a tutorial. *Journal of neural engineering*, 11(3):035001.

Townsend, G., LaPallo, B., Boulay, C., Krusienski, D., Frye, G., Hauser, C., Schwartz, N., Vaughan, T., Wolpaw, J., and Sellers, E. (2010). A novel P300-based brain-computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clinical neurophysiology*, 121(7):1109–1120.

Townsend, G., Shanahan, J., Ryan, D. B., and Sellers, E. W. (2012). A general P300 brain–computer interface presentation paradigm based on performance guided constraints. *Neuroscience letters*, 531(2):63–68.

Tzourio-Mazoyer, N., Landeau, B., Papathanassiou, D., Crivello, F., Etard, O., Delcroix, N., Mazoyer, B., and Joliot, M. (2002). Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage*, 15(1):273–289.

Vallabhaneni, A., Wang, T., and He, B. (2005). Brain-computer interface. In *Neural engineering*, pages 85–121. Springer.

van Mierlo, P., Carrette, E., Hallez, H., Raedt, R., Meurs, A., Vandenberghe, S., Roost, D., Boon, P., Staelens, S., and Vonck, K. (2013). Ictal-onset localization through connectivity analysis of intracranial EEG signals in patients with refractory epilepsy. *Epilepsia*, 54(8):1409–1418.

van Mierlo, P., Papadopoulou, M., Carrette, E., Boon, P., Vandenberghe, S., Vonck, K., and Marinazzo, D. (2014). Functional brain connectivity from EEG in epilepsy: Seizure prediction and epileptogenic focus localization. *Progress in neurobiology*, 121:19–35.

Vecchiato, G., Astolfi, L., Fallani, F. D. V., Salinari, S., Cincotti, F., Aloise, F., Mattia, D., Marciani, M. G., Bianchi, L., Soranzo, R., et al. (2009). The study of brain activity during the observation of commercial advertsing by using high resolution EEG techniques. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 57–60. IEEE.

Verhoeven, T., Buteneers, P., Wiersema, J.-R., Dambre, J., and Kindermans, P.-J. (2015). Towards a symbiotic brain-computer interface: exploring the application-decoder interaction. *Journal of Neural Engineering*, 12(6):066027.

Verhoeven, T., Hübner, D., Tangermann, M., Müller, K.-R., Dambre, J., and Kindermans, P.-J. (2017). Improving zero-training brain-computer interfaces by mixing model estimators. *Journal of Neural Engineering*, 14(3):036021.

Vidal, J. J. (1973). Toward direct brain-computer communication. *Annual review of Biophysics and Bioengineering*, 2(1):157–180.

Vidaurre, C., Kawanabe, M., von Bünau, P., Blankertz, B., and Müller, K.-R. (2011a). Toward unsupervised adaptation of LDA for brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 58(3):587–597.

Vidaurre, C., Sannelli, C., Müller, K.-R., and Blankertz, B. (2011b). Co-adaptive calibration to improve BCI efficiency. *Journal of neural engineering*, 8(2):025009.

Von Bünau, P., Meinecke, F. C., Király, F. C., and Müller, K.-R. (2009). Finding stationary subspaces in multivariate time series. *Physical review letters*, 103(21):214101.

Wilke, C., Van Drongelen, W., Kohrman, M., and He, B. (2009). Identification of epileptogenic foci from causal analysis of ECoG interictal spike activity. *Clinical Neurophysiology*, 120(8):1449–1456.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain-computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791.

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(3):252–259.

Wostyn, S., Staljanssens, W., De Taeye, L., Strobbe, G., Gadeyne, S., Van Roost, D., Raedt, R., Vonck, K., and van Mierlo, P. (2016). EEG derived brain activity reflects treatment response from vagus nerve stimulation in patients with epilepsy. *International Journal of Neural Systems*, page 1650048.

Wronkiewicz, M., Larson, E., and Lee, A. K. (2015). Leveraging anatomical information to improve transfer learning in brain-computer interfaces. *Journal of neural engineering*, 12(4):046027.

Xu, P., Yang, P., Lei, X., and Yao, D. (2011). An enhanced probabilistic LDA for multi-class brain computer interface. *PloS one*, 6(1):e14634.

Yang, Z., Choupan, J., Reutens, D., and Hocking, J. (2015). Lateralization of temporal lobe epilepsy based on resting-state functional magnetic resonance imaging and machine learning. *Frontiers in neurology*, 6:184.

Yin, E., Zhou, Z., Jiang, J., Chen, F., Liu, Y., and Hu, D. (2014). A speedy hybrid BCI spelling approach combining P300 and SSVEP. *IEEE Transactions on Biomedical Engineering*, 61(2):473–483.

Yin, E., Zhou, Z., Jiang, J., Yu, Y., and Hu, D. (2015). A dynamically optimized SSVEP brain-computer interface (BCI) speller. *IEEE Transactions on Biomedical Engineering*, 62(6):1447–1456.

Zhang, Z., Lu, G., Zhong, Y., Tan, Q., Liao, W., Wang, Z., Wang, Z., Li, K., Chen, H., and Liu, Y. (2010). Altered spontaneous neuronal activity of the default-mode network in mesial temporal lobe epilepsy. *Brain research*, 1323:152–160.