

**DISJUNCTIVE NORMAL UNSUPERVISED LDA FOR P300-BASED
BRAIN-COMPUTER INTERFACES**

by
Majed Elwardy

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of
Master of Science

Sabanci University

August 2016

DISJUNCTIVE NORMAL UNSUPERVISED LDA FOR P300-BASED
BRAIN-COMPUTER INTERFACES

APPROVED BY

Assoc. Prof. Dr. Müjdat ÇETİN
(Thesis Supervisor)

Assoc. Prof. Dr. Tolga TAŞDiZEN

Assoc. Prof. Dr. Kemal KILIÇ

DATE OF APPROVAL:

© Majed Elwardy 2016
All Rights Reserved

...to my wonderful wife Zaynab, without you I would have graduated two years ago.

Acknowledgments

I would like to thank my supervisor Müjdat Çetin for his guidance, motivation, suggestions and freedom encouragement throughout my graduate studies. It was a great and unforgettable experience to work with him.

I would like to thank Tolga Taşdizen for his guidance, precious suggestions and support on my thesis study and his participation in the Thesis committee.

I would like to thank Kemal Kılıç for his support on my graduate courses, motivation and his participation in the Thesis committee.

I am also thankful to TÜBİTAK for providing the financial support for my graduate education and my living support¹.

My special thanks to Ozan Özdenizci, Sezen Yağmur Günay, Mastaneh Torkamani Azar, and Abdullahi Adamu for sharing their BCI experiences with me, for their non-stop help and continuous support during my graduate. I am also thankful to all other members of SPIS group specially Oğuzcan Zengin, Muhammad Usman Ghani, and Muhammed Burak Alver for their unconditional help.

I owe special thanks to my parents (Mahmoud Elwardy & Manal Ali), my wife Zeinab, and my son Omar for their limitless love and support. Without them, I would never have achieved to this point. Although my son is 11 months old during my thesis defence, but he add a lot to me with his laughing, smiling, and crying.

¹This work has been supported by a graduate fellowship from the Scientific and Technological Research Council of Turkey (TÜBİTAK).

DISJUNCTIVE NORMAL UNSUPERVISED LDA FOR P300-BASED BRAIN-COMPUTER INTERFACES

Majed Elwardy

EE, M.Sc. Thesis, 2016

Thesis Supervisor: Müjdat Çetin

Keywords: Brain-computer interface, P300 Speller, calibration session, unsupervised classifier, unlabelled data, LDA, BLDA

Abstract

Can people use text-entry based brain-computer interface (BCI) systems and start a free spelling mode without any calibration session? Brain activities differ largely between people and across sessions for the same user. Thus, how can the text-entry system classify the target character among the other characters in the P300-based BCI speller matrix? In this thesis, we introduce a new unsupervised classifier for a P300-based BCI speller, which uses a disjunctive normal form representation to define an energy function involving a logistic sigmoid function for classification. Our proposed classifier updates the initialized random weights performing classification for the P300 signals from the recorded data exploiting the knowledge of the sequence of row/column highlights. To verify the effectiveness of the proposed method, we performed an experimental analysis on data from 7 healthy subjects, collected in our laboratory and used public BCI competition datasets. We compare the proposed unsupervised method to a baseline supervised linear discriminant analysis (LDA) classifier and Bayesian linear discriminant analysis (BLDA) and demonstrate its performance. Our analysis shows that the proposed approach facilitates unsupervised learning from unlabelled test data.

P300 TABANLI BEYİN-BİLGİSAYAR ARAYÜZLERİ İÇİN AYIRICI NORMAL GÖZETİMSİZ DAA

Majed Elwardy

EE, Yüksek Lisans Tezi, 2016

Tez Danışmanı: Müjdat Çetin

Anahtar Kelimeler: Beyin-bilgisayar arayüzü, P300 heceleyicisi, ayarlama oturumu, gözetimsiz sınıflandırıcı, etiketsiz veri, DAA, BDAA

Özet

İnsanlar metin yazma amaçlı beyin-bilgisayar arayüzü (BBA) sistemleri için ayarlama oturumuna ihtiyaç duymadan doğrudan heceleme moduna geçebilirler mi? Beyin aktiviteleri insanlar arasında ve aynı kullanıcının farklı oturumları arasında büyük değişkenlik göstermektedir. Bu durumda metin yazma sistemleri P300 tabanlı BBA heceleme matrisindeki hedef harfi diğerlerinden ayırt ederek nasıl sınıflandırabilir? Biz bu tezde P300 tabanlı BBA heceleyicileri için yeni bir gözetimsiz sınıflandırıcı öneriyoruz. Bu sınıflandırıcı lojistik sigmoid fonksiyonuna dayalı bir enerji fonksiyonu tanımlamak için ayırıcı normal form temsili kullanıyor. Önerdiğimiz sınıflandırıcı satır/sütunların parlaklaştırılarak vurgulanma dizisine dair bilgileri kullanarak, kaydedilen verilerden P300 sinyallerini sınıflandırmak için rastgele olarak başlatılan ağırlıkları günceller. Önerilen yöntemin geçerliliğini doğrulamak için kendi laboratuvarımızda toplanan ve kamuya açık BBA yarışması veri kümelerinde bulunan 7 sağlıklı kullanıcıya ait veriler üzerinde bir deneysel analiz gerçekleştirdik. Önerdiğimiz gözetimsiz yöntemi temel düzeyde birer gözetimli sınıflandırıcı olan doğrusal ayırtaç analizi (DAA) ve Bayesçi doğrusal ayırtaç analizi (BDAA) ile karşılaştırıp başarımını gösterdik. Analizimiz önerilen yaklaşımın etiketsiz test verilerinden gözetimsiz öğrenmeyi kolaylaştırdığını gösterdi.

Table of Contents

Acknowledgments	v
Abstract	vi
Özet	vii
1 Introduction	1
1.1 Scope	1
1.2 Motivation	3
1.3 Contributions	4
1.4 Outline	5
2 Background on BCI and P300 Spellers	6
2.1 Introduction	6
2.2 Electroencephalography (EEG) Signals	7
2.2.1 Electrodes	7
2.3 A Journey to General BCI Systems	10
2.3.1 Motor Imagery	10
2.3.2 Event Related Potentials	11
2.4 P300-based BCI Systems	14
2.4.1 Decoding the Brain Signals for A P300-based BCI System . .	18
2.5 Machine Learning for P300 Speller	19
2.5.1 Supervised Learning	19
2.5.2 Unsupervised Learning	21
2.6 Summary	24
3 Disjunctive Normal Unsupervised Classifier	25
3.1 Linear Discriminant Analysis (LDA)	25
3.2 Bayesian Linear Discriminant Analysis (BLDA)	26
3.3 Disjunctive Normal Unsupervised LDA classifier (DNUL)	29
3.3.1 Model Architecture	29
3.3.2 Model Initialization	30
3.3.3 Model Optimization	31
3.4 Regularized Disjunctive Normal Unsupervised LDA Classifier (RD- NUL)	31
3.5 Toy Examples for the Proposed Unsupervised Classifier	32

4	Offline and Simulated Online Analysis Experiments	38
4.1	Background	38
4.2	Terminology	39
4.3	P300 Classification Problem	40
4.4	Methods	41
	4.4.1 Data pre-processing	41
	4.4.2 Classification	43
4.5	Dataset	43
4.6	Experimental Setup	44
4.7	Experimental Results	45
	4.7.1 Offline Analysis	46
	4.7.2 Simulated Online Analysis	72
4.8	Comparison with the State-of-the-art in Unsupervised Classification in BCIs	78
5	Conclusion and Future Work	86
A	Datasets	89
B	Sensitivity Parameter (β)	92
	Bibliography	92

List of Figures

1.1	(a) A BCI-based motor imagery system. (b) A BCI-based P300 speller.	2
1.2	Interface of P300-based speller matrix used in this study.	3
2.1	A typical EEG based BCI system consists of electrodes, cables, amplifier, and a computer that processes the data. Taken from [1]. . . .	8
2.2	64-channel electrode cap using international 10-20 electrode distribution. Taken from [2]	9
2.3	Active electrodes and gel used in this study. Taken from [3]	9
2.4	The international 10-20 system. Taken from [1].	9
2.5	A typical BCI system model. Taken from [4].	11
2.6	Electrode placement layout according to the 10-20 electrode system. The dashed black circles show the location of highest amplitude of the P300 component. The golden electrodes are used in this work. CMS and DRL electrodes form a feedback loop, which drive the average potential of the subject (the Common Mode voltage) as close as possible to the ADC reference voltage. Taken from [5]	13
2.7	Average of (Subject 5) brain signals over trials following a visual stimulus obtained from different electrode sites. The solid red line is the average response of trials where a P300 wave is visible, the blue dashed line shows the average response of trials where no P300 wave is elicited.	14
2.8	First P300 speller paradigm used by Farwell and Donchin.	15
2.9	Two different paradigms (a) Hex-o-Spell interface. (b) RSVP keyboard interface.	16

2.10	The user interface of the AMUSE paradigm. Each circle encodes one out of six tones/tone directions relative to the user, who is positioned in the middle of the ring of speakers [6]	17
2.11	Elements of the user’s screen. Text To Spell indicates the pre-defined text. The speller will analyze evoked responses, and will append the selected text to Text Result. [7]	17
2.12	SU-BCI P300 Stimulus Software used in this study.	18
2.13	A hyperplane which separates two classes. Taken from [8]	20
2.14	Three examples of data generated by the LDA model. Note that in LDA both classes share the same covariance structure. To show the influence of the covariance structure on the direction of the decision boundary, we have used the same means per class in all three examples. By changing the covariance structure over the three examples, we rotate the decision boundary. An example can be seen in [9].	20
2.15	Example of data generated from a three component Gaussian Mixture Model. Note that unlike data generated by an LDA model, each cluster has its own covariance structure. Example in [9].	23
3.1	A toy example on a synthetic dataset with a standard deviation = 0.2. The DNUL classifier classified the data successfully with 100% classification accuracy. The second figure shows the energy function for the initialized 2 random-weight vectors for one of the classifiers among 10 classifiers (which is the highest).	34
3.2	A toy example on a synthetic dataset with a standard deviation = 0.4. The classifier achieved 97.76% classification accuracy with true positive rate = 94.23% and true negative rate = 97.69%.	35
3.3	A toy example on a synthetic dataset with a standard deviation = 0.5. The classifier achieved 95.19% classification accuracy with true positive rate = 82.69% and true negative rate = 98.46%.	35
3.4	A toy example on a synthetic dataset with a standard deviation = 0.6. The classifier achieved 91.67% classification accuracy with true positive rate = 71.15% and true negative rate = 95.76%.	36

3.5	A toy example on a synthetic dataset with a standard deviation = 0.7. The classifier achieved 90.71% classification accuracy with true positive rate = 69.23% and true negative rate = 95%.	36
3.6	A toy example on a synthetic dataset with a standard deviation = 0.8. The classifier achieved 87.18% classification accuracy with true positive rate = 63.46% and true negative rate = 91.92%.	37
3.7	A toy example on a synthetic dataset with a standard deviation = 0.9. The classifier achieved 86.56% classification accuracy with true positive rate = 61.53% and true negative rate = 91.53%.	37
4.1	Offline analysis results for subject 1 with (Batch-26) configuration. . .	47
4.2	Offline analysis results for subject 2 with (Batch-26) configuration. . .	48
4.3	Offline analysis results for subject 3 with (Batch-26) configuration. . .	48
4.4	Offline analysis results for subject 4 with (Batch-26) configuration. . .	48
4.5	Offline analysis results for subject 5 with (Batch-26) configuration. . .	49
4.6	Offline analysis results for subject 6 with (Batch-26) configuration. . .	49
4.7	Offline analysis results for subject 7 with (Batch-26) configuration. . .	49
4.8	Average classification performance over 7 subjects with (Batch-26). Error shadows show 95% confidence intervals from the mean with sample size = 7.	50
4.9	Offline analysis results for subject 1 with (Batch-14) configuration. . .	51
4.10	Offline analysis results for subject 2 with (Batch-14) configuration. . .	51
4.11	Offline analysis results for subject 3 with (Batch-14) configuration. . .	52
4.12	Offline analysis results for subject 4 with (Batch-14) configuration. . .	52
4.13	Offline analysis results for subject 5 with (Batch-14) configuration. . .	52
4.14	Offline analysis results for subject 6 with (Batch-14) configuration. . .	53
4.15	Offline analysis results for subject 7 with (Batch-14) configuration. . .	53
4.16	Average classification performance over 7 subjects with (Batch-14). Error shadows show 95% confidence intervals from the mean with sample size = 7.	53
4.17	Offline analysis results for subject 1 with (N-Batch-26) configuration.	55
4.18	Offline analysis results for subject 2 with (N-Batch-26) configuration.	55
4.19	Offline analysis results for subject 3 with (N-Batch-26) configuration.	55

4.20	Offline analysis results for subject 4 with (N-Batch-26) configuration.	56
4.21	Offline analysis results for subject 5 with (N-Batch-26) configuration.	56
4.22	Offline analysis results for subject 6 with (N-Batch-26) configuration.	56
4.23	Offline analysis results for subject 7 with (N-Batch-26) configuration.	57
4.24	Average classification performance over 7 subjects with (N-Batch-26). Error shadows show 95% confidence intervals from the mean with sample size = 7.	57
4.25	Offline analysis results for subject 1 with (N-Batch-14) configuration.	58
4.26	Offline analysis results for subject 2 with (N-Batch-14) configuration.	59
4.27	Offline analysis results for subject 3 with (N-Batch-14) configuration.	59
4.28	Offline analysis results for subject 4 with (N-Batch-14) configuration.	59
4.29	Offline analysis results for subject 5 with (N-Batch-14) configuration.	60
4.30	Offline analysis results for subject 6 with (N-Batch-14) configuration.	60
4.31	Offline analysis results for subject 7 with (N-Batch-14) configuration.	60
4.32	Average classification performance over 7 subjects with (N-Batch-14). Error shadows show 95% confidence intervals from the mean with sample size = 7.	61
4.33	Average classification performance comparing the regularized DNUL classifier with DNUL. Error shadows corresponding to the point show 95% confidence intervals from the mean with sample size = 7.	62
4.34	Bit rate performance comparing the regularized DNUL classifier with DNUL. Error shadows corresponding to the point show 95% confidence intervals from the mean with sample size = 7.	63
4.35	Average classification performance for subject A over 5 classifier groups using 12 electrodes with a batch mode configuration.	65
4.36	Average classification performance for subject A over 5 classifier groups using 64 electrodes with a batch mode configuration.	65
4.37	Classification performance for subject A using 12 electrodes with a N-batch mode configuration.	66
4.38	Classification performance for subject A using 64 electrodes with a N-batch mode configuration.	66

4.39	Average classification performance for subject B over 5 classifier groups using 12 electrodes with a batch mode configuration.	67
4.40	Average classification performance for subject B over 5 classifier groups using 64 electrodes with a batch mode configuration.	67
4.41	Classification performance for subject B using 12 electrodes with a N-batch mode configuration.	68
4.42	Classification performance for subject B using 64 electrodes with a N-batch mode configuration.	68
4.43	Average classification performance for subject C over 5 classifier groups using 12 electrodes with a batch mode configuration.	69
4.44	Average classification performance for subject C over 5 classifier groups using 64 electrodes with a batch mode configuration.	69
4.45	Classification performance for subject C using 12 electrodes with a N-batch mode configuration.	70
4.46	Classification performance for subject C using 64 electrodes with a N-batch mode configuration.	70
4.47	An arbitrary example shows a sequence of letters for simulated online spelling.	72
4.48	Simulated online spelling (sequential mode) showing the performance averaged over the 7 subjects.	74
4.49	Simulated online spelling using 12 electrodes and 64 electrodes for subject A.	75
4.50	Simulated online spelling using 12 electrodes and 64 electrodes for subject B.	76
4.51	Simulated online spelling using 12 electrodes and 64 electrodes for subject C.	77
B.1	A sensitivity parameter analysis showing the classifier performance averaged over the 7 subjects. The x-axis represents the beta value (β). The y-axis represents the classifier accuracy. The shaded band shows the standard error from the mean. The vertical green dashed line intercepts the beta value at 0.1 which gives the maximum classifier accuracy among all values.	92

List of Tables

2.1	Methods used for brain-computer interfaces. Taken from [10].	7
3.1	Influence of the tuning parameter.	32
4.1	Percentage of correctly classified characters for each subject obtained with different values of trial groups for (Batch-26).	47
4.2	Percentage of correctly classified characters for each subject obtained with different values of trial groups for (Batch-14).	50
4.3	Percentage of correctly classified characters for each subject obtained with different values of trial groups for (N-Batch-26).	54
4.4	Percentage of correctly classified characters for each subject obtained with different values of trial groups for (N-Batch-14).	58
4.5	Batch mode: percentage of correctly classified characters for subjects A, B, and C for 12 electrodes. The values in braces are the standard deviation.	71
4.6	Batch mode: percentage of correctly classified characters for subjects A, B, and C for 64 electrodes. The values in braces are the standard deviation.	71
4.7	N-Batch mode: percentage of correctly classified characters for subjects A, B, and C for 12 and 64 electrodes configuration.	72
4.8	Averaged accuracies of our proposed unsupervised classifier (DNUL). The numbers show the percentage of correctly classified characters. The values in braces are the standard deviation.	80
4.9	Averaged accuracies of the competing unsupervised classifier. The numbers show the percentage of correctly classified characters. The values in braces are the standard deviation. Taken from Kindermans et al. [11]	81

4.10	Accuracies of different supervised classifiers. The numbers show the percentage of correctly classified characters. eSVM, SUP, and OA-SUP are taken from Kindermans et al. [11].	81
4.11	OFF-US: percentage of correctly classified characters through 10 classifier groups.	83
4.12	OFF-US-T: percentage of correctly classified characters through 10 classifier groups.	83
4.13	ON-US-T: percentage of correctly classified characters through 10 classifier groups.	84
4.14	OA-US-T: percentage of correctly classified characters through 10 classifier groups.	84
4.15	RE-OA-US-T: percentage of correctly classified characters through 10 classifier groups.	85
4.16	OA-US: percentage of correctly classified characters through 10 classifier groups.	85
A.1	Target words in training and test datasets for SU datasets	90
A.2	Target words in training and test datasets for BCI Competition II	90
B.1	Sensitivity parameter (β): SU Dataset (S1)	93
B.2	Sensitivity parameter (β): SU Dataset (S2)	94
B.3	Sensitivity parameter (β): SU Dataset (S3)	95
B.4	Sensitivity parameter (β): SU Dataset (S4)	96
B.5	Sensitivity parameter (β): SU Dataset (S5)	97
B.6	Sensitivity parameter (β): SU Dataset (S6)	98
B.7	Sensitivity parameter (β): SU Dataset (S7)	99

Chapter 1

Introduction

Across all ages and cultures, people are doing their best to use multiple means to communicate and control, from the beginning of creation to the current era. Talking, writing, and gesture have been the most common ways to interact with each other throughout time. Interestingly, in each era, intellectuals thought about potential barriers and how to overcome them, at least theoretically. For example, a short time ago, back in the 60s, communicating and controlling devices with brain waves was listed as science fiction. In today's world, it became true and people can communicate with the computer through brain waves. A brain-computer interface (BCI) aims to establish a direct communication channel between the brain and a computer or machine so disabled individuals can interact with the real-world [12].

In this thesis, we introduced a new unsupervised classifier for a P300-based BCI, which tackle some main problems for the text entry systems. These problems can be briefly addressed under the calibration sessions where they are tedious, time-consuming, and annoying sessions for the subjects especially the disabled individuals. Furthermore, we demonstrate the use of our proposed approach on offline and simulated online analysis in order to verify the effectiveness of the proposed method.

1.1 Scope

Technology should always improve life quality. A significant number of individuals suffer from losing all voluntary muscle control due to amyotrophic lateral sclerosis (ALS), traumatic brain injuries, or spinal cord injuries [13]. Although the motor pathway is lost, neuronal activity of the brain still works in many of these cases. Therefore, one direction for raising the life quality of the disabled individuals

is to create a channel between a brain and a computer which it can be used for various applications. Thus, BCI returns hope to many people.

Over the last two decades, a large body of work has been performed for recording activity from the brain either invasively or non-invasively for the purpose of brain-computer interfacing. The electroencephalogram (EEG) is a non-invasive technique involving electrical signals measured through the scalp and can be used as the cornerstone for BCI [14]. Along with EEG [15], magnetoencephalography (MEG) [16], positron emission topography (PET) [17], functional magnetic resonance imaging (fMRI) [18], and optical imaging, functional near infrared spectroscopy (fNIRS) [19] provide other ways to monitor brain activity non-invasively. In an EEG-based BCI system, incoming signals from an EEG amplifier are processed and classified to decode the user's intent [20]. Furthermore, it can be used to provide input signals in many applications including text entry systems [21], robotic arm control [22], and cursor control [23]. Figure 1.1 shows how data are collected from subjects.

One of the most common application related to BCI is the text-entry systems. They allow subjects to select characters from a symbolic grid matrix containing characters and symbols on a computer screen while recording the brain waves. The P300 speller is one of the most common BCI-based text-entry systems, which allows subjects to write text on the computer screen. Farwell and Donchin [21] demonstrated the first P300 speller paradigm which is also called the oddball paradigm. P300 is an event-related potential (ERP) elicited in the brain as a response to a visual or auditory stimulus. It is a positive deflection measured around the parietal

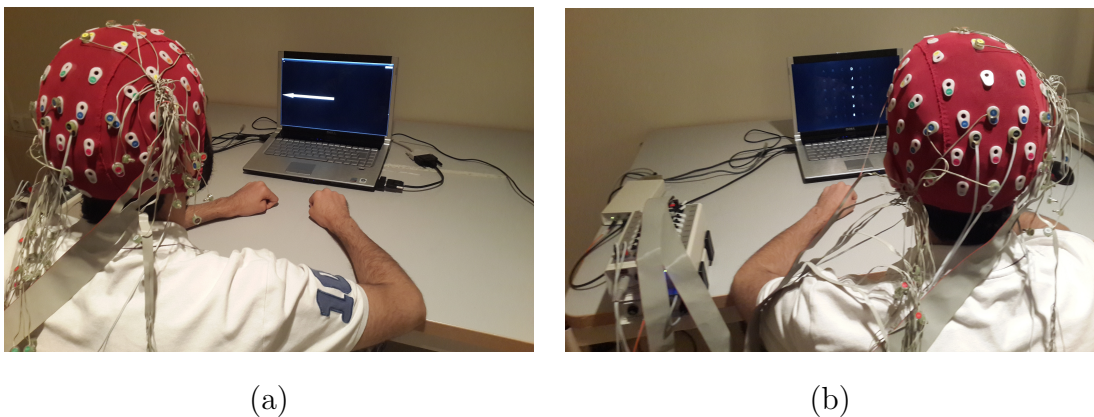


Figure 1.1: (a) A BCI-based motor imagery system. (b) A BCI-based P300 speller.



Figure 1.2: Interface of P300-based speller matrix used in this study.

lobe, nearly 300 ms to 600 ms after the occurrence of the attended stimulus [24]. The system allows people to spell words and numbers by focusing on the desired character or number in a matrix shown on the screen (see Figure 1.2). When the desired character is highlighted, the subject attends to the unexpected stimulus and a P300 wave is generated. The character which the user intends to type can be inferred from the intersection of the detected P300 responses in the sequence of row/column highlights. Machine learning algorithms can be used to classify and learn the attended and non-attended highlights for rows and columns. Thus, the character can be estimated from the intersection of attended highlights.

EEG signals suffer from low signal to noise ratio (SNR) due to several factors including the variability in brain activities, changes in electrode positions in long sessions, meta-activities in the brain, and artifacts due to eye movements and muscular activities. Therefore, P300 spellers need several stimulus repetitions to increase the classification accuracy [25] [26].

1.2 Motivation

One of the most common problems in BCIs is the calibration process. Subjects have to go through tedious, time-consuming and annoying calibration sessions before they can start using a BCI system for communication purposes. The brain signals vary across people and across sessions for the same user [27]. For this reason, supervised training methods based on calibration sessions involving labelled training data are usually used. Furthermore, the BCI system should be trained for a specific

person. The downsides of having to use such sessions include the consumption of additional time and increased fatigue for the users. Even for healthy people the calibration is still an annoying process. Furthermore, such sessions might have to be repeated to account for any non-stationary behaviour of the brain signals over the course of system use. The aforementioned problems imply another inherent problem, namely the collected training data may sometimes be unreliable. For example, during the data collection for this thesis, many users reported that they felt sleepy, lost concentration, and probably could not focus on the target letters. Mainly the labelled data is what we expect the user to write, not the user actual writes. Healthy subjects can report mistakes and express feelings during the experiment, then we can decide how reliable the data are. On the other hand, paralyzed people can not express when they made mistakes and it is hard to measure the reliability of the data.

There have been few pieces of work on unsupervised methods for P300-based BCI spellers to tackle the problems raised above. An unsupervised method was proposed by Lu et al. [28]. Although that unsupervised classifier has also been applied to P300 data, it still needs some labelled data from many previous subjects to train a subject independent classification model (SICM), which allows EEG from a new subject to be classified first by the SICM then goes through adaptation process. Another recent unsupervised classification method, based on a Bayesian model, has been proposed by Kindermans et al. [11]. The classifier can be trained unsupervisedly using an Expectation Maximization (EM) approach, eliminating the use of calibration sessions. Up to my knowledge, it was the only paper which is able to train a P300 classifier without any labelled data. There also exist semi-supervised adaptation methods which involve supervised training followed by adaptation of the classifier with the incoming EEG data [29].

1.3 Contributions

The work done in this thesis provides a contribution towards addressing the mentioned problems by proposing a new unsupervised classifier for P300-based spellers. In this approach, the disjunctive normal form plays a role in forming an energy function, which allows to update the randomly initialized classifier weights by using

the logistic sigmoid function for classification and by exploiting the knowledge of the sequence of row/column highlights [30]. The idea is that one round of row/column highlights in the speller matrix should evoke a P300 response only after two (one row and one column) of the highlights. Note that exploiting this fact does not require knowledge of the labels of the data, hence this idea can be a basis for unsupervised learning.

To achieve this goal, we propose a disjunctive normal unsupervised LDA for P300-based brain-computer interfaces ¹. The thesis makes several contributions, which can be summarized as follows. We developed a novel unsupervised method based on the disjunctive normal form for P300-based BCI speller systems, which allows us to run the classifier without using any calibration process and without any labelled data. Moreover, we demonstrated the use of our proposed approach on both offline and simulated online analysis experiments. Besides, we compared our classifier with BCI competition datasets (BCI Competition II [32] and BCI competition III [33]).

1.4 Outline

Chapter 2 presents introductory background information about BCI, P300 speller paradigm, stimulus software used in this work, and a survey of machine learning techniques used for P300.

Chapter 3 presents the proposed unsupervised classifier and all the supervised classification technical pieces involved in this work together with their mathematical preliminaries.

Chapter 4 presents the offline and simulated online experiments for P300 speller, in order to demonstrate the proposed classifier effectiveness. The performance and detailed report results can be found in this chapter.

Chapter 5 provides a summary of the contributions made and indicates possible directions for future work, motivated by the limitations and advantages of the proposed methods.

¹A preliminary portion of this work was published at the IEEE 24th Signal Processing and Communication Application Conference (SIU) 2016. [31]

Chapter 2

Background on BCI and P300 Spellers

This chapter aims to provide the basic concepts of EEG signals processing, brain-computer interfaces (BCIs), P300 signals of event-related potentials (ERP) and P300-based spellers, motor imagery systems, and classification methods in both supervised and unsupervised domains. It also includes a survey of published works, methods, and results.

2.1 Introduction

BCI might use brain signals recorded by a variety of methodologies. These include invasive and non-invasive methods. Scalp-recorded EEG provides the most practical and widely used non-invasive access to the brain activity. However, its signal resolution is low. On the other hand, using invasive techniques such as electrocorticography (ECoG), require access to the cortical surface of the brain. Although this technique provides high resolution signals, it is prohibitively expensive and might involve risks for the patient [20].

Most BCIs rely on sensors outside the head to measure the electrical activity of the brain. Magnetoencephalography (MEG), functional magnetic resonance imaging (fMRI), and positron emission tomography (PET) are non-invasive rather than (ECoG) which is an invasive. However, they are expensive, hard to utilise outside the laboratory, and not applicable in daily life [20]. In contrast, EEG is relatively cheap, applicable with many different paradigms and offers non-muscular communication and control mechanisms. Table 2.1 shows the methods used for brain-computer Interfaces.

	EEG	MEG	fMRI	ECoG
Deployment	Noninvasive	Noninvasive	Noninvasive	Invasive
Measured activity	Electrical	Magnetic	Hemodynamic	Electrical
Temporal resolution	Medium	Medium	Low	High
Spatial resolution	Low	Low	Medium	Medium
Portability	High	Low	Low	High
Cost	Low	High	High	High

Table 2.1: Methods used for brain-computer interfaces. Taken from [10].

2.2 Electroencephalography (EEG) Signals

EEG is one of the well-known technique due to its advantages of low cost, non-invasiveness, portability, and ease of use. Because of that, it has become the most commonly used BCI signal acquisition tool. EEG has been mainly used for clinical diagnosis of neurological disorders. EEG signals provide a good temporal resolution (in milliseconds). It measures the electrical activity as voltage changes on the scalp via the electrodes attached to it. Hans Berger introduced human EEG in 1929 [34]. However, EEG is not without disadvantages: EEG signals suffer from poor spatial resolution because of measuring the electrical activities from the scalp as it is an invasive method.

To establish a BCI channel, the electrodes must be placed on the scalp by the cap to detect the EEG signals (5-20 microvolt range). Then, it can be connected to the amplifiers to magnify the EEG signals. Finally, actual brain signals are recorded by a device and converted to a digital format. A typical EEG based BCI system is illustrated in Figure 2.1.

2.2.1 Electrodes

One of the most important components of BCI systems are electrodes. Electrodes are little pin-pads of Ag/AgCl attached to the scalp with the aid of a headcap consisting of an elastic cap with plastic, electrode holders. An example of the headcap used in our laboratory can be seen in Figure 2.2. In order to decrease

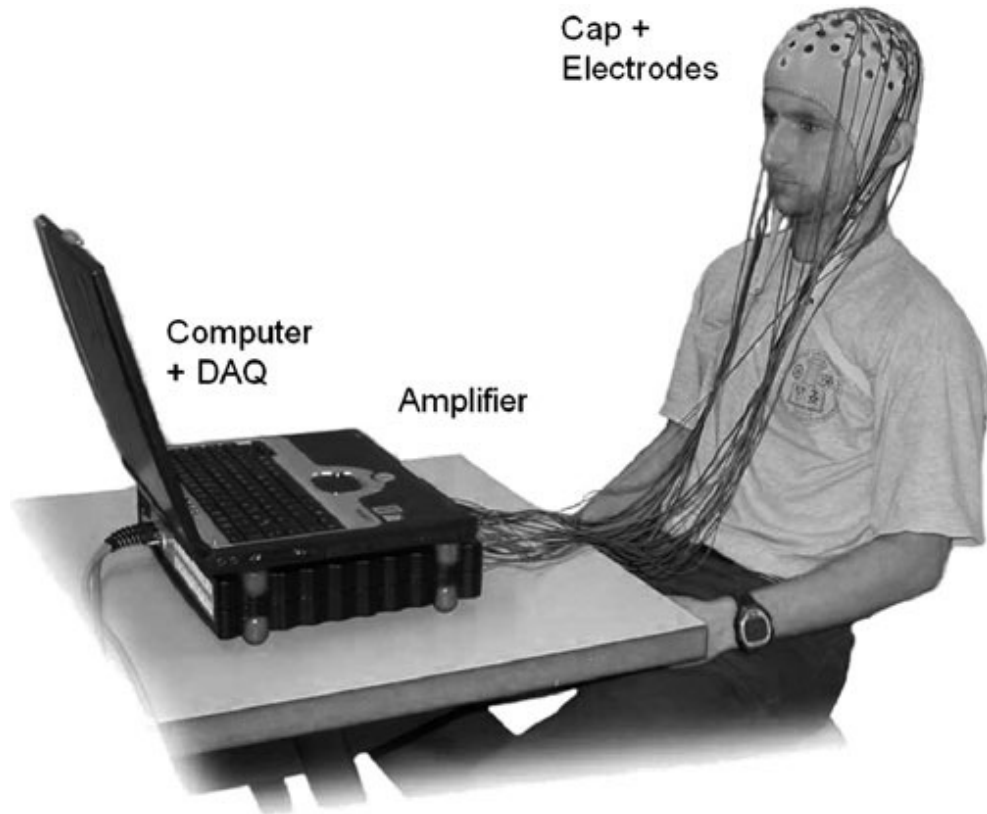


Figure 2.1: A typical EEG based BCI system consists of electrodes, cables, amplifier, and a computer that processes the data. Taken from [1].

the skin resistance or voltage offset and to have a stable, stationary conductive medium for proper measurements, usually a conductive gel is applied to fill the plastic holes before clicking the active electrodes as shown in Figure 2.3. However, electromagnetic interference, power cable noise, and signal degradation, need for skin preparation, etc., are problems for practical usage of these electrodes outside the laboratory [35]. To reduce some of these effects, high active electrode impedance and cable shielding is used as shown in Figure 2.3.

The electrodes are placed on the head of the subject according to an international system called 10-20 system, proposed by the American EEG society [36]. It is widely used in clinical EEG recording and EEG research as well as BCI research. This system proposes that the electrodes are placed in a 10% - 20% distance from each other with respect to the total distance between the nasion and inion of the subject. The labels of the electrode sites are usually also the labels of the recorded channels. Figure 2.4 depicts the electrode placement according to the 10 - 20 system.



Figure 2.2: 64-channel electrode cap using international 10-20 electrode distribution. Taken from [2]

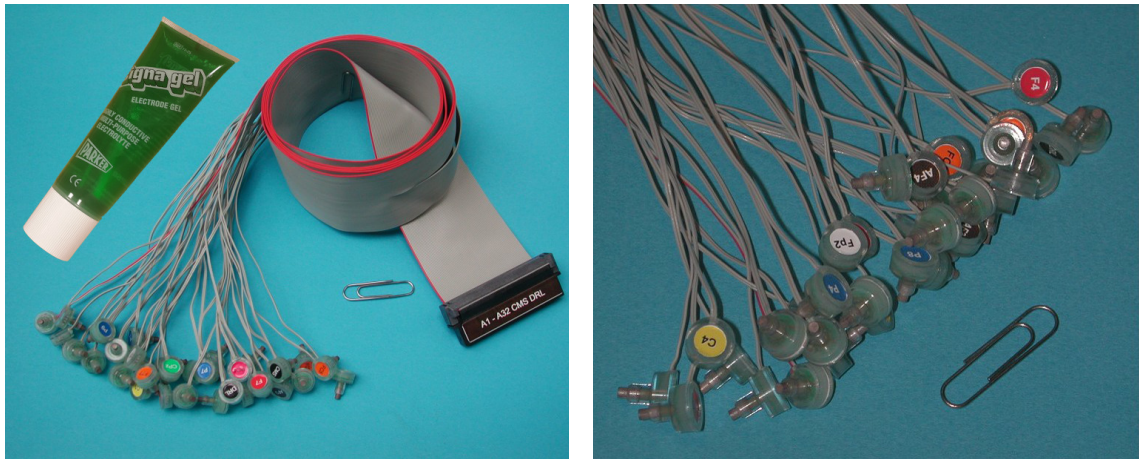


Figure 2.3: Active electrodes and gel used in this study. Taken from [3]

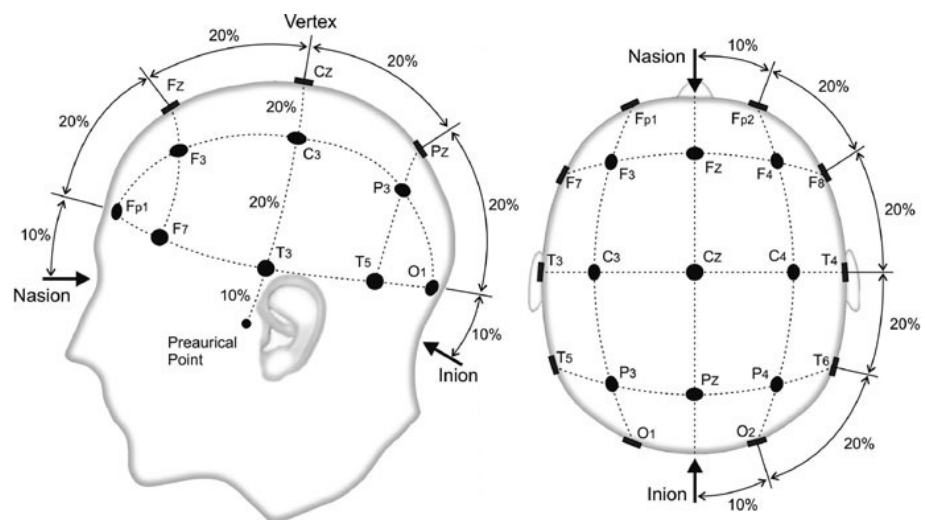


Figure 2.4: The international 10-20 system. Taken from [1].

2.3 A Journey to General BCI Systems

In order to activate the neurons in the brain to elicit electrical signals, it does not require an interference or an effort from the user. Daily life activities such as thinking, moving, feeling something, etc., unconsciously do that itself. Furthermore, the neurons will translate the activities of the physiological and pathological information in terms of electrical signals [37]. Those electrical signals recorded by electrodes can be measured and acquired by a bio-signal acquisition system. Then the signals are analyzed to translate the brain activities into command signals using a computer algorithm. These commands can be used to provide input signals in many applications including text entry [21], robotic arm control [22], cursor control [23]. As shown in Figure 2.5, the BCI system usually consists of three main parts: Signal Acquisition, Signal Processing, and Application Interface.

Signal Acquisition involves collecting temporal EEG data and amplifying brain signals. It uses active or passive electrodes to transmit the electrical activity of the subject to a high sensitivity, low-noise amplifier, namely the EEG amplifier. The Signal Processing part then processes the acquired brain signals in three steps sequentially: data pre-processing, feature extraction, and classification or detection. The processed data is transmitted into the Application Interface part for further control of external devices or any other useful clinical applications like treating stroke, autism, and other disorders.

2.3.1 Motor Imagery

Many BCI systems rely on imagined movement. The brain activity changes in the EEG recording either with real or imagined movement. That advance the ability of people to record and use BCIs with their mental state tasks. The EEG signals are recorded multiple times from the brain processes. The information is averaged over the different recordings to filter out redundant brain activity and to keep the relevant information [38]. Most of the motor imagery data belong to two classes such as mentally thinking about moving left and right hand[39]. Besides, it allows subjects to have binary communication to choose between two categories by just thinking of right and left hand movements.

The EEG has many regular rhythms. The most well-known are the occipital

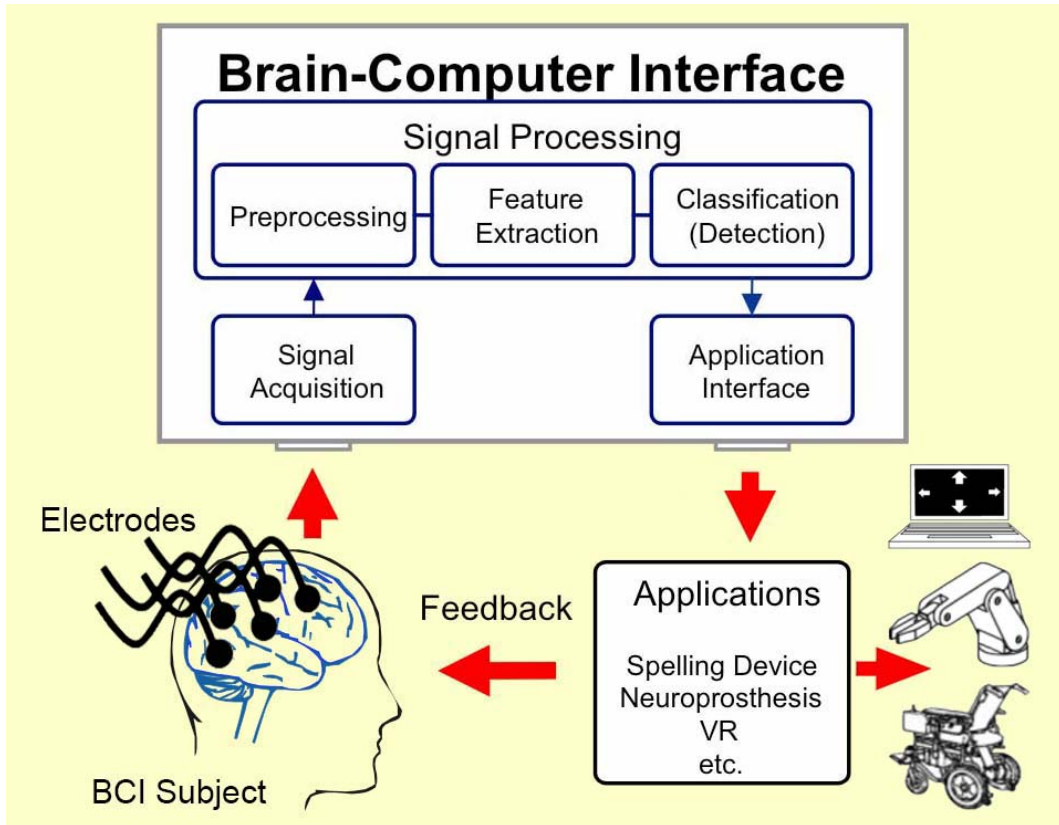


Figure 2.5: A typical BCI system model. Taken from [4].

alpha rhythm and the central mu and beta rhythms. In motor imagery studies, spectral power densities around 16-24 Hz for beta, 12-16 Hz for sigma, 8-12 Hz for alpha bands are used. A calibration session or adaptivity method is needed to train or adapt the computer to differentiate between the different classes.

2.3.2 Event Related Potentials

An event-related potential (ERP) is a specific brain response to an event such as the presentation of a visual or auditory stimulus (e.g., a specific flash or sound), a mistake, a motor event [40]. It is an unconsciously stereotyped brain wave recorded on the scalp. ERPs can be measured before, during or after sensory, motor or psychological events and usually have a fixed time delay after (or before) these events, named stimuli. In 1964, Walter et al. discovered that when a subject was required to press a button after detecting a target in a visual stimulus, they elicited a large negative voltage at frontal electrodes that happen just before the subject presses the button [41]. This voltage, ERP component called Contingent Negative

Variation (CNV) indicated the subject's mental preparation to press the button. One of the most extensively used ERP component in BCI research is P300, usually named P300 component discovered by Sutton et al. in 1965 [42]. This thesis is focused and completely related to the P300 component. This component will be discussed in the next section.

P300 component

P300 is an event-related potential (ERP) elicited in the brain as a response to a visual or auditory stimulus. It is a large positive deflection measured around the parietal lobe, nearly 300 ms to 600 ms after the occurrence of the attended stimulus [24]. This idea generated another paradigm known as the 'oddball paradigm', where the subject is stimulated with two categories of events: relevant and irrelevant [43]. The relevant events occur rarely with respect to irrelevant events, and due to the complete random order of events, elicit a large P300 response in ERPs. For the first time in 1988, Farwell and Donchin used the oddball paradigm to devise a communication system which allows users to type letters on a screen by thoughts with P300 signals rather than using muscular output [21].

Donchin and his colleagues developed a 6×6 matrix of letters, numbers, and/or other symbols. The individual rows and columns flash in a block-randomized fashion and the user attends to the desired item and counts how many items it flashes. Here the row and column which contain the target letter are the relevant events or target stimuli, while other are irrelevant events or non-target stimuli.

The P300 component is located along the mid-line scalp sites. The highest amplitudes of the P300 component can be recorded from central-parietal (Cz) and mid-frontal (Fz) electrode locations [44]. Figure 2.6 shows the electrode placement layout used in this work according to the 10-20 electrode placement system. Figure 2.7 shows a typical P300 response of a single trial and averaged over trials recorded at several electrode sites used in this work.

As can be seen from that section, recording the P300 component looks easy. Nevertheless, the quality of P300 signals is affected by various factors as following:

- Positioning a cap toward the correct location plays a role with signals quality including P300 components.

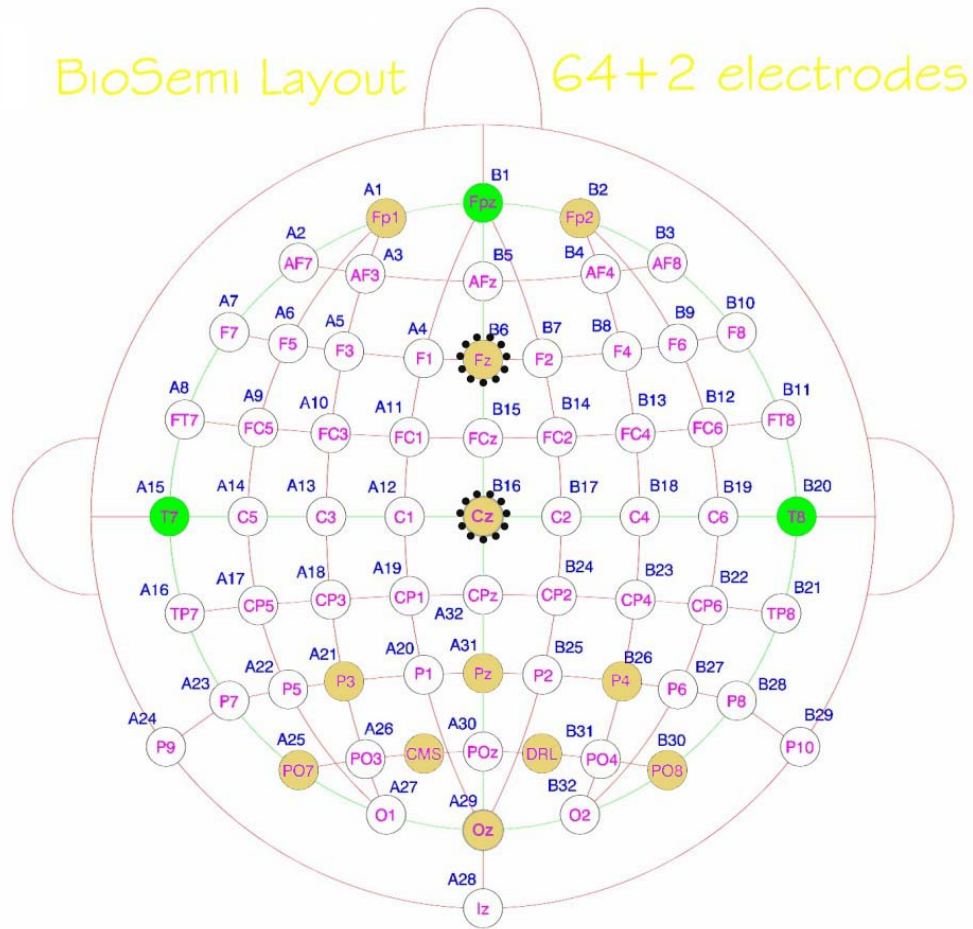


Figure 2.6: Electrode placement layout according to the 10-20 electrode system. The dashed black circles show the location of highest amplitude of the P300 component. The golden electrodes are used in this work. CMS and DRL electrodes form a feedback loop, which drive the average potential of the subject (the Common Mode voltage) as close as possible to the ADC reference voltage. Taken from [5]

- A subject's mental state, emotion, psychological activities, degree of fatigue, and concentration will all affect the result of P300 recordings.
- The recording environment of the temporal EEG data will also influence the final acquisition of a P300 signal. The surrounding noise should be reduced in order to achieve a high quality P300 signal. A P300 signal is always averaged by several measurements due to its small amplitude (in μV).

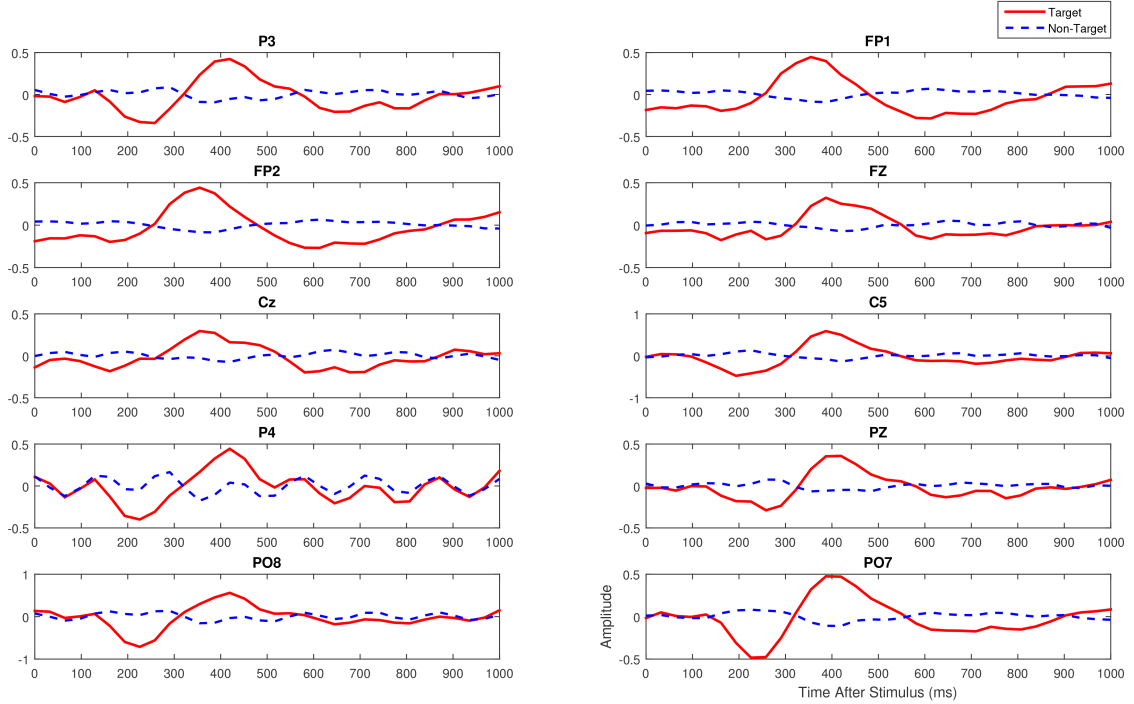


Figure 2.7: Average of (Subject 5) brain signals over trials following a visual stimulus obtained from different electrode sites. The solid red line is the average response of trials where a P300 wave is visible, the blue dashed line shows the average response of trials where no P300 wave is elicited.

2.4 P300-based BCI Systems

The first ERP-based BCI system that was produced is the P300-based BCI by Farwell and Donchin (1988), also known as the matrix speller. The matrix speller consists of a 6×6 grid of symbols which are shown on the computer screen. Figure 2.8 shows the first prototype of the P300 speller paradigm. Donchin’s first P300 speller has become the most widely studied P300 based BCI system [21]. It can be seen from the Figure 2.8 that it has 36 cells, involves 26 letters, 6 commands, and 4 symbols. The task here is to spell the word “B-R-A-I-N” letter by letter using the matrix speller paradigm shown at the top of Figure 2.8. By focussing attention on a specific symbol, the subject is able to select that symbol. The speller matrix is covered by a random flash sequence. If the the row or column with the target character is flashed, then a P300 signal will occur. The other flashes with the non-target characters are then considered as irrelevant targets and no P300

MESSAGE					
BRAIN					
Choose one letter or command					
A	G	M	S	Y	*
B	H	N	T	Z	*
C	I	O	U	*	TALK
D	J	P	V	FLN	SPAC
E	K	Q	W	*	BKSP
F	L	R	X	SPL	QUIT

Figure 2.8: First P300 speller paradigm used by Farwell and Donchin.

signals will appear. Besides, using feature selection and classification techniques for P300 signals aid the system to predict the character and display it on the computer screen. In a few cases the computer displayed a letter other than the one in which a subject was focusing, then the subject focused on the BKSP (backspace) command to delete the character in order to correct the error. At the end, the subject selected the TALK command and a computer read the spelled word [21].

P300-based BCI systems and related technologies have been highly developed and improved recently. Many paradigms including flashing, stimulus techniques, and interfaces have been introduced. One example of this is the hexagonal two-level hierarchy of the Berlin BCI known as “Hex-o-Spell” [45], where characters are clustered based on hexagons appealing visualization as shown in Figure 2.9 (a). Another introduced paradigm is the rapid serial visual presentation (RSVP) keyboard. It allows visual stimulus sequences to be displayed on a screen over time on a fixed focal area and in rapid succession [46]. RSVP keyboard paradigm can be seen in Figure 2.9 (b).

An auditory ERP-based paradigm, called AMUSE, is introduced in [47]. In AMUSE, each stimulus is a specific tone that originates from a specific direction. Kindermans et al., used 6 unique tones forming auditory stimuli, each produced by one of six speakers that are positioned on a ring of 130 cm diameter around the user. Figure 2.10 shows the user interface used in this study [6].

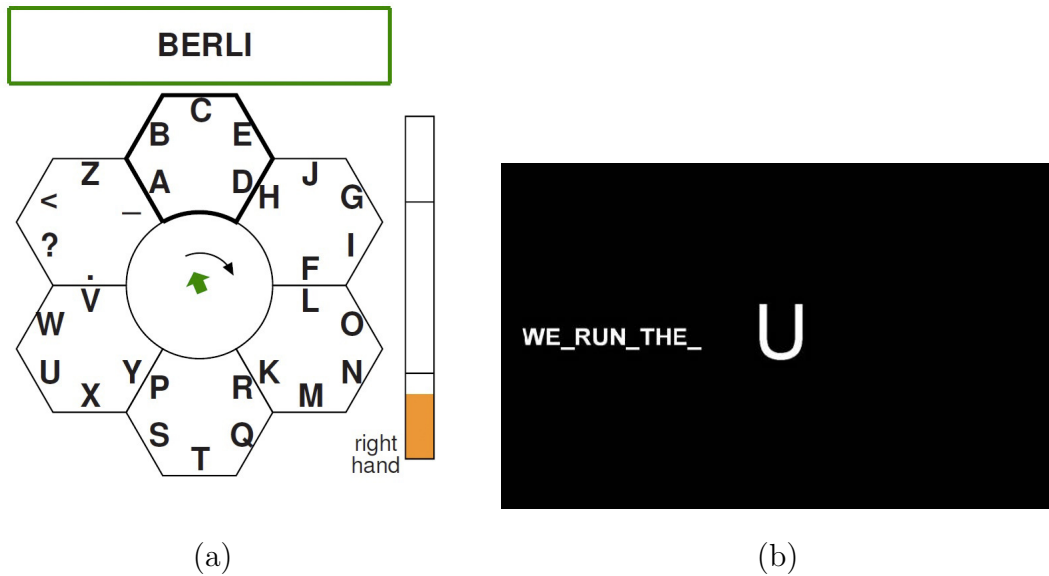


Figure 2.9: Two different paradigms (a) Hex-o-Spell interface. (b) RSVP keyboard interface.

Another popular software tool based on BCI spelling is BCI2000. It is a complete package of tools used by many EEG research groups around the world. It was first developed by the members of Schalk laboratory and presented in [48]. BCI2000 can incorporate alone or in combination with any brain signals, signal processing methods, output devices, and operating protocols. See Figure 2.11.

In this work, the SU-BCI P300 stimulus software is used to deliver the subject with the required visuals, or directions, to evoke the desired stimulus. SU-BCI was previously developed at the Signal Processing and Information System (SPIS) Laboratory [49]. It is essentially a matrix based system similar to the one introduced by Donchin. The software allows any matrix size, cell content customization (letters or shapes), various colouring, stimulation schemes, and timings as shown in Figure 2.12.

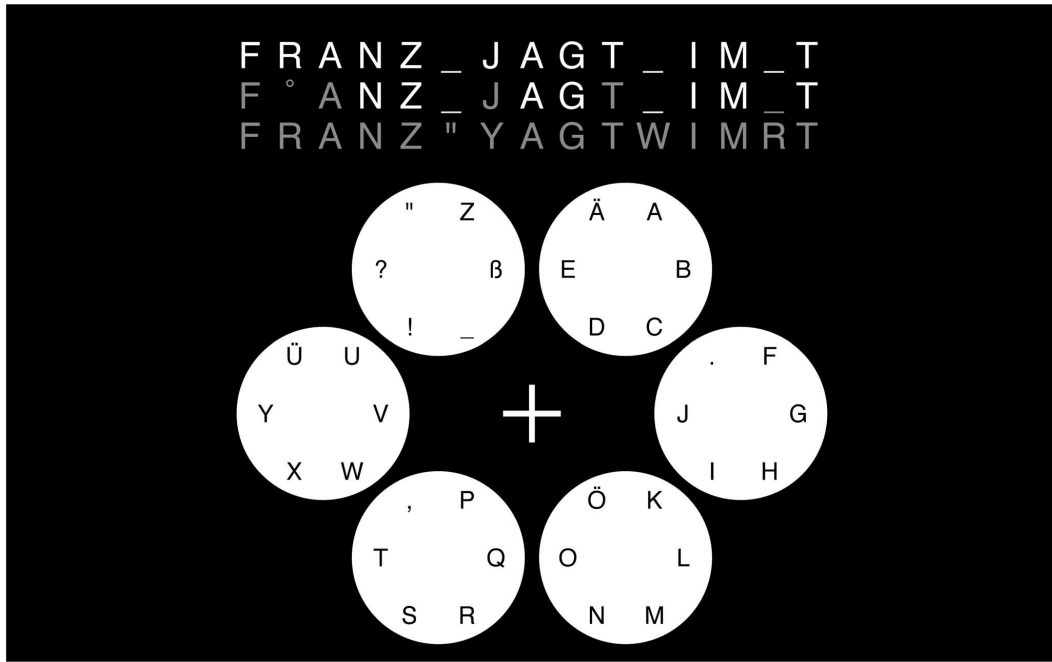


Figure 2.10: The user interface of the AMUSE paradigm. Each circle encodes one out of six tones/tone directions relative to the user, who is positioned in the middle of the ring of speakers [6]

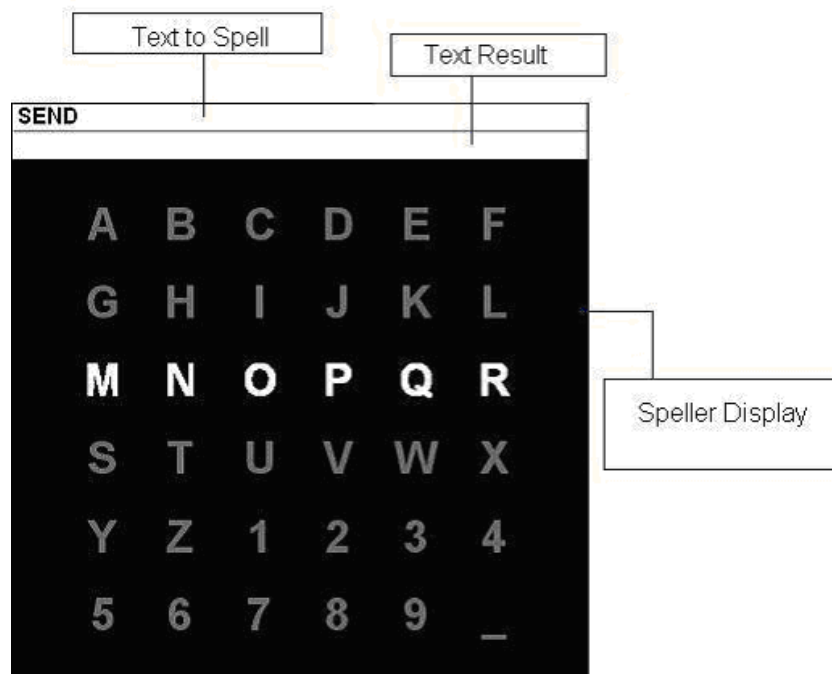


Figure 2.11: Elements of the user's screen. Text To Spell indicates the pre-defined text. The speller will analyze evoked responses, and will append the selected text to Text Result. [7]

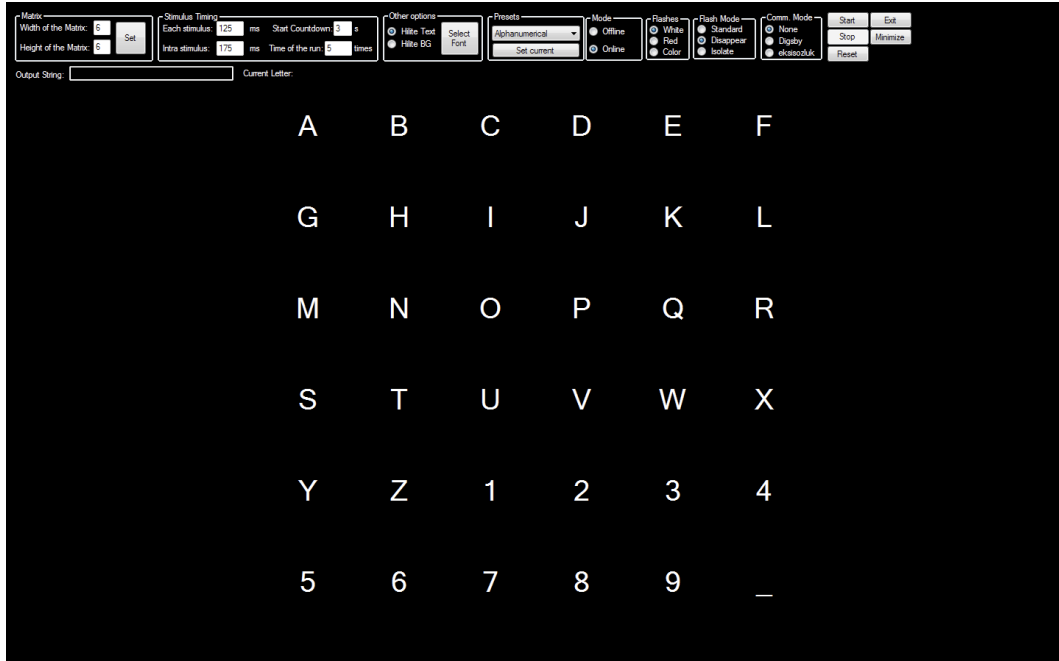


Figure 2.12: SU-BCI P300 Stimulus Software used in this study.

2.4.1 Decoding the Brain Signals for A P300-based BCI System

The goal of a BCI is not only recording a raw data, but also the recording should be informative and contains meaningful signals. Decoding the user’s P300 component from the EEG data reflects the characteristic parameters of the subject. Further, it can be decided if the recording is done in a well prepared setup atmosphere or not. After decoding the signals, the extracted features are converted to commands to control external devices through computer algorithms matching with the task [44].

Recorded EEG data require three main steps in order to process the data: data pre-processing, classification, and post processing (detection). Raw EEG data are converted to digital signals using an ADC converter. The digital signals are pre-processed for classification. First, the signals are band-pass filtered and decimated or sub-sampled by a factor to eliminate the artifacts. Noise and artifacts refer to information that reduces the signal quality. The signals are divided into one-second epochs corresponding to individual row and column flashes which are used as the feature vectors for classification. Features might be peaks, actual or special waveforms or deflections at specific times, spectral density, etc. In order to obtain a good feature representation, a feature extraction process might be applied. In

this thesis, the amplitude of the signals represents the feature vector [49]. Second, the classification step, the classifier learned the P300 wave pattern by giving the formed feature vector to the classifier. Finally, the extracted features are translated into P300 detection for every row and column and combined to detect the desired character. The character which the user intends to type can be inferred from the intersection of the detected P300 responses in the sequence intersection of row and column highlights.

2.5 Machine Learning for P300 Speller

The goal of this section is to introduce the most closely related machine learning methods used in P300 speller in BCI context. Classical machine learning methods are mainly divided into the following groups: supervised, semi-supervised, and unsupervised methods. Previous works have been done in our laboratory for developing supervised and semi-supervised algorithms for the P300 Speller [37] [50]. The methods developed in this thesis are for unsupervised methods.

2.5.1 Supervised Learning

We start with the most common machine learning models, those models require labelled data for training. Furthermore, test data can be applied to the learned method for validation through the model accuracy. In this section, some of the supervised classifiers are briefly presented with their mathematical preliminaries such as: Linear Discriminant Analysis (LDA), Fisher's Linear Discriminant Analysis (FLDA), and Bayesian Linear Discriminant Analysis (BLDA).

Linear Discriminant Analysis

The aim of Linear Discriminant Analysis (LDA) is to use hyperplanes to separate the data representing the different classes. For a two-class problem, LDA looks for a linear combination of features that characterizes or separates two classes (see Figure 2.13) [51] [52]. Where x is the feature vector and w is a classification weight vector. LDA assumes a normal distribution of the data, with equal covariance matrices for both classes (see Figure 2.14) [8].

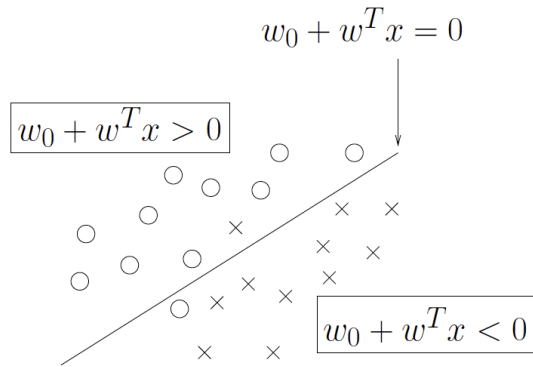


Figure 2.13: A hyperplane which separates two classes. Taken from [8]

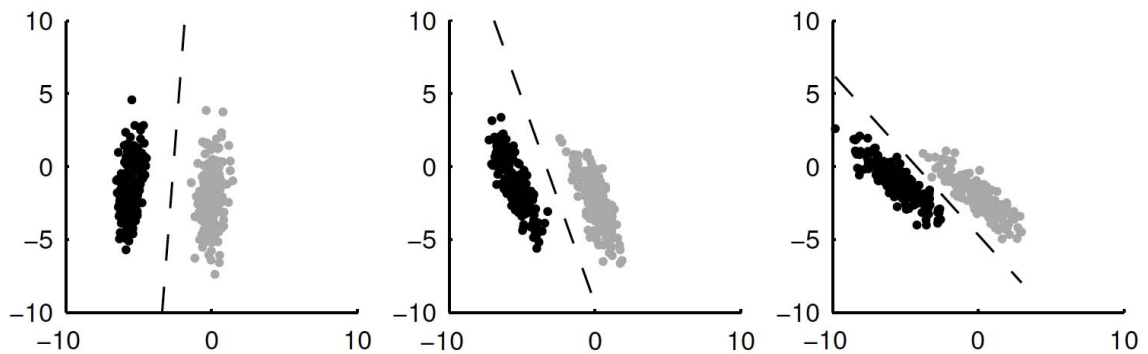


Figure 2.14: Three examples of data generated by the LDA model. Note that in LDA both classes share the same covariance structure. To show the influence of the covariance structure on the direction of the decision boundary, we have used the same means per class in all three examples. By changing the covariance structure over the three examples, we rotate the decision boundary. An example can be seen in [9].

The separating hyperplane is obtained by seeking the projection that maximizes the distance between the two classes means and minimizes the interclass variance [52]. This classifier is simple to use and generally provides good results with a very low computational requirement. Consequently, LDA has been used with success in the P300 speller [53] [54]. More practical descriptions are given in Chapter 3.

Fisher's Linear Discriminant Analysis

Fisher's linear discriminant analysis (FLDA) is the benchmark method for determining the optimal separating hyperplane between two classes [55]. It uses a

different weight calculation process compared to LDA. Fisher's LDA aims at finding a set of weights w that maximize the ratio:

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \quad (2.1)$$

where S_B is the scatter matrix between classes and S_w is the scatter matrix within a class. S_B and S_w are defined as:

$$S_B = \sum_c (\mu_c - \bar{x})(\mu_c - \bar{x})^T \quad (2.2)$$

$$S_w = \sum_c \sum_{i \in c} (\bar{x}_i - \mu_c)(\bar{x}_i - \mu_c)^T \quad (2.3)$$

FLDA is simple in calculations and provides a robust classification when the two classes are Gaussian with equal covariance [54]. A detailed description of FLDA is given in Appendix A of [12]. This method has been extensively used in P300 studies [56].

Bayesian Linear Discriminant Analysis

BLDA can be seen as an extension of Fisher's Linear Discriminant Analysis (FLDA). In contrast to FLDA, in BLDA regularization is used to prevent overfitting to high dimensional and possibly noisy datasets. Through a Bayesian analysis the degree of regularization can be estimated automatically and quickly from training data without the need for time consuming cross-validation [12]. Besides, BLDA is one of the main classifiers which has been used widely in BCI applications [57]. A detailed description of BLDA is shown in Chapter 3.

2.5.2 Unsupervised Learning

Unsupervised learning methods are used to discover hidden structures or to exploit known patterns in the data. Those methods do not require labelled data and they start learning directly with the unlabelled data. In this section, some of the unsupervised classifiers are briefly presented with their mathematical preliminaries such as: K-means clustering, Gaussian Mixture Models (GMM), and Expectation Maximization (EM).

K-means Clustering

K-means is an efficient unsupervised learning algorithm that groups the data into clusters. The aim of the K-means algorithm is to divide M points in N dimensions into K clusters so that the within-cluster sum of squares is minimized, where K is chosen before the algorithm starts pointing to the number of clusters. K-means clustering algorithm is described in detail by Hartigan (1975) [58]. It is also called Lloyd's algorithm [59].

K-means clustering is one of the most popular clustering techniques due to its simplicity and efficiency in speed. The first step in K-means algorithm is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because different locations causes different results, hence it is a non-deterministic algorithm. The better choice is to place them as far away from each other as possible [60]. The algorithm tries to minimize the objective function:

$$\arg \min_{\mathbf{c}} \sum_{i=1}^k \sum_{\mathbf{x} \in c_i} \|\mathbf{x} - \mu_i\|_2^2 \quad (2.4)$$

where c_i is the set of points that belong to cluster i . The K-means clustering typically uses Euclidean distance metric for computing the distance between data points and the cluster centers. It can also uses different distance separation measures [61].

The main disadvantage of that method is that it needs to assign the initial seeds to start the algorithm. Furthermore, k-means++ has been developed to start the algorithm automatically without the need of specifying the initial seeds. It is a method to initialize the number of k cluster which is given as an input to the k-means algorithm. Since choosing the right value for k centroids in advance is difficult, this algorithm provides a method to find the value for k centroids before proceeding to cluster the data [62].

Gaussian Mixture Models

A two-component Gaussian Mixture model (GMM) is almost identical to the LDA model discussed in supervised learning section, the only difference is that a GMM assumes that each group of data-points has its own covariance matrix and

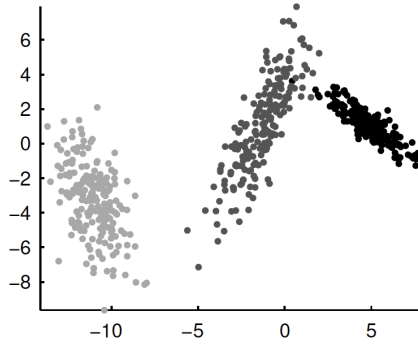


Figure 2.15: Example of data generated from a three component Gaussian Mixture Model. Note that unlike data generated by an LDA model, each cluster has its own covariance structure. Example in [9].

that GMM models are trained unsupervised without any label information. In a GMM model, there are N data points x_n , where the generative model specifies that each data point belongs to one of the k groups (or clusters). The data in group k is distributed as a multivariate Gaussian with mean μ_k and covariance Σ_k (see Figure 2.15). Apart from that, the LDA parameters are selected using maximum likelihood; this is not possible for GMM, therefore the Expectation Maximization algorithm is applied to select the parameters for GMM [9].

Expectation Maximization

The Expectation Maximization (EM) framework can be used to optimize latent variable models of missing or hidden data, such as the GMM, where it is difficult to maximize the likelihood [63]. Each iteration of the EM algorithm consists of two processes: The expectation step, and the maximization step. In the expectation, or E-step, the missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E-step are used instead of the actual missing data. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration [64].

2.6 Summary

A general introduction and background knowledge about the BCI topics have been covered in this chapter. Especially, some concepts are described such as: EEG signals, recording concepts, Event-related potential (ERP), P300 component, and P300-based BCI systems. P300-based BCI took most of the focus including information about the application, interface, working principles, and flashing paradigms, since the P300 speller is one of the most common BCI-based text-entry systems. The general procedure needed for the subject to type letters with thoughts through the brain signals was outlined in this chapter. A survey of classification techniques have has been briefly introduced. We mentioned the most classification techniques used before with the P300 speller for supervised and unsupervised learning. We introduced Linear discriminant Analysis, Fisher's Linear Discriminant Analysis, and Bayesian Linear Discriminant Analysis as examples for supervised learning. For unsupervised learning techniques we mentioned K-means, Gaussian Mixture Models, and Expectation Maximization. Some of these algorithms are going to be used in the following chapters to develop, analyze, and compare the proposed classifier for P300 speller based BCI.

In this work, we aim at proposing a new unsupervised classifier for P300-based spellers which allow us to run the classifier without using any calibration process and without any labelled data. In addition, it will be compared with the main supervised classifiers to demonstrate its effectiveness.

Chapter 3

Disjunctive Normal Unsupervised Classifier

In this chapter, we present in detail the proposed methodology of the developed unsupervised classifier. Several supervised classifiers are introduced in detail to give the reader a background on the used supervised classifiers such as Linear discriminant analysis (LDA) and Bayesian Linear Discriminant Analysis (BLDA). Afterwards, we introduce some techniques used for improving the proposed unsupervised classifier. At the end of this chapter, toy examples are presented for the proposed unsupervised classifier on a synthetic data to demonstrate the effectiveness.

3.1 Linear Discriminant Analysis (LDA)

LDA supervised classifier has been used as a baseline classifier model for comparison with the proposed unsupervised method. LDA can be derived from simple probabilistic models which model the class conditional distribution of the data $P(X|y = k)$ for each class k . Predictions can then be obtained by using Bayes' rule:

$$P(y = k|X) = \frac{P(X|y = k)P(y = k)}{P(X)} = \frac{P(X|y = k)P(y = k)}{\sum_K P(X|y = k)P(y = k)} \quad (3.1)$$

We choose the class k which maximizes the conditional probability. $P(X|y = k)$ can be modelled as a multivariate Gaussian distribution with density:

$$P(X|y = k) = \frac{1}{(2\pi)^n |\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(X - \mu_k)\Sigma_k^{-1}(X - \mu_k)^t\right) \quad (3.2)$$

The estimate of the class mean and shared covariance matrix for unweighted data are:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{y_i=k} x_i \quad (3.3)$$

$$\hat{\Sigma} = \frac{1}{n - k} \sum_{k=1}^K \sum_{y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T \quad (3.4)$$

this leads to linear decision surface and can used to predict the classes with the learned weights [65].

Consider a two-class classification problem $\mathbf{K} = \{0, 1\}$, where $k = 1$ corresponds to row/column containing the target letter and $k = 0$ corresponds to row/column not containing the target letter. Given the EEG data (\mathbf{X}) , where the model assumes (\mathbf{X}) has a Gaussian distribution. The model has the same covariance matrix for each class; only the means vary as mentioned in Chapter 2. Under this modelling assumption, the classifier infers the mean and covariance parameter of each class; it computes the sample mean of each class. Then, it computes the sample covariance by first subtracting the sample mean of each class from the observations of that class, and taking the empirical covariance matrix of the result.

In this work, we calculate the estimated mean for each class and the shared covariance matrix. Then the classifier generates the weight vector which is a linear combination of the components of x and used for classification decisions to predict the classes by using the using of Sigmoid function that has real-valued and a differentiable function which produces a curve with an S shape and takes the value 0.5 in the middle of the classification line between the two classes. Derivatives of the sigmoid function are employed in learning algorithms.

3.2 Bayesian Linear Discriminant Analysis (BLDA)

The other supervised classifier used for comparison with the proposed unsupervised classifier is BLDA. The algorithm was proposed in [12], and the actual code was developed by Ulrich Hoffmann of the EPFL BCI group in 2006. This section exactly follows the summary given in Appendix B of [12]. The extended explanation can be founded in [66].

BLDA can be seen as an extension of Fisher's Linear Discriminant Analysis (FLDA) described in (Chapter 2). In contrast to FLDA, in BLDA regularization is

used to prevent overfitting to high dimensional and possibly noisy datasets. Through a Bayesian analysis, the degree of regularization can be estimated automatically and quickly from training data without the need for the time consuming cross-validation process.

Least squares regression is equivalent to FLDA if regression targets are set to N/N_1 for examples from class 1 and to $-N/N_2$ for examples from class -1; where N is the total number of training examples, N_1 is the number of examples from class 1 and N_2 is the number of examples from class -1. Given the connection between regression and FLDA, BLDA performs regression in a Bayesian framework and sets the targets mentioned above.

The assumption in Bayesian regression is that targets t and feature vectors x are linearly related with additive white Gaussian noise n .

$$t = \mathbf{w}^T \mathbf{x} + n \quad (3.5)$$

Given this assumption, the likelihood function for the weights \mathbf{w} used in regression is

$$p(\mathbf{D}|\beta, \mathbf{w}) = \left(\frac{\beta}{2\pi}\right)^{N/2} \exp\left(-\frac{\beta}{2}\|\mathbf{X}^T \mathbf{w} - \mathbf{t}\|^2\right) \quad (3.6)$$

Here, \mathbf{t} denotes the vector containing the regression targets, \mathbf{X} denotes the matrix that is obtained from the horizontal stacking of the training feature vectors, \mathbf{D} denotes the pair $\{\mathbf{X}, \mathbf{t}\}$, β denotes the inverse variance of the noise, and N denotes the number of examples in the training set.

To perform inference in a Bayesian setting, one has to specify a prior distribution for the latent variables, i.e., for the weight vector \mathbf{w} . The expression for the prior distribution we consider and use here is

$$p(\mathbf{w}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{D/2} \left(\frac{\epsilon}{2\pi}\right)^{1/2} \exp\left(-\frac{1}{2}\mathbf{w}^T \mathbf{I}'(\alpha) \mathbf{w}\right) \quad (3.7)$$

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

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

and D is the number of features. Hence, the prior for the weights is an isotropic, zero-mean Gaussian distribution. The effect of using a zero-mean Gaussian prior for the weights is similar to the effect of regularization term used in ridge regression and regularized FLDA. The estimates for \mathbf{w} are shrunk towards the origin and the danger of over-fitting is reduced. The prior for the bias (the last entry in \mathbf{w}) is a zero-mean univariate Gaussian. Setting ϵ to a very small value, the prior for the bias is practically flat. This expresses the fact that a priori there are no assumptions made about the value of the bias parameter.

Given the likelihood and the prior, the posterior distribution can be computed using Bayes rule.

$$p(\mathbf{w}|\beta, \alpha, \mathbf{D}) = \frac{p(\mathbf{D}|\beta, \mathbf{w})p(\mathbf{w}|\alpha)}{\int p(\mathbf{D}|\beta, \mathbf{w})p(\mathbf{w}|\alpha)d\mathbf{w}} \quad (3.8)$$

Since both the prior and the likelihood are Gaussian, the posterior is also Gaussian and its parameters can be derived from the likelihood and the prior by completing the square. The mean \mathbf{m} and covariance \mathbf{C} of the posterior satisfy the following equations.

$$\mathbf{m} = \beta(\beta\mathbf{X}\mathbf{X}^T + \mathbf{I}'(\alpha))^{-1}\mathbf{X}t \quad (3.9)$$

$$\mathbf{C} = (\beta\mathbf{X}\mathbf{X}^T + \mathbf{I}'(\alpha))^{-1} \quad (3.10)$$

By multiplying the likelihood function Eq. 3.6 for a new input vector $\hat{\mathbf{x}}$ with the posterior distribution Eq. 3.8 followed by integration over \mathbf{w} , we obtain the predictive distribution, i.e., the probability distribution over regression targets conditioned on an input vector,

$$p(\hat{t}|\beta, \alpha, \hat{\mathbf{x}}, \mathbf{D}) = \int p(\hat{t}|\beta, \hat{\mathbf{x}}, \mathbf{w})p(\mathbf{w}|\beta, \alpha, \mathbf{D})d\mathbf{w} \quad (3.11)$$

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

$$\mu = \mathbf{m}^T \hat{\mathbf{x}} \quad (3.12)$$

$$\sigma^2 = \frac{1}{\beta} + \hat{\mathbf{x}}^T \mathbf{C} \hat{\mathbf{x}} \quad (3.13)$$

In this work, only mean values used for taking decisions in order to classify P300 signals which containing the target letter versus non-P300 signals which containing

non-target letters by calculating score; mean value of the predictive distribution. Mean values are summed over trials and the decision made by selecting the max summed mean. In order to calculate the character accuracy, score of the trials that contain the corresponding character should be summed individually. Scores are added up in consecutive repetitions of stimuli (called trial groups) for typing a particular character. The classifier chooses the character with the maximum score. In this work, we use the scores, rather than the classification decisions of BLDA as in [67].

3.3 Disjunctive Normal Unsupervised LDA classifier (DNUL)

The following sections provide the details of our proposed unsupervised classification method based on the disjunctive normal form [30]. The first section propose the model architecture of the unsupervised classifier which mainly focuses on integrating the proposed idea to classify P300 signals unsupervisedly without using any calibration session and labelled data. The second section shows how the classifier parameters can be initialized and configured. The last section provides the model optimization procedures in order to learn and update the classifier weights.

3.3.1 Model Architecture

Consider a two-class classification problem: $\mathbf{C} = \{0, 1\}$, for which we observe the data samples (x_1, x_2, \dots, x_n) where n is the number of samples. Let us assume one row/column flash among a full sequence of flashes comes from the class $\mathbf{C} = 1$ and all other $(n - 1)$ row/column flashes in that sequence come from the class $\mathbf{C} = 0$ where $\mathbf{C} = 1$ corresponds to row/column containing the target letter and $\mathbf{C} = 0$ corresponds to row/column not containing the target letter. Let $y_j = f(x_j)$ for $j \in \{1, \dots, n\}$ where $y \in \{0, 1\}$ and $f(x_j)$ is the classification function. Let us define the following Boolean indicator function, which we will call the one-vs-all function $g(\mathbf{y})$.

$$g(y_1, y_2, \dots, y_n) = \begin{cases} 1, & \text{if only one argument is 1;} \\ 0, & \text{otherwise.} \end{cases} \quad (3.14)$$

Any Boolean function can be written as a disjunction of conjunctions, also known as the disjunctive normal form [68].

$$\begin{aligned} E(x) &= g(y_1, y_2, \dots, y_n) \\ &= (y_1, y'_2, \dots, y'_n) \cup (y'_1, y_2, \dots, y'_n) \cup \dots \cup (y'_1, y'_2, \dots, y_n) \end{aligned} \quad (3.15)$$

Furthermore, allowing M repeated observations, we define:

$$E(x) = \sum_i^M g(f(x_{1i}), f(x_{2i}), \dots, f(x_{ni})) \quad (3.16)$$

where we can relax the function f so it has real valued outputs in $[0, 1]$ rather than binary. We perform such relaxation through a logistic sigmoid function, where β is a sensitivity parameter and S is the dimensionality of the feature vector.

$$f(x_{ji}) = \frac{1}{1 + e^{-\beta(\sum_{k=1}^{S+1} w(k)x_{ji}(k))}} \quad (3.17)$$

Using De Morgan's laws and products of conjunctions yields the following differentiable energy function [68].

$$E(x) = \sum_i^M \left(1 - \prod_j^n \underbrace{\left(1 - f(x_{ji}) \prod_{k \neq j}^n (1 - f(x_{ki})) \right)}_{Q_j} \right) \quad (3.18)$$

where M denotes the number of rounds for row/column highlights.

3.3.2 Model Initialization

Let us consider a P300 speller paradigm with $\Gamma = \{(x, L(x))\}$, where x denotes the data and $L(x)$ denotes the binary class label corresponding to x . Furthermore, let Γ^+ denote class $L = 1$ corresponding to the desired target and Γ^- denote class $L = 0$ corresponding to the non-desired target.

Since the model is designed to work in an unsupervised fashion, the labels for learning the model will not be available to the algorithm. We will use the disjunctive normal form-based energy function in (3.18) to classify the two classes without using any labels. The weights w of the disjunctive normal unsupervised linear discriminant (DNUL) classifier are randomly initialized and the bias term is set to 1 as it considers a good initial seed point and chooses arbitrary in order to reduce the computational time. Consider a speller matrix as in Fig. 1.2. We have 6×6 characters, which

means the target character needs a set of row/column intensifications (highlights) to cover the matrix in order to be classified. We call the set of intensifications covering the entire array a trial group. Therefore, $n = 6$ in our algorithm. Trial group is one of the important terms used for P300 spellers, which defined in details in terminology section (chapter 4).

The sigmoid function in (3.17) takes the value 0.5 in the middle of the classification line between two classes. The goal is to design a classifier to put the data from the desired class in Γ^+ when $f(x_{ji}) \geq 0.5$ and data from non-desired class in Γ^- when $f(x_{ji}) < 0.5$ by optimizing the energy function in (3.18).

3.3.3 Model Optimization

In order to learn the DNUL classifier, we use gradient ascent to maximize the energy function by taking the partial derivatives of the energy function with respect to the weights. The gradient of the energy function in (3.18) is given by:

$$\begin{aligned}
\frac{\partial E}{\partial w} &= - \sum_i^M \frac{\partial}{\partial w} \left(\prod_j^n Q_j \right) \\
&= - \sum_i^M \sum_j^n \left(\frac{\partial Q_j}{\partial w} \prod_{l \neq j}^n Q_l \right) \\
\frac{\partial Q_j}{\partial w} &= - \frac{\partial}{\partial w} f(x_{ji}) \prod_{k \neq j}^n (1 - f(x_{ki})) \\
&\quad - f(x_{ji}) \sum_{p \neq j}^n \left(- \frac{\partial}{\partial w} f(x_{pi}) \prod_{k \neq p, j}^n (1 - f(x_{ki})) \right)
\end{aligned} \tag{3.19}$$

The model performs iterations till the DNUL classifier converges updating the weights at each iteration: Equation (3.20) shows that where α is the step size or learning rate. The bias term is included in the weight vector.

$$w^{new} = w^{old} + \alpha \frac{\partial E}{\partial w} \tag{3.20}$$

3.4 Regularized Disjunctive Normal Unsupervised LDA Classifier (RD-NUL)

In this section, a new regularization term is introduced. Regularization is used to prevent overfitting to high dimensional data and possibly noisy datasets. Adding

Lambda	Influence on the model
$\lambda = 0$	Downgrade to the original energy function
$\lambda = \infty$	Only cares about penalizing w which has the large influence
λ in between	Balance data fit against the magnitude of the coefficients

Table 3.1: Influence of the tuning parameter.

the regularization term to the energy function which is introduced in Equation (3.18) produces the regularized DNUL (RDNUL) energy function as follows:

$$E(x) = \sum_i^M \left(1 - \prod_j^n \left(1 - \underbrace{f(x_{ji}) \prod_{k \neq j}^n (1 - f(x_{ki}))}_{Q_j} \right) \right) - \frac{\lambda}{2} \|\mathbf{w}\|^2 \quad (3.21)$$

Where the penalty term in ridge regression composed of $\|\mathbf{w}\|^2$ which is the L2 norm of the weight vector and λ which is the tuning parameter or constant regularization parameter [69]. Table 3.1 shows the general influence of the tuning parameter on the model.

By taking the partial derivative with respect to the weights, the gradient of the regularization term is given by:

$$\frac{\partial \|\mathbf{w}\|^2}{\partial w} = 2w \quad (3.22)$$

Equation 3.23 shows the learning process including updating the weights at each iteration where λ is the tuning parameter.

$$w^{new} = w^{new} + \alpha \left(\frac{\partial E}{\partial w} - \lambda w \right) \quad (3.23)$$

For the tuning parameter, as it is an unsupervised problem we can't use validation set (for large datasets) or cross validation (for smaller datasets) to tune the parameter. In this work we chose that value empirically by trial and error.

3.5 Toy Examples for the Proposed Unsupervised Classifier

In this section, different scenarios of a toy example with synthetic data are introduced for the proposed disjunctive normal unsupervised LDA classifier (DNUL).

The aim of this section is to introduce the efficiency of the proposed DNUL classifier, where discussed in detail in section 3.3.

The synthetic data are simulated for P300 spellers; we assume we have one row/column flash contains a P300 component among a full sequence of flashes and all other $(n - 1)$ rows/columns flashes in that sequence don't contain P300 components. In other words, the aim of the classifier is to classify a P300 instance on one side and the other five instances which do not contain P300 on the other side in case of 6×6 matrix as shown in Figure 1.2. The data are generated based on a two class classification problem, one class centred with a zero mean and the other class with a mean vector of ones. The variance of the classes are chosen with different diversity in order to control the difficulty of the problem.

The DNUL classification model is one of the most challenging as it starts initially unlearned without using labels. In this case, the model fed by the synthetic data directly in order to classify the P300 signals without any single labelled instance. The initialization parameters of the DNUL classifier for the toy example initialized as following: for each classifier, we perform 10 optimizations. For each optimization, we initialize 2 weight vectors drawn from normally distributed random numbers $\sim \mathcal{N}(0, 1)$, one with \mathbf{w} and one with $-\mathbf{w}$. In total, we have 20 classifiers and we pick the classifier with the highest energy function. The number of iterations is set to 500, the step size is set to $\alpha = 0.2$, and the sensitivity parameter $\beta = 1$. We generated two dimensional (2D) data with 312 artificial instances.

First, we started with a very ideal simple toy example to show the strength of the proposed model. In order to change the difficulty of the classification data, we increased the data variance. Hence, the intersection between classes increases. Figure 3.1 illustrates a detailed example with an easy case.

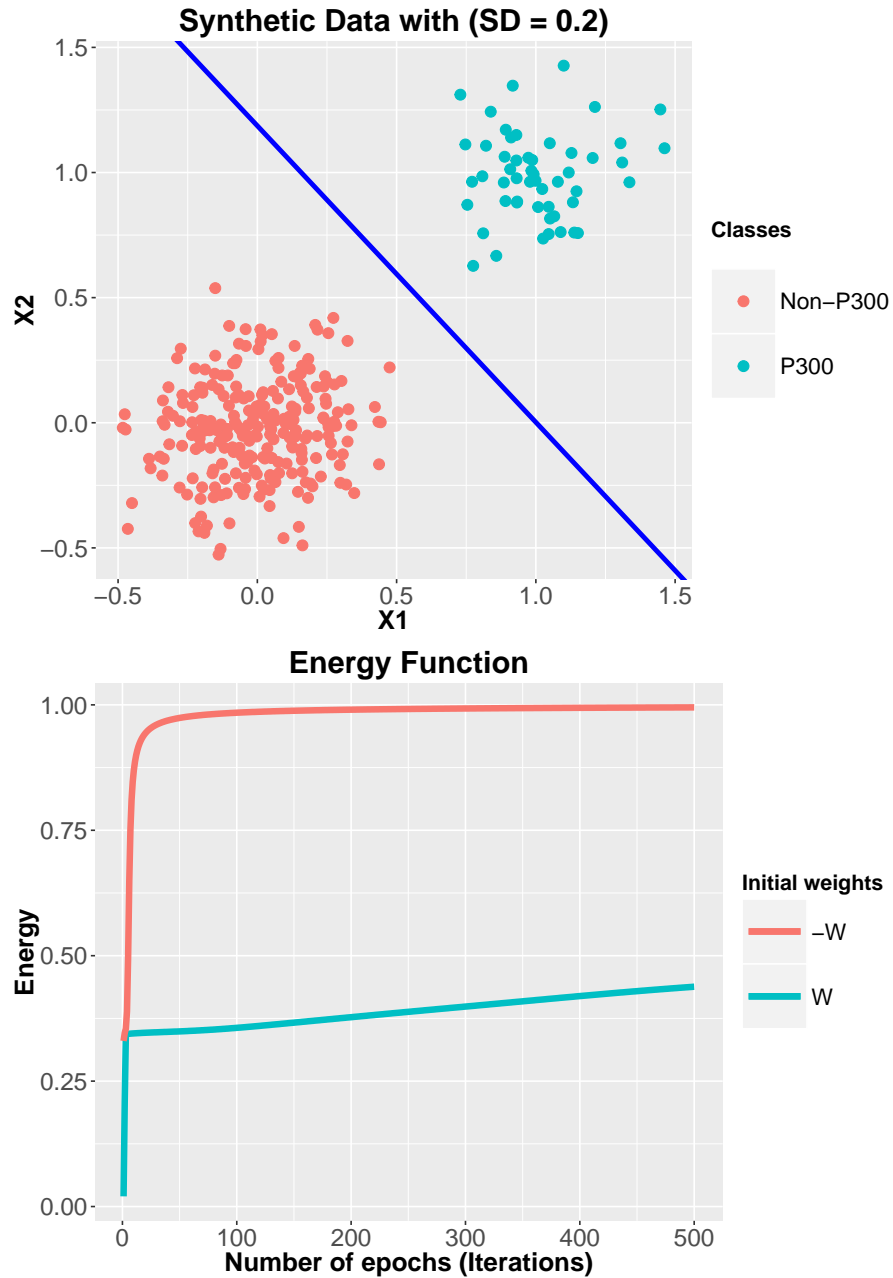


Figure 3.1: A toy example on a synthetic dataset with a standard deviation = 0.2. The DNUL classifier classified the data successfully with 100% classification accuracy. The second figure shows the energy function for the initialized 2 random-weight vectors for one of the classifiers among 10 classifiers (which is the highest).

As we can see from the Figure 3.1, the energy function with $-\mathbf{w}$ converged to 1 (Energy = 1) and the other one with \mathbf{w} be unable to reach a high energy and failed to classify the instances. The following figures show the different scenarios with increasing the variance and how the DNUL classifier is able to classify the synthetic data with different hard levels.

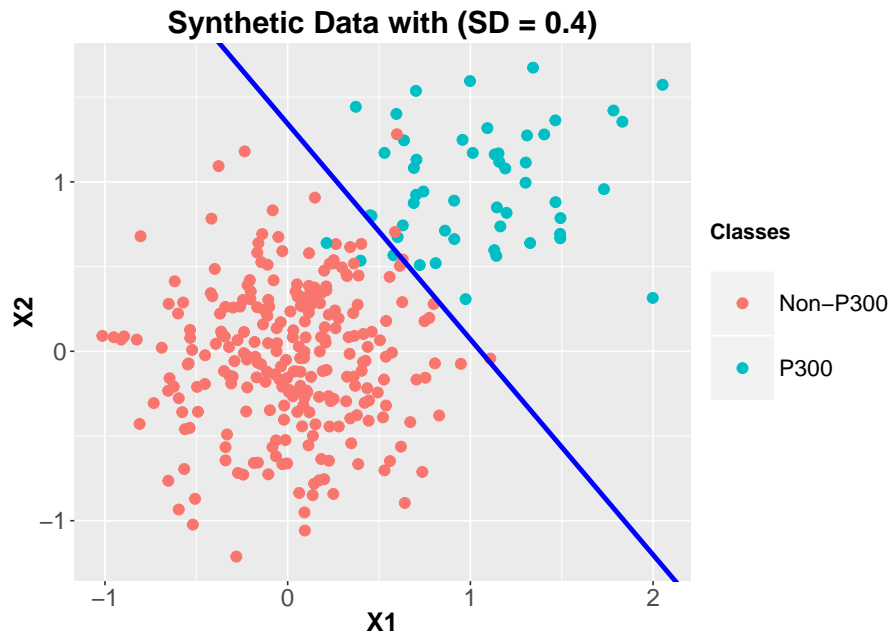


Figure 3.2: A toy example on a synthetic dataset with a standard deviation = 0.4. The classifier achieved 97.76% classification accuracy with true positive rate = 94.23% and true negative rate = 97.69%.

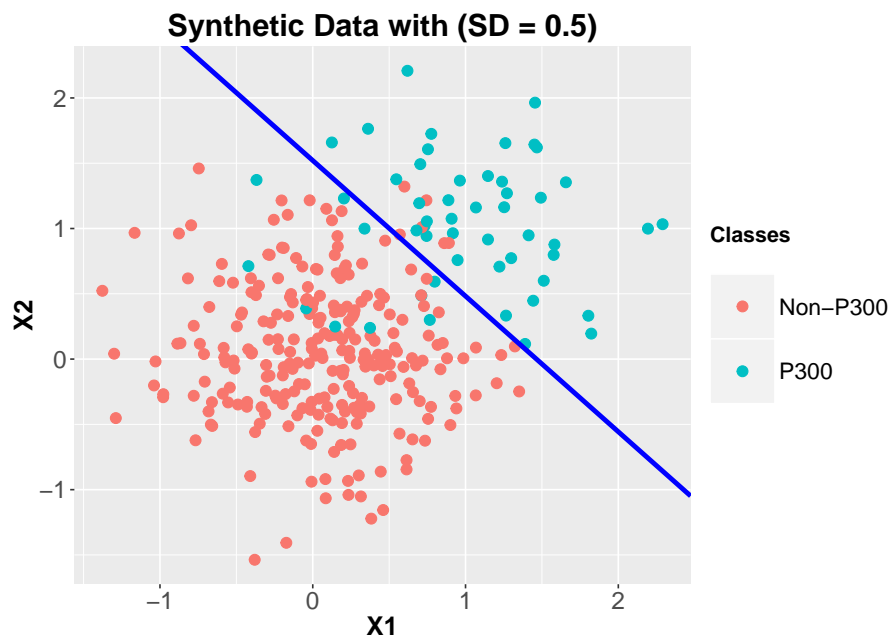


Figure 3.3: A toy example on a synthetic dataset with a standard deviation = 0.5. The classifier achieved 95.19% classification accuracy with true positive rate = 82.69% and true negative rate = 98.46%.

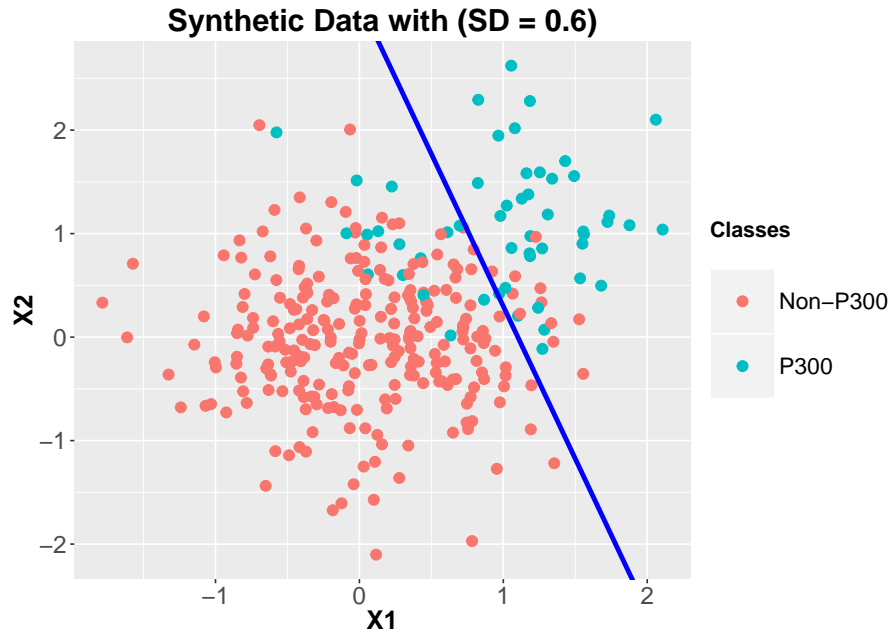


Figure 3.4: A toy example on a synthetic dataset with a standard deviation = 0.6. The classifier achieved 91.67% classification accuracy with true positive rate = 71.15% and true negative rate = 95.76%.

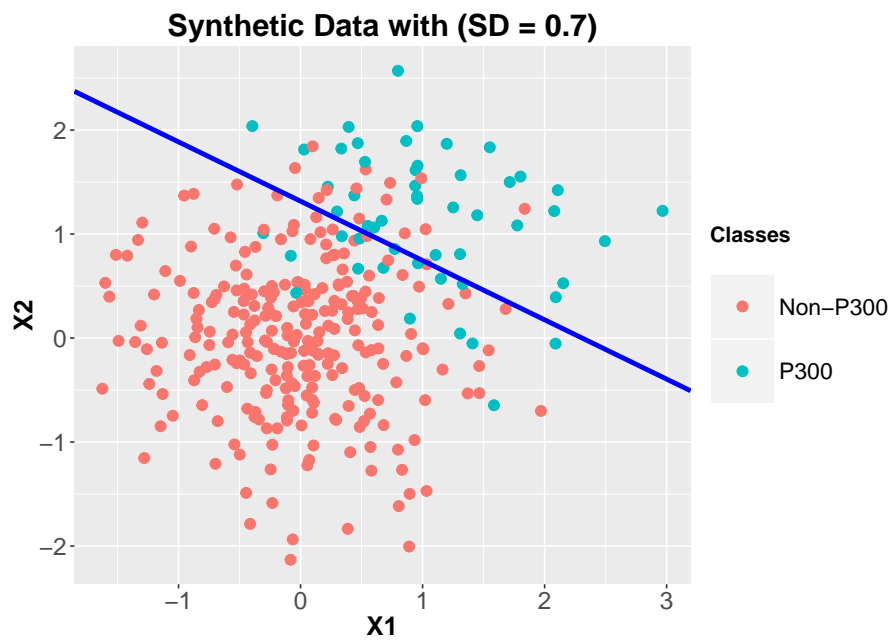


Figure 3.5: A toy example on a synthetic dataset with a standard deviation = 0.7. The classifier achieved 90.71% classification accuracy with true positive rate = 69.23% and true negative rate = 95%.

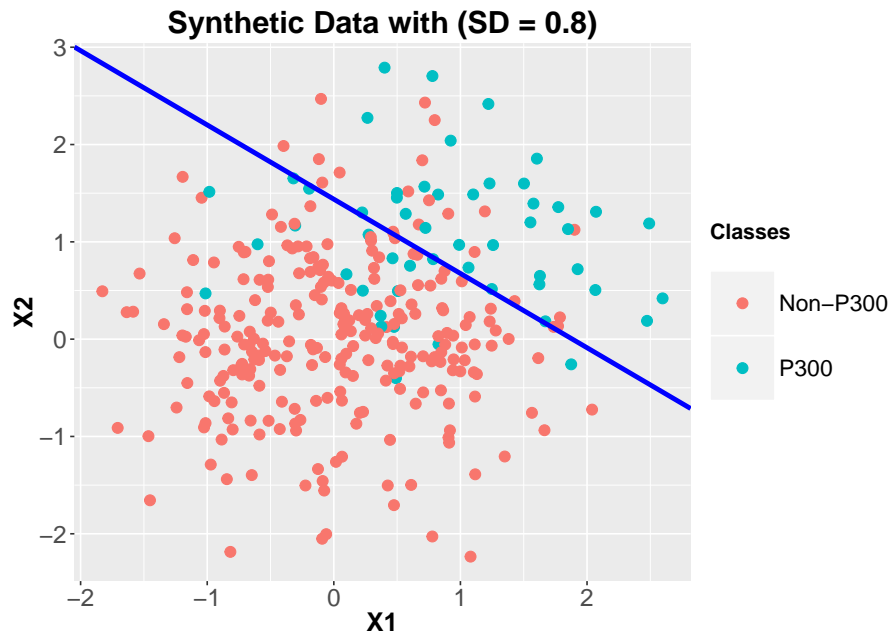


Figure 3.6: A toy example on a synthetic dataset with a standard deviation = 0.8. The classifier achieved 87.18% classification accuracy with true positive rate = 63.46% and true negative rate = 91.92%.

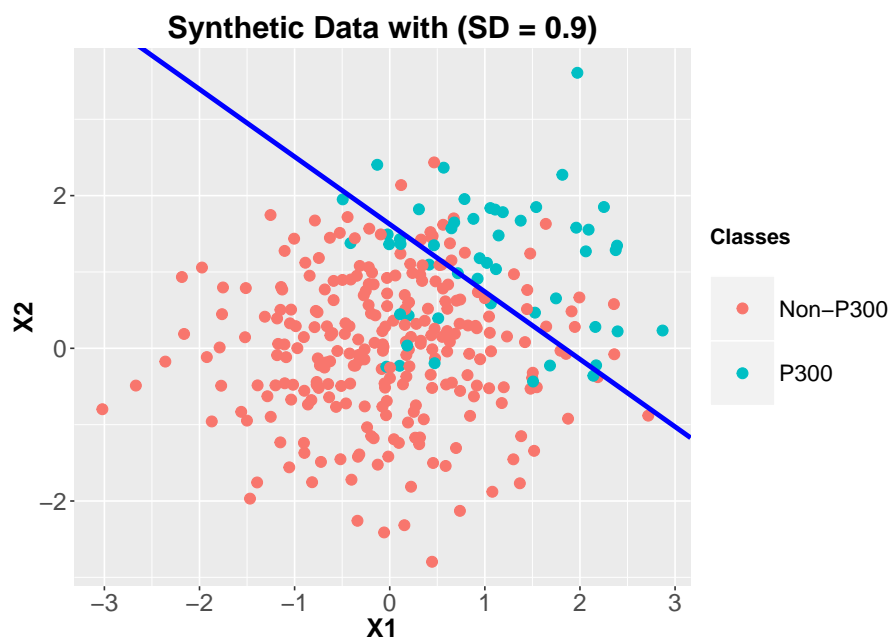


Figure 3.7: A toy example on a synthetic dataset with a standard deviation = 0.9. The classifier achieved 86.56% classification accuracy with true positive rate = 61.53% and true negative rate = 91.53%.

Chapter 4

Offline and Simulated Online Analysis Experiments

This chapter includes the procedures and steps for the signal processing which has been applied on the EEG data for this study. In addition, it contains the description of offline and simulated online analysis. At the end of the chapter, the experimental results for both of these analyses and comparison with other relevant studies have been shown.

4.1 Background

In BCI research, there are many scenarios for analysing the EEG data. The most common scenarios for P300 spellers are offline, online simulation analysis, and real online analysis. Each of these types of analysis has its own purpose and laboratory configuration.

Offline analysis is one of the main common analysis used for P300 speller where the data collection performed separately from the experimental analysis. In other words, the data collection can be recorded in one laboratory in a broad period of time, and then the analysis is done in another place at different time, with all the experimental data as a chunk. The purpose of this analysis is to develop a classification algorithms and techniques for P300-based BCIs, that is because the whole dataset is available for the classifier. For this case, there is no feedback capability to indicate the classification results; which allows the subject to see the chosen character. Offline analysis considered as a first step toward the improvement of a real P300 speller system, where the subject can see the written characters in a real-time.

In online analysis, generally, data are pre-processed and analysed concurrently.

In other words, for P300 spellers the characters are pre-processed and classified one by one in a real time and we call this by real online analysis. The offline recorded data can also be online analysed by simulating the offline recorded characters individually. In this case, characters classified sequentially one by one and the classification results are instantly displayed to the subject. For this case we call this scenario by simulated online analysis. Hence, feedback can be provided for the subject for these analysis. Online analysis is useful toward understanding the strength of the classifier as it starts with less amount of data and increases sequentially which make it harder for the classifier to learn with small amount of data especially in the beginning.

4.2 Terminology

Some extensively used terms should be defined in order to make it possible for readers to understand clearly the work presented in this thesis. This section exactly follows the same terminology used in [37] [49] .

- A **target letter** is the letter that the subject is informed to focus on at a time instant.
- A **trial** denotes the intensification of each row or column, the timing of which is marked by trigger signals in the recording. We also use the term “flash” in this thesis to imply a trial.
- A **trial group** is the group of trials that includes each row and column intensification that is flashed exactly once. For example, with a speller matrix dimension of 6×6 , a trial group consists of 12 individual flashes in which there are no rows or columns that are flashed more than once. With this in mind, a trial group is the smallest data set for a P300 classification problem. In this thesis, a trial group is sometimes referred as repetition or one set of flashes.
- An **epoch** is a determined period of recorded data that includes a trial. In P300 studies, this period is usually from 600 ms to 1000 ms starting from the time when a stimulus event (flash) occurs.
- A **run** is the collection of several trial groups. A run is recorded for each letter defined in a session to be spelled. There can be a period of a few seconds

between each run, but the recording is not interfered with, and is continues.

- A **session** is the time period in which the recordings for all previously defined number of target letters are done.
- A **session group** is the collection of all sessions recorded with one subject during the course of a day with a time break in between each session.

To increase the performance of the classifier, the number of recorded trial groups (repetitions) in a run in the training set has to be increased. However, that leads to low information rate and therefore the speed of the BCI system is decreased. Hence, researchers on BCI try to come up with efficient signal processing and classification algorithms to achieve performance improvements in accuracy and speed simultaneously.

4.3 P300 Classification Problem

The aim of the classification problem is to detect or determine whether the received epoch contains P300 component or not. It can be seen that, it is the first step toward classification since a trial group (the smallest dataset which contains 12 trials) contains two P300 components and the other 10 trials not containing a P300 component. The intersection between row and column which corresponds to P300 components can be deduced, in order to classify a character. If the classifier makes a single mistake for a trial group then we will lose the character. In this case, the classifier accuracy may appear higher than the character accuracy. All the results reporting in this thesis are for character accuracies.

The data are recorded in two sessions, one for training the classifier and the other used for testing; to see how well the classifier learnt. In this thesis, classification algorithms are divided in two categories, supervised and unsupervised classification. For the supervised classification, we have used the Linear Discriminant Analysis (LDA) classifier and Bayesian Linear Discriminant Analysis (BLDA), which were described in Sections 3.1 and 3.2, respectively. The classifier uses a classical machine learning methodology where the first session is used as the training set and the other session is given to the classifier as the test set for evaluation. MATLAB was used to perform all the analysis of the experimental data in this study and to read the

raw EEG data file in BioSemi Data File (BDF) format. It also creates the necessary data structures for the recorded data. As the data structures are formed after the recorded data file is read, data epochs of 1-second (1000 ms) periods that follow each trigger signal are extracted and each epoch is labelled as **1** or **-1** according to the target stimuli. In addition, R programming was used to represent and illustrate the results.

DNUL, our unsupervised classifier proposed in section 3.3, has a different methodology in learning compared to supervised classifiers. It is unsupervisedly trained on the test set. The learning process looks different compared to the supervised classifiers but as a rule with unsupervised algorithms, the learning process starts with the test data and the learning (training) perform without using any ground truth label. This methodology was used to enable comparison with other supervised algorithms on the same data.

4.4 Methods

4.4.1 Data pre-processing

The 6×6 spelling matrix uses the most common stimulus type as shown in Figure 1.2. The intensification covers the rows and columns of the matrix in a block-randomized fashion. Each intensification flashes exactly once with an inter-stimulus interval (ISI) of 125 ms; the intensification duration of 50 ms and the remaining 75 ms waiting for the next intensification. As it mentioned before, EEG signals suffer from low signal to noise ratio (SNR) due to several factors including the variability in brain activities. Therefore, P300 spellers need several stimulus repetitions (trial groups) to increase the classification accuracy. In this work we fixed the number of trial groups to 15. Each trial group consists of 12 trials; 2 of the trials are relevant to stimuli (contain P300 component) and the other 10 trials are irrelevant stimuli (not contain P300 component). When a particular experiment involves the use of multiple trial groups we average the EEG data over the trial groups to obtain the data points to be used in classification. The pre-processing of the data in this work follows the steps which described in [70].

Temporal EEG data was recorded from 12 active channels during the experiment

which were placed at Fp1, Fp2, Fz, Cz, Pz, Oz, P3, P4, Po7, and Po8 locations according to the international 10-20 system, in addition to the two auxiliary electrodes located on the mastoid channels for reference as shown in Figure 2.6. The data were recorded with the BioSemi acquisition ActiView software. It is an open source program written in LabVIEW. It saves the recordings as a BDF file, which uses a format originally developed by BioSemi. MATLAB is used to extract the saved BDF file via the code developed by Alois Schloegl in 1998. After that, the raw data pass through several signal processing steps before the classification step.

Trigger channel plays a role in dividing the data into epochs where it can be extracted from the raw data. In order to do that, times (sample numbers in the sequence) and values (actual trigger values) of each trigger signal are obtained from the data in the trigger channel and stored in a key-value pair. After obtaining this information, a sequence of row/column can be obtained. In this work, exploiting this information also motivates us to classify P300 components with knowing the sequence of row/column trail.

First step to do with the raw data in order to obtain a better representation of P300 component is filtering. The whole data are filtered with a 3rd order Butterworth bandpass filter with cut-off frequencies 1-12 Hz. The aim of filtering is to get rid of irrelevant frequency components such as background noise (e.g. power cable frequency) and DC offset that occurs between electrodes and the skin due to sweating. There are other different irrelevant artifacts that come from the body potentials of the subject due to muscle movement and irrelevant EEG signals that are not relevant to P300 component. Hence, as mentioned before the mean of two recorded mastoid channels are used as a reference signal by subtracting these from the other channels, we obtain a higher SNR and the data are referenced.

The next step after the data have been filtered is to reshape the whole data into epochs. Each epoch is a data set of 1 second (1000 ms). Since, the data are sampled at 2048 Hz; each epoch contains 2048 samples. The recorded data are decimated by 64 in order to reduce the feature space. After the preparation step, the data are normalized to remove the negative effects of the electrode-skin resistance which plays a role in amplitude changes. However, if the waveform contains very high and extreme values, normalization may result in a poor performance. To eliminate that

effect, the data are windsorized in a 10% window. Windsorizing the data removes the extremities by clipping the samples that are out of this window and provides a healthy normalization. At the end, we have a better representation of the data and we got a feature vector contains 32 samples after decimation. Hence, the data are ready for classification.

The pre-processing method described in section 4.4.1 are applied equally to all 7 subjects included in SU dataset as shown in Appendix A, and the results for each subject is presented separately [37].

4.4.2 Classification

Classification is one of the most important steps in the scope of this thesis. After the data pre-processing procedure, we get a feature vector for each epoch by concatenating the filtered data from each electrode. Sequel to the above pre-processing section, in case of 10 electrodes we obtain a feature vector of 320 samples. At the end, we reshape the data in order to obtain a matrix of size $r \times t$ where r is the size of feature vector (320 samples) and t is the number of feature vectors (epochs) usually it equals to 180 (12 trials \times 15 trial groups).

The data now are ready for classification. As shown in Chapter 3, we used two different learning paradigms for classifying the P300 component: supervised and unsupervised learning. We used supervised techniques in this thesis to compare with our proposed unsupervised classifier.

4.5 Dataset

In our offline validation, this work used three different datasets involving data from a 6×6 visual matrix speller. In the first one, data are recorded in our own laboratory by 7 male healthy subjects who performed offline spelling, whose ages are between 18 and 30. Only two of the subjects had prior BCI experience. The 6×6 spelling matrix uses the most common stimulus type. The intensification covers the rows and columns of the matrix in a block-randomized fashion. Each subject recorded two sessions: one for training and one for testing. The training session involved spelling 14 characters forming 2 Turkish words. The test session involved spelling 26 characters forming 4 Turkish words. In this work, we generate

two versions of test data, one with 14 characters and the other one with the entire 26 characters as shown in Table A.1. The data were recorded with the BioSemi ActiView software. We used the data pre-processing methods described in section 4.4.1. Details of this dataset are given in Appendix A.

The second and third datasets are publicly available and most widely used in BCI research. These datasets are: BCI Competition II (Dataset Iib) [32] and BCI Competition III (Dataset II) [33]. The first one involves one subject called subject C and it is considered as an easy dataset. The second dataset involves 2 subjects called subjects A and B. Subject A is harder than subject B. Details of these datasets are given in Appendix A.

4.6 Experimental Setup

In order to evaluate the proposed disjunctive normal unsupervised LDA classifier (DNUL) for the P300 based speller, accuracy and bit-rate performance have been evaluated. These two main criteria can demonstrate our classifier effectiveness. Classifier accuracy is calculated by dividing the total number of correctly classified characters in a session by the total number of characters to be classified in that session. To evaluate the speed of communication, information transfer rate, which is commonly called bit-rate (B) in bits/min is computed as in [71]:

$$B = \frac{60}{T} \left(\log_2(n) + p \log_2(p) + (1 - p) \log_2 \left(\frac{1 - p}{n - 1} \right) \right) \quad (4.1)$$

where n is the number of characters in the speller matrix (36 in this case), p is the classification accuracy and T is the time in seconds that is needed to spell one symbol calculated for SU datasets as in Appendix A by $(3.5 + 0.125 \times 12 \times N_t)$, where N_t is the number of available trial groups. Since one set of flashes takes 1.5 s and assuming that 3.5 s is needed to display the target letter to the subject, there can be 12 characters at maximum that a subject can type in a minute. Hence, the maximum bit-rate of our system using a perfect classifier for offline classification is 62.04 bits/min, which is calculated by $12 \times \log_2 36$, for using one trial group in SU datasets. Using more trial groups will reduce the bit-rate as for 15 trial groups the perfect classifier for offline classification is 11.93 bits/min. In a similar way, for the

BCI competitions dataset as shown in Appendix A, one set of flashes takes 2.1 s. If we take into account the 2.5 s period for the target letter displaying, at most 13.04 characters can be typed in a minute by the subject. Hence, the maximum bit-rate for offline classification for both BCI competition II and BCI competition III are 67.42 bits/min.

Two other supervised classifiers, LDA and BLDA, are used for comparison with the proposed unsupervised classifier and all of them will be evaluated using the same criteria.

4.7 Experimental Results

The DNUL classification model is one of the most challenging as it starts initially unlearned without using labels. In this case, there is no need for the training session, the approach just evaluates the model on the upcoming EEG data as discussed in Chapter 3. Most systems, including ours, classify the individual intensifications and combine the outputs to predict the spelled character and that makes the problem harder, because if the model fails to classify one intensification it will lead to losing a character.

The number of trial groups for spelling a character was pre-defined, the maximum number of trial groups recorded in these datasets was 15. Our experiments are divided into two main categories, offline analysis (batch mode) and simulated online analysis (sequential mode). The sequential mode is designed to simulate online spelling in order to evaluate the sequential adaptation process of the classifier. The configurations of these setups are shown in the next corresponding sections.

Two extensively used terms should be clearly described in order to understand the results presented in this thesis.

- An **optimization**: for each optimization, we initialize 2 weight vectors drawn from normally distributed random numbers $\sim \mathcal{N}(0,1)$, one with \mathbf{w} and one with $-\mathbf{w}$.
- A **Classifier group**: For each classifier group, we perform 10 optimizations. In total, we have 20 classifiers and we pick the classifier with the highest energy function.

The initialization parameters of the DNUL model is the same for all experiments. The number of iterations is set to 500 and the step-size is set to $\alpha = 0.2$. The sensitivity parameter $\beta = 0.1$ was chosen based on an analysis performed for 7 subjects and is used for the whole other dataset as shown in detail in Appendix B. The variation of β will alter the classification accuracy as depicted in Figure B.1. Sensitivity parameter β controls the steepness of the sigmoid; the bigger β is, the steeper the sigmoid. We are working on a mechanism to set this parameter automatically based on data.

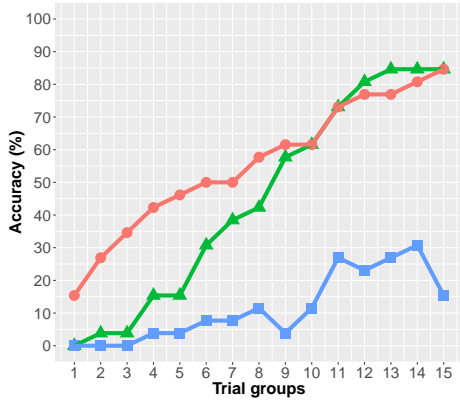
4.7.1 Offline Analysis

To start, we compare our proposed approach with the supervised LDA and BLDA classifiers. As a rule, the LDA and BLDA classifiers always learns (trains) with the training data and the DNUL classifier learns (trains) with the test data which will be used for validating the supervised classifiers in order to enable comparison. In this section, two main offline experiments are performed to demonstrate the efficiency of the unsupervised classifier.

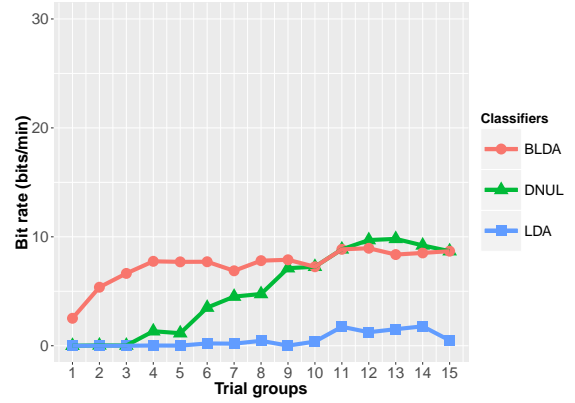
- **Batch mode:** In this configuration mode, experiments are carried out by averaging the EEG dataset with a chunk of the whole trial groups (15 in this case) for supervised or unsupervised learning and then the classifier is evaluated on the sequence of trial groups starting from 1 to 15.
- **N-Batch mode:** In this mode, the number of trial groups for supervised and unsupervised learning respectively matches the number of trial groups used for testing.

SU-Dataset: Batch mode with 26 characters in the test set

In this experiment, all classifiers are evaluated on the (Test 2) dataset as shown in Table A.1. The detailed accuracies for individual subjects corresponding to the experiments are given in Table 4.1 and can be shown in Figures 4.1 to 4.8. The curves display classification accuracy and bit-rate as a function of the number of trial groups involved in each data sample used to test the classifiers.



(a) Classifier Accuracy



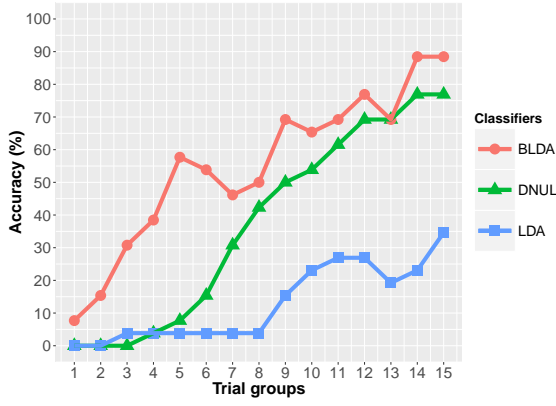
(b) Bit rate

Figure 4.1: Offline analysis results for subject 1 with (Batch-26) configuration.

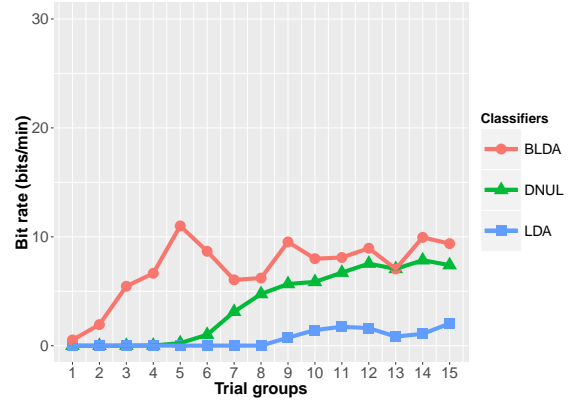
As it can be seen, using more trial groups, DNUL improves compared to LDA and getting closer in performance to BLDA. The overall improvement from LDA to DNUL is 34.61% and 4.47 bits/min for accuracy and bit-rate respectively in case of 15 trial groups. There is a small difference between BLDA and DNUL in case of 15 groups. The BLDA improved compared to DNUL by 13.74% and 2.25 bits/min for accuracy and bit-rate respectively and that is because of the non-stationarity of the EEG signals.

Table 4.1: Percentage of correctly classified characters for each subject obtained with different values of trial groups for (Batch-26).

Subjects	5 trial groups			10 trial groups			15 trial groups		
	DNUL	LDA	BLDA	DNUL	LDA	BLDA	DNUL	LDA	BLDA
S1	15.38	3.85	46.15	61.54	11.54	61.54	84.62	15.38	84.62
S2	7.69	3.85	57.69	53.85	23.08	65.38	76.92	34.62	88.46
S3	11.54	7.69	61.54	50	30.77	84.62	76.92	34.62	96.15
S4	46.15	7.69	80.77	76.92	30.77	92.31	88.46	61.54	92.31
S5	42.31	11.54	53.85	88.46	23.08	84.62	100	50	100
S6	7.69	3.85	57.69	19.23	23.08	57.69	26.92	50	84.62
S7	65.38	3.85	96.15	96.15	34.62	100	96.15	61.54	100
Average	28.02	6.04	64.84	63.74	25.27	78.02	78.57	43.96	92.31

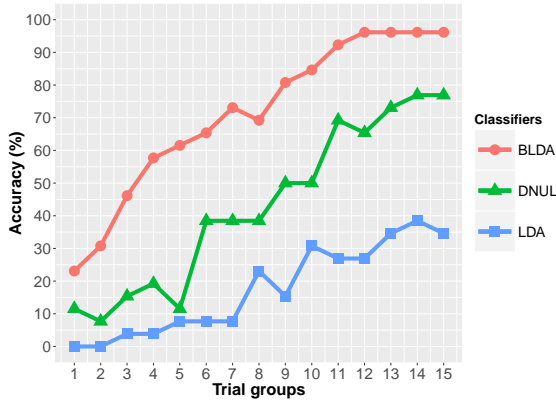


(a) Classifier Accuracy

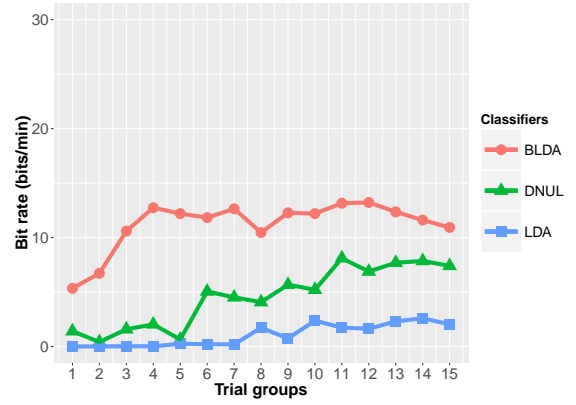


(b) Bit rate

Figure 4.2: Offline analysis results for subject 2 with (Batch-26) configuration.

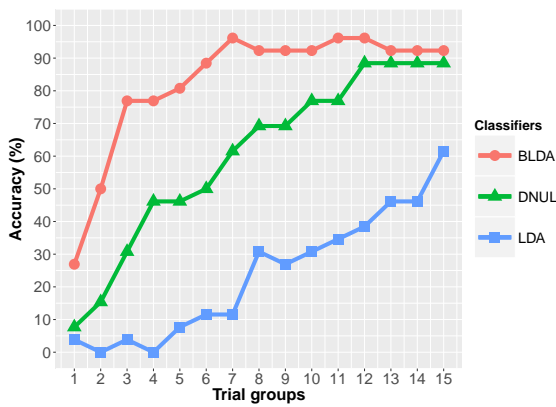


(a) Classifier Accuracy

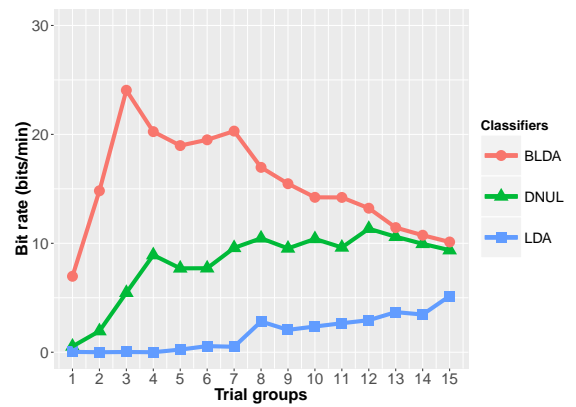


(b) Bit rate

Figure 4.3: Offline analysis results for subject 3 with (Batch-26) configuration.

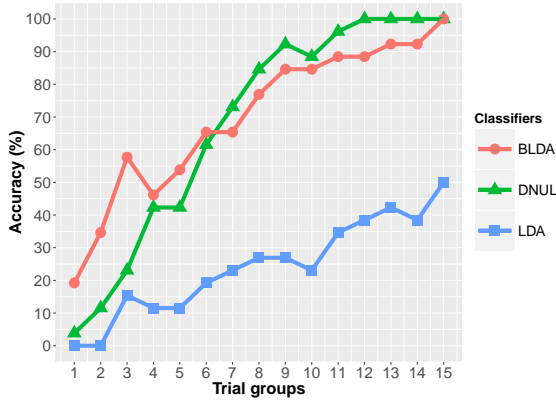


(a) Classifier Accuracy

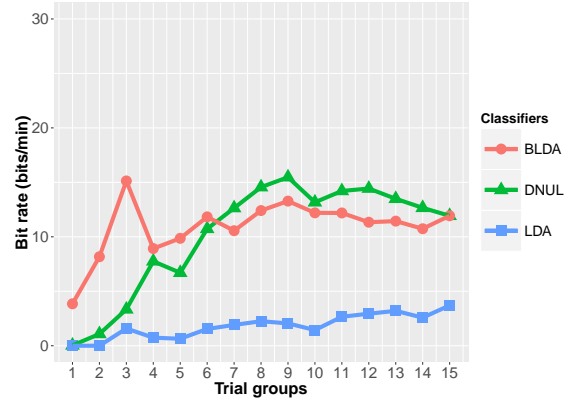


(b) Bit rate

Figure 4.4: Offline analysis results for subject 4 with (Batch-26) configuration.

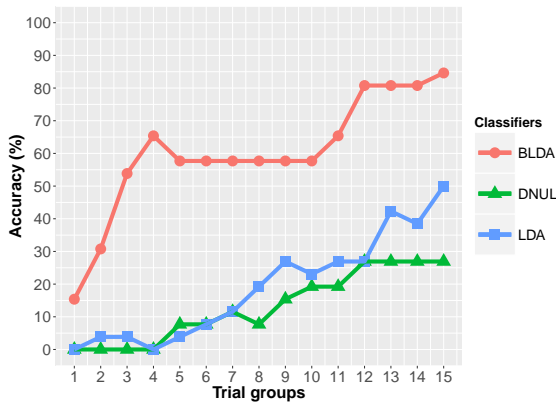


(a) Classifier Accuracy

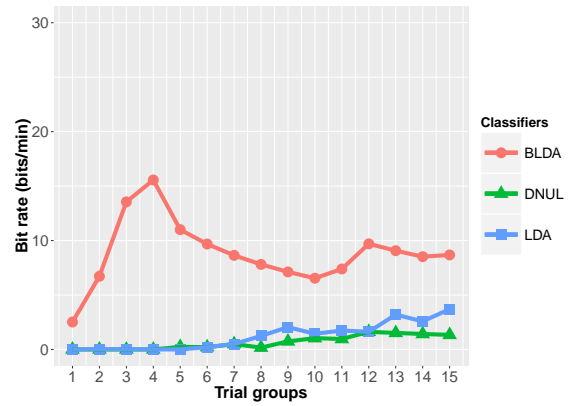


(b) Bit rate

Figure 4.5: Offline analysis results for subject 5 with (Batch-26) configuration.

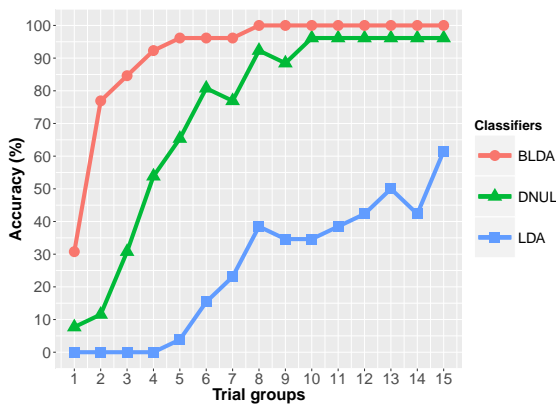


(a) Classifier Accuracy

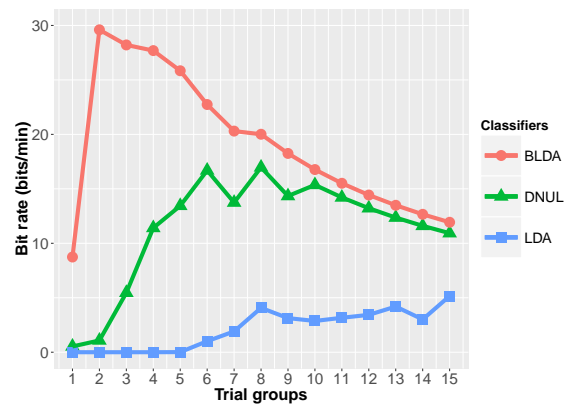


(b) Bit rate

Figure 4.6: Offline analysis results for subject 6 with (Batch-26) configuration.



(a) Classifier Accuracy



(b) Bit rate

Figure 4.7: Offline analysis results for subject 7 with (Batch-26) configuration.

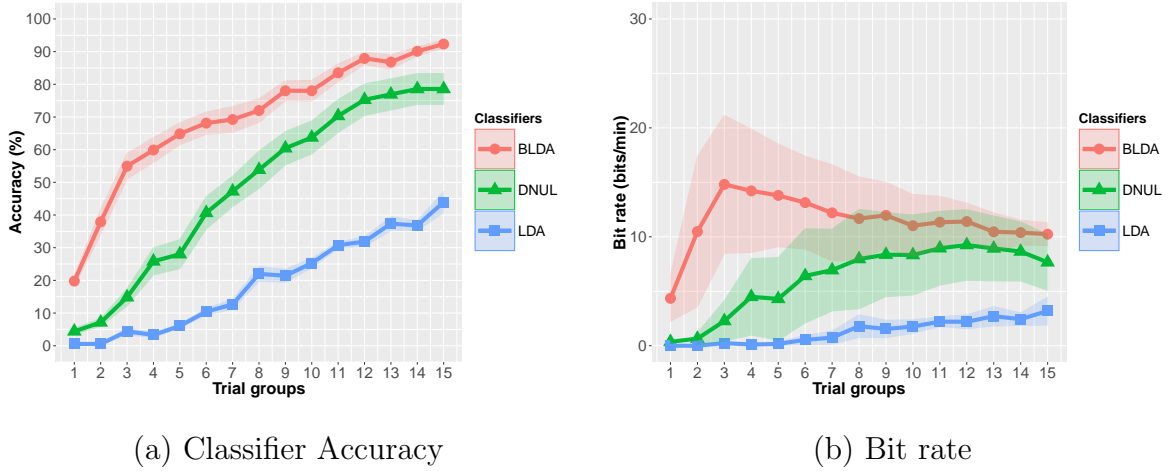


Figure 4.8: Average classification performance over 7 subjects with (Batch-26). Error shadows show 95% confidence intervals from the mean with sample size = 7.

SU-Dataset: Batch mode with 14 characters in the test set

For this experiment, all classifiers are evaluated on the (Test 1) dataset as shown in Table A.1. The idea behind this analysis is that we came up with a version which has equal number of characters to learn and evaluate with the supervised classifiers. The detailed accuracies for individual subjects corresponding to the experiments are given in Table 4.2 and can be shown in Figures 4.9 to 4.16.

As shown in the table 4.2, there is no change in behaviour of the classifiers among

Table 4.2: Percentage of correctly classified characters for each subject obtained with different values of trial groups for (Batch-14).

Subjects	5 trial groups			10 trial groups			15 trial groups		
	DNUL	LDA	BLDA	DNUL	LDA	BLDA	DNUL	LDA	BLDA
S1	21.43	7.14	50	85.71	21.43	71.43	85.71	28.57	85.71
S2	14.29	0	57.14	28.57	28.57	64.29	42.86	28.57	85.71
S3	14.29	0	64.29	71.43	21.43	85.71	71.43	35.71	92.86
S4	42.86	7.14	78.57	85.71	35.71	92.86	92.86	35.71	92.86
S5	21.43	14.29	42.86	71.43	21.43	85.71	85.71	57.14	100
S6	7.14	0	42.86	7.14	14.29	42.86	21.43	42.86	85.71
S7	42.86	0	92.86	85.71	28.57	100	85.71	50	100
Average	23.47	4.08	61.22	62.24	24.49	77.55	69.39	39.8	91.84

each others. DNUL overcomes in performance compared to LDA. The performance outcome of this experiment is generally smaller than the previous experiment specially for DNUL as it learns with less number of characters than the previous one. For the supervised classifiers, the accuracies are fairly comparable with DNUL as they learn with the same amount of characters. Besides, the performance of the bit-rate shows the speed of typing reduces a little bit compared to the previous experiment.

The next two experiments related to SU-datasets are going to use different configuration mode which we call N-batch mode as mentioned in the offline analysis section.

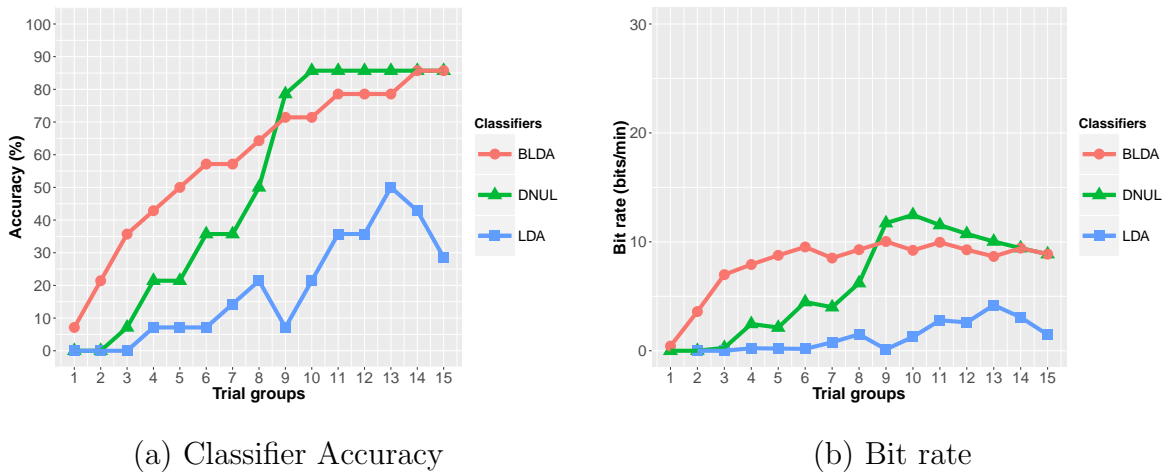


Figure 4.9: Offline analysis results for subject 1 with (Batch-14) configuration.

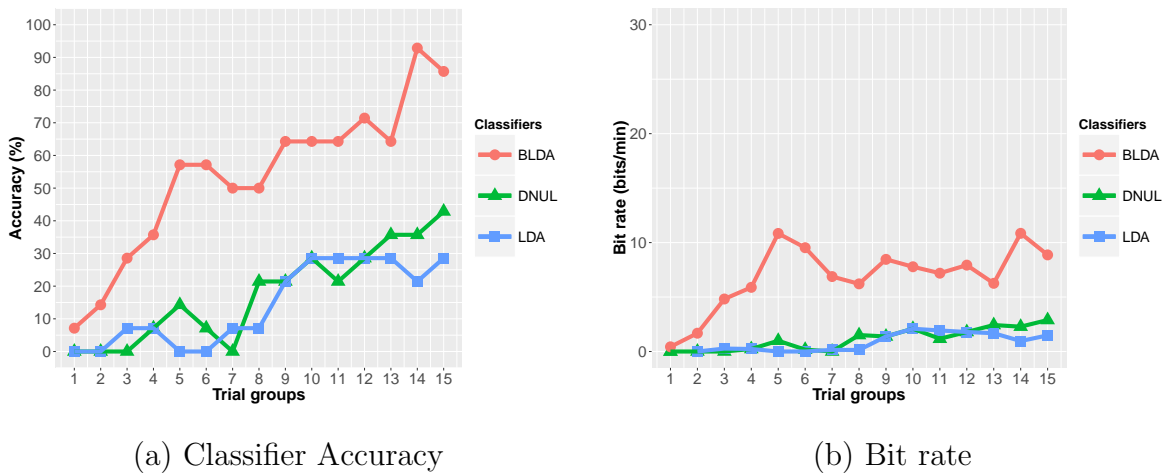
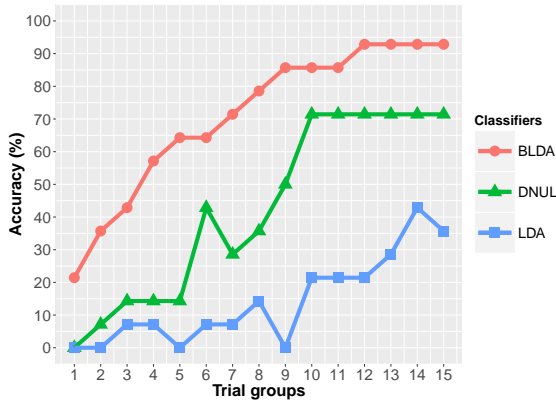
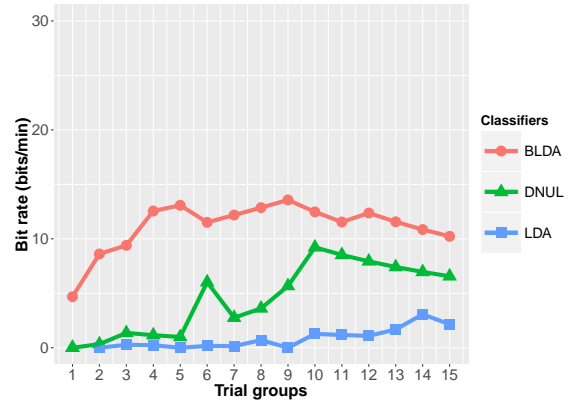


Figure 4.10: Offline analysis results for subject 2 with (Batch-14) configuration.

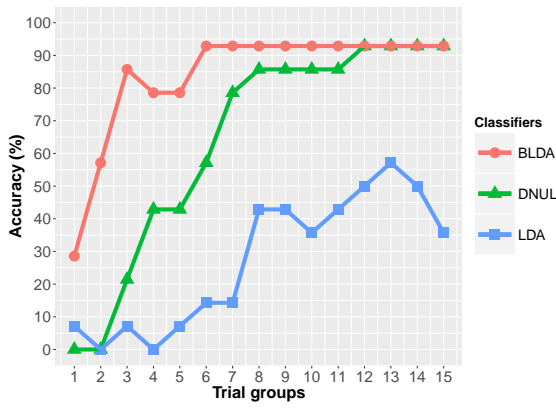


(a) Classifier Accuracy

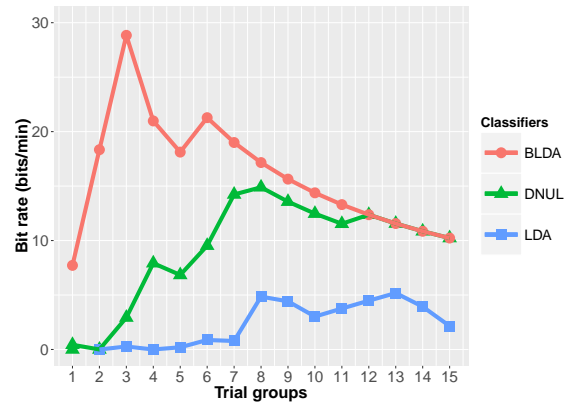


(b) Bit rate

Figure 4.11: Offline analysis results for subject 3 with (Batch-14) configuration.

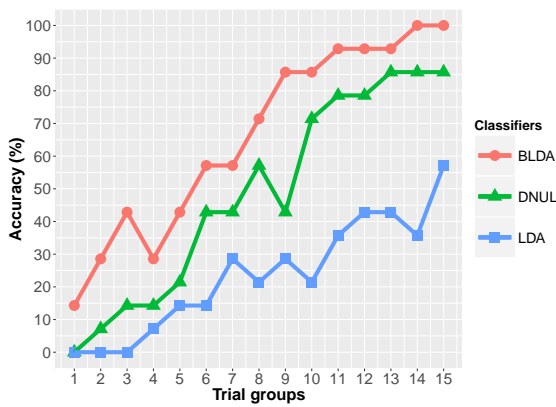


(a) Classifier Accuracy

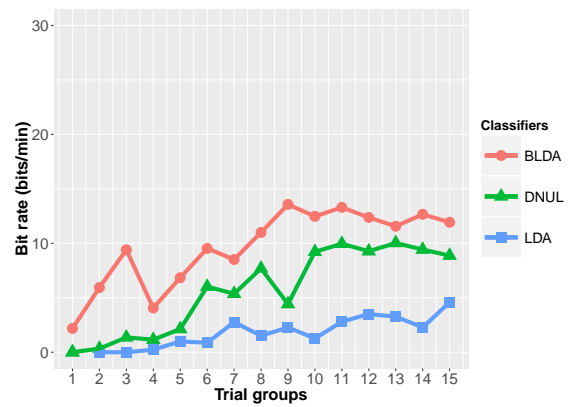


(b) Bit rate

Figure 4.12: Offline analysis results for subject 4 with (Batch-14) configuration.

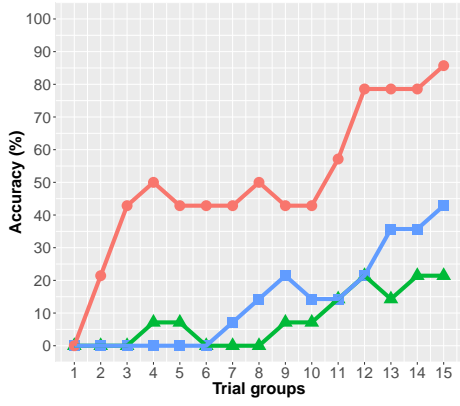


(a) Classifier Accuracy

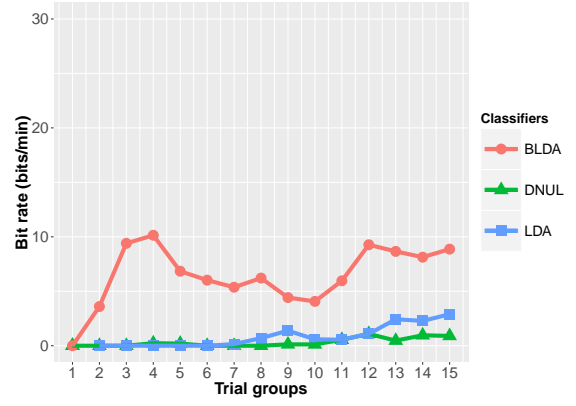


(b) Bit rate

Figure 4.13: Offline analysis results for subject 5 with (Batch-14) configuration.

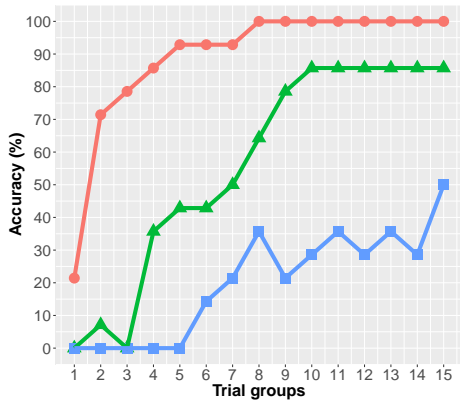


(a) Classifier Accuracy

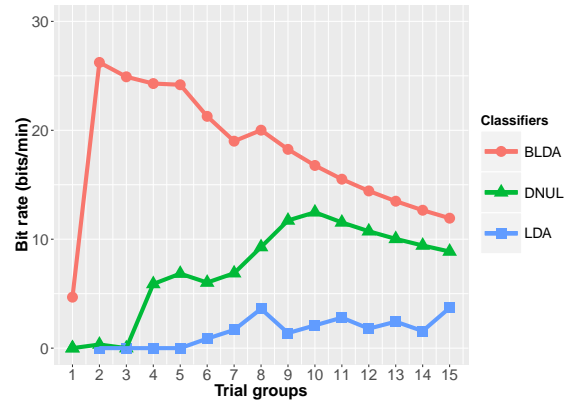


(b) Bit rate

Figure 4.14: Offline analysis results for subject 6 with (Batch-14) configuration.

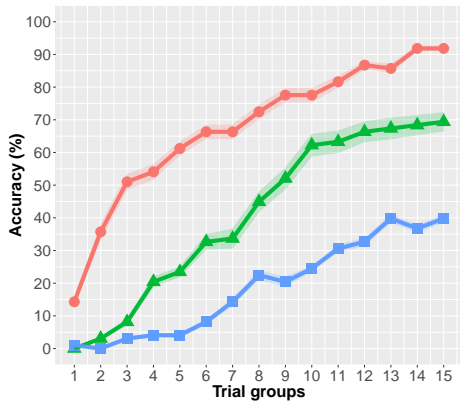


(a) Classifier Accuracy

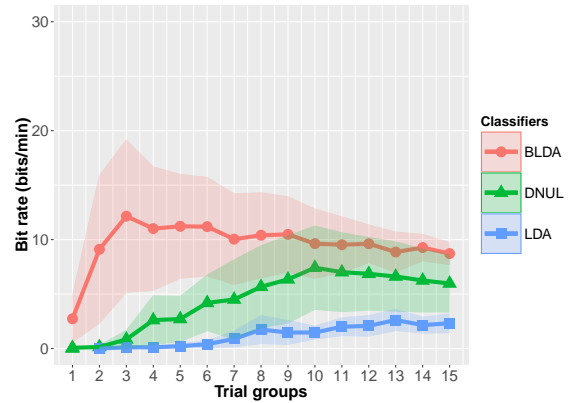


(b) Bit rate

Figure 4.15: Offline analysis results for subject 7 with (Batch-14) configuration.



(a) Classifier Accuracy



(b) Bit rate

Figure 4.16: Average classification performance over 7 subjects with (Batch-14). Error shadows show 95% confidence intervals from the mean with sample size = 7.

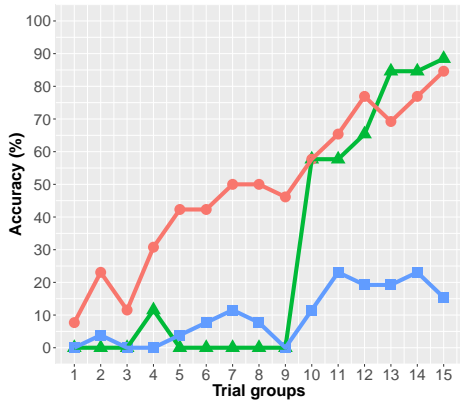
SU-Dataset: N-Batch mode with 26 characters in the test set

In this experiment, all classifiers are evaluated on the (Test 2) dataset as shown in Table A.1. This experiment follows the same methodology of Batch mode with 26 characters, but apart from this, it has a different configuration. With this configuration the number of trial groups the classifiers techniques use for supervised and unsupervised learning respectively matches the number of trial groups used for testing. The detailed accuracies for individual subjects corresponding to the experiments are given in Table 4.3 and are shown in Figures 4.17 to 4.24. The curves display classification accuracy and bit-rate as a function of the number of trial groups involved in each data sample used to test the classifiers.

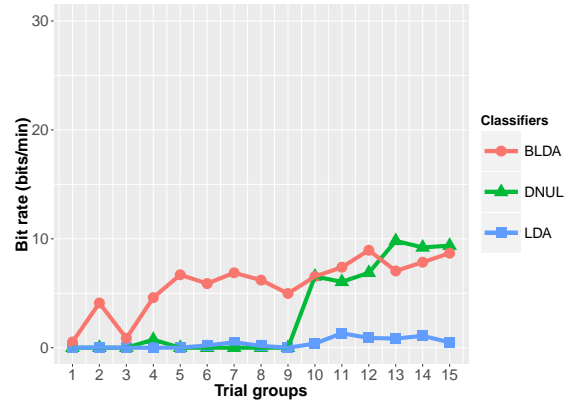
For this mode, it can be expected that the performance of most classifiers for the first few trial groups would be worse than the other mode. For instance, with 5 trial groups, the overall deterioration from batch mode to N-batch mode for accuracy of DNUL and BLDA is 19.78% and 8.8% respectively. Interestingly, the LDA classifier improved 8.25% with 5 trial groups. Table 4.3 gives more details for 10 and 15 trial groups.

Table 4.3: Percentage of correctly classified characters for each subject obtained with different values of trial groups for (N-Batch-26).

Subjects	5 trial groups			10 trial groups			15 trial groups		
	DNUL	LDA	BLDA	DNUL	LDA	BLDA	DNUL	LDA	BLDA
S1	0	3.85	42.31	57.69	11.54	57.69	88.46	15.38	84.62
S2	3.85	7.69	42.31	3.85	23.08	61.54	69.23	34.62	88.46
S3	15.38	11.54	50	61.54	23.08	88.46	76.92	34.62	96.15
S4	19.23	19.23	80.77	92.31	30.77	96.15	88.46	61.54	92.31
S5	0	7.69	53.85	65.38	19.23	88.46	100	50	100
S6	0	3.85	23.08	3.85	15.38	65.38	42.31	50	84.62
S7	19.23	46.15	100	3.85	38.46	100	96.15	61.54	100
Average	8.24	14.29	56.04	41.21	23.08	79.67	80.22	43.96	92.31

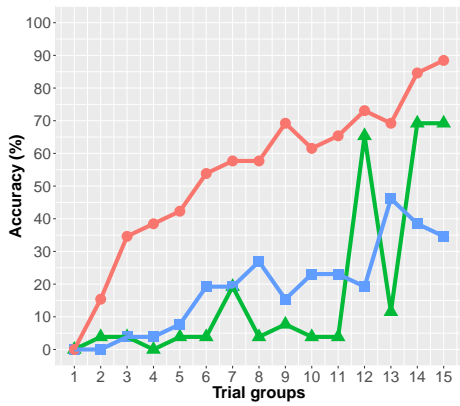


(a) Classifier Accuracy

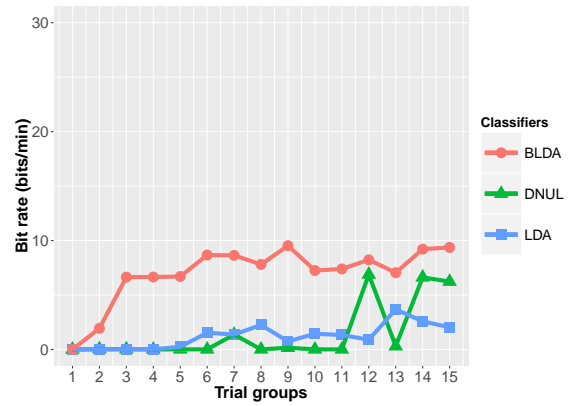


(b) Bit rate

Figure 4.17: Offline analysis results for subject 1 with (N-Batch-26) configuration.

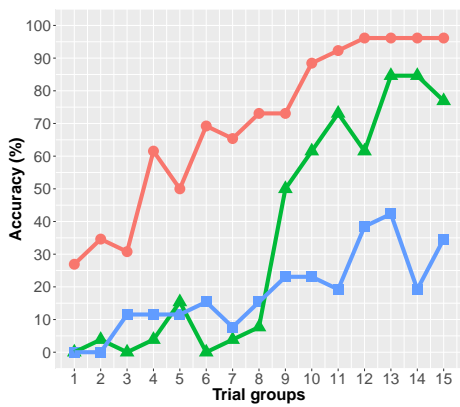


(a) Classifier Accuracy

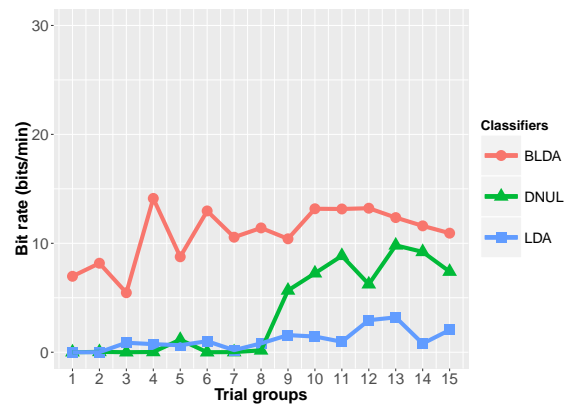


(b) Bit rate

Figure 4.18: Offline analysis results for subject 2 with (N-Batch-26) configuration.

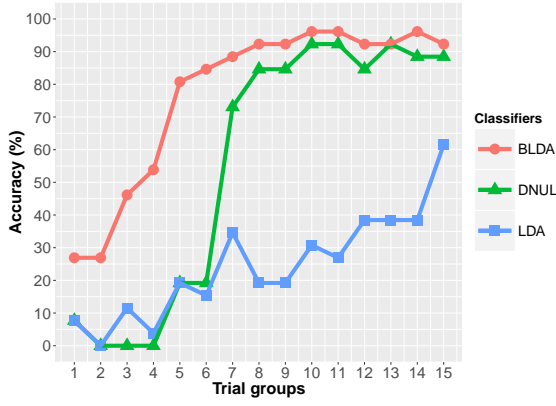


(a) Classifier Accuracy

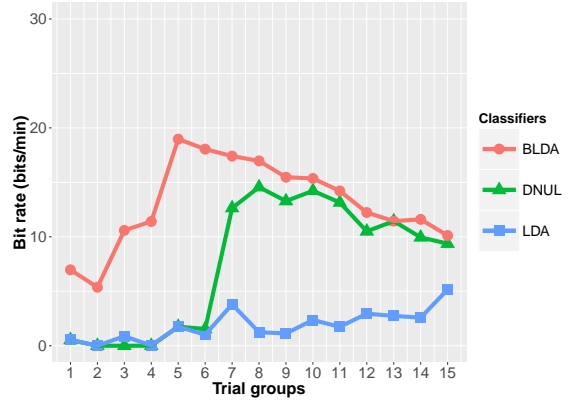


(b) Bit rate

Figure 4.19: Offline analysis results for subject 3 with (N-Batch-26) configuration.

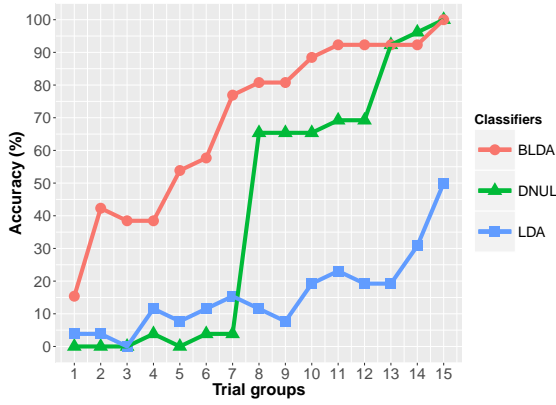


(a) Classifier Accuracy

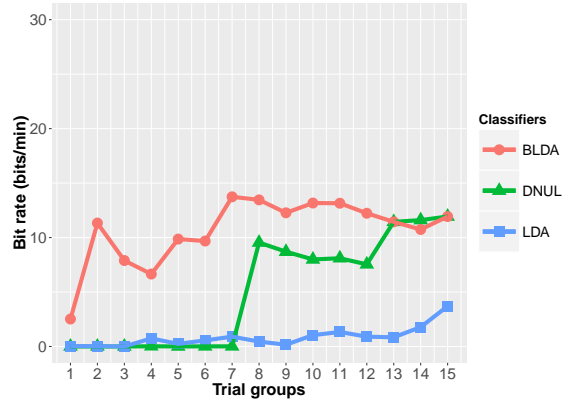


(b) Bit rate

Figure 4.20: Offline analysis results for subject 4 with (N-Batch-26) configuration.

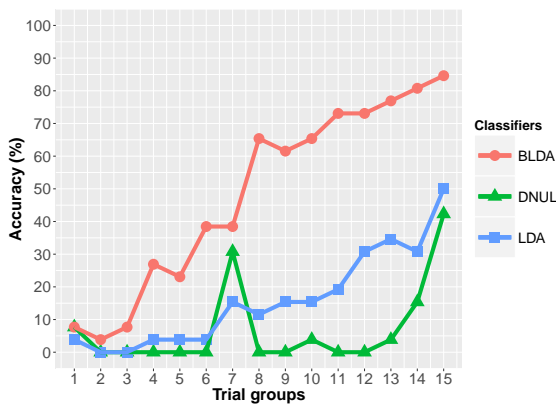


(a) Classifier Accuracy

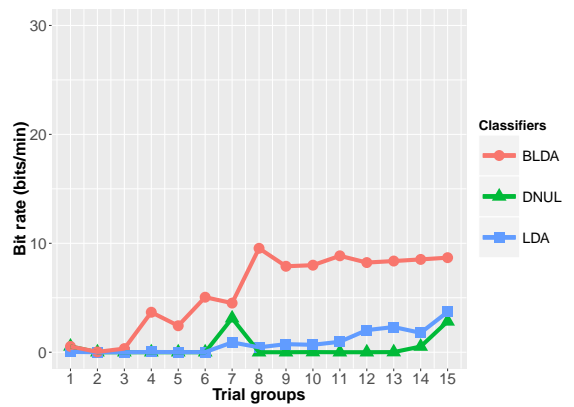


(b) Bit rate

Figure 4.21: Offline analysis results for subject 5 with (N-Batch-26) configuration.

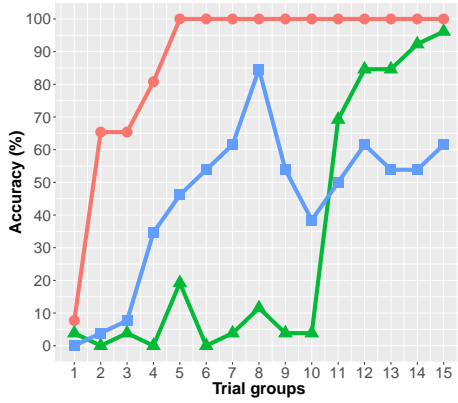


(a) Classifier Accuracy

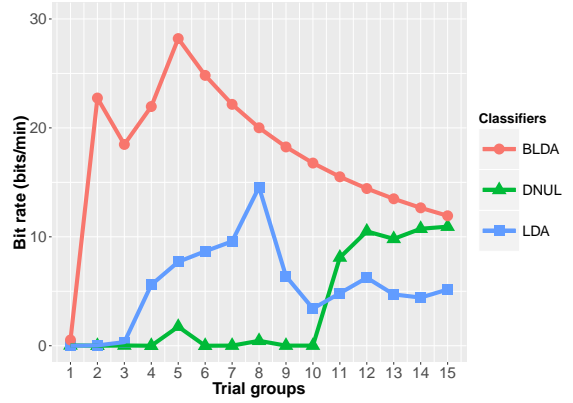


(b) Bit rate

Figure 4.22: Offline analysis results for subject 6 with (N-Batch-26) configuration.

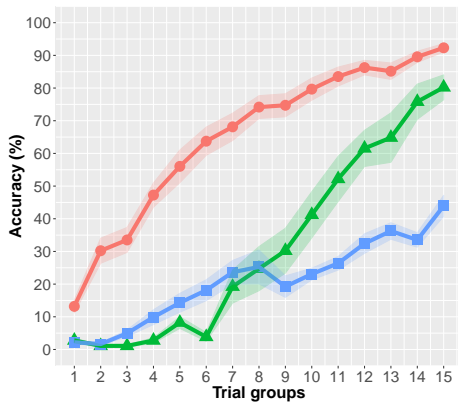


(a) Classifier Accuracy

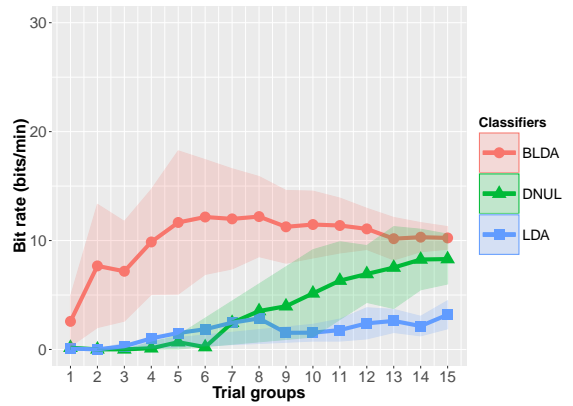


(b) Bit rate

Figure 4.23: Offline analysis results for subject 7 with (N-Batch-26) configuration.



(a) Classifier Accuracy



(b) Bit rate

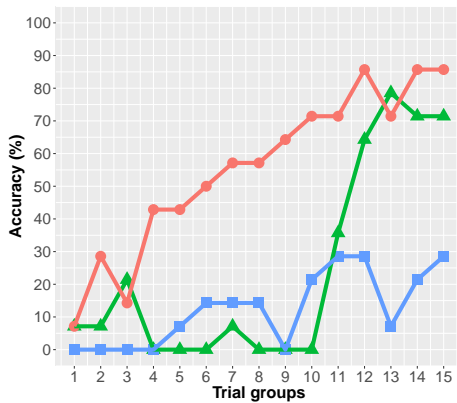
Figure 4.24: Average classification performance over 7 subjects with (N-Batch-26). Error shadows show 95% confidence intervals from the mean with sample size = 7.

SU-Dataset: N-Batch mode with 14 characters in the test set

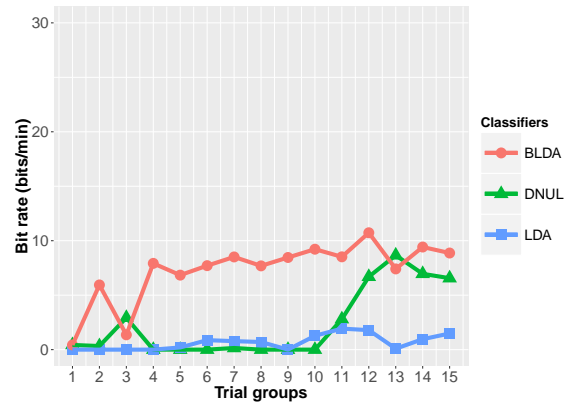
In this experiment, all classifiers are evaluated on the (Test 2) dataset as shown in Table A.1. This experiment uses the N-batch mode with 14. The detailed accuracies for individual subjects corresponding to the experiments are given in Table 4.4 and can be shown in Figures 4.25 to 4.32. The curves display classification accuracy and bit-rate as a function of the number of trial groups involved in each data sample used to test the classifiers.

Table 4.4: Percentage of correctly classified characters for each subject obtained with different values of trial groups for (N-Batch-14).

Subjects	5 trial groups			10 trial groups			15 trial groups		
	DNUL	LDA	BLDA	DNUL	LDA	BLDA	DNUL	LDA	BLDA
S1	0	7.14	42.86	0	21.43	71.43	71.43	28.57	85.71
S2	7.14	7.14	50	0	28.57	64.29	0	28.57	85.71
S3	7.14	14.29	50	57.14	28.57	85.71	71.43	35.71	92.86
S4	0	21.43	78.57	92.86	35.71	100	92.86	35.71	92.86
S5	0	0	28.57	78.57	14.29	100	85.71	57.14	100
S6	7.14	7.14	21.43	21.43	21.43	64.29	28.57	42.86	85.71
S7	7.14	50	100	7.14	42.86	100	85.71	50	100
Average	4.08	15.31	53.06	36.73	27.55	83.67	62.24	39.8	91.84

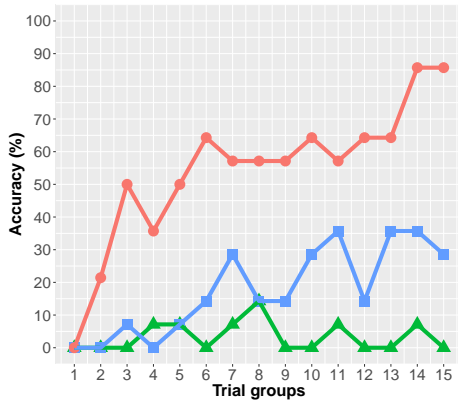


(a) Classifier Accuracy

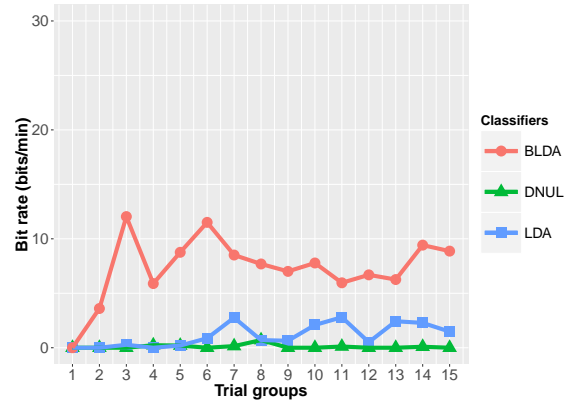


(b) Bit rate

Figure 4.25: Offline analysis results for subject 1 with (N-Batch-14) configuration.

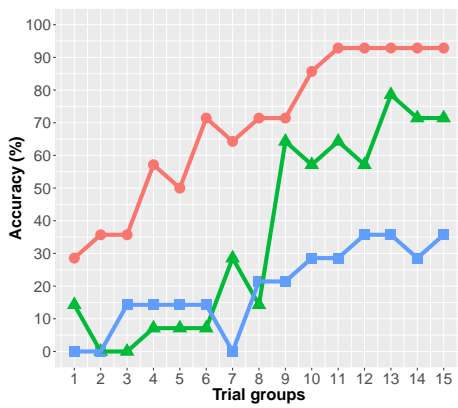


(a) Classifier Accuracy

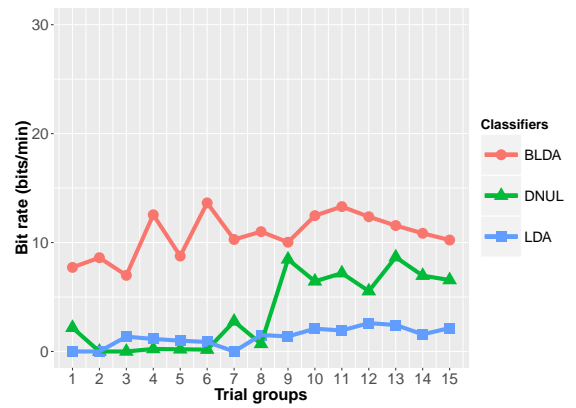


(b) Bit rate

Figure 4.26: Offline analysis results for subject 2 with (N-Batch-14) configuration.

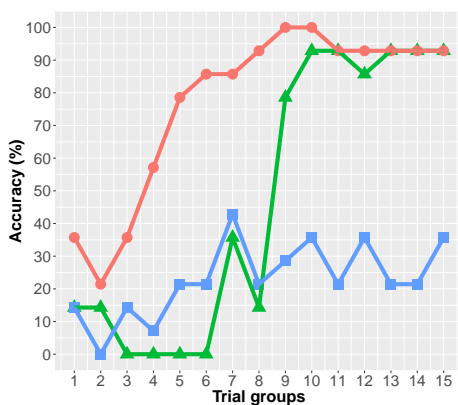


(a) Classifier Accuracy

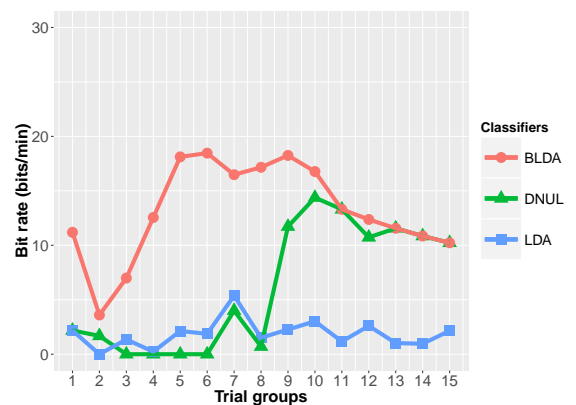


(b) Bit rate

Figure 4.27: Offline analysis results for subject 3 with (N-Batch-14) configuration.

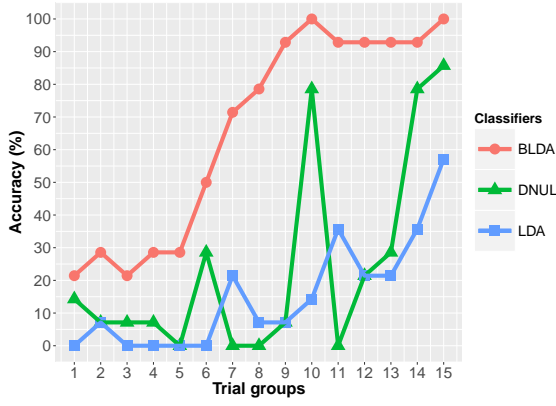


(a) Classifier Accuracy

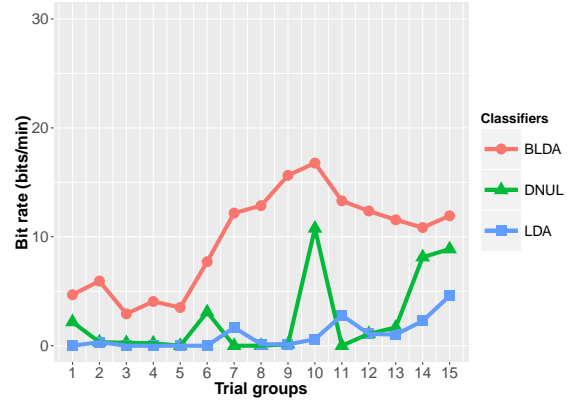


(b) Bit rate

Figure 4.28: Offline analysis results for subject 4 with (N-Batch-14) configuration.

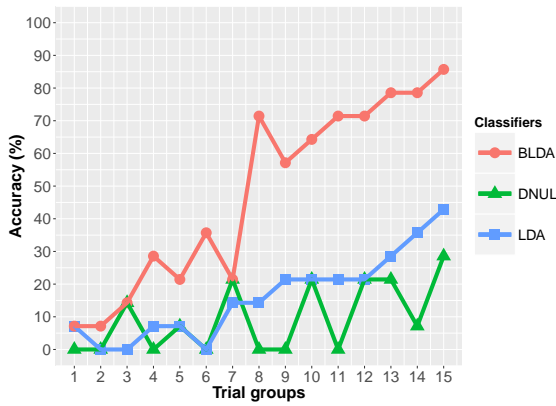


(a) Classifier Accuracy

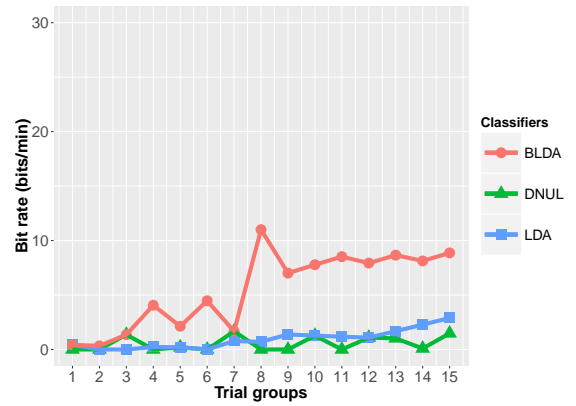


(b) Bit rate

Figure 4.29: Offline analysis results for subject 5 with (N-Batch-14) configuration.

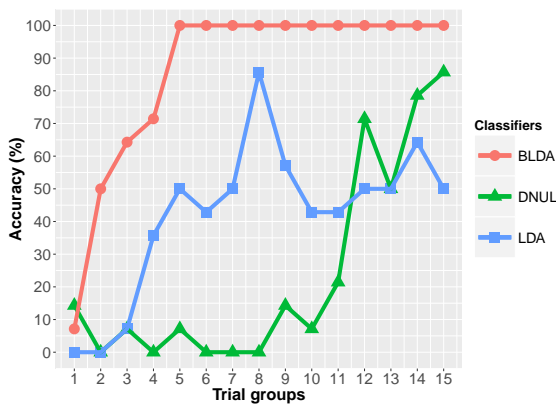


(a) Classifier Accuracy

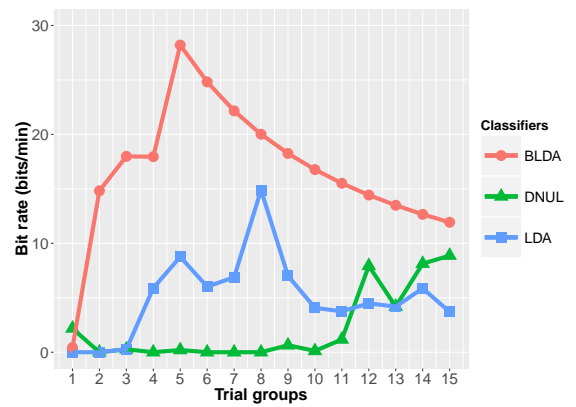


(b) Bit rate

Figure 4.30: Offline analysis results for subject 6 with (N-Batch-14) configuration.



(a) Classifier Accuracy



(b) Bit rate

Figure 4.31: Offline analysis results for subject 7 with (N-Batch-14) configuration.

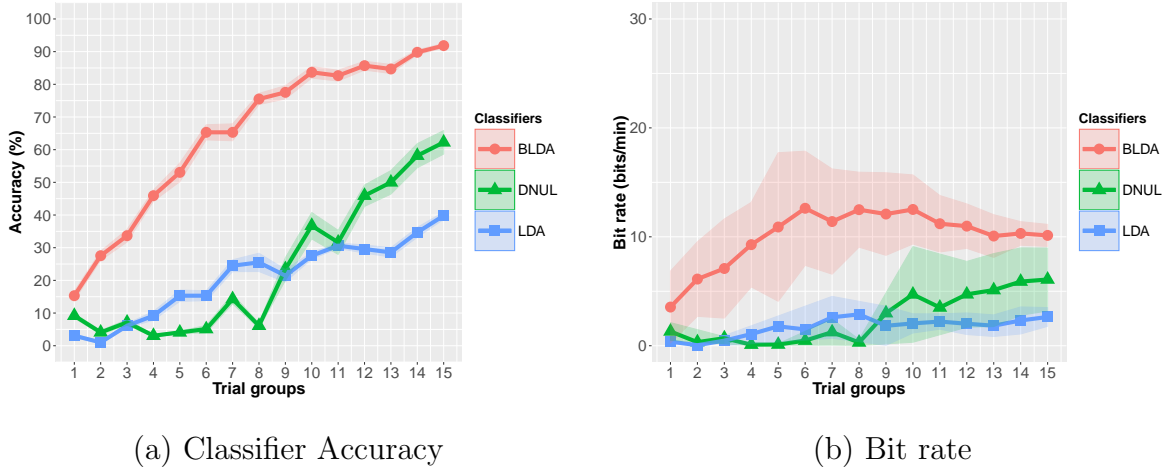


Figure 4.32: Average classification performance over 7 subjects with (N-Batch-14). Error shadows show 95% confidence intervals from the mean with sample size = 7.

Overall, these results demonstrate the unsupervised classification capability of DNUL against the LDA; the most comparable supervised classifier in terms of classification decisions and simplicity. Although the BLDA performed better than DNUL, BLDA is one of the most powerful classifiers for P300 since it uses regularization, still our proposed classifier learned without any labelled data in comparison with BLDA and gives a high accuracy especially with larger number of trial groups. Interestingly, when the unlabelled data quality (through more repetitions) and quantity (through more characters) is sufficiently high, DNUL appears to provide better performance than supervised LDA which is trained on labelled data from a separate session. We speculate this might be due to the non-stationary nature of the EEG data across the sessions. Same for BLDA, but for this case our DNUL classifiers appears to provide a high accuracies with some of the subjects (For instance, some subjects with DNUL exceeded the BLDA performance in the last 5 trial groups, and one subject exceeded the BLDA accuracy from the 9th trial group as shown in Figure 4.10) and these variations in accuracies appears due to the non-stationary nature of the EEG data across the sessions.

Regularized DNUL (RDNUL)

A direction toward improving the DNUL performance is to add a regularization term as proposed in section 3.4. As it can be seen form the figures below, regular-

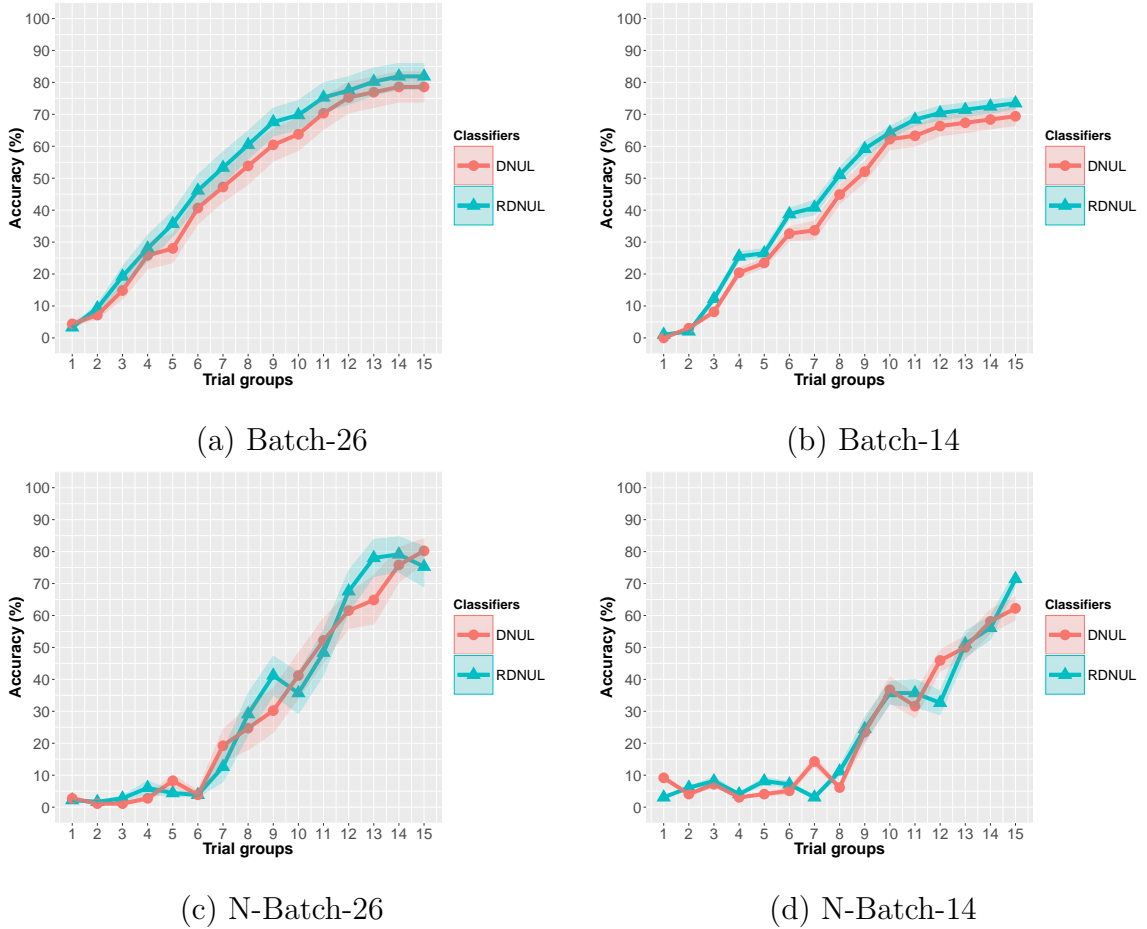


Figure 4.33: Average classification performance comparing the regularized DNUL classifier with DNUL. Error shadows corresponding to the point show 95% confidence intervals from the mean with sample size = 7.

ization term slightly increased the accuracies for all the experiment described above in comparison with DNUL.

Figure 4.33 shows the overall accuracies from the DNUL and the RDNUL which follow the same experiments methodology. For Batch mode, we can see that RDNUL accuracies exceeded the DNUL among all number of trial groups. On the other hand, with N-batch mode, accuracies remain close across trial groups. Moving to the bit-rate, Figure 4.34 shows an overview of bit-rates for the described experiments with RDNUL bitrates. Logically, It can be proved that the bit-rate is directly proportional with the accuracy performance of the classifier. That demonstrates that RDNUL bit-rates for batch mode slightly increased compared to DNUL.

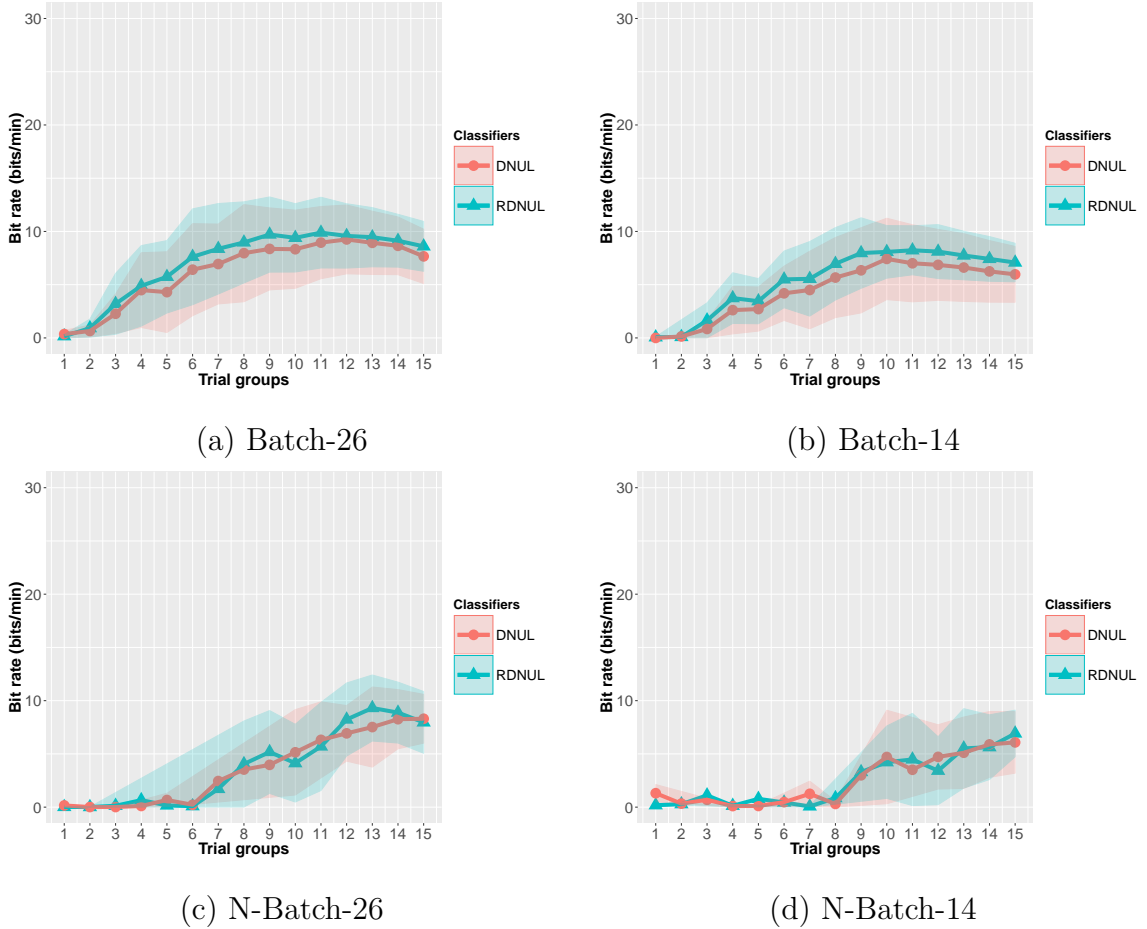


Figure 4.34: Bit rate performance comparing the regularized DNUL classifier with DNUL. Error shadows corresponding to the point show 95% confidence intervals from the mean with sample size = 7.

BCI Competition Dataset II & III Results

Before going through the performance results for the proposed method with the BCI competition datasets, different pre-processing techniques are applied to the BCI competition dataset in order to get a good representation of P300 signals. Appendix A describes the properties of the datasets, time, and channel parameters. In this section we are going to evaluate the proposed classifier with two datasets from BCI competition which contains 3 subjects A, B and C. The pre-processing procedure used for these datasets is listed below:

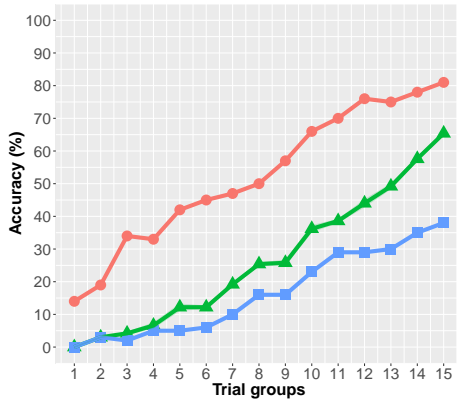
- A **version** with 12 electrodes out of 64 was chosen by the position of central, parietal, and occipital lobes where the P300 signal is known to be more apparent. The selected channels are Fz,Cz,Pz,Oz,O1,O2,P3,P4,CPz,FCz,PO7 and

PO8. Note that 8 of these channels were also used in our own EEG recordings (SU-Dataset) as in [37].

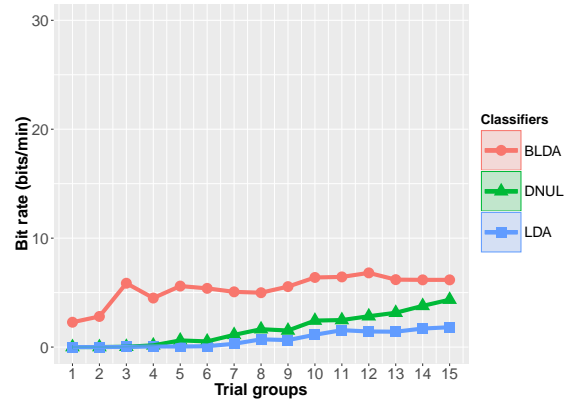
- Another **version** with the whole 64 electrodes was chosen and used in comparison with the relevant work in [11].
- Averaging all 64 channels signal is computed and then subtracted from each channel for referencing. This method called a Common Average Reference (CAR) as in [72]. Unlike SU-Dataset recordings which we had the mastoid channels that were not used in this dataset for reference purpose.
- A band-pass filter with cut off frequencies 1-30 Hz was applied to the whole data and the data were normalized and windsorized.
- An epoch was extracted with a time frame of 0-667 ms post stimulus, starting from the sample where the stimulus is presented. The dimensionality was reduced by sub-sampling with a factor 8 and 20 samples for each channel was retained.

The following experiments evaluated the proposed method with different datasets using the previous pre-processing technique. The experiments below will follow the same methodology for batch mode and N-batch mode configurations. Subject A is considered as the hardest subject in terms of classifying the P300 component. Subject C is considered an easy subject where classifying the P300 component is easier. Bit-rate is also depicted in the figures as a function of the number of trial groups.

BCI Competition III Dataset: Subject A (Batch-mode)

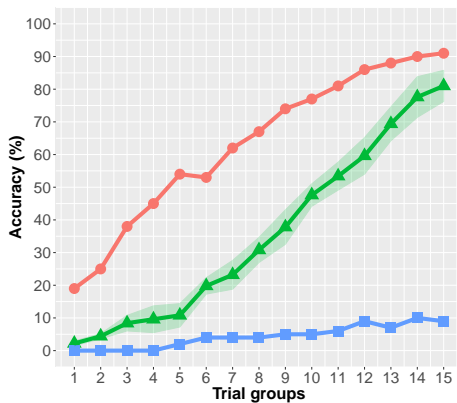


(a) Classifier Accuracy

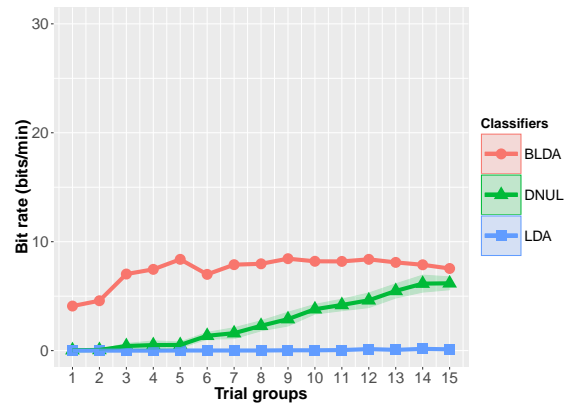


(b) Bit rate

Figure 4.35: Average classification performance for subject A over 5 classifier groups using 12 electrodes with a batch mode configuration.



(a) Classifier Accuracy

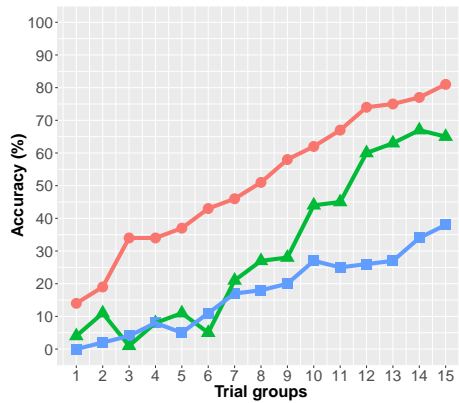


(b) Bit rate

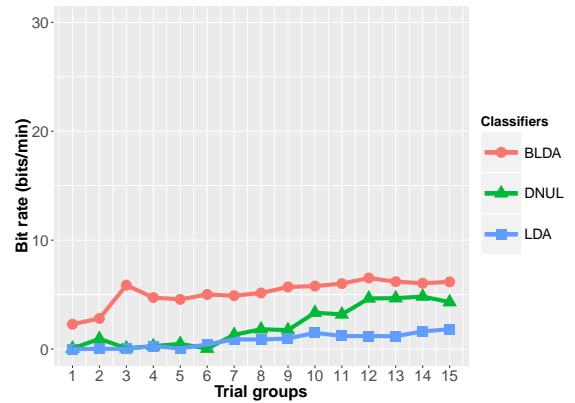
Figure 4.36: Average classification performance for subject A over 5 classifier groups using 64 electrodes with a batch mode configuration.

As seen in the Figure 4.35 and 4.36, the classification accuracies reached the maximum with 15 trial groups. DNUL shows its effectiveness compared with the LDA classifier. It can be noticed that DNUL performance is slightly higher when using more electrodes. Interestingly, LDA classifier totally failed in case of 64 electrodes due to high dimensionality of the feature vector against other classifiers. Bit rates are increased gradually with more trial received compared to LDA. However, it is almost stable with BLDA. Table 4.5 and 4.6 show the percentage accuracies.

BCI Competition III Dataset: Subject A (N-Batch-mode)

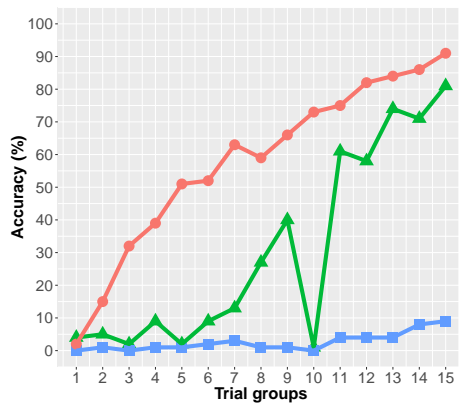


(a) Classifier Accuracy

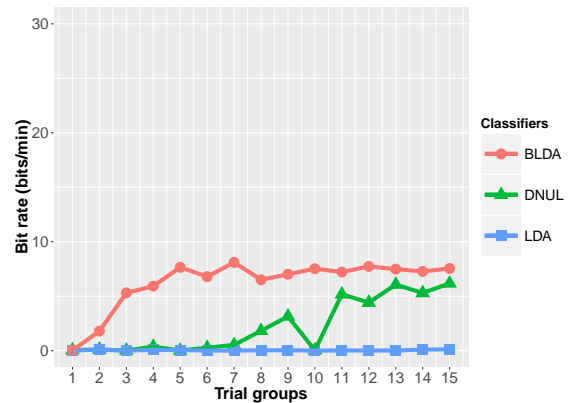


(b) Bit rate

Figure 4.37: Classification performance for subject A using 12 electrodes with a N-batch mode configuration.



(a) Classifier Accuracy

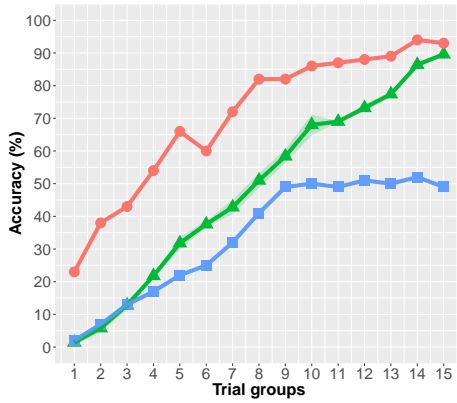


(b) Bit rate

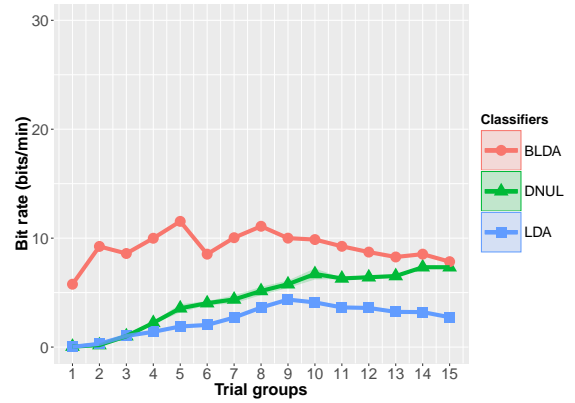
Figure 4.38: Classification performance for subject A using 64 electrodes with a N-batch mode configuration.

Figure 4.37 and 4.38 show the classification accuracies for the N-batch mode configurations. It can be seen in case of using 64 electrodes that the DNUL failed with the 10th trial group and got 0% accuracy. The reason is the 10 optimizations are not sufficient enough to reach a high energy function for that case. That does not imply we always get 0% accuracy with this amount of trial groups. Table 4.7 shows the percentage accuracies in detail for the used classifiers.

BCI Competition III Dataset: Subject B (Batch-mode)

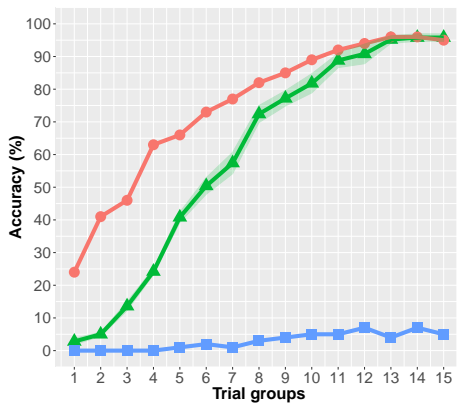


(a) Classifier Accuracy

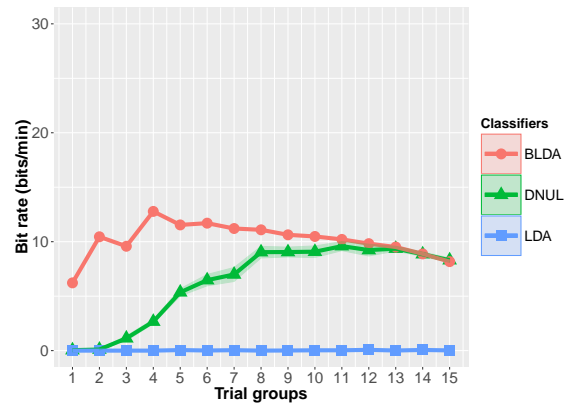


(b) Bit rate

Figure 4.39: Average classification performance for subject B over 5 classifier groups using 12 electrodes with a batch mode configuration.



(a) Classifier Accuracy

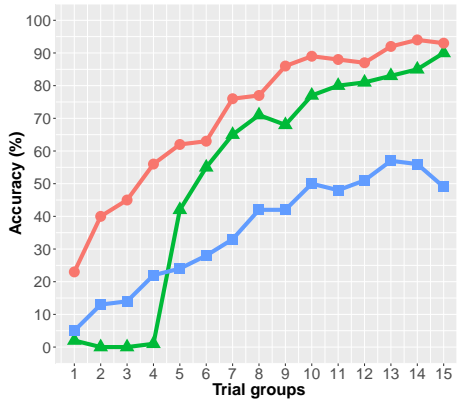


(b) Bit rate

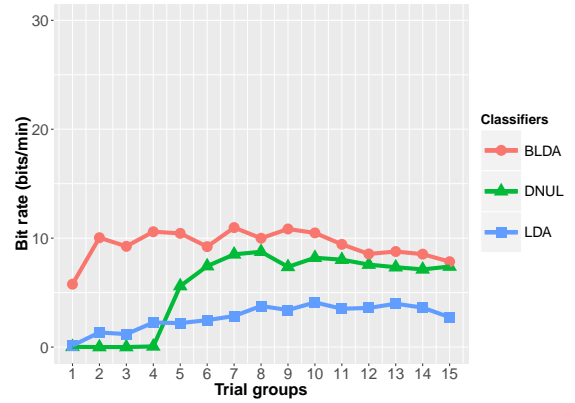
Figure 4.40: Average classification performance for subject B over 5 classifier groups using 64 electrodes with a batch mode configuration.

As seen in the Figure 4.39 and 4.40, this time DNUL reached the highest accuracy compared to other supervised classifiers with 64 electrodes. Furthermore, the BLDA and DNUL have close accuracies in case of 12 electrodes. Table 4.5 and 4.6 show the percentage accuracies in details for the used classifiers. Note that, Table 4.5 and 4.6 shows the 5 classifier groups in detail.

BCI Competition III Dataset: Subject B (N-Batch-mode)

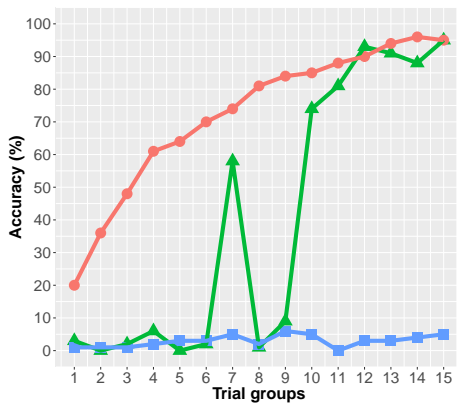


(a) Classifier Accuracy

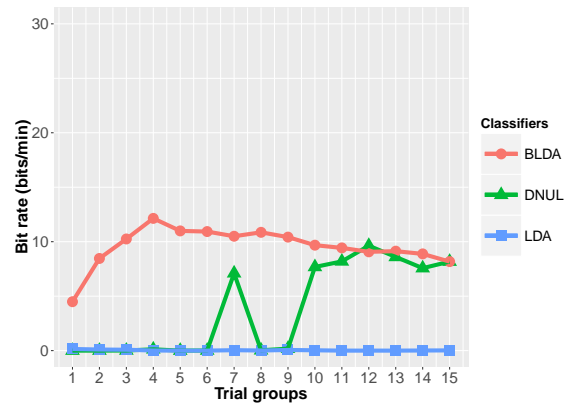


(b) Bit rate

Figure 4.41: Classification performance for subject B using 12 electrodes with a N-batch mode configuration.



(a) Classifier Accuracy

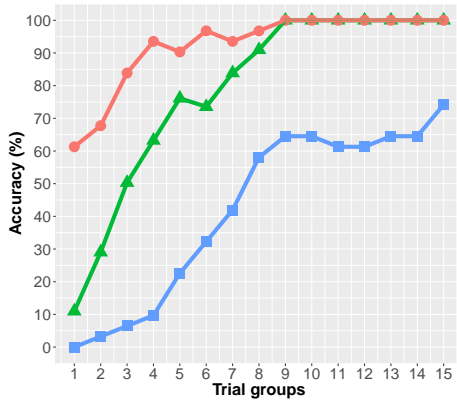


(b) Bit rate

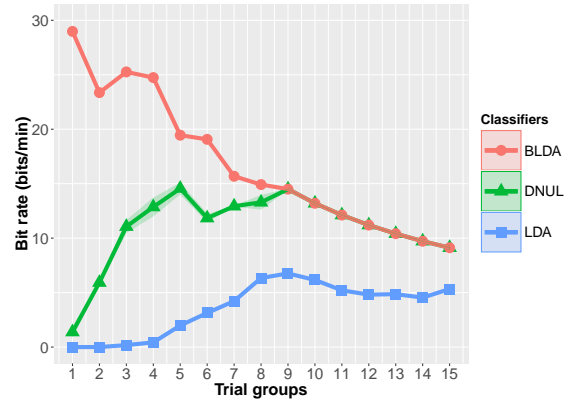
Figure 4.42: Classification performance for subject B using 64 electrodes with a N-batch mode configuration.

It can be seen in this case of using 64 electrodes that the DNUL failed with some trial groups and as we said the reason is the about 10 optimizations are not sufficient to reach a high energy function as shown in Figure 4.41 and 4.42. Table 4.7 shows the percentage accuracies in detail for the used classifiers.

BCI Competition II Dataset: Subject C (Batch-mode)

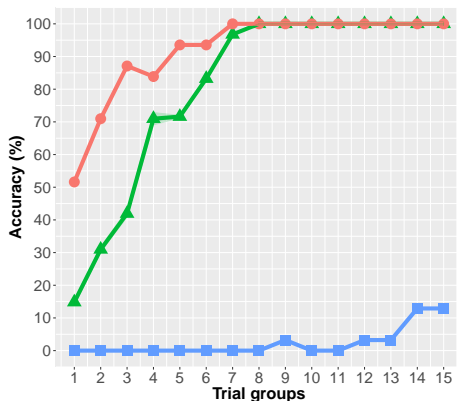


(a) Classifier Accuracy

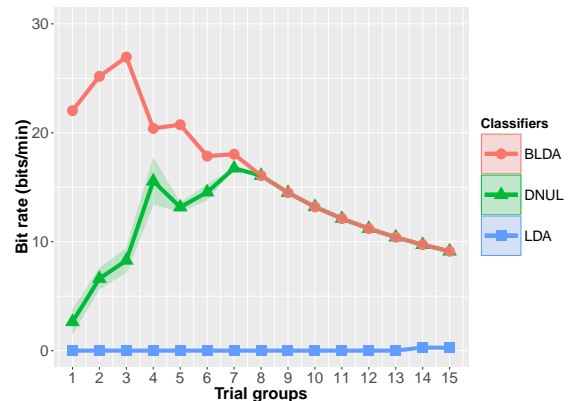


(b) Bit rate

Figure 4.43: Average classification performance for subject C over 5 classifier groups using 12 electrodes with a batch mode configuration.



(a) Classifier Accuracy

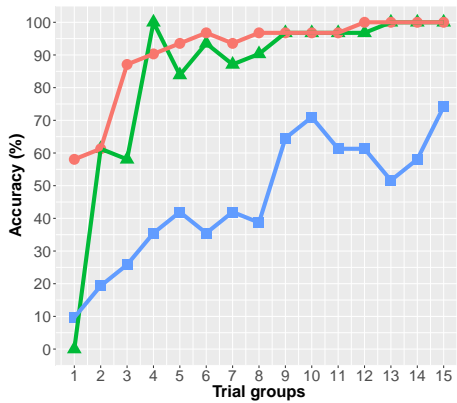


(b) Bit rate

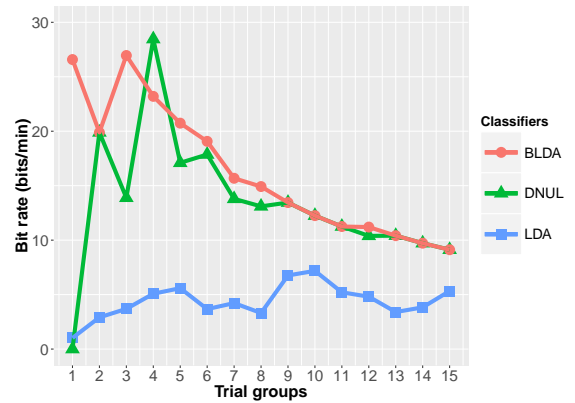
Figure 4.44: Average classification performance for subject C over 5 classifier groups using 64 electrodes with a batch mode configuration.

Subject C is considered as an easy subject. DNUL classifier reached to 100% classification accuracy with this subject. Using the whole set of electrodes slightly improved the character classification with smaller number of trial group. Bit rates are high and increased gradually with more trial received compared to LDA. Table 4.5 and 4.6 show the percentage accuracies in detail using 5 classifier groups for the used classifiers.

BCI Competition II Dataset: Subject C (N-Batch-mode)

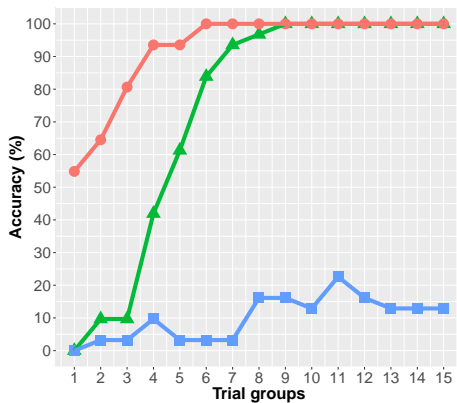


(a) Classifier Accuracy

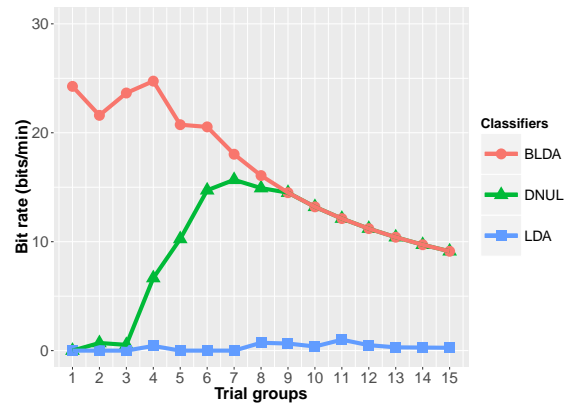


(b) Bit rate

Figure 4.45: Classification performance for subject C using 12 electrodes with a N-batch mode configuration.



(a) Classifier Accuracy



(b) Bit rate

Figure 4.46: Classification performance for subject C using 64 electrodes with a N-batch mode configuration.

As can be seen with N-batch configuration for subject C, the DNUL classifier reached 100% classification accuracy with only 4 trial groups. That demonstrates that our DNUL classifier is able to compete with other supervised classification accuracies. Table 4.7 shows the percentage accuracies in detail for the used classifiers.

BCI Competition II & III Results Summary Tables

Table 4.5: **Batch mode:** percentage of correctly classified characters for subjects A, B, and C for 12 electrodes. The values in braces are the standard deviation.

	R	LDA	BLDA	DNUL(1)	DNUL(2)	DNUL(3)	DNUL(4)	DNUL(5)	DNUL
A	5	5	42	11	12	14	12	12	12.2 (1.1)
	10	23	66	35	37	35	37	37	36.2 (1.1)
	15	38	81	66	65	65	65	66	65.4 (0.54)
B	5	22	66	31	31	35	30	32	31.8 (1.92)
	10	50	86	70	70	62	69	69	68 (3.39)
	15	49	93	90	89	90	90	89	89.6 (0.55)
C	5	22.58	90.32	74.19	77.42	77.42	77.42	74.19	76.13 (0.55)
	10	64.51	100	100	100	100	100	100	100 (0.0)
	15	74.19	100	100	100	100	100	100	100 (0.0)

Table 4.6: **Batch mode:** percentage of correctly classified characters for subjects A, B, and C for 64 electrodes. The values in braces are the standard deviation.

	R	LDA	BLDA	DNUL(1)	DNUL(2)	DNUL(3)	DNUL(4)	DNUL(5)	DNUL
A	5	2	54	10	4	13	14	13	10.8 (4.09)
	10	5	77	43	44	52	49	50	47.6 (3.91)
	15	9	91	80	72	85	83	85	81 (5.43)
B	5	1	66	41	43	42	39	39	40.8 (1.8)
	10	5	89	83	82	84	76	84	81.8 (3.34)
	15	5	95	96	98	95	95	95	95.8 (1.3)
C	5	0	93.55	70.96	70.96	74.19	70.96	70.96	71.61 (0.45)
	10	0	100	100	100	100	100	100	100 (0.0)
	15	12.9	100	100	100	100	100	100	100 (0.0)

Table 4.7: **N-Batch mode:** percentage of correctly classified characters for subjects A, B, and C for 12 and 64 electrodes configuration.

		12 electrodes			64 electrodes		
	R	LDA	BLDA	DNUL	LDA	BLDA	DNUL
A	5	5	37	11	1	51	2
	10	27	62	44	0	73	1
	15	38	81	65	9	91	81
B	5	24	62	42	3	64	0
	10	50	89	77	5	85	74
	15	49	93	90	5	95	95
C	5	41.94	93.55	83.87	3.22	93.5	61.29
	10	70.97	96.77	96.77	12.9	100	100
	15	74.19	100	100	12.9	100	100

4.7.2 Simulated Online Analysis

For this experiment an online simulation has been done to test the adaptation process of DNUL. We design and update the classifier after the data are received for each character and perform classification. Finally, we also perform an offline retest, that is we classify each previously seen character with the final classifier as shown in Figure 4.47. This experiment demonstrates how DNUL can in principle be adapted and refined as more test data are received.

We evaluated the online experiment by using the same dataset as used before with the offline experiments as shown in Appendix A. The pre-processing steps are done as described previously in this chapter. The only difference is we pre-processed the data character by character in order to have a fair online simulation procedure.

Letters **KALEM_YOLCULUK**
 DNUL **KALEM_YOLCULUK**
 Retest **KALEM_YOLCULUK**

Figure 4.47: An arbitrary example shows a sequence of letters for simulated online spelling.

One step further, the configuration of the classifier is also the same as the previous experiments.

The sequence of the online experiment is as follows. The classifier starts in the beginning without using any training data. In order to classify a character we are adding the EEG data for the received character. Then we apply the DNUL classifier with 10 optimizations in order to classify the character. For the next character, we increase the amount of training dataset for the classifier to classify the next character. Note that for BCI competition III, we used 3 optimizations to perform the online experiments in a reasonable time because the large data size.

Overall, we observe that the classifier is improved when we either receive more data or increase the number of used electrodes. As expected, the classifier performs better if the data involve more trial groups. Figure 4.48 shows the average online spelling for the SU dataset with the 7 subjects. The horizontal axis represents the number of processed characters. The vertical axis represents the number of characters that were classified correctly. The dashed line is an upper bound showing the number of the seen characters. Figures from 4.49 to 4.51 show the performance of online spelling with the BCI competition datasets.

One of the observations is about the retest classifier, it should perform better than the adaptive process (DNUL), as shown in Figure 4.51. However, in some cases it gives a lower performance as in Figure 4.50 with the case of 10-trial groups using 64 electrodes. The reason is the adaptivity process failed at some point as the classifier couldn't reach a good position and that reflects on the slope of the DNUL classifier. The higher slope we got, the faster adaptivity process is performed. That imply the last classifier not in a good position and failed to classify the character because the number of optimizations was not enough or due to the non-stationarity of the data.

SU subjects: Average of 7 subjects (Simulated online analysis)

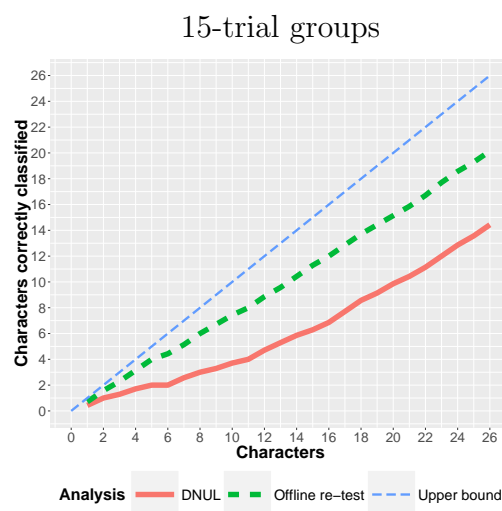
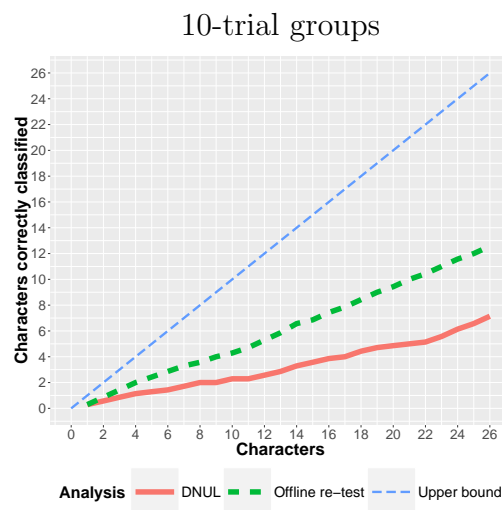
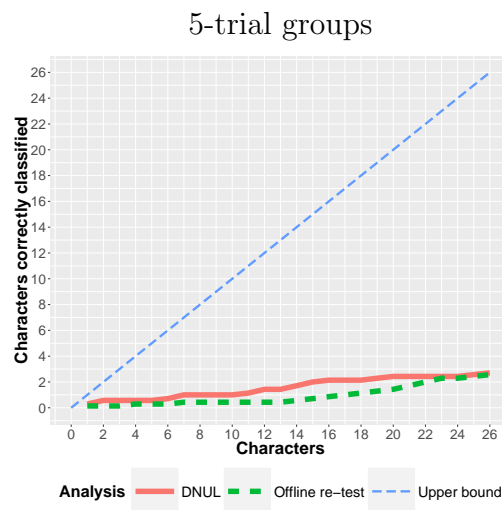


Figure 4.48: Simulated online spelling (sequential mode) showing the performance averaged over the 7 subjects.

BCI Competition III Dataset: Subject A (Simulated online analysis)

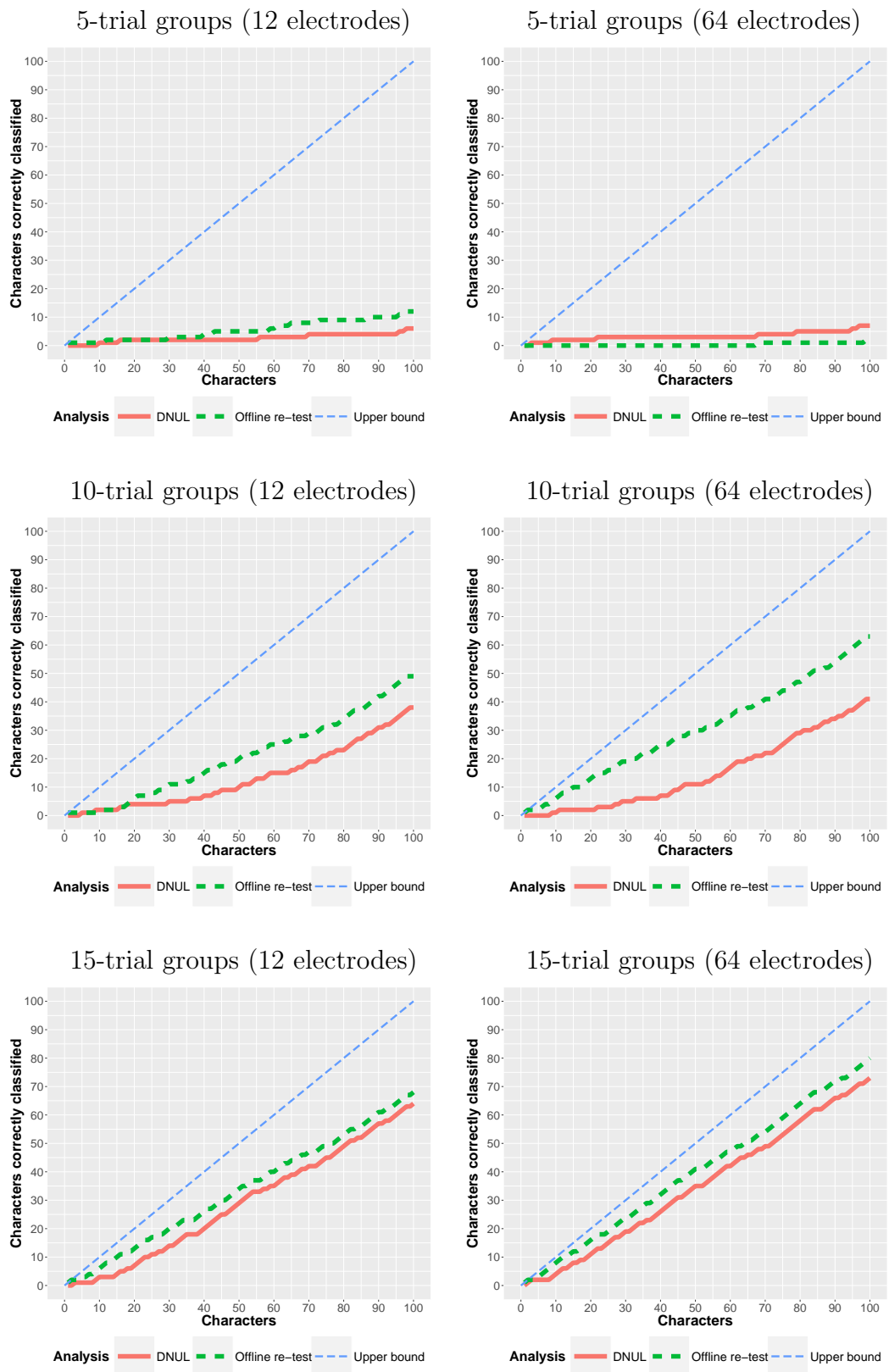


Figure 4.49: Simulated online spelling using 12 electrodes and 64 electrodes for subject A.

BCI Competition III Dataset: Subject B (Simulated online analysis)

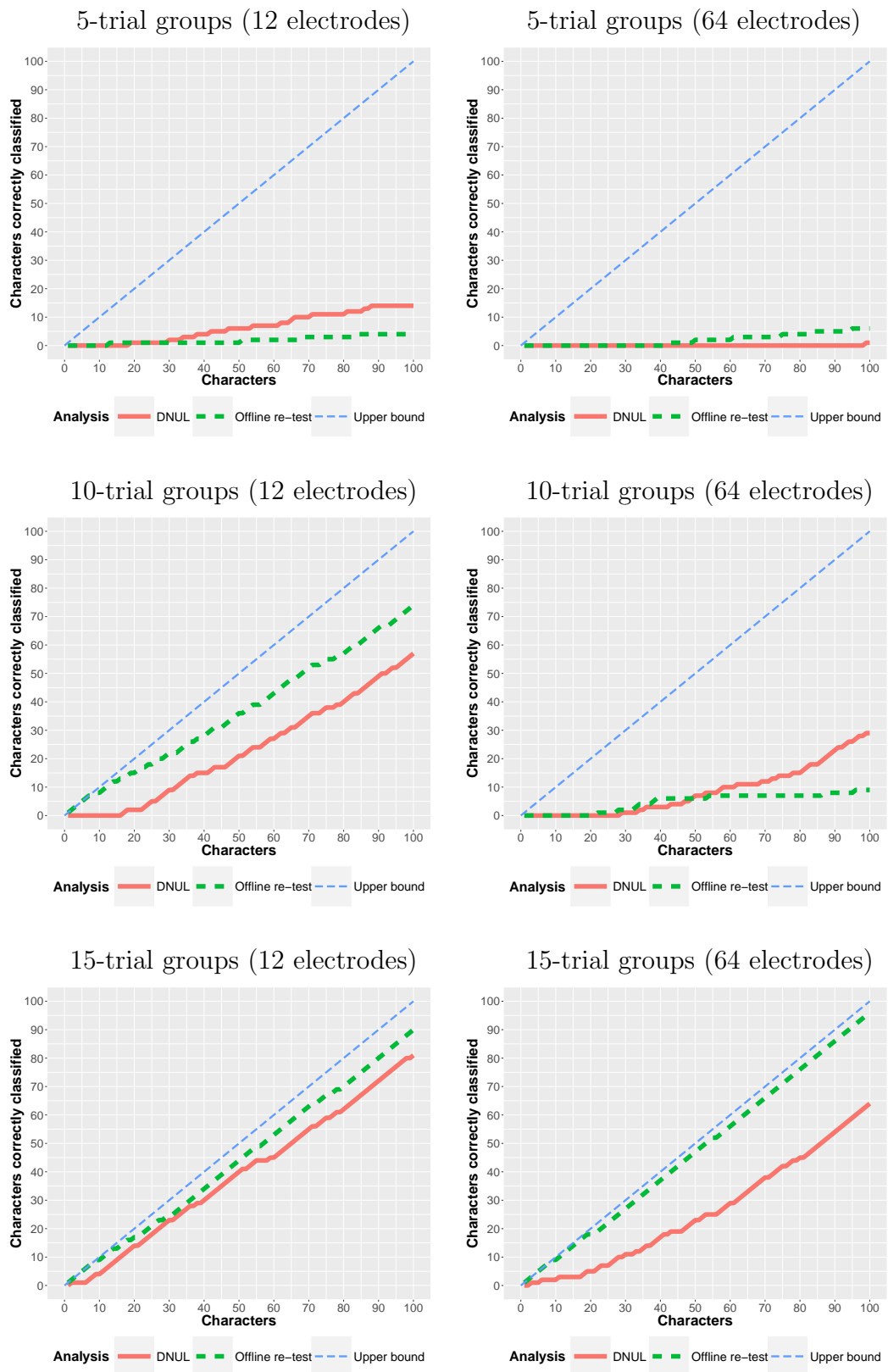


Figure 4.50: Simulated online spelling using 12 electrodes and 64 electrodes for subject B.

BCI Competition II Dataset: Subject C (Simulated online analysis)

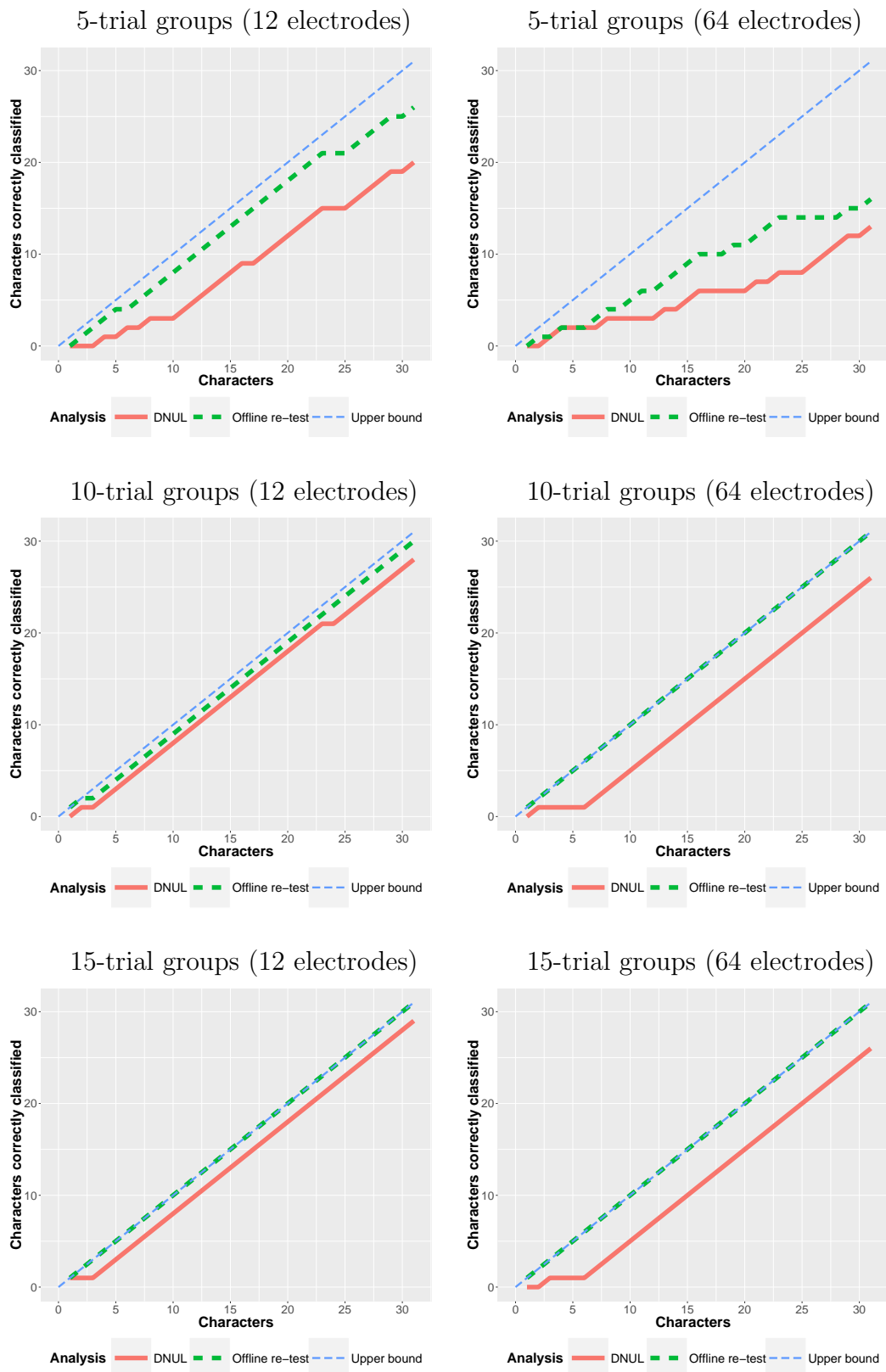


Figure 4.51: Simulated online spelling using 12 electrodes and 64 electrodes for subject C.

4.8 Comparison with the State-of-the-art in Unsupervised Classification in BCIs

In this section, a comparison is performed in order to compare our proposed unsupervised classifier with another unsupervised classifier for a P300-based speller introduced by Kindermans et al. in [11]. It is the only existing method which is able to train a P300 classifier without any labelled data till that time. That motivates us to compare with his method to demonstrate the effectiveness of our method and show the merits and demerits of our proposed method.

The introduced classifier by Kindermans et al. uses the combination of the Expectation Maximization (EM) algorithm for classification and a direct maximization of the likelihood as a criterion to find the best classifier as described in [11] and [9]. This classifier has been used to increase the classification accuracies for P300 spellers by combining inter-subject transfer learning and language models with the classifier [73].

Before proceeding to compare the accuracies of the classifiers, some important points should be introduced for both of them. First of all, both of the classifiers are purely unsupervised classifiers without using any single label. Second, our classifier does not use any clustering or statistical modelling techniques. The competing classifier uses EM algorithm and Bayesian linear analysis. It also uses regularization while ours does not use any regularization. We came up with the regularized version later as described at section 3.4 chapter 3. Third, both of the classifiers start with initial seeds drawn from normally distributed random numbers forming two weight vectors one with \mathbf{w} and one with $-\mathbf{w}$. Finally, the classifiers get detached at the decision making manner. Our proposed method selects the best classifier corresponding to the highest energy function. On the other hand, the competing classifier finds the maximum log likelihood in order to find the best classifier.

For this comparison, different types of experiments have been separated into two categories, offline and online. Details about the experiments will be mentioned briefly for comparison and the detailed experiments can be found in [11]. The aim of these experiments is to evaluate the classifier performance for different configurations, for instance, the amount of unlabelled data, speller adaptivity, and spelling without any prior knowledge.

Experiments

Before presenting and interpreting the results, different configurations we will consider should be mentioned and described. This section follows the experiment section as in [11].

OFF-US: Unsupervised Training on the Test Set

The first experiment uses the entire test data for learning the classifier without using any labels. This procedure was used to enable comparison with existing methods on the same data. This experiment is identical to the batch mode which was described before.

OFF-US-T: Increasing the Amount of Unlabelled Data

This experiment is the same as the first experiment except that, it was designed to evaluate the performance of the unsupervised classifier by increasing the amount of unlabelled data using the training data.

ON-US-T: A Non-adaptive but Unsupervised Online System

In this online experiment, the OFF-US classifier is used which is trained unsupervisedly on the train set and tested on unseen data from the same subject.

OA-US-T: Improving the Online Spelling Trough Adaptation

This experiment re-uses the ON-US-T to initialize the system to increase the performance by adapting the classifier with the new sessions. While evaluating the classifier on the test set, the classifier receives the EEG data for the character that it has to classify. Hence, the the training data grows.

RE-OA-US-T: Evaluation of OA-US-T

A retest evaluation of the OA-US-T classifier is done after the entire test set was processed. These results are not representative for online classification but they show if the classifier has improved or not.

OA-US: The True Challenge, Spelling Without any Prior Knowledge

This is the most challenging experiment. The classifier starts initially untrained. To classify a character we add the EEG data for that character and learn the classifier. After that, the next character comes to increase the training set and learn the classifier. Note, training set does not mean the training session, it refers to the data which the classifier unsupervisedly learns with.

Comparison and discussion

Before we start to compare our unsupervised classifier with the introduced classifier by Kindermans et al. [11] we will give some information about the pre-processing steps and what the differences between the two classifiers are. BCI competition dataset II & III are pre-processed character by character allowing the online simulation in the system. The only difference allocated in the pre-processing step is the downsampling. In our data we reduced the dimensionality by sub-sampling with a factor 8 and 20 samples of each channel was returned. The competing method sub-sampled the data by a factor 6. The comparison applied on two competition datasets as shown in Appendix A. Table 4.8 shows the results of the averaged accuracies for our proposed classifier comparable with the other classifier [11].

Table 4.8: Averaged accuracies of our proposed unsupervised classifier (DNUL). The numbers show the percentage of correctly classified characters. The values in braces are the standard deviation.

	R	OFF-US	OFF-US-T	ON-US-T	OA-US-T	RE-OA-US-T	OA-US
A	5	11.1 (2.9)	16.9 (2.37)	12.3 (2.21)	12.8 (1.55)	11.9 (1.79)	5.3 (3.16)
	10	45.4 (5.29)	52.7 (3.62)	36.6 (2.95)	39.5 (2.32)	44.2 (3.01)	5.2 (4.54)
	15	76.5 (4.88)	85.5 (1.96)	54.4 (2.36)	71.4 (2.37)	74.4 (1.84)	41.2 (16.3)
B	5	32.7 (2.8)	37.4 (3.53)	23.8 (3.35)	28.5 (2.46)	27.9 (3.66)	5.1 (2.37)
	10	81.8 (1.28)	81.1 (3.03)	52.5 (3.74)	61.7 (3.86)	72 (4.22)	4.7 (3.13)
	15	96.5 (0.71)	96.6 (0.69)	63.6 (3.83)	85.7 (2.11)	91.1 (2.38)	48.5 (22.3)
C	5	76.8 (1.03)	75.16 (1.06)	45.81 (1.23)	60.04 (2.29)	66.14 (3.14)	10.32 (15.4)
	10	100 (0.0)	100 (0.0)	86.13 (0.82)	100 (0.0)	100 (0.0)	74.84 (4.24)
	15	100 (0.0)	100 (0.0)	92.90 (0.67)	100 (0.0)	100 (0.0)	83.87 (11.3)

Table 4.9: Averaged accuracies of the competing unsupervised classifier. The numbers show the percentage of correctly classified characters. The values in braces are the standard deviation. Taken from Kindermans et al. [11]

	R	OFF-US	OFF-US-T	ON-US-T	OA-US-T	RE-OA-US-T	OA-US
A	5	46.8 (4.0)	69.0 (0.0)	64.2 (0.0)	66.5 (0.5)	69.0 (0.0)	9.0 (7.4)
	10	89.4 (1.1)	91.0 (0.0)	86.0 (0.0)	87.0 (0.0)	88.0 (0.0)	62.4 (4.1)
	15	95.8 (1.3)	96.0 (0.0)	94.0 (0.0)	96.0 (0.0)	96.0 (0.0)	86.6 (1.6)
B	5	76.3 (1.6)	79.0 (0.0)	75.0 (0.0)	75.0 (0.0)	79.0 (0.0)	53.0 (2.1)
	10	92.1 (1.3)	95.0 (0.0)	91.0 (0.0)	94.0 (0.0)	95.0 (0.0)	87.9 (0.6)
	15	95.2 (0.6)	95.0 (0.0)	92.0 (0.0)	94.0 (0.0)	95.0 (0.0)	87.3 (1.1)
C	5	98.7 (1.7)	96.8 (0.0)	96.8 (0.0)	96.8 (0.0)	96.8 (0.0)	56.5 (5.5)
	10	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	83.5 (1.1)
	15	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	100 (0.0)	92.3 (1.7)

Table 4.10: Accuracies of different supervised classifiers. The numbers show the percentage of correctly classified characters. eSVM, SUP, and OA-SUP are taken from Kindermans et al. [11].

	R	eSVM	SUP	OA-SUP	BLDA	LDA
A	5	72	67	68	51	3
	10	83	88	91	75	2
	15	97	96	95	89	4
B	5	97	84	84	69	0
	10	75	93	93	87	3
	15	96	96	93	95	8
C	5	100	100	100	93.54	0
	10	100	100	100	100	0
	15	100	100	100	100	16.1

Let us start with comparing the OFF-US classifier which is the main method and comparable to the supervised classifiers as well. Although our classifier is too simple, it performs well and get some results comparable with the other classifier. Overall, our classifier has a drawback with limited number of trial groups. We got too low accuracies compared to other classifier. That is because of the non-stationarity of the EEG signals which the classifier got affected. While increasing the number of

trial groups, almost higher than 10 trials, the classifier works well and gives a high accuracy. It can be seen from Table 4.8, with 15 trial groups we have high character accuracies comparable with the competing classifier and other supervised classifiers.

The results for subject A with 5 trial groups is a very poor compared to the other classifier. When we increase the number of trial groups, the averaged accuracy increases till it reaches 76.5%. Subject A is considered as a hard subject for classification compared with subjects B and C. Moving to subject B, we found that with 10 trial groups it shows a good averaged accuracy with 81.8% and still it is somewhat low compared to the other unsupervised classifier. However, with 15 trial groups we recorded an average accuracy of **96.5%** which considered the highest accuracy achieved by our unsupervised classifier compared with the competing classifier which achieved accuracy of 95.2% as shown in Table 4.9 and all other supervised classifiers. The winners of BCI competition achieved an 96% accuracy for subject B with 15 trial groups with eSVM [74] and SUP supervised classifiers [75] as shown in Table 4.10. For subject C, as it can be seen, our approach has achieved an **100%** averaged accuracy with 10 and 15 trail groups but a low accuracy with 5 trial groups. Subject C EEG dataset is considered classifiable where most of the classifiers classify the whole characters successfully.

Overall, for the other experimental configurations, our classifier has a poor accuracy compared to the competing classifier especially with the small number of trial groups. The reason is with fewer trial groups less information can be inferred due to the low signal to noise ratio (SNR). The influence of the amount of data can be shown in the second experiment performance where we are merging the training and testing data. Tables 4.11 to 4.16 show the accuracies for the individual experiments with 10 classifier groups for each.

Table 4.11: **OFF-US**: percentage of correctly classified characters through 10 classifier groups.

		Classifier groups									
		1	2	3	4	5	6	7	8	9	10
A	5	16	9	11	11	14	11	11	5	10	13
	10	53	41	41	48	56	44	44	41	42	44
	15	86	74	73	74	84	77	76	73	71	77
B	5	30	30	30	34	35	34	30	37	31	36
	10	80	80	80	81	82	83	80	83	80	82
	15	95	95	95	95	96	96	95	96	95	96
C	5	74.19	77.42	74.19	77.41	80.64	77.41	80.64	70.96	74.19	80.64
	10	100	100	100	100	100	100	100	100	100	100
	15	100	100	100	100	100	100	100	100	100	100

Table 4.12: **OFF-US-T**: percentage of correctly classified characters through 10 classifier groups.

		Classifier groups									
		1	2	3	4	5	6	7	8	9	10
A	5	21	18	14	14	16	14	18	17	19	18
	10	9	56	55	53	59	46	52	53	51	53
	15	88	88	84	83	87	83	84	86	85	87
B	5	37	31	36	41	38	32	38	40	40	41
	10	83	79	74	83	80	85	83	81	81	82
	15	95	96	94	95	94	97	94	96	95	97
C	5	77.74	70.96	77.41	74.19	70.96	74.19	70.96	80.64	77.41	77.41
	10	100	100	100	100	100	100	100	100	100	100
	15	100	100	100	100	100	100	100	100	100	100

Table 4.13: **ON-US-T**: percentage of correctly classified characters through 10 classifier groups.

		Classifier groups									
		1	2	3	4	5	6	7	8	9	10
A	5	9	13	13	9	15	15	14	11	11	13
	10	39	42	38	34	38	38	36	32	35	34
	15	5	54	58	58	55	51	56	50	55	55
B	5	25	24	27	21	22	21	20	21	30	27
	10	50	46	57	52	49	57	57	51	54	52
	15	61	59	62	71	64	67	68	61	62	61
C	5	45.16	48.38	41.93	38.71	45.16	51.61	48.38	48.38	41.93	48.38
	10	87.09	80.64	83.87	87.09	87.09	87.09	90.32	83.87	87.09	87.09
	15	93.54	93.54	90.32	93.54	90.32	93.54	93.54	93.54	93.54	93.54

Table 4.14: **OA-US-T**: percentage of correctly classified characters through 10 classifier groups.

		Classifier groups									
		1	2	3	4	5	6	7	8	9	10
A	5	12	13	13	13	15	13	13	13	9	14
	10	38	40	41	38	38	40	40	45	37	38
	15	73	74	69	75	71	73	70	72	69	68
B	5	31	27	31	27	30	24	27	28	32	28
	10	60	56	63	59	62	69	67	61	59	61
	15	88	84	83	85	85	88	88	85	83	88
C	5	58.06	58.06	58.06	61.74	58.06	64.52	61.29	58.06	61.29	61.29
	10	100	100	100	100	100	100	100	100	100	100
	15	100	100	100	100	100	100	100	100	100	100

Table 4.15: **RE-OA-US-T**: percentage of correctly classified characters through 10 classifier groups.

		Classifier groups									
		1	2	3	4	5	6	7	8	9	10
A	5	10	14	12	15	13	11	13	10	11	10
	10	46	42	46	46	47	48	42	44	38	43
	15	76	75	74	76	74	75	74	77	72	71
B	5	30	27	34	27	31	28	27	30	26	29
	10	69	66	77	66	74	78	75	73	70	72
	15	92	88	92	92	94	91	89	94	87	92
C	5	64.52	67.74	64.52	67.74	67.74	67.74	61.29	61.29	70.97	67.74
	10	100	100	100	100	100	100	100	100	100	100
	15	100	100	100	100	100	100	100	100	100	100

Table 4.16: **OA-US**: percentage of correctly classified characters through 10 classifier groups.

		Classifier groups									
		1	2	3	4	5	6	7	8	9	10
A	5	3	2	3	6	11	7	7	3	2	9
	10	3	8	0	3	3	4	2	5	16	8
	15	34	54	39	3	51	34	62	40	52	43
B	5	8	4	5	5	4	0	7	4	6	8
	10	5	4	4	6	0	4	10	0	6	8
	15	67	0	30	67	44	46	67	35	70	59
C	5	12.90	0	3.22	12.90	0	51.61	3.22	12.90	3.22	3.22
	10	77.42	70.96	70.96	77.42	77.42	70.97	80.64	77.42	67.75	77.42
	15	90.33	90.33	87.09	74.19	87.09	87.09	90.33	87.09	90.33	54.84

Chapter 5

Conclusion and Future Work

Recently, researchers have been trying to move toward reducing the need of the labelled data and the calibration sessions in order to improve the accuracy and the decoding speed of ERP based on BCI. Although, massive efforts have been working for reducing the usage of labelled data, it was reported as an impossible process until recently. Over the last several years, some researches demonstrated methods and algorithms that have been developed toward the true zero training in different areas in BCI [76].

In this thesis, we presented and evaluated a novel unsupervised classifier based on the disjunctive normal form for P300-based BCI speller systems which allows us to run the classifier without using any calibration process and without any labelled data. In this direction, we proposed the model architecture and how this idea can be embedded into the P300 paradigm. Then, we showed how the proposed classifier can be initialized and uses the initialized parameters for classification. At last, we demonstrated the model optimization in order to learn the classifier weights with the unlabelled data and updating the weights with performing many iterations. Besides, we showed how we can improve the proposed method accuracies by using a regularization term. In order to test the model, we generated simple toy examples to demonstrate the idea and the classifier efficiency. We called our classifier the disjunctive normal unsupervised LDA classifier (DNUL).

To demonstrate our classifier efficiency, many experiments have been performed in order to find the limitations and advantages of our classifier. Moreover, offline analysis experiments including both batch mode and N-batch mode are defined and used in a wide scale in this thesis and simulated online analysis experiments which

demonstrate a classifier in a real mode which involved in different useful experiments. Initially, we illustrated the efficiency of the SU-dataset contains 7 healthy subjects which their temporal EEG signals recorded in the Signal Processing and Information System Laboratory (SPIS). In addition to two datasets from BCI competition II & III. Bit-rates have also been calculated to demonstrate the classifier speed. We also compared our results with two main supervised classifiers used in P300 which are LDA and BLDA.

In general, our results demonstrate the efficiency of our unsupervised classifier (DNUL), although conversely its simplicity as we are not using any statistical models. For the SU-dataset classification with 15 trial groups produces high accuracies and with some subjects we got higher accuracies compared to BLDA and LDA classifiers. The overall accuracy is slightly less than the BLDA classifier in all experiments. We do not forget that BLDA classifier uses labelled data and regularization in term of learning a classifier. On the other hand, in comparison with LDA we exceed the classification accuracies in all cases especially with using many repetitions or trial groups. Simultaneously, with the BCI competition II & III we have 3 subjects as mentioned before. Interestingly, dataset of subject C got 100% accuracy with 10 and 15 trial groups compared with the supervised classifiers and BCI winners. In addition, with subject B, we achieved an average accuracy of 96.5% which considered the highest accuracy achieved by our unsupervised classifier compared with the competing classifier and supervised classifiers where they achieved 95.2% and 96% accuracy respectively with 15 trial groups. In contrary, with subject A we got very poor accuracies with small number of trial groups. Our classifier gives very poor accuracies with small number of trial groups compared to BLDA and mainly the same or a slightly higher accuracy with LDA classifier. As number of trial groups increases, the accuracies of our classifier increases as well. To sum up, we have a promising unsupervised classifier that achieves good accuracies with high number of trial groups despite its simplicity without using any single label.

There are many further directions of this work to improve the classification accuracies. Adding statistical information could be one possible way for improving the model efficiency assuming that ERP features are normally distributed as in

[77]. Following this further, a mechanism for the sensitivity parameter should be developed to set this parameter automatically with different sets of trial groups especially with N-batch mode experiments where the number of trial groups matches the number of trial groups used for testing. Equally important, other factors could affect the model, such as initialization seeds. We can find good initialization seeds for the initial classifier. That will probably increase the classification accuracies and reduce the number of iterations in order to find the best classifier. That could be used as a form of transfer learning, which would transfer information in the form of initializations from other subjects. Additionally, the benefit of the use of a language model in BCI based typing could be one of the directions in terms of increasing the accuracies as in [67] [73]. Another line of future work could be aimed at increasing the efficiency of the online adaptation process by updating the classifier with new test data rather than recomputing it.

Several portions of P300 speller have been studied and developed recently including speller matrix sizes [78], flashing paradigms [79], and inter-stimulus intervals [80] in order to increase the classification accuracy. That may lead to different directions for further work that can be focused on the speller matrix and the flashing paradigms.

Appendix A

Datasets

In this appendix, specific details on the datasets used in this work will be presented.

SU Datasets

- These **datasets** were recorded in our lab at Sabanci University
- **Paradigm:** 6 by 6 matrix speller recorded using BioSemi software ActiView
- **Subjects:** 7
- **EEG recording:** 12 active channels at 2048 Hz
- **Training Set:** 14 characters, 15 trial groups
- **Test Set:** 26 characters, 15 trial groups
- **Inter Stimulus Interval (ISI):** 125 ms
- **Stimulus Duration:** 50 ms
- **Pause between Stimuli:** 75 ms
- **Reference:** C. Ulas et al. (2013) [67], A. Amcalar et al. (2010) [35]

Table A.1: Target words in training and test datasets for SU datasets

Dataset	Spelled characters	Characters
Training	KALEM_YOLCULUK	14
Test 1	KITAP_MASA_AGL	14
Test 2	KITAP_MASA_AGLAMAK_SIKINTI	26

BCI Competition II - (Subject C)

- These **datasets** are provided by Wadsworth Center, NYS Department of Health
- **Paradigm:** 6 by 6 matrix speller recorded using BCI2000
- **Subjects:** 1
- **EEG Recording:** 64 channels at 240 Hz
- **Training Set:** 42 characters, 15 trial groups
- **Testset:** 31 characters, 15 trial groups
- **Inter Stimulus Interval (ISI):** 75 ms
- **Stimulus Duration:** 100 ms
- **Pause between Characters:** 5 ms
- **Reference:** Blankertz et al. (2004) [32]

Table A.2: Target words in training and test datasets for BCI Competition II

Dataset	Target words
Training 1	CAT DOG FISH WATER BOWL
Training 2	HAT HAT GLOVE SHOES FISH RAT
Test	FOOD MOOT HAM PIE CAKE TUNA ZYGOT

BCI Competition III - (Subject A & B)

- These **datasets** are provided by Wadsworth Center, NYS Department of Health
- **Paradigm:** 6 by 6 matrix speller recorded using BCI2000
- **Subjects:** 2
- **EEG Recording:** 64 Channels at 240 Hz
- **Training Set:** 85 characters, 15 trial groups
- **Test Set:** 100 characters, 15 trial groups
- **Inter Stimulus Interval (ISI):** 75 ms
- **Stimulus Duration:** 100 ms
- **Pause between Characters:** 5 ms
- **Reference:** Blankertz et al. (2006) [33]

Appendix B

Sensitivity Parameter (β)

In this appendix, an analysis was done for the sensitivity parameter to show the effect of the beta (β) value on the classifier accuracy for DNUL classifier.

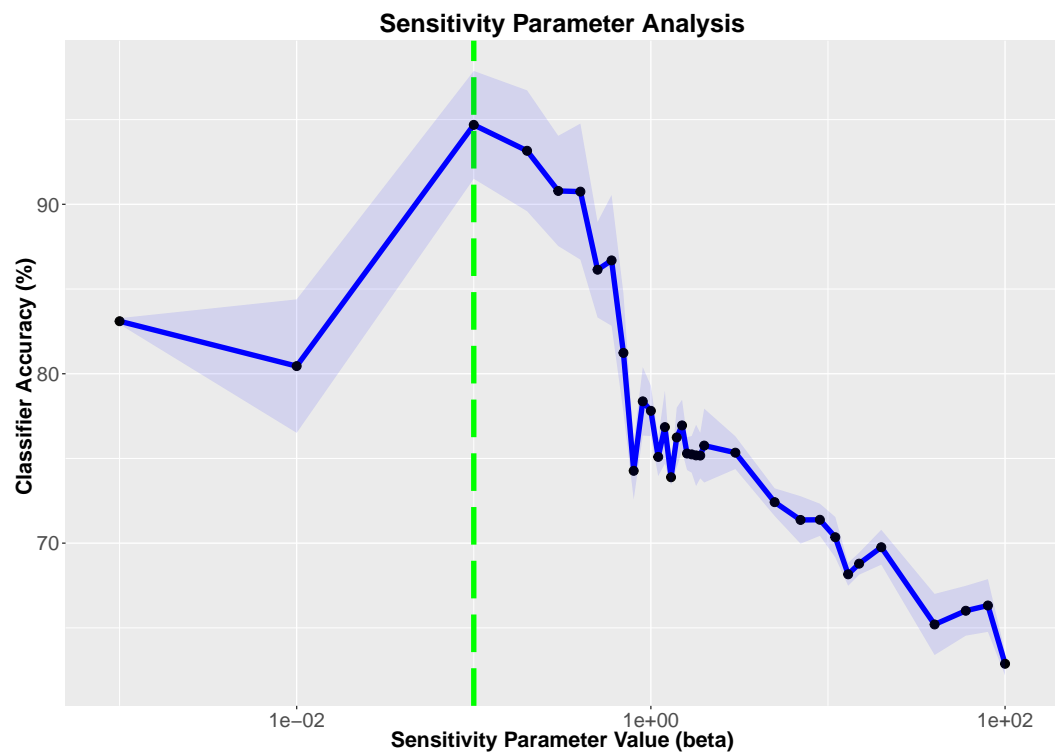


Figure B.1: A sensitivity parameter analysis showing the classifier performance averaged over the 7 subjects. The x-axis represents the beta value (β). The y-axis represents the classifier accuracy. The shaded band shows the standard error from the mean. The vertical green dashed line intercepts the beta value at 0.1 which gives the maximum classifier accuracy among all values.

Table B.1: Sensitivity parameter (β): SU Dataset (S1)

β	Classifier group I		Classifier group II		Classifier group III		Average
	Energy	Accuracy	Energy	Accuracy	Energy	Accuracy	Mean (SD)
0.001	0.3293	83.33	0.3293	83.33	0.3293	83.33	83.33 (0.00)
0.01	0.5385	91.99	0.5611	92.95	0.5382	93.27	92.74 (0.67)
0.1	0.9893	98.08	0.9892	98.08	0.9895	97.44	97.86 (0.37)
0.2	0.9944	96.79	0.9944	96.79	0.9948	98.08	97.22 (0.74)
0.3	0.9953	95.51	0.9932	84.62	0.9962	99.36	93.16 (7.64)
0.4	0.9976	98.08	0.9962	93.59	0.9966	95.51	95.73 (2.25)
0.5	0.9964	87.82	0.9969	94.23	0.9954	88.46	90.17 (3.53)
0.6	0.9964	78.21	0.9955	82.69	0.9958	66.67	75.85 (8.27)
0.7	0.9956	74.36	0.9965	67.95	0.9961	67.95	70.09 (3.70)
0.8	0.9969	69.23	0.9960	70.51	0.9965	75.00	71.58 (3.03)
0.9	0.9970	82.69	0.9977	90.38	0.9970	78.21	83.76 (6.16)
1.0	0.9983	93.59	0.9978	75.64	0.9967	69.87	79.70 (12.4)
1.1	0.9967	72.44	0.9973	77.56	0.9971	69.87	73.29 (3.91)
1.2	0.9971	76.92	0.9969	70.51	0.9972	78.85	75.43 (4.36)
1.3	0.9978	70.51	0.9972	79.49	0.9972	71.15	73.72 (5.00)
1.4	0.9976	74.36	0.9975	71.15	0.9967	76.92	74.15 (2.89)
1.5	0.9975	76.28	0.9965	71.79	0.9979	76.28	74.79 (2.59)
1.6	0.9788	74.04	0.9983	74.36	0.9974	75.64	74.68 (0.84)
1.7	0.9977	68.59	0.9977	78.85	0.9982	78.85	75.43 (5.92)
1.8	0.9979	76.28	0.9979	75.00	0.9987	77.56	76.28 (1.28)
1.9	0.9984	77.56	0.9978	69.87	0.9788	76.60	74.68 (4.19)
2	0.9980	76.28	0.9792	72.76	0.9978	69.87	72.97 (3.21)
3	0.9797	83.33	0.9798	71.15	0.9798	69.55	74.68 (7.53)
5	0.9036	68.91	0.9415	67.31	0.9227	71.79	69.34 (2.27)
7	0.8076	75.32	0.8651	75.00	0.8843	76.92	75.75 (1.03)
9	0.8076	77.24	0.8267	69.23	0.8268	66.35	70.94 (5.64)
11	0.7499	70.19	0.8268	70.19	0.7883	69.87	70.09 (0.18)
13	0.6728	75.00	0.8069	68.27	0.6923	66.35	71.05 (3.51)
15	0.6346	71.47	0.6731	68.91	0.6731	67.95	69.44 (1.82)
20	0.6538	68.27	0.5385	66.35	0.5192	69.55	68.06 (1.61)
40	0.4423	67.63	0.4808	59.62	0.5577	68.27	65.17 (4.82)
60	0.4231	72.12	0.4038	66.99	0.4423	66.67	68.59 (3.05)
80	0.5192	68.91	0.4615	62.18	0.3269	53.85	61.65 (7.54)
100	0.3462	61.22	0.4615	68.91	0.2692	62.50	64.21 (4.12)

Table B.2: Sensitivity parameter (β): SU Dataset (S2)

β	Classifier group I		Classifier group II		Classifier group III		Average
	Energy	Accuracy	Energy	Accuracy	Energy	Accuracy	Mean (SD)
0.001	0.3231	83.33	0.3231	83.33	0.3231	83.33	83.33 (0.00)
0.01	0.4433	89.74	0.4335	89.10	0.4164	85.26	88.03 (2.42)
0.1	0.9830	95.51	0.9848	96.15	0.9828	95.51	95.72 (11.5)
0.2	0.9925	96.15	0.9902	92.95	0.9916	94.87	94.66 (1.61)
0.3	0.9944	92.95	0.9946	94.87	0.9916	83.97	90.60 (5.81)
0.4	0.9949	86.54	0.9936	82.05	0.9930	67.31	78.63 (10.0)
0.5	0.9944	80.13	0.9936	76.92	0.9950	73.08	76.71 (3.53)
0.6	0.9956	86.54	0.9950	85.26	0.9955	76.28	82.69 (5.59)
0.7	0.9976	92.31	0.9957	69.87	0.9960	72.44	78.21 (12.2)
0.8	0.9954	70.51	0.9955	71.79	0.9950	67.95	70.09 (1.95)
0.9	0.9969	71.15	0.9938	73.08	0.9952	68.59	70.94 (2.25)
1.0	0.9956	71.15	0.9965	73.08	0.9965	70.51	71.58 (1.33)
1.1	0.9963	71.79	0.9967	81.41	0.9964	69.23	74.15 (6.42)
1.2	0.9965	69.87	0.9974	71.79	0.9963	71.15	70.94 (0.97)
1.3	0.9967	69.87	0.9961	76.92	0.9977	78.21	75.00 (4.48)
1.4	0.9968	70.51	0.9967	71.15	0.9968	73.08	71.58 (1.34)
1.5	0.9969	69.23	0.9971	78.85	0.9974	73.72	73.93 (4.81)
1.6	0.9962	73.08	0.9979	76.28	0.9974	69.87	73.08 (3.21)
1.7	0.9974	71.15	0.9782	72.76	0.9976	76.92	73.61 (2.97)
1.8	0.9784	80.45	0.9976	80.77	0.9978	69.87	77.03 (6.20)
1.9	0.9977	69.87	0.9976	69.23	0.9963	67.31	68.80 (1.34)
2	0.9979	68.59	0.9978	73.72	0.9964	71.79	71.37 (2.59)
3	0.9786	70.83	0.9407	76.92	0.9797	83.65	77.14 (6.41)
5	0.9610	78.85	0.9801	66.99	0.9411	72.76	72.86 (5.93)
7	0.8459	74.04	0.8649	64.42	0.8844	67.31	68.59 (4.93)
9	0.7883	69.87	0.9228	69.87	0.8076	72.12	70.62 (1.29)
11	0.8461	72.44	0.7884	70.19	0.8653	77.88	73.50 (3.95)
13	0.7115	67.31	0.6923	66.03	0.6346	65.06	66.13 (1.12)
15	0.6730	67.63	0.6538	65.06	0.6731	67.31	66.67 (1.39)
20	0.6538	75.32	0.5192	76.92	0.7692	68.59	73.61 (4.42)
40	0.4808	69.55	0.4808	59.62	0.5385	63.46	64.21 (5.01)
60	0.5192	65.38	0.4615	68.59	0.5385	67.63	67.20 (1.64)
80	0.4808	75.32	0.4808	77.88	0.5385	65.71	72.97 (6.42)
100	0.3846	61.22	0.5385	71.15	0.3846	57.69	63.35 (6.98)

Table B.3: Sensitivity parameter (β): SU Dataset (S3)

β	Classifier group I		Classifier group II		Classifier group III		Average
	Energy	Accuracy	Energy	Accuracy	Energy	Accuracy	Mean (SD)
0.001	0.3240	83.33	0.3240	83.33	0.3240	83.33	83.33 (0.00)
0.01	0.7102	93.59	0.7262	93.27	0.7107	93.27	93.38 (0.18)
0.1	0.9839	94.87	0.9849	96.15	0.9822	94.23	95.09 (0.98)
0.2	0.9920	94.23	0.9930	94.23	0.9933	96.79	95.09 (1.48)
0.3	0.9955	96.79	0.9953	96.79	0.9942	93.59	95.73 (1.85)
0.4	0.9958	92.95	0.9955	89.74	0.9960	94.87	92.52 (2.59)
0.5	0.9965	93.59	0.9961	92.95	0.9962	89.10	91.88 (2.42)
0.6	0.9970	91.03	0.9967	92.31	0.9956	82.69	88.68 (5.22)
0.7	0.9968	91.67	0.9979	94.87	0.9959	89.74	92.09 (2.59)
0.8	0.9955	71.79	0.9971	90.38	0.9960	71.15	77.78 (10.9)
0.9	0.9971	70.51	0.9962	69.87	0.9970	83.97	74.79 (7.96)
1.0	0.9963	83.33	0.9977	87.18	0.9972	73.72	81.41 (6.93)
1.1	0.9969	82.69	0.9961	73.08	0.9969	73.08	76.28 (5.55)
1.2	0.9971	88.46	0.9967	73.08	0.9961	72.44	77.99 (9.07)
1.3	0.9965	71.79	0.9783	79.17	0.9971	72.44	74.47 (4.08)
1.4	0.9966	75.00	0.9783	77.56	0.9969	71.79	74.79 (2.89)
1.5	0.9788	83.65	0.9788	77.88	0.9970	82.69	81.41 (3.09)
1.6	0.9974	70.51	0.9977	87.82	0.9974	73.08	77.14 (9.34)
1.7	0.9782	77.24	0.9789	76.60	0.9782	79.49	77.78 (1.51)
1.8	0.9979	83.33	0.9991	88.46	0.9790	79.49	83.76 (4.50)
1.9	0.9981	85.90	0.9982	83.97	0.9977	71.15	80.34 (8.01)
2	0.9978	69.23	0.9982	69.23	0.9973	71.79	70.09 (1.48)
3	0.9986	78.21	0.9988	67.95	0.9787	74.68	73.61 (5.21)
5	0.9610	72.12	0.9798	72.12	0.9610	73.08	72.44 (0.55)
7	0.9036	77.56	0.8840	66.67	0.9804	75.64	73.29 (5.81)
9	0.7692	75.00	0.7884	68.91	0.8649	80.77	74.89 (5.93)
11	0.7307	79.81	0.8076	70.51	0.8268	70.83	73.72 (5.27)
13	0.5961	70.19	0.8269	65.71	0.7499	66.99	67.63 (2.31)
15	0.6731	69.55	0.6731	64.74	0.7115	72.44	68.91 (3.88)
20	0.6154	62.18	0.5000	66.35	0.6346	71.15	66.56 (4.49)
40	0.5577	70.19	0.5769	61.54	0.4038	63.46	65.06 (4.54)
60	0.5192	67.63	0.5000	76.28	0.4808	65.71	69.87 (5.63)
80	0.3654	69.87	0.5577	73.72	0.5577	68.91	70.83 (2.54)
100	0.3269	57.69	0.4231	69.55	0.3462	54.81	60.68 (7.81)

Table B.4: Sensitivity parameter (β): SU Dataset (S4)

β	Classifier group I		Classifier group II		Classifier group III		Average
	Energy	Accuracy	Energy	Accuracy	Energy	Accuracy	Mean (SD)
0.001	0.3078	83.33	0.3078	83.33	0.3079	83.33	83.33 (0.00)
0.01	0.4769	72.44	0.4767	72.12	0.4929	71.47	72.01 (0.49)
0.1	0.9881	98.08	0.9882	98.08	0.9881	98.08	98.08 (0.00)
0.2	0.9944	98.08	0.9944	98.08	0.9944	98.08	98.08 (0.00)
0.3	0.9964	98.08	0.9962	97.44	0.9953	97.44	97.65 (0.37)
0.4	0.9963	98.08	0.9972	98.08	0.9972	98.08	98.08 (0.00)
0.5	0.9978	97.44	0.9962	94.23	0.9963	95.51	95.73 (1.61)
0.6	0.9970	96.15	0.9966	95.51	0.9972	96.79	96.15 (0.64)
0.7	0.9954	67.31	0.9957	86.54	0.9966	96.15	83.33 (14.7)
0.8	0.9986	98.72	0.9952	72.44	0.9967	76.92	82.69 (14.1)
0.9	0.9980	95.51	0.9966	68.59	0.9977	91.03	85.04 (14.4)
1.0	0.9970	70.51	0.9961	69.87	0.9965	85.90	75.43 (9.07)
1.1	0.9958	69.87	0.9962	78.21	0.9970	76.28	74.79 (4.36)
1.2	0.9963	75.64	0.9974	90.38	0.9961	74.36	80.13 (8.91)
1.3	0.9972	71.15	0.9967	73.72	0.9968	81.41	75.43 (5.34)
1.4	0.9973	74.36	0.9982	89.74	0.9980	85.90	83.33 (8.01)
1.5	0.9976	69.87	0.9790	91.99	0.9971	67.31	76.39 (13.6)
1.6	0.9985	91.67	0.9978	72.44	0.9977	68.59	77.56 (12.4)
1.7	0.9974	76.28	0.9971	76.92	0.9970	78.21	77.14 (0.98)
1.8	0.9978	69.23	0.9978	73.08	0.9967	71.15	71.15 (1.92)
1.9	0.9973	70.51	0.9980	82.69	0.9985	68.59	73.93 (7.64)
2	0.9983	87.18	0.9983	88.46	0.9968	71.15	82.26 (9.64)
3	0.9795	77.24	0.9981	81.41	0.9606	77.88	78.85 (2.24)
5	0.9419	71.15	0.9795	74.04	0.9596	73.08	72.76 (1.47)
7	0.8458	67.63	0.8841	70.19	0.9032	66.35	68.06 (1.96)
9	0.7306	70.19	0.8075	66.67	0.8460	70.83	69.23 (2.24)
11	0.7692	67.95	0.7115	61.54	0.7307	68.91	66.13 (4.01)
13	0.5768	63.46	0.6923	75.96	0.5961	66.03	68.48 (6.60)
15	0.5769	65.71	0.7307	75.32	0.7883	68.27	69.76 (4.98)
20	0.3846	69.87	0.5385	78.21	0.7115	69.55	72.54 (4.90)
40	0.4808	61.54	0.3654	48.72	0.6346	64.10	58.12 (8.24)
60	0.4231	60.26	0.3654	58.97	0.3269	58.33	59.19 (0.97)
80	0.3269	63.14	0.2885	58.97	0.4038	66.35	62.82 (3.69)
100	0.1923	42.31	0.5962	82.37	0.2308	55.45	60.04 (20.4)

Table B.5: Sensitivity parameter (β): SU Dataset (S5)

β	Classifier group I		Classifier group II		Classifier group III		Average
	Energy	Accuracy	Energy	Accuracy	Energy	Accuracy	Mean (SD)
0.001	0.3121	83.01	0.3122	83.01	0.3121	83.01	83.01 (0.00)
0.01	0.4139	73.08	0.4232	74.04	0.4210	73.72	73.61 (0.49)
0.1	0.3121	100	0.9895	100	0.9898	100	100 (0.00)
0.2	0.9899	99.36	0.9891	85.26	0.9952	100	94.87 (8.33)
0.3	0.9947	75.00	0.9968	100	0.9952	96.79	90.60 (13.6)
0.4	0.9932	98.72	0.9962	96.15	0.9963	98.08	97.65 (1.33)
0.5	0.9970	75.64	0.9962	94.87	0.9956	84.62	85.04 (9.62)
0.6	0.9946	98.72	0.9958	82.05	0.9975	96.79	92.52 (9.11)
0.7	0.9980	96.15	0.9972	94.87	0.9969	89.74	93.53 (3.39)
0.8	0.9980	71.15	0.9961	67.95	0.9968	67.31	72.01 (4.54)
0.9	0.9965	80.77	0.9976	92.31	0.9960	74.36	82.48 (9.09)
1.0	0.9963	75.64	0.9963	77.56	0.9970	81.41	78.21 (2.93)
1.1	0.9970	72.44	0.9963	79.62	0.9969	72.44	73.93 (2.59)
1.2	0.9968	73.08	0.9974	80.77	0.9779	66.03	73.29 (7.37)
1.3	0.9973	73.08	0.9974	72.44	0.9875	71.15	72.22 (0.97)
1.4	0.9974	76.28	0.9966	69.87	0.9983	88.46	78.21 (9.44)
1.5	0.9972	83.97	0.9984	91.67	0.9971	75.00	83.55 (8.34)
1.6	0.9979	69.87	0.9972	73.72	0.9975	79.49	74.36 (4.83)
1.7	0.9973	67.95	0.9973	73.08	0.9972	67.95	69.66 (2.96)
1.8	0.9966	91.03	0.9976	69.87	0.9972	69.23	76.71 (12.4)
1.9	0.9984	70.51	0.9790	74.04	0.9979	82.05	75.53 (5.91)
2	0.9978	78.21	0.9788	79.81	0.9975	76.28	78.10 (1.76)
3	0.9977	69.87	0.9792	75.32	0.9987	78.85	74.68 (4.52)
5	0.9989	71.47	0.9608	66.35	0.9609	76.60	71.47 (5.12)
7	0.9600	75.00	0.8074	66.35	0.8313	76.60	72.65 (5.51)
9	0.8460	65.06	0.7884	67.63	0.9422	75.96	69.55 (5.69)
11	0.7115	66.35	0.7691	72.44	0.6537	72.44	70.41 (3.51)
13	0.7882	65.06	0.7307	66.35	0.7115	66.35	65.92 (0.74)
15	0.6346	71.47	0.6731	61.22	0.5768	68.59	67.09 (5.29)
20	0.5769	63.78	0.5962	66.35	0.6153	73.08	67.74 (4.88)
40	0.6923	63.46	0.4423	61.54	0.3077	62.18	62.39 (0.97)
60	0.4038	65.06	0.5000	65.06	0.3462	55.77	61.97 (5.36)
80	0.3846	61.86	0.4231	74.68	0.3462	59.94	65.49 (8.01)
100	0.4038	69.55	0.3846	60.90	0.3462	64.74	65.06 (4.33)

Table B.6: Sensitivity parameter (β): SU Dataset (S6)

β	Classifier group I		Classifier group II		Classifier group III		Average
	Energy	Accuracy	Energy	Accuracy	Energy	Accuracy	Mean (SD)
0.001	0.3104	82.05	0.3104	82.05	0.3104	82.05	82.05 (0.00)
0.01	0.4534	73.40	0.4474	73.08	0.4548	73.40	73.29 (0.18)
0.1	0.9691	73.72	0.9703	76.28	0.9744	78.21	76.07 (2.25)
0.2	0.9869	74.36	0.9887	72.44	0.9860	69.87	72.22 (2.25)
0.3	0.9924	69.87	0.9913	71.79	0.9919	75.00	72.22 (2.59)
0.4	0.9931	71.79	0.9926	71.79	0.9940	75.00	72.86 (1.85)
0.5	0.9945	76.92	0.9940	80.77	0.9944	70.51	76.07 (5.18)
0.6	0.9944	72.44	0.9952	69.87	0.9948	73.72	72.01 (1.95)
0.7	0.9958	73.72	0.9965	68.59	0.9963	69.87	70.73 (2.66)
0.8	0.9953	74.36	0.9960	72.44	0.9959	76.28	74.36 (1.92)
0.9	0.9961	75.64	0.9778	82.37	0.9959	69.23	75.75 (6.57)
1.0	0.9969	80.77	0.9971	73.08	0.9961	72.44	75.43 (4.63)
1.1	0.9968	74.36	0.9969	69.23	0.9965	72.44	72.01 (2.59)
1.2	0.9968	78.85	0.9782	68.27	0.9958	70.51	72.54 (5.57)
1.3	0.9972	70.51	0.9974	71.79	0.9973	75.00	72.44 (2.31)
1.4	0.9969	71.15	0.9898	71.15	0.9972	70.51	70.94 (0.37)
1.5	0.9973	73.08	0.9965	71.79	0.9967	73.08	72.65 (0.74)
1.6	0.9970	73.72	0.9970	69.23	0.9973	71.79	71.58 (2.25)
1.7	0.9975	81.41	0.9974	75.00	0.9975	76.28	77.56 (3.39)
1.8	0.9788	70.51	0.9980	72.44	0.9976	69.23	70.73 (1.16)
1.9	0.9784	75.96	0.9975	78.21	0.9982	71.15	75.11 (3.60)
2	0.9788	69.55	0.9982	69.23	0.9971	75.00	71.26 (3.24)
3	0.9790	72.76	0.9600	71.15	0.9601	69.87	71.26 (1.44)
5	0.8650	77.24	0.9034	77.88	0.9597	74.36	76.50 (1.87)
7	0.8844	68.59	0.8459	66.67	0.7306	63.46	66.24 (2.59)
9	0.7499	67.95	0.7500	66.03	0.7306	74.36	69.44 (4.36)
11	0.8076	67.63	0.7884	66.99	0.7884	64.10	66.24 (1.87)
13	0.6346	63.78	0.7500	66.99	0.6731	75.96	68.91 (6.31)
15	0.6538	68.91	0.7691	66.67	0.6923	68.27	67.95 (1.15)
20	0.6154	69.87	0.5577	70.83	0.5769	64.42	68.38 (3.45)
40	0.5961	65.06	0.4808	67.31	0.4808	71.15	67.84 (3.07)
60	0.4615	65.71	0.3846	61.86	0.5000	75.64	67.74 (7.11)
80	0.4808	65.71	0.3654	68.91	0.5000	61.86	65.28 (3.53)
100	0.3269	61.22	0.5769	66.67	0.3846	61.22	63.03 (3.14)

Table B.7: Sensitivity parameter (β): SU Dataset (S7)

β	Classifier group I		Classifier group II		Classifier group III		Average
	Energy	Accuracy	Energy	Accuracy	Energy	Accuracy	Mean (SD)
0.001	0.3321	83.33	0.3321	83.33	0.3321	83.33	83.33 (0.00)
0.01	0.6132	70.19	0.6221	70.19	0.6127	69.87	70.09 (0.18)
0.1	0.9903	100	0.9902	100	0.9899	100	100 (0.00)
0.2	0.9953	100	0.9953	100	0.9953	100	100 (0.00)
0.3	0.9969	100	0.9969	100	0.9957	98.72	95.57 (0.74)
0.4	0.9977	100	0.9975	99.36	0.9976	100	99.79 (0.37)
0.5	0.9982	100	0.9909	76.92	0.9921	85.26	87.39 (11.7)
0.6	0.9978	98.72	0.9979	98.72	0.9985	99.36	98.93 (0.37)
0.7	0.9949	87.18	0.9958	84.62	0.9941	69.87	80.56 (9.34)
0.8	0.9938	76.28	0.9914	70.51	0.9961	67.31	71.37 (4.55)
0.9	0.9953	78.21	0.9947	76.92	0.9958	72.44	75.85 (3.03)
1.0	0.9953	74.36	0.9988	98.72	0.9958	75.64	82.91 (13.7)
1.1	0.9953	77.56	0.9971	85.26	0.9955	80.77	81.20 (3.87)
1.2	0.9778	86.22	0.9783	85.58	0.9974	91.03	87.61 (2.98)
1.3	0.9954	74.36	0.9960	74.36	0.9964	73.08	73.93 (0.74)
1.4	0.9982	94.87	0.9780	68.27	0.9961	78.85	80.66 (13.4)
1.5	0.9781	75.96	0.9957	78.85	0.9958	73.08	75.96 (2.89)
1.6	0.9764	72.44	0.9966	74.36	0.9972	89.10	78.63 (9.12)
1.7	0.9787	81.41	0.9789	75.96	0.9963	69.23	75.53 (6.10)
1.8	0.9971	71.79	0.9782	70.19	0.9977	69.87	70.62 (1.03)
1.9	0.9768	73.40	0.9598	80.77	0.9787	79.17	77.78 (3.88)
2	0.9791	89.42	0.9782	73.72	0.9970	89.74	84.29 (9.16)
3	0.9771	82.69	0.9790	74.04	0.9775	74.68	77.14 (4.82)
5	0.9601	71.79	0.9416	66.99	0.9419	75.96	71.58 (4.49)
7	0.9226	79.49	0.9225	73.08	0.9228	72.44	75.00 (3.90)
9	0.8651	65.38	0.8843	73.40	0.8652	86.22	75.00 (10.5)
11	0.8268	77.88	0.7883	66.03	0.8845	73.08	72.33 (5.96)
13	0.7498	60.90	0.7884	73.72	0.7307	72.44	69.02 (7.06)
15	0.8076	75.32	0.6730	66.35	0.7692	73.04	71.69 (4.66)
20	0.7115	77.24	0.7500	65.71	0.7308	71.47	71.47 (5.77)
40	0.5385	73.40	0.5000	74.36	0.5577	73.08	73.61 (0.67)
60	0.4615	73.08	0.5000	70.51	0.4038	58.97	67.52 (7.52)
80	0.5000	74.04	0.3077	58.33	0.4231	63.14	65.17 (8.05)
100	0.3654	58.68	0.3462	64.42	0.3462	68.27	63.78 (4.83)

Bibliography

- [1] B. Graimann, B. Z. Allison, and G. Pfurtscheller, *Brain-computer interfaces: Revolutionizing human-computer interaction*. Springer Science & Business Media, 2010.
- [2] “Biosemi headcaps, <http://www.biosemi.com/headcap.htm>.”
- [3] “Biosemi active electrodes, http://www.biosemi.com/pin_electrode.htm.”
- [4] “A commonly used typical brain-computer interface system diagram, <http://www.dewalldlab.com/research/bci.htm>.”
- [5] “Biosemi active electrodes, http://www.biosemi.com/active_electrode.htm.”
- [6] P.-J. Kindermans, M. Schreuder, B. Schrauwen, K.-R. Müller, and M. Tangermann, “True zero-training brain-computer interfacing—an online study,” *PloS one*, vol. 9, no. 7, p. e102504, 2014.
- [7] “Bci2000 speller.” [Online]. Available: http://www.bci2000.org/wiki/index.php/User_Reference:P3SpellerTask
- [8] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, “A review of classification algorithms for eeg-based brain-computer interfaces,” *Journal of neural engineering*, vol. 4, no. 2, p. R1, 2007.
- [9] P.-J. Kindermans, “A bayesian machine learning framework for true zero-training brain-computer interfaces,” Ph.D. dissertation, Ghent University, 2014.
- [10] H. Grkk and A. Nijholt, “Brain computer interfaces for multimodal interaction: A survey and principles,” *International Journal of Human-*

- Computer Interaction*, vol. 28, no. 5, pp. 292–307, 2012. [Online]. Available: <http://dx.doi.org/10.1080/10447318.2011.582022>
- [11] P.-J. Kindermans, D. Verstraeten, and B. Schrauwen, “A bayesian model for exploiting application constraints to enable unsupervised training of a p300-based bci,” *PLoS ONE*, vol. 7, no. 4, pp. 1–12, 04 2012. [Online]. Available: <http://dx.doi.org/10.1371/journal.pone.0033758>
- [12] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, “An efficient p300-based brain–computer interface for disabled subjects,” *Journal of Neuroscience methods*, vol. 167, no. 1, pp. 115–125, 2008.
- [13] F. Nijboer, E. Sellers, J. Mellinger, M. Jordan, T. Matuz, A. Furdea, S. Halder, U. Mochty, D. Krusienski, T. Vaughan *et al.*, “A p300-based brain–computer interface for people with amyotrophic lateral sclerosis,” *Clinical neurophysiology*, vol. 119, no. 8, pp. 1909–1916, 2008.
- [14] J.-J. Vidal, “Toward direct brain-computer communication,” *Annual review of Biophysics and Bioengineering*, vol. 2, no. 1, pp. 157–180, 1973.
- [15] C. Guger, S. Daban, E. Sellers, C. Holzner, G. Krausz, R. Carabalona, F. Gramatica, and G. Edlinger, “How many people are able to control a p300-based brain–computer interface (bci)?” *Neuroscience letters*, vol. 462, no. 1, pp. 94–98, 2009.
- [16] D. Cohen, “Magnetoencephalography: detection of the brains electrical activity with a superconducting magnetometer,” *Science*, vol. 175, no. 4022, pp. 664–666, 1972.
- [17] Z. T. Harmany, R. F. Marcia, and R. M. Willett, “Spiral out of convexity: Sparsity-regularized algorithms for photon-limited imaging,” in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2010, pp. 75 330R–75 330R.
- [18] R. Sitaram, A. Caria, R. Veit, T. Gaber, G. Rota, A. Kuebler, and N. Birbaumer, “Fmri brain-computer interface: a tool for neuroscientific research and treatment,” *Computational intelligence and neuroscience*, vol. 2007, 2007.

- [19] S. M. Coyle, T. E. Ward, and C. M. Markham, “Brain–computer interface using a simplified functional near-infrared spectroscopy system,” *Journal of neural engineering*, vol. 4, no. 3, p. 219, 2007.
- [20] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, “Braincomputer interfaces for communication and control,” *Clinical Neurophysiology*, vol. 113, no. 6, pp. 767 – 791, 2002.
- [21] L. A. Farwell and E. Donchin, “Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials,” *Electroencephalography and clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.
- [22] R. T. Lauer, P. H. Peckham, K. L. Kilgore, and W. J. Heetderks, “Applications of cortical signals to neuroprosthetic control: a critical review.” *IEEE transactions on rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 8, no. 2, pp. 205–208, 2000.
- [23] J. R. Wolpaw and D. J. McFarland, “Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 51, pp. 17 849–17 854, 2004.
- [24] T. W. Picton, “The p300 wave of the human event-related potential.” *Journal of clinical neurophysiology*, vol. 9, no. 4, pp. 456–479, 1992.
- [25] U. Orhan, K. Hild, D. Erdogmus, B. Roark, B. Oken, and M. Fried-Oken, “Rsvp keyboard: An eeg based typing interface,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 645–648.
- [26] U. Orhan, D. Erdogmus, B. Roark, B. Oken, S. Purwar, K. Hild, A. Fowler, and M. Fried-Oken, “Improved accuracy using recursive bayesian estimation based language model fusion in erp-based bci typing systems,” in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, Aug 2012, pp. 2497–2500.

- [27] J. R. Wolpaw and D. J. McFarland, “Multichannel eeg-based brain-computer communication,” *Electroencephalography and clinical Neurophysiology*, vol. 90, no. 6, pp. 444–449, 1994.
- [28] S. Lu, C. Guan, and H. Zhang, “Unsupervised brain computer interface based on intersubject information and online adaptation,” *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 17, no. 2, pp. 135–145, 2009.
- [29] D. J. McFarland, W. A. Sarnacki, and J. R. Wolpaw, “Should the parameters of a bci translation algorithm be continually adapted?” *Journal of neuroscience methods*, vol. 199, no. 1, pp. 103–107, 2011.
- [30] M. Sajjadi, M. Seyedhosseini, and T. Tasdizen, “Disjunctive normal networks,” *CoRR*, vol. abs/1412.8534, 2014.
- [31] M. Elwardy, T. Taşdizen, and M. Çetin, “Disjunctive normal unsupervised lda for p300-based brain-computer interfaces,” in *2016 24th Signal Processing and Communication Application Conference (SIU)*. IEEE, 2016, pp. 2261–2264.
- [32] B. Blankertz, K.-R. Muller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlogl, C. Neuper, G. Pfurtscheller, T. Hinterberger *et al.*, “The bci competition 2003: progress and perspectives in detection and discrimination of eeg single trials,” *IEEE transactions on biomedical engineering*, vol. 51, no. 6, pp. 1044–1051, 2004.
- [33] B. Blankertz, K.-R. Muller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlogl, G. Pfurtscheller, J. R. Millan, M. Schroder, and N. Birbaumer, “The bci competition iii: Validating alternative approaches to actual bci problems,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 14, no. 2, pp. 153–159, 2006.
- [34] S. Sanei and J. A. Chambers, *EEG signal processing*. John Wiley & Sons, 2013.
- [35] A. Amcalar and M. Cetin, “Design, implementation and evaluation of a real-time p300-based brain-computer interface system,” in *Pattern Recognition*

- (ICPR), 2010 20th International Conference on. IEEE, August.2010, pp. 117–120.
- [36] E. Niedermeyer and F. H. L. da Silva, *EEG Recording and Operation of the Apparatus*, 5th ed. Lippincott Williams and Wilkins, 2005.
- [37] C. Ulas, “Incorporation of a language model into a brain computer interface based speller,” Master thesis, Sabanci University, 2013.
- [38] H. Verschore, “A brain-computer interface combined with a language model: the requirements and benefits of a p300 speller,” Master thesis, Universiteit Gent, 2012.
- [39] G. Pfurtscheller, C. Neuper, D. Flotzinger, and M. Pregenzer, “Eeg-based discrimination between imagination of right and left hand movement,” *Electroencephalography and clinical Neurophysiology*, vol. 103, no. 6, pp. 642–651, 1997.
- [40] S. J. Luck, *An introduction to the event-related potential technique*. MIT press, 2014.
- [41] W. G. WALTER, R. COOPER, V. J. ALDRIDGE, W. C. McCALLUM, and A. L. WINTER, “Contingent negative variation : An electric sign of sensori-motor association and expectancy in the human brain,” *Nature*, vol. 203, no. 4943, pp. 380–384, 07 1964. [Online]. Available: <http://dx.doi.org/10.1038/203380a0>
- [42] S. Sutton, M. Braren, J. Zubin, and E. John, “Evoked-potential correlates of stimulus uncertainty,” *Science*, vol. 150, no. 3700, pp. 1187–1188, 1965.
- [43] N. K. Squires, K. C. Squires, and S. A. Hillyard, “Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli in man,” *Electroencephalography and clinical neurophysiology*, vol. 38, no. 4, pp. 387–401, 1975.
- [44] M. Wang, “Design of a modified p300 speller system based on prediction by partial matching language model,” Ph.D. dissertation, University of Cincinnati, 2012.

- [45] B. Blankertz, M. Krauledat, G. Dornhege, J. Williamson, R. Murray-Smith, and K.-R. Müller, *A Note on Brain Actuated Spelling with the Berlin Brain-Computer Interface*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 759–768. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73281-5_83
- [46] K. Hild, U. Orhan, D. Erdogmus, B. Roark, B. Oken, S. Purwar, H. Nezamfar, and M. Fried-Oken, “An erp-based brain-computer interface for text entry using rapid serial visual presentation and language modeling,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*. Association for Computational Linguistics, 2011, pp. 38–43.
- [47] M. Schreuder, T. Rost, and M. Tangermann, “Listen, you are writing! speeding up online spelling with a dynamic auditory bci,” *Frontiers in neuroscience*, vol. 5, p. 112, 2011.
- [48] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, “Bci2000: a general-purpose brain-computer interface (bci) system,” *IEEE Transactions on biomedical engineering*, vol. 51, no. 6, pp. 1034–1043, 2004.
- [49] A. Amcalar, “Design, implementation and evaluation of a real-time p300-based brain-computer interface system,” in *Master thesis*. Sabanci university, Feb.2010.
- [50] I. Yilmaz, “Adaptation in p300 and motor imagery-based bci systems,” Master thesis, Sabanci University, 2015.
- [51] R. Duda, P. E. Hart, and D. G. Stork, “pattern classification, second edition,” *Address: Wiley-Interscience*, 2001.
- [52] K. Fukunaga, “Statistical pattern recognition, seconde edition,” *Academic Press, San Diego, CA*, vol. 1, p. 990, 1990.
- [53] V. Bostanov, “Bci competition 2003-data sets ib and iib: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram,” *IEEE Transactions on Biomedical engineering*, vol. 51, no. 6, pp. 1057–1061, 2004.

- [54] D. J. Krusienski, E. W. Sellers, F. Cabestaing, S. Bayouth, D. J. McFarland, T. M. Vaughan, and J. R. Wolpaw, “A comparison of classification techniques for the p300 speller,” *Journal of neural engineering*, vol. 3, no. 4, p. 299, 2006.
- [55] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [56] A. Selim, M. A. Wahed, and Y. Kadah, “Machine learning methodologies in brain-computer interface systems,” in *2008 Cairo International Biomedical Engineering Conference*. IEEE, 2008, pp. 1–5.
- [57] J. Jin, E. W. Sellers, and X. Wang, “Targeting an efficient target-to-target interval for p300 speller brain–computer interfaces,” *Medical & biological engineering & computing*, vol. 50, no. 3, pp. 289–296, 2012.
- [58] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [59] S. Lloyd, “Least squares quantization in pcm,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [60] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.
- [61] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.
- [62] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.
- [63] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

- [64] S. Borman, “The expectation maximization algorithm—a short tutorial,” *Submitted for publication*, pp. 1–9, 2004.
- [65] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.
- [66] U. Hoffmann, G. Garcia, J.-M. Vesin, and T. Ebrahimi, “Application of the evidence framework to brain-computer interfaces,” in *Engineering in Medicine and Biology Society, 2004. IEMBS’04. 26th Annual International Conference of the IEEE*, vol. 1. IEEE, 2004, pp. 446–449.
- [67] C. Ulas and M. Cetin, “Incorporation of a language model into a brain computer interface based speller through hmms,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1138–1142.
- [68] M. Hazewinkel, *Encyclopaedia of Mathematics: Volume 6*. Springer Science & Business Media, 2013.
- [69] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [70] “Amcalar a, design, implementation and evaluation of a real-time p300-based braincomputer interface system, master thesis, sabanci university, 2010.”
- [71] S. T. Ahi, H. Kambara, and Y. Koike, “A dictionary-driven p300 speller with a modified interface,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 19, no. 1, pp. 6–14, 2011.
- [72] M. J. Alhaddad, “Common average reference (car) improves p300 speller,” *International Journal of Engineering and Technology*, vol. 2, no. 3, p. 21, 2012.
- [73] P.-J. Kindermans, H. Verschore, D. Verstraeten, and B. Schrauwen, “A p300 bci for the masses: Prior information enables instant unsupervised spelling,” in *Advances in Neural Information Processing Systems*, 2012, pp. 710–718.

- [74] A. Rakotomamonjy and V. Guigue, “Bci competition iii: dataset ii-ensemble of svms for bci p300 speller,” *IEEE transactions on biomedical engineering*, vol. 55, no. 3, pp. 1147–1154, 2008.
- [75] P.-J. Kindermans, D. Verstraeten, P. Buteneers, and B. Schrauwen, “How do you like your p300 speller: adaptive, accurate and simple?” in *5th international brain-computer interface conference, Proceedings*, 2011, p. 4.
- [76] M. Krauledat, M. Tangermann, B. Blankertz, and K.-R. Müller, “Towards zero training for brain-computer interfacing,” *PloS one*, vol. 3, no. 8, p. e2967, 2008.
- [77] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K.-R. Müller, “Single-trial analysis and classification of erp components a tutorial,” *NeuroImage*, vol. 56, no. 2, pp. 814–825, 2011.
- [78] B. Z. Allison and J. A. Pineda, “Erps evoked by different matrix sizes: implications for a brain computer interface (bci) system,” *IEEE transactions on neural systems and rehabilitation engineering*, vol. 11, no. 2, pp. 110–113, 2003.
- [79] G. Townsend, B. LaPallo, C. Boulay, D. Krusienski, G. Frye, C. Hauser, N. Schwartz, T. Vaughan, J. Wolpaw, and E. Sellers, “A novel p300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns,” *Clinical Neurophysiology*, vol. 121, no. 7, pp. 1109–1120, 2010.
- [80] D. J. McFarland, W. A. Sarnacki, G. Townsend, T. Vaughan, and J. R. Wolpaw, “The p300-based brain–computer interface (bci): effects of stimulus rate,” *Clinical neurophysiology*, vol. 122, no. 4, pp. 731–737, 2011.