POLITECNICO DI MILANO

Facoltà di Ingegneria dei Sistemi

Master of Science in Biomedical Engineering

ONE-CLASS SUPPORT VECTOR MACHINE FOR OUTLIER DETECTION IN BRAIN-COMPUTER INTERFACE

Master's Thesis of

Guilherme Augusto Zimeo Morais

ID 764134

Supervisor: Prof. Alessandra Laura Giulia Pedrocchi

Co-Supervisor: Wojciech Samek (TU-Berlin)

Academic Year 2011/2012

Table of Contents

Acknowledgments

First of all, I would like to thank Wojciech Samek, who was my tutor at the *Technische Universität Berlin*. From the beginning of our contact, he always gave me full attention for every kind of issue, even when it had nothing to do with my thesis, but rather with my life in Berlin. He was a person always available for me to help me with my questions. We shared the room during my stay in Berlin and it was extremely helpful for the development of the current thesis, since I could ask whenever and whatever I would like to, without hesitating. Moreover, in the very few days when he could not come to the University, I could send him an e-mail that he would answer in the very same day; or even call him through Skype. Along these 4 months of thesis, he gave me a lot of ideas for this project and always helped me with the results interpretation. Sometimes, when I have not said something for a while, he would come to me to ask whether I had some question, or whether I would like to discuss something. Finally, he was also extremely patient in reviewing my thesis. He gave me extremely important hints to improve it, both regarding the English grammar and the thesis content itself. To sum up everything exposed above, Wojciech provided me a supervision better than I could ever have expected from an advisor. I am immensely thankful for that. I am sure that, if someday I supervise someone, I will remember and follow the example he gave me along these months.

I also would like to thank all the people who are somehow responsible to make this project possible:

Prof. Giancarlo Ferrigno, who believed in my potential and first suggested me a thesis topic to work with him; and afterwards recommended me to Prof. Pedrocchi, since her projects seemed to fulfill better my research ambitions.

Dr. Emilia Ambrosini, who received me to discuss the possibilities of thesis topics and explained carefully every possibility. Moreover, she was the one who made the contact with the group in Berlin to ask them whether it would be possible to receive me, since here I had indeed found the topic on Brain-Computer Interface, on which I wanted to work since a few years ago after reading a paper from Miguel Nicolelis.

Dr. Carmen Vidaurre, who first received my application and subsequently indicated Wojciech as my tutor.

Prof. Benjamin Blankertz, who was the responsible for the Berlin Brain-Computer Interface research group regarding the acceptance of my application and gave me full attention when I was insecure about it.

Prof. Alessandra Pedrocchi, who was my supervisor from the Politecnico di Milano, supported my application to Berlin and more recently reviewed my thesis, giving me remarkably helpful hints for it.

Prof. Cinthia Itiki, who is my tutor from the Universidade de São Paulo for the double degree program with the Politecnico di Milano; she always provided me every kind of information that I could need. And, more than a tutor for academic issues, she was a friend for me since I could talk with her about everything and she always gave me great advices. She also reviewed my motivation letter for the group in Berlin, what I think was deciding for my acceptance.

Prof. Luciano Zocchi, who could understand me when I thought I wouldn’t be able to proceed with so many tasks to do at the same time. Indeed, this thesis would not have been completed by this date without his comprehension and attention.

Laura Acqualagna; who, even though was very busy with her PhD activities, helped me with the corrections of the Italian language on the 12-page summary for the current thesis in the very last day for its submission. More than that, she stayed with me literally the whole day (until 11pm) helping me with every single detail.

Finally, I am deeply and eternally thankful for all the people that have made this period easier for me, either by my side or thousands of kilometers away from me.

These people are the friends that I met in Milan, that I met in Berlin, that came to visit me or that I simply met along some trip around the world and, even for a day, could taught me things for the whole life.

But these people are also those who are still in Brazil and, in spite of the distance, didn't seem to be away from me at all.

My friends:

Aline Limieri Dualibe,

Ana Flora Felix de Souza Pontes,

Francisco Spinelli Rodrigues,

Gustavo Hernandes Vieira;

Nanly Katsumoto Ong.

My so beloved family:

my mom, Adelia Ester Maame Zimeo;

my father, Antonio Manuel Pamplona Morais;

my aunt, Antonieta Maame Zimeo;

my grandma, Albertina Maame.

And those who are not materially here anymore: my grandma Alda and my grandpas Fernando and Luca.

I love you all and you were always by my side and in my thoughts. Thanks for being part of my life.

Abstract

Brain-Computer Interface (BCI) represents the communication between the brain and a computer, which may be used to read and interpret the electrical activity of the brain through signal processing and machine learning. Such communication offers a great gamma of possibilities for the medical research, for example the development of neuroprosthetics controlled by motor imagery, wheelchair control and virtual spelling.

The brain's activity can be obtained from invasive – based on surgery for the electrode's placement – or non-invasive methods, e.g. Electroencephalogram (EEG), functional Magnetic Resonance Imaging (fMRI), Magnetoencephalography (MEG) and Near-Infrared Spectroscopy (NIRS).

For the EEG experiments, on which the current work focuses, two important phases of data collection are performed. The first is for the classifier's learning (calibration phase), while the second is carried out in order to evaluate the classification performance during an online test with visual feedback (testing).

The signal processing area has been widely explored and currently presents innumerous combinations of approaches regarding feature extraction through temporal and spatial filters, classification techniques and outliers removal. These techniques are focused on improving the BCI classification performance in order to assure a better reliability of the BCI-based devices.

This report elucidates a novel method for the pre-processing of the EEG event-related (de)synchronization (ERS/ERD) features that results in an increase of the BCI's final classification performance when combined with Common Spatial Patterns (CSP) spatial filters.

Moreover, the current work will also illustrate how this method showed to be efficient for diverse machine signal processing choices, e.g. the number of CSP filters per class and the usage of shrinkage covariance matrix.

Sommario

L’interfaccia cervello-computer (BCI) rappresenta la comunicazione tra il cervello ed un calcolatore, che può essere utilizzato per leggere ed interpretare l’attività elettrica del cervello tramite tecniche di elaborazione di segnali ed apprendimento automatico. Tale communicazione offre un’ampia gamma di possibilità per la ricerca medica; ad esempio, sviluppo delle neuroprotesi controllate dal pensiero, controllo di carrozzelle e spelling virtuale.

L’attività del cervello può essere ottenuta attraverso metodi invasivi, basati sulla chirurgia per il collocamento degli elettrodi; oppure quelli non-invasivi, come ad esempio l’elettroencefalograma (EEG), la risonanza magnetica funzionale (fMRI), il magnetoencefalografia (MEG) e la spettroscopia nel vicino infrarosso (NIRS).

Per gli esperimenti basati su EEG, su cui il presente progetto si basa, vi sono due fasi fondamentali per la registrazione dei dati. La prima è basata sull’apprendimento del classificatore, chiamata fase di calibrazione; mentre la seconda è eseguita per valutare la performance del classificatore durante un test online in presenza di feedback visivo per il soggetto.

Il settore di elaborazione di segnali è stato molto esplorato in ricerca e perciò attualmente presenta diverse combinazioni di possibili approcci riguardo l’estrazione dei features attraverso filtri temporali e spaziali; oltre metodi di classificazioni e per la rimozione di outliers. Tali tecniche si concentrano nell’incremento della performance di classificazione di un sistema BCI per una maggiore affidabilità dei dispositivi.

La presente tesi espone un nuovo metodo per la pre-elaborazione dei EEG-features chiamate (de)sincronizzazioni evento relate (ERS/ERD). L’applicazione di questo metodo, quando associata ai filtri spaziali Common Spatial Patterns (CSP), produce un considerabile incremento della performance finale.

Inoltre sarà presentata anche la sua efficacia al cambiare delle tecniche di apprendimento automatico, ad esempio il numero di filtri utilizzati per il CSP oppure l’uso di matrice di covarianza con restringimento (shrinkage covariance matrix).

List of Figures

List of Tables

Estratto della Tesi

L’interfaccia cervello-computer (BCI) rappresenta la comunicazione tra il cervello ed un calcolatore, che può essere utilizzato per leggere ed interpretare l’attività elettrica del cervello tramite tecniche di elaborazione di segnali ed apprendimento automatico. Tale communicazione offre un’ampia gamma di possibilità per la ricerca medica; ad esempio, sviluppo delle neuroprotesi controllate dal pensiero, controllo di carrozzelle e spelling virtuale.

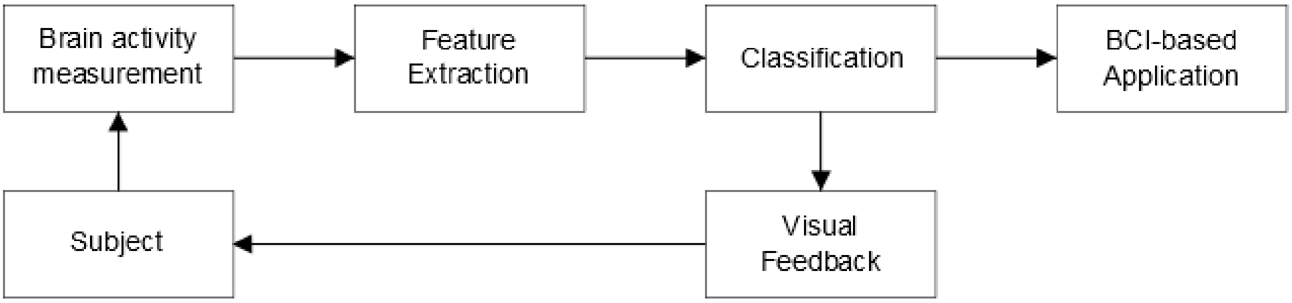


Figura E.1: Schema generale delle fasi del brain-computer interface (BCI). Dopo il blocco di classificazione, il soggetto ha la possibilità di ricevere un feedback visivo (fase online: sezione 1.2) che può facilitare la successiva modulazione dell’attività del cervello verso il risultato desiderato.

Uno dei principali problemi delle BCIs riguarda la misura dei segnali di interesse. L’attività del cervello può essere misurata attraverso metodi invasivi, impiantando chirurgicamente degli elettrodi sulla corteccia celebrale; oppure non-invasivi, come ad esempio l’elettroencefalografia (EEG), la risonanza magnetica funzionale (fMRI), il magnetoencefalografia (MEG) e la spettroscopia nel vicino infrarosso (NIRS).

Nella presente tesi vengono prese in esame le BCIs basate su EEG, per vari motivi. In primo luogo, l'EEG è la tecnologia utilizzata dal gruppo di ricerca presso cui questo lavoro è stato sviluppato (Berlin Brain-Computer Interface – BBCI). Essa é la tecnica di misura per BCI più diffusa, perché non è invasiva e gli esperimenti sono poco costosi rispetto alle altre tecniche. Inoltre, l'EEG ha il vantaggio di avere un’elevata risoluzione temporale, che è importante per un applicazione BCI in tempo reale.

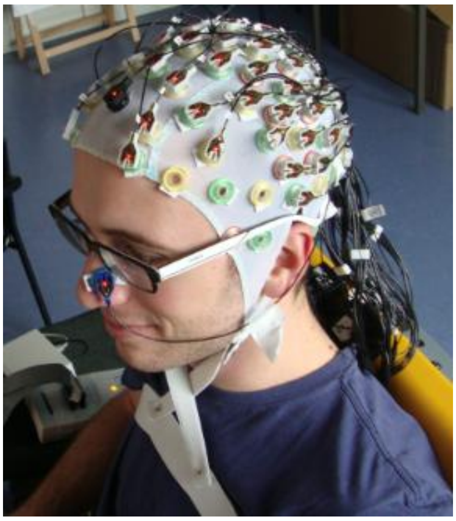


Figura E.2: Cuffia con 62 elettrodi collocati sullo scalpo dell’autore per un esperimento di BCI basato su EEG.

Dopo aver deciso il metodo di acquisizione dell’attività celebrale, si devono determinare le caratteristiche del segnale che rappresentano l’informazione utile per la BCI. Ossia, nel segnale misurato deve esserci un’evidenza dell’attività di interesse. Tale evidenza, chiamata *feature*, può essere suddivisa in tre classi: i potenziali evento-correlati (ERP), i potenziali corticali lenti (SCP) ed i ritmi sensorimotori (SMR).

Nel presente lavoro vengono analizzati i SMR. Essi sono principalmente localizzati nelle bande di frequenze µ (9 – 13Hz) oppure ß (15 – 30Hz). La scelta dipende delle caratteristiche specifiche del soggetto. Una riduzione oppure un incremento dello spettro di potenza corrispondente riflette il compito eseguito dal soggetto. Tale cambiamento nella potenza del segnale è la risposta alla variazione di sincronizzazione delle popolazioni neurali che sono misurate dagli elettrodi del EEG. A seconda di queste risposte, il segnale misurato viene classificato.

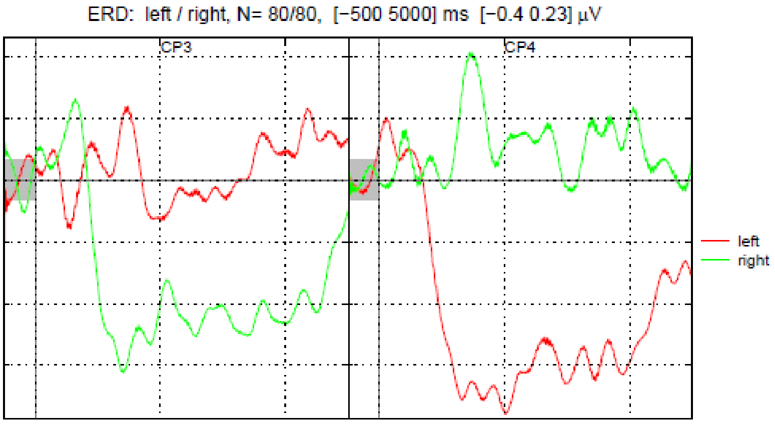


Figura E.3: Desincronizzazioni evento-correlate registrate dall’elettrodo CP3 (sinistra) e CP4 (destra) in funzione del tempo (ms). In corrispondenza dell'immaginazione del movimento della mano destra (verde) e quello della sinistra (sinistra) i due segnali sono chiaramente discriminabili. Figura adattata da [54].

Per le BCIs basate sulle desincronizzazioni evento-correlate, l'immaginazione del movimento di un dato arto corrisponde ad una riduzione dello spettro di potenza nella zona controlaterale della corteccia, mentre lo spettro nella zona ipsilaterale non subisce significanti variazioni . L’algoritmo chiamato Common Spatial Pattern (CSP) è stato utilizzato per l’estrazione di tale caratteristiche. Esso si basa sull’amplificazione delle differenze degli spettri di potenza osservati in ogni zona sensorimotora.

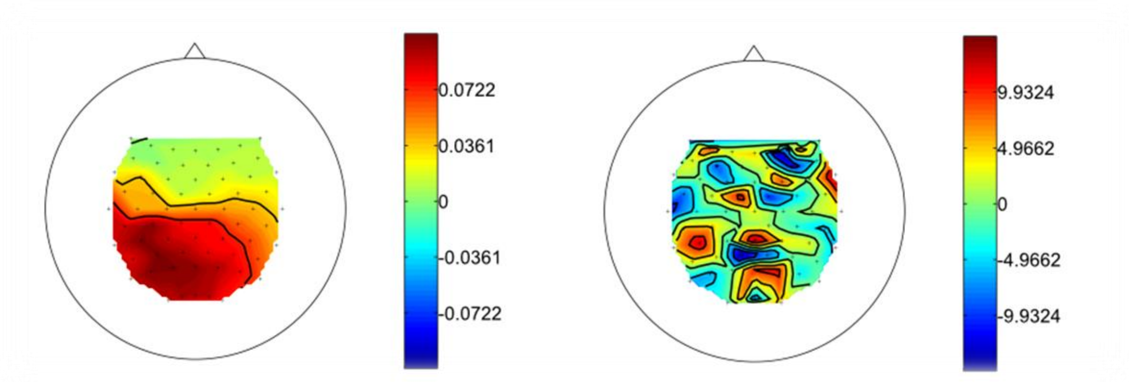


Figura E.4: Distribuzioni sullo scalpo del pattern (sinistra) e del filtro (destra) del CSP corrispondenti all'immaginazione del movimento della mano destra (*right-hand motor imagery*).

Come mostrato in figura E.4, gli output del CSP sono i pattern e i filtri. I pattern rappresentano la distribuzione originale dell’attività celebrale sullo scalpo del soggetto. I filtri, quando moltiplicati per il segnale originale, risultano in un'amplificazione delle differenze tra le varianze osservate in ogni zona sensorimotora. Dunque, l'applicazione del CSP comporta un incremento di discriminabilità tra i compiti eseguiti dal soggetto, migliorando la classificazione finale.

In generale, tali compiti, chiamati “classi”, sono rappresentati da tre tipi di immaginazione: movimento della mano destra, movimento della mano sinistra, movimento dei piedi (entrambi).

Il CSP è efficiente nell'incrementare il rendimento della classificazione soltanto di due classi. Come sarà successivamente descritto, gli esperimenti BCI realizzati scelgono due classi durante la fase di calibrazione perché sia possibile applicare questo metodo.

Una volta che le features sono state estratte, esse vengono separate da un classificatore nelle diverse classi. Nel presente progetto é stata utilizzata la *Linear Discriminant Analysis* - LDA, che è un approccio lineare. Lo spazio n-dimensionale a cui appartengono i campioni viene separato nei sotto-spazi corrispondenti alle varie classi da un iper-piano. La sua direzione e posizione vengono definiti dal vettore normale ***w*** e da una soglia *b* :

$$f(x) = \ sgn(\boldsymbol{w}.\boldsymbol{x} + b)$$

La LDA calcola i parametri (***w**, b*) in modo da massimizzare i margini medi, in cui “margine” è la minima distanza tra un campione e l’iperpiano. Inizialmente si assume che tutti i campioni siano uniformemente distribuiti in forma gaussiana con la stessa covarianza quando proiettati sulla direzione di ***w***.

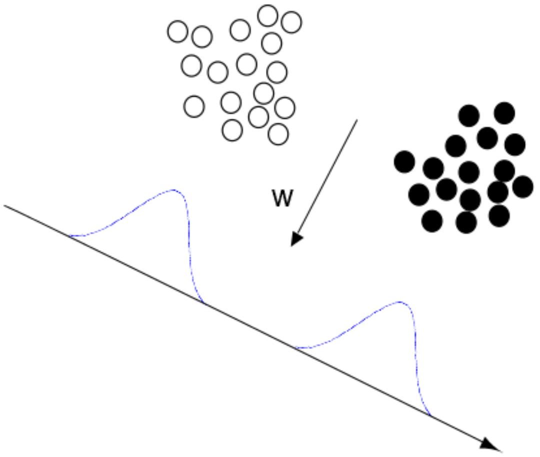


Figura E.5: Proiezione di due classi (bianco e nero) sulla direzione del vettore ottimale ***w***, ossia la direzione in cui le distribuzioni gaussiane hanno media massima e varianza minima; ciò assicura maggiore separabilità tra le classi e, di conseguenza, una migliore performance del classificatore.

L’efficacia della classificazione della LDA è basata sulla discriminazione dei campioni che appartengono ad ogni classe. Essa è fatta stimando la direzione ottimale di proiezione che risulta nella massima media e minima varianza delle gaussiane. Nel caso ideale, la suddivisione delle classi avverrebbe senza difficoltà. Tuttavia, negli esperimenti di BCI, spesso sono presenti degli “*outliers”* nei campioni raccolti. Con il termine outliers vengono definiti quei campioni il cui valore dista chiaramente dalle altre osservazioni disponibili. A seconda della quantità e posizione di tali outliers nello spazio, le caratteristiche del classificatore possono cambiare drasticamente, risultando in una peggiore classificazione finale (figura E.6).

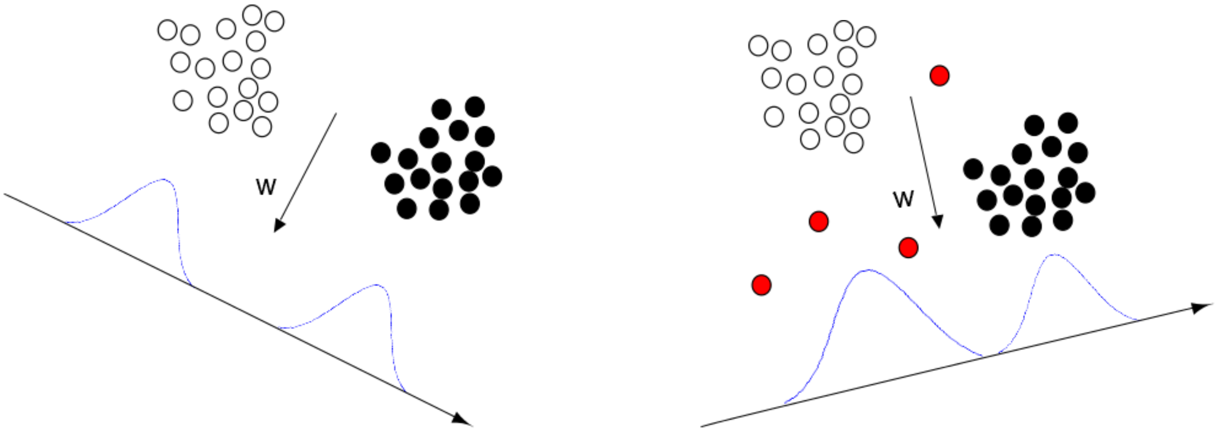


Figura E.6: Influenza della presenza di outliers nei campioni. A sinistra, la distribuzione in assenza di outliers. A destra, quattro outliers (in rosso) sono stati aggiunti. Essi appartengono alla classe rappresentata dai campioni bianchi. Di conseguenza, il vettore ottimale ***w*** che controlla la direzione di proiezione viene cambiato, risultando in un avvicinamento tra le classi e una conseguente riduzione della separabilità tra di loro.

Nei dati presi qui in considerazione, outliers sono quei campioni le cui caratteristiche differiscono dalle reali features dei potenziali ERD/ERS. La loro presenza è tradotta in una notevole riduzione della performance.

Gli outliers, oltre a rappresentare un problema per il classificatore LDA, possono influenzare intensamente il calcolo dei filtri e dei patterns ottimi del CSP. Ciò può avvenire poiché l'algoritmo CSP è basato sula stima delle matrici di covarianza che rappresentano le due classi. L'elevata sensibilitá di tale stima alla presenza di outliers è riportata in letteratura [68].

La presente tesi propone un nuovo metodo per l'individuazione di outliers nei campioni raccolti negli esperimenti di BCI, basati su EEG e SMR. Tale metodo si chiama one-class Support Vector Machine (1-SVM).

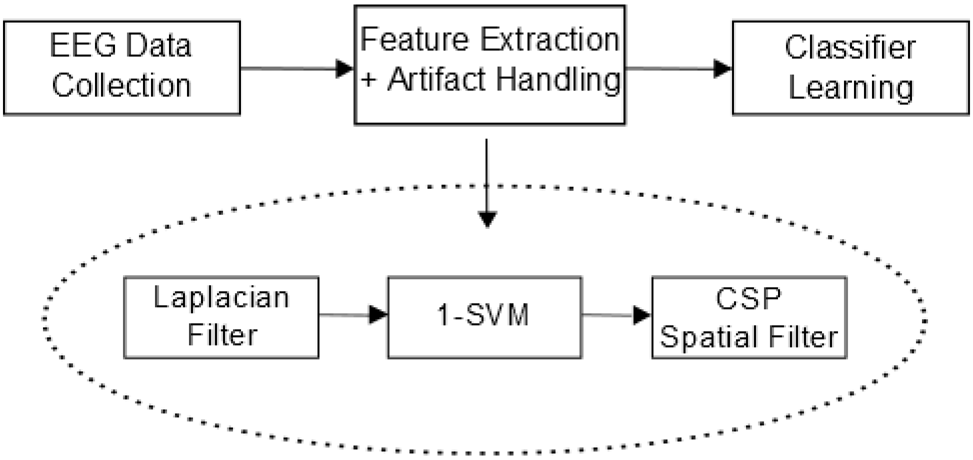


Figura E.7: Schema della fase di calibrazione proposta nella tesi. Dopo che il segnale EEG é stato raccolto, passo che include anche un filtro passa-banda (capitolo 3), vi sono ulteriori tre blocchi di elaborazione per la selezione delle features per l'apprendimento del classificatore: il filtro laplaciano per attenuare il rumore di background nel segnale EEG; 1-SVM per individuare gli outliers (che verrano successivamente rimossi o penalizzati); filtri spaziali CSP per ridurre la dimensione dei dati mentre si amplifica la differenza tra le varianze delle classi.

L’idea iniziale del progetto era di classificare gli outliers del gruppo di calibrazione e successivamente rimuoverli, oppure penalizzarli prima della stima della matrice di covarianza per la generazione dei filtri CSP, che dipendono dei campioni. Infatti, l’apprendimento del classificatore LDA potrebbe essere fortemente influenzato dalla loro presenza.

Per tale proposito, sono stati utilizzati dei dati di quaranta soggetti senza esperienza precedente in esperimenti BCI. Sono stati selezionati soggetti senza precedenti episodi di malattie neurologiche, dato che tale presenza avrebbe potuto interferire nello svolgimento del compito basato sull'immaginazione dei movimenti. La raccolta dei dati comprendeva due fasi: calibrazione (150 campioni) e feedback (300 campioni).

In realtà la fase di calibrazione era inizialmente composta di 225 campioni, 75 per ogni classe possibile (mano destra, mano sinistra, piedi). Tuttavia, essendo il classificatore binario (discriminazione di due classi), vengono considerati i campioni di due classi per volta (150). Alla fine del processo di calibrazione, vengono selezionati i 2 gruppi che presentano un minor errore di classificazione. Tale errore viene stimato tramite cross-validation.

Un altro punto riguarda la raccolta di dati nella fase online (di feedback). Poiché i dati utilizzati erano stati precedentemente raccolti per un altro progetto, nella presente tesi non si è potuto approfittare dei vantaggi del feedback visivo e della conseguente possibilità di modulazione del compito da parte del soggetto.

Sebbene un'analisi offline non sia ideale, essa rappresenta un primo passo per dimostrare l’efficacia del metodo proposto nell’incremento della performance dei BCIs. Inoltre, se esso è in grado di migliorare la classificazione dei dati test (dalla fase “online”), si ipotizza che possa produrre una performance ancora migliore nel caso real-time.

Come accennato precedentemente, il metodo si basa sulla SVM di una classe (1-SVM). Un approccio simile è stato riportato in letteratura per identificare degli intrusi (outliers) in una network [105]. Tuttavia, tale metodo ancora non era stato valutato per BCIs prima di questo progetto.

La 1-SVM crea una frontiera tra i campioni classificati come puri e quelli come outliers. La tipologia di tale contorno, che in questo caso presenta 68 dimensioni (il numero di elettrodi usati per la raccolta di dati durante gli esperimenti), dipende dalla scelta del parametro *nu*, che rappresenta la probabilità di un dato campione di essere un outlier. Esso è definito dall’esperto che esegue l’esperimento BCI.

Una delle valutazioni eseguite in questo progetto é stata quella riguardo il miglior approccio per scegliere il parametro *nu.* Per ogni caso, il suo valore ottimale era conosciuto perché l’analisi eseguita era offline. Ossia, tutti i dati dalla fase di feedback potevano essere usati finché il miglior valore di nu, cioè quello che risultasse nella miglior performance di classificazione, non fosse incontrato. Tuttavia, tale approccio non è riproducibile in un esperimento real-time e perciò non è valido per tale scelta.

Nonostante ciò, conoscere la performance ottimale di ogni soggetto è stato importante come mezzo di paragone. Come si osserva nella tabella E.1, la performance finale media tra i 40 soggetti è stata riportata per diversi metodi di elezione del nu. Questi risultati sono stati confrontati con quelli ottimali per avere un’idea della differenza tra di loro. Si osserva in tale tabella che il metodo di cross-validation, in cui il valore di nu varia in base al soggetto, è risultato migliore rispetto a quelli in cui nu era fisso per tutti i partecipanti (soglia fissa); in ogni caso, sempre lontano dal valore ottimale.

CSP choices	Best nu	CV nu	nu = 0.20	nu = 0.45	nu = 0.70
Filters per class: 1 **Shrinkage: No**	22.442	**24.987**	26.675	27.737	28.667
Filter per class: 1 **Shrinkage: Yes**	23.179	**26.001**	26.775	26.771	27.421
Filter per class: 3 **Shrinkage: Yes**	22.100	25.879	25.375	**25.275**	27.592
Filter per class: 3 **Shrinkage: No**	22.612	**26.308**	26.871	27.137	29.312

Tabella E.1: Errori percentuale medi per 40 soggetti. Il valore ottenuto dai metodi di cross-validation e di soglia fissa è stato confrontato per diverse scelte di nu. La colonna in arancione rappresenta i risultati ottimali e non riproducibili; i valori messi in evidenza in azzurro sono quelli risultanti nella miglior performance, ossia l'errore percentuale minore. Si può concludere che la cross-validation (CV) è stata la miglior alternativa.

La cross-validation è un metodo basato sulla stima della performance usando soltanto il set di calibrazione. È realizzato attraverso la divisione di tale set in cinque subsets. Quattro di loro, quindi 120 campioni, sono forniti alla 1-SVM per ogni valore di nu possibile (sono stati usati 20 valori ugualmente distanti tra 0 e 0.95). I campioni che sono classificati come outliers sono penalizzati e successivamente tutto il set di 120 è fornito per l'algoritmo CSP e poi per l'apprendimento del classificatore LDA.

Infine, i 30 campioni restanti sono forniti al LDA, che esegue la loro classificazione. Essa viene poi confrontata con i veri labels dei 30 campioni. L'errore percentuale rappresenta quanti campioni sono stati erroneamente classificati. Quindi il valore di nu corrispondente al minore errore percentuale sarà scelto per la vera valutazione degli outliers in tutto il set di calibrazione (150 campioni).

Un'altra scelta per la 1-SVM riguarda quali campioni devono essere forniti come input all'algoritmo. Le due alternative valutate in questo lavoro sono: fornire tutto il set di calibrazione (150 campioni) oppure una classe (75 prove) per volta. Nel primo caso, potrebbe presentarsi il problema che un outlier di una data classe possa non essere individuato perché in realtà si trova nel sottospazio della seconda classe (figura E.8).

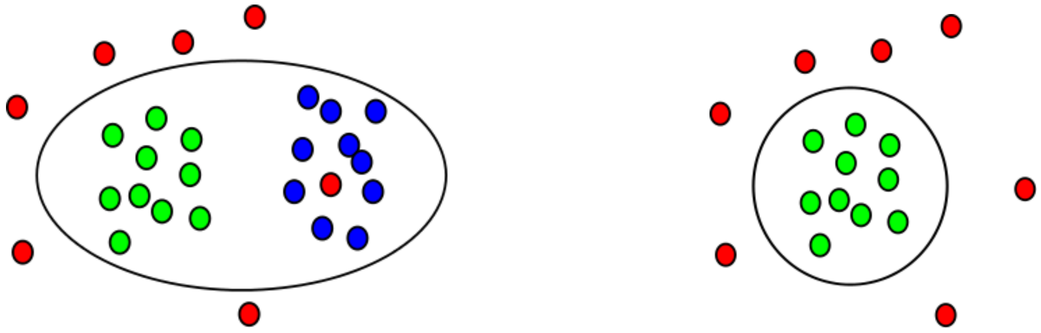


Figura E.8: Confronto delle frontiere create per due (sinistra) oppure una classe (destra) fornite come input alla 1-SVM. I cerchi verdi rappresentano la prima classe, mentre quelli blu la seconda. I cerchi rossi sono gli outliers della prima classe. A sinistra, vi è un outlier che si trova in realtà in mezzo ai campioni della seconda classe e perciò non è stato individuato. Tuttavia, a destra, caso in cui soltanto i campioni della prima classe sono stati forniti, tutti gli outliers sono stati correttamente individuati.

Nonostante, la tabella 5.2 mostra come il caso in cui tutti i due classi sono forniti insieme risulta meglio.

	Both classes (best nu)	One class (best nu)	Both classes (CV nu)	One class (CV nu)
Filters per class: 1 Shrinkage: No	**-5.6583**	-4.8375	**-3.1125**	-2.8042
Filter per class: 1 Shrinkage: Yes	**-5.0333**	-4.5167	-2.6042	**-2.7500**
Filter per class: 3 Shrinkage: No	**-6.0750**	-5.4292	-2.3792	**-2.4958**
Filter per class: 3 Shrinkage: Yes	**-4.5167**	-3.9583	**-1.1375**	-0.6542

Tabella E.2: Media (40 soggetti) della riduzione degli errori percentuali per diverse possibilità di scelta del CSP. Più il valore è negativo, più la performance media è stata migliorata. Le prime due colonne (verde) confrontano le alternative di input per il valore ottimale del parametro nu, mentre le ultime due (arancione) il nu scelto dalla cross-validation (CV). Il miglior valore di riduzione dell'errore osservato per ogni coppia di colonne è presentato in blu.

È stato ipotizzato che i risultati sopra presentati siano conseguenza dell’uso dello stesso valore di nu per entrambe le classi, anche se loro fossero fornite separatamente alla 1-SVM. Il problema è che può esserci il caso in cui una classe abbia molti più outliers che l’altra; così se entrambe sono valutate con la stessa probabilità che un campione sia un outlier (stesso nu), vi saranno sicuramente dei campioni individuati erroneamente.

Tuttavia, la decisione di usare lo stesso *nu* è basata su un’altra ipotesi, cioè la possibilità di un drawback nell’efficacia della cross-validation, dato che il nu eletto difficilmente raggiunge il valore ottimale, ossia il valore che risulta nella migliore performance. Perciò, scegliere due valori diversi di nu da un processo che non è sufficientemente preciso potrebbe risultare in una performance ancora peggio.

Dopo che gli outliers sono stati individuati, si deve decidere come trattarli. Ossia, è possibile rimuoverli completamente oppure penalizzarli con pesi proporzionali alla distanza che loro presentano dal confine creato dalla 1-SVM. L’idea di non rimuoverli completamente è basata sull’ipotesi che loro possano avere delle caratteristiche importanti per l’apprendimento del classificatore LDA.

Per tale penalizzazione, è stata usata una funzione sigmoide (sezione 4.3) con nove diverse configurazioni (Appendix A). Otto sono state create da un metodo statistico, in cui si è scelto la percentuale di campioni che sono stati precedentemente individuati come outliers, che hanno ricevuto dei pesi tra 0.01 e 0.50. Tale approccio è stato utilizzato al fine di controllare meglio lo distribuzione degli outliers lungo la funzione sigmoide rispetto a una percentuale predefinita (attraverso quantili).

Inoltre, sono stati considerati dei casi in cui anche i campioni classificati come non outliers sono stati penalizzati, sempre a seconda della distanza dal confine creato. Tale decisione si basa sulla possibilità di correggere una eventuale classificazione sbagliata. Cioè un outlier che è stato classificato come campione puro verrà così penalizzato perché probabilmente si trova vicino al confine della 1-SVM.

Tutte queste funzioni di penalizzazione degli outliers sono state confrontate con l’alternativa della completa rimozione (funzione #10) nella tabella E.3.

Per la tabella seguente, sono stati usati i metodi sotto (1 – 3) che fanno uso delle variabili presenti in (i – iv):

1) $\sum_{i=1}^{N} \frac{(CV_i - opt_i)}{N}$

2) $\sum_{i=1}^{N} \frac{(opt_i - \min(remov_i, orig_i))}{N}$

3) $\sum_{i=1}^{N} \frac{(CV_i - \min(remov_i, orig_i))}{N}$

i) **CV**: 1-SVM + nu chosen by cross-validation

ii) **opt**: 1-SVM + optimal, theoretical nu

iii) **remov**: removing outliers labeled through a filter method (section 2.3)

iv) **orig**: considering all the trials, without any outlier detection

Function Index	**Evaluation Method 1**	**Evaluation Method 2**	**Evaluation Method 3**
#1	3.3792	-3.0750	0.3042
#2	3.1958	-2.9042	0.2917
#3	3.0875	-2.9917	0.0958
#4	3.2875	-2.7000	0.5875
#5	3.6167	-2.9583	0.6583
#6	2.6042	-2.5708	0.0333
#7	2.9167	-2.9708	-0.0542
#8	2.7417	-2.6667	0.0750
#9	2.9875	-2.9542	0.0333
#10	**3.7792**	**-3.3708**	0.4083

Tabella E.3: Valutazione delle funzioni che trattano gli outliers individuati dalla 1-SVM. Il risultato in blu rappresenta la miglior performance per il nu ottimale (teorico), mentre il risultato in rosso mostra la maggiore distanza tra i risultati ottenuti dalla cross-validation (reale) e quello ottimale (teorico).

Nonostante la funzione che rimuove completamente gli outliers abbia presentato il miglior risultato per il metodo che valuta il potenziale massimale della 1-SVM, ossia rispetto i valori ottimali di nu, essa ha anche presentato il peggior risultato per il metodo che valuta la distanza tra i risultati teorici e quelli reali, basati sull’elezione del parametro nu dalla cross-validation. Ossia, in un esperimento real-time, in cui il nu ottimale non è disponibile, la funzione che rimuove completamente gli outliers risulterebbe in realtà in una dei peggiori performances.

Quindi è stato ipotizzato che la 1-SVM può essere in grado di identificare tutti gli outliers da essere rimossi completamente soltanto in un caso ideale che non è riproducibile in pratica. Però considerando il caso reale della cross-validation per la elezione del nu, un approccio più sicuro sarebbe quello di penalizzare le prove e quindi mantenere le sue informazioni per assicurare una migliore classificazione finale della BCI.

La valutazione finale dell’errore percentuale per ogni soggetto mostrata in figure E.9 considera le scelte della 1-SVM finora esposte: l’applicazione della cross-validation per l’elezione del parametro nu, l’impiego di tutto il set di calibrazione come input e l’uso di una funzione che penalizza i campioni anziché la loro rimozione.

Oltre le scelte riguardo la 1-SVM, si devono decidere anche delle alternative per il CSP, ad esempio il numero di filtri da essere considerati e la presenza o assenza dello shrinkage per la stima della matrice di covarianza. Lo shrinkage comporta una stima più robusta in presenza di outliers [101]. Per valutare il potenziale della 1-SVM rispetto alla scelta che ci si aspetta essere più stabile, è stato quindi impiegato lo shrinkage. Inoltre, riguardo il numero di filtri per classe, ne sono stati scelti 3, perché si ipotizza che si abbia una minore perdita di informazione.

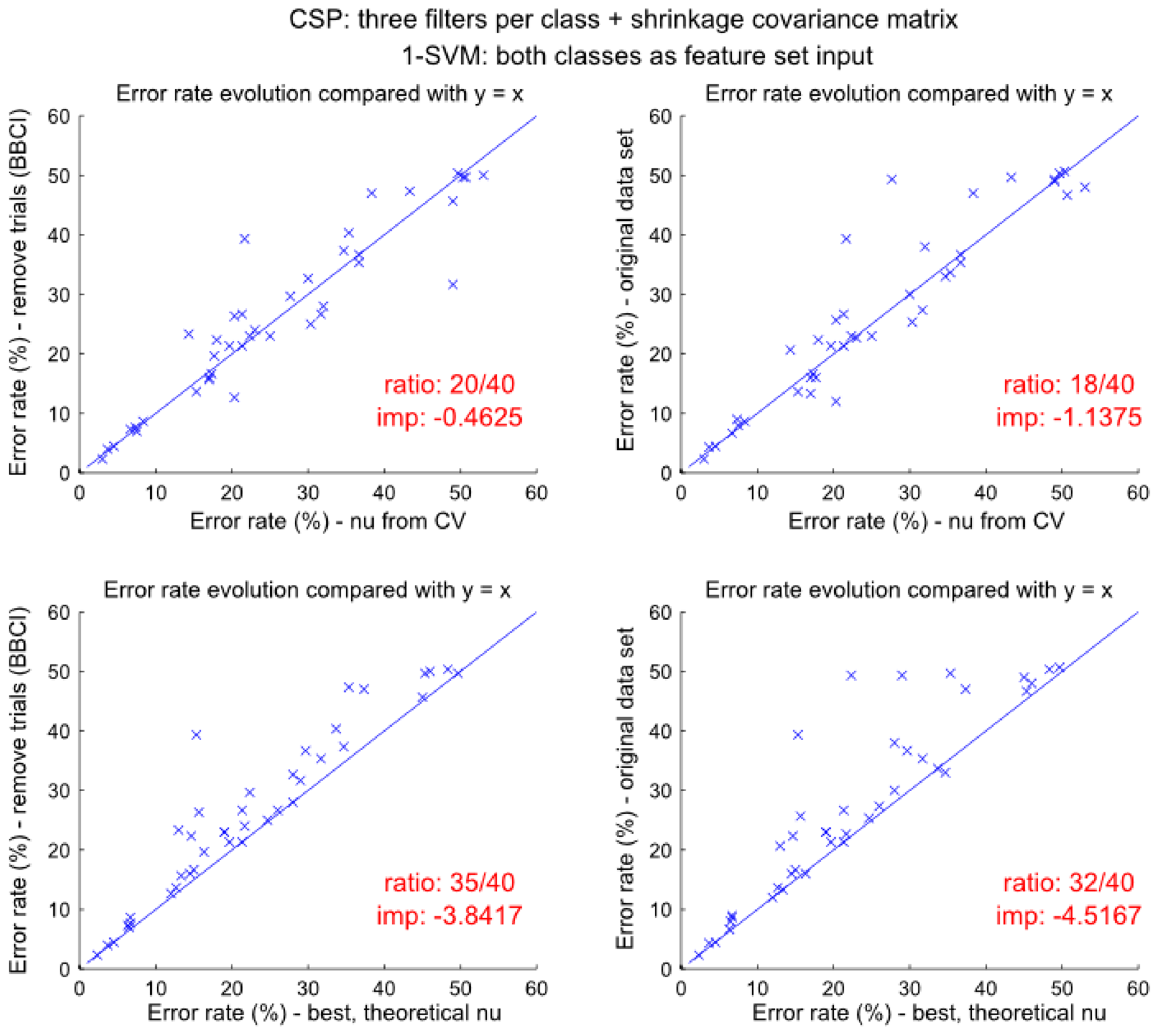


Figura E.9: Confronto dell'errore percentuale tra l'impiego e l'assenza della 1-SVM nel trattamento dei dati. Vengono considerati lo shrinkage, il CSP con tre filtri per classe ed entrambe le classi sono fornite come input alla 1-SVM. Ogni croce rappresenta un soggetto diverso e la sua posizione spaziale segue l'errore percentuale registrato nel metodo di riferimento (y-axis) paragonato con quello del metodo proposto (x-axis). Nei grafici a sinistra, l'asse y rappresenta il metodo di filtraggio per trattare gli outliers presentato nella sezione 2.3. Ogni grafico presenta inoltre una retta y = x che serve come una soglia visuale che illustra quali soggetti abbiano in realtà presentato una riduzione dell'errore percentuale per il metodo proposto (croci localizzate sopra la retta).

I valori in rosso nella figura E.9 rappresentano numericamente il miglioramento osservato nella presenza della 1-SVM quando paragonata con il metodo di riferimento. "Ratio" rappresenta il numero di soggetti che hanno presentato uno netto incremento, mentre "imp" (*improvement*) mostra l'attenuazione risultante nell'errore percentuale in termini di media di tutti i 40 soggetti.

Dai grafici in figura E.9, è possibile concludere che la 1-SVM ha il potenziale di ridurre l'errore percentuale osservato nella classificazione finale di un sistema basato su BCI. Il quarto grafico, cioè quello che paragona il set di dati originali con l'impiego del nu ottimale, mostra chiaramente come il metodo proposto potrebbe effettivamente migliorare la performance. Tale potenziale è tradotto nel numero di soggetti che hanno presentato uno stretto incremento di performance: 32/40 (80%).

Inoltre, da un'ispezione visiva dello stesso grafico, si può vedere che l'altro 20% dei soggetti presenta una performance finale praticamente uguale con quella iniziale, cioè l'errore rimane stabile. Questo è un altro punto a favore della 1-SVM.

Un risultato simile può essere trovato paragonando l'errore percentuale del metodo 1-SVM (figura E.9; grafico inferiore a sinistra) con il metodo di filtraggio per individuazione degli outliers (che vengono successivamente rimossi: sezione 2.3). In tale confronto, nessun soggetto ha presentato una riduzione della performance. L'87.5% dei soggetti ha avuto un miglioramento.

Nell'applicazione della 1-SVM con cross-validation, il rapporto dei soggetti con un incremento di performance assoluto paragonato al set di dati originale (grafico superiore a destra) è ridotto al 45%; inoltre, si potrebbe presentare anche un peggioramento della performance fino a 10%.

Nonostante ciò, esso è stato in grado di ridurre l'errore percentuale di due soggetti in particolare fino a circa il 20%. Risultato non ovvio, se si consideri il fatto che il CSP impiega 3 filtri per classe con lo shrinkage per la stima della matrice di covarianza, che comporta una elevata immunità alla presenza di outliers.

Ulteriori analisi per il soggetto etichettato come VPjh_08_05_27 nella banca dati del BBCI sono stati realizzati e discussi. Tale soggetto è uno di quelli che avevano presentato un espressivo miglioramento di performance dopo l'impiego della 1-SVM con cross-validation.

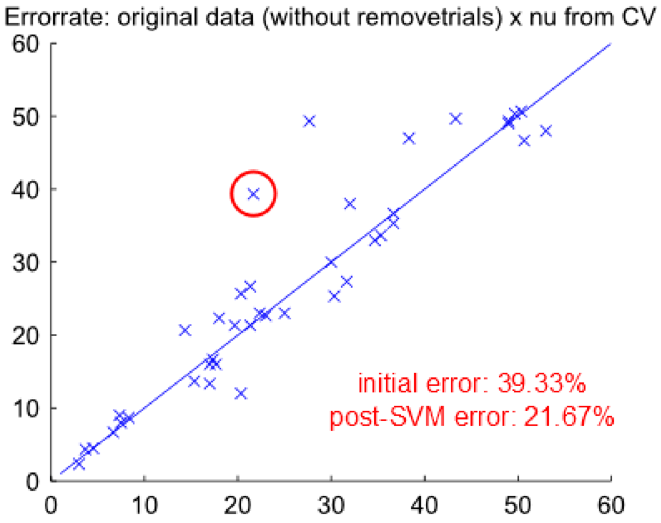


Figura E.10: Performance del soggetto scelto (cerchio rosso) per le ulteriori analisi. Il grafico mostra la riduzione dell'errore percentuale dopo l'impiego della 1-SVM. Il confronto è fatto nel contesto dell'applicazione dello shrinkage per la stima della matrice di covarianza, CSP con tre filtri per classe e l'uso di entrambe le classi come input per la 1-SVM.

Per valutare l'influenza di un singolo campione nella performance finale, i grafici che mettono in correlazione l'errore percentuale con il peso di penalizzazione sono esposti nella sezione 6.2 per diversi set di riferimento. Da questi grafici si possono trarre delle osservazioni e conclusioni importanti:

a) La miglior performance realizzabile non avviene necessariamente per i pesi estremi (0.0 o 1.0), ma può piuttosto essere trovata per pesi intermedi, ad esempio 0.2 (figura 6.2);
b) L'assenza dello shrinkage per la stima della covarianza è risultata, come si aspettava, in un maggiore cambiamento di performance che è tradotto nell'elevata transizione di colori (figure 6.3);
c) Soltanto uno step di variazione del nu (0.05) può risultare in un'espressiva perda di performance (ad esempio 15%, paragonando le figure 6.4 e 6.5); ciò mostra l'elevata sensibilità nell'output del 1-SVM e di conseguenza un ostacolo alla cross-validation per raggiungere una performance ottimale;
d) Un singolo campione, a seconda del set di riferimento, può influenzare considerevolmente la performance, dato che è stato registrato sia un miglioramento che una riduzione del 15% (fig 6.5);
e) La rimozione ed inclusione di campioni che, quando realizzati separatamente avevano migliorato la performance (tabelle 6.1 e 6.2), può anche portare ad un peggioramento della stessa se considerati congiuntamente; ciò riflette la dipendenza del processo di penalizzazione dal set di riferimento.

Per comprendere meglio il punto (e) sopra presentato, la sezione 6.3 presenta delle analisi riguardo il cambiamento della posizione del sottospazio definito dai filtri CSP. Ciò è stato valutato per la rimozione di due particolari campioni (#005, #117) oppure inclusioni di altri tre (#101, #105, #107). Tale valutazione è stata realizzata in seguito ai risultati presentati nelle tabelle E.4 e E.5.

	Initial	Removing #5	Removing #117	Removing both
Weight #005	1.0	0.0	1.0	0.0
Weight #117	1.0	1.0	0.0	0.0
Error rate	**31.00%**	**17.00%**	**18.67%**	**14.67%**

Tabelle E.4: Analisi dell'evoluzione dell'errore percentuale a seconda della rimozione dei campioni suggeriti dalla figura 6.5. La prima colonna mostra i pesi iniziali di tali campioni e il corrispondente errore percentuale. La seconda e la terza colonna presentano la rimozione di soltanto un campione, già risultando in una considerevole riduzione dell'errore. Nella quarta colonna entrambi i campioni vengono rimossi, con l'aspettativa che tale approccio potrebbe migliorare maggiormente la performance. Infatti, l'errore è stato così ulteriormente ridotto.

	Initial	Including #101	Including #105	Including #107	Including all
Weight #101	0.0001	1.0	0.0001	0.0001	1.0
Weight #105	0.0102	0.0102	1.0	0.0102	1.0
Weight #107	0.0016	0.0016	0.0016	1.0	1.0
Error rate	**31.00%**	**20.33%**	**19.00%**	**20.00%**	**16.67%**

Tabella E.5: Analisi dell'evoluzione dell'errore percentuale a seconda dell'inclusione dei campioni suggeriti dalla figura 6.5. La prima colonna mostra i pesi iniziali di tali campioni ed il errore percentuale. La seconda, la terza e la quarta colonna presentano l'inclusione di soltanto un campione, già risultando in una considerabile riduzione dell'errore. La quinta colonna include tutti i tre campioni, con l'aspettativa che tale approccio potrebbe migliorare ancora di più la performance. Infatti, l'errore è stato ulteriormente ridotto.

In particolare, la tabella E.6 mostra che i filtri della prima classe (mano sinistra) abbiano subito una elevata modifica spaziale dopo la rimozione e l'inclusione dei campioni. Ciò si riflette nell'angolo dei filtri rispetto a quelli originali. In particolare, l'angolo rispetto al primo filtro ha subito un'ampia variazione (74 e 85º).

	Angle: original vs. removal	Angle: original vs. inclusion
1st filter	74.582	85.273
2nd filter	52.674	53.486
3rd filter	19.539	85.830
4th filter	8.805	5.943
5th filter	14.026	10.665
6th filter	2.904	7.457

Tabella E.6: Confronto della posizione spaziale dei filtri CSP attraverso gli angoli tra gli stessi, prima e dopo la rimozione (prima colonna) e l'inclusione (seconda colonna) dei campioni.

È stato ipotizzato che tale cambiamento nella posizione è responsabile di una peggiore performance come mostrato in tabella E.7. Il problema può essere dovuto al fatto che i filtri CSP ottenuti dal set di calibrazione sono successivamente utilizzati anche per il set di dati di test (feedback).

	Initial	Removing outliers	Including inliers	Both methods
Weight #005	1.0	0.0	1.0	0.0
Weight #117	1.0	0.0	1.0	0.0
Weight #101	0.0001	0.0001	1.0	1.0
Weight #105	0.0102	0.0102	1.0	1.0
Weight #107	0.0016	0.0016	1.0	1.0
Error rate	**31.00%**	**14.67%**	**16.67%**	**30.33%**

Tabella E.7: Analisi dell'evoluzione dell'errore percentuale a seconda della rimozione e dell'inclusione dei campioni suggeriti dalla figura 6.5. La prima colonna mostra i pesi iniziali di tali campioni e il corrispondente errore percentuale. La seconda colonna presenta l'inclusione dei campioni #005 e #117, mentre la terza include i campioni #101, #105, #107. La quarta colonna impiega entrambi i metodi insieme con l'aspettativa che tale approccio possa migliorare ulteriormente la performance. Tuttavia, in questo caso, la combinazione dei metodi è risultata in un peggioramento dell'errore percentuale, quasi ritornando al valore iniziale.

Infatti, se tali filtri già subiscono un'elevato cambiamento sia per la rimozione che per l'inclusione dei campioni, è possibile in realtà aspettarsi che, al combinare i metodi, i filtri CSP risultanti possano non essere più efficaci e di conseguenza interferire negativamente sulla classificazione finale.

Infine, la tabella E.8 mostra la corrispondenza tra il campione ed il compito (mano sinistra o piedi) realizzato. Nonostante ci si potrebbe aspettare, ad esempio per il caso della rimozione dei campioni, che entrambi corrispondano alla mano sinistra (dato che i filtri corrispondenti, indici 1 – 3, sono stati molto alterati dopo la loro rimozione), in realtà il campione #005 appartiene alla classe corrispondente ai piedi.

#005	#117	#101	#105	#107
Feet	Left	Left	Feet	Left

Tabella E.8: Rappresentazione della classe al quale ogni campione appartiene. I campioni rimossi sono in blu, mentre quelli inclusi in verde.

È stato ipotizzato che il campione della mano sinistra (#117) fosse decisivo nella definizione dei filtri CSP corrispondenti, mentre quello riguardo i piedi no. Nonostante ciò, quest'ultimo è stato in grado di migliorare la performance del 14% quando considerato singolarmente. Tale risultato suggerisce che l'apprendimento del classificatore e il calcolo dei filtri CSP non sono sempre correlati, sebbene entrambi possano influire sulla performance finale di un sistema basato su BCI.

1. Introduction

1.1 What is a BCI?

Brain-Computer Interface (BCI) represents the communication between the brain and a computer that may be used to interpret the real-time activity of the brain through signal processing and machine learning. Such communication offers a great gamma of possibilities for the medical focused research, for example the development of prosthetics controlled by motor imagery [1, 2, 3], wheelchair control [4, 5, 6], and virtual spelling [7, 8]. These applications are all based on the understanding of the subject's intention through BCI and, when effective, could represent a great improvement on the daily life of a person who has lost the movement of an upper and/or a lower limb; and including patients with e.g. amyotrophic lateral sclerosis, whose communication could be facilitated by online interaction through such virtual spellers.

Moreover, BCIs can also be applied into other contexts, e.g. internet browsing by paralyzed patients [9, 10], composition and performance of music [11], exploration of a virtual environment [12] and games playing [13, 14]. The employment of BCIs on these areas also represents a better insertion of e.g. spinal cord injured patients on every kind of activity, in which they could use their intention to conduct and control the corresponding machine that will act in the place of their limb. However, unlike the limb prosthetic control, these operations are not necessarily based on the motor imagery. The intention of the subject can be decoded for example from the direction of the gaze and its corresponding intensity, translated on the amplitude of the EEG [15].

Furthermore the BCI measure and control of the subject's affective and cognitive state can provide to the computer further information about attention, anxiety, pleasure and frustration; offering a further control regarding the current task load [16]. Within the measure of affective states, a BCI can be projected for games applications [17] and for the control of hearing-neuroprosthetics [18]. More recently, some groups have also worked on the prediction of emergency braking situations during simulated driving [19], which may represent a novel technology to decrease the number of car accidents in the future.

As it has been previously exposed, the use of BCIs may stand for a great enhancement on the daily life of patients with diverse kinds of disorders, such as spinal cord injury, amyotrophic lateral sclerosis, and hearing problems. Nevertheless, such approach still presents various difficulties to its implementation, related to each element of the BCI system closed loop (when there is a feedback to the subject: figure 1.1). These challenges will be exposed and further explained along the next topics of this chapter.

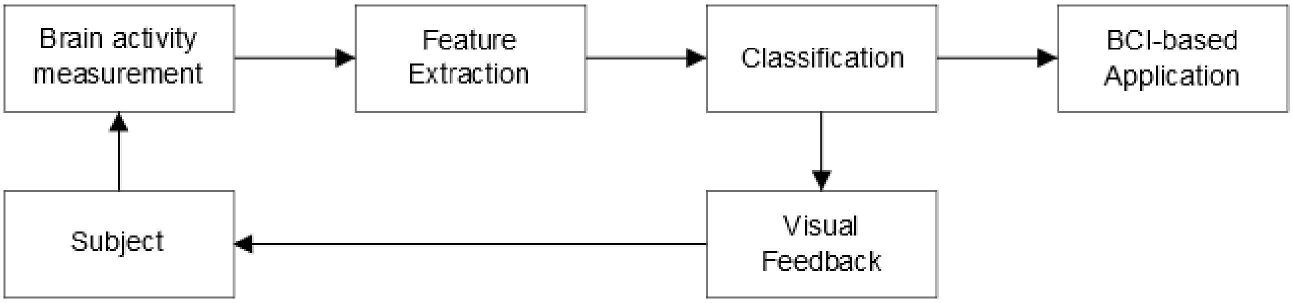


Figure 1.1: A general scheme of the stages on brain-computer interface (BCI). After the classification block, there is the possibility of a visual feedback to the subject (online phase, see section 1.2) that can facilitate the succeeding modulation of the brain activity (e.g. motor imagery, emotional states) towards the desired result.

1.2 Signal measurement

One of the major issues on the feasibility of BCI is the signal measurement, the first step to be performed. The brain's activity can be obtained by invasive – based on surgery for the electrode's placement – or non-invasive methods, e.g. Electroencephalogram (EEG), functional Magnetic Resonance Imaging (fMRI), Magnetoencephalography (MEG) and Near-Infrared Spectroscopy (NIRS). Each approach has its advantages and disadvantages, which will be summarized below along with the methods general operation.

Invasive techniques

The invasive methods regarding BCI are divided into two different approaches: microelectrodes and electrocorticography (ECoG), with the common characteristic of being both implanted through surgery.

i) Microelectrodes

Microelectrodes are electrodes whose width usually is lower than 100µm and that are used to monitor the activity at the microscopic level, where the signals measured can represent the potentials of a single neuron or the activity of a local population of neurons, also measured in the cortex. Therefore, this approach has the advantage of a high resolution on the recording of neuron activity, which is translated into a higher accuracy for the BCI concerning the interpretation of the subject's intention or affective state. However, it presents, besides the great surgery risks the patient must undergo, also the long-term problems regarding the stability of the electrodes position and the arise of tissue around the electrode due to the scarring processes, which results on a decrease of the signal-to-noise ratio (SNR). Another point to be depicted about this technique concerns the current possibility of applications, which are limited to non-human animals, or human patients who consent to be subjected to it. There are some research groups working on this approach and that have also reported improvements on it [20, 21, 22].

ii) Electrocortigoraphy (ECoG)

Electrocorticography is based on the recording of electrical signals that are normally acquired under the dura matter. Unlike the microelectrodes, the signals are obtained from a great area of the brain and spatially averaged. It represents a middle-term invasive technique, standing between the microelectrodes and the non-invasive methods. It is depicted as a choice to attenuate the risks and stability problems presented by the microelectrodes. Moreover, in comparison to the EEG, it has to a better spatial resolution and a lower influence of environmental noise, because it is placed under the patient's skull, resulting in a greater SNR. Nonetheless, considering it is invasive, there are still surgery risks for the patient, including tissue damage and consequent infection. ECoG-based BCI research [23, 24, 25, 26] is realized with epileptic patients that had previously underwent the placement of electrodes onto their cortex surface in order to localize the epileptic focus.

Non-invasive techniques

The non-invasive methods for BCI can be divided into two different groups namely imaging techniques and electrophysiological techniques. The first group is formed by functional Magnetic Resonance Imaging (fMRI) and Near-Infrared Spectroscopy (NIRS), while the second consists of Magnetoencephalography (MEG) and Electroencephalogram (EEG).

i) functional Magnetic Resonance Imaging (fMRI)

In order to map the neural activity, i.e. the subject's motor imagery or cognitive state, the fMRI uses the blood oxygenation level dependent (BOLD) change, which translates into images the task that the subject is performing. For the particular BCI application with online feedback, the real-time technique is used; which allows the concurrent acquisition and visualization of the images. Since this approach is based on magnetic techniques, it requires an isolated room and a large equipment to generate the magnetic field, and therefore may not be feasible for the traditional BCI applications, such some of those described on (1.1) that demand a portable system. Nevertheless, concerning the affective and cognitive states of the brain, this method may expand the applications frontiers due to its imaging visualization and high spatial resolution. Some studies evaluated the possibilities of fMRI employment in the BCI context [27, 28]. Furthermore, in the literature, there are also reviews covering the decoding of mental states obtained with fMRI and the exploitation for self-regulation and the control of emotion disorders and pain [29, 30].

ii) Near-Infraread Spectroscopy (NIRS)

NIRS measures also rely on the contrast caused by the level of blood oxygenation in different areas of the brain. Although, this recent studied method for BCI presents the advantages of do not depend on magnetic sources to create the images, which means the independence of shielded rooms and of large and costly equipments. The NIRS devices are actually portable and with comparable costs to the EEG, since the changes are measured with optical recording devices. However, they represent a decrease on the spatial and temporal resolution. The major drawback of its application relies on the consequent BCI classification performance, which reflects the limited transfer rate of information caused by the hemodynamic latency of change. Therefore, the final accuracy is still reduced compared to the EEG approach, which presents some similar advantages of usage. Nonetheless, diverse research groups are highly encouraged by this new technique for BCI applications [31, 32, 33, 34].

iii) Magnetoencephalography (MEG)

MEG measures the magnetic fields induced by electrical current that is flowing normally on the brain, in particular resulted from ions flowing along the synapses connections. It represents an independence from a further magnetic field application as seen on the fMRI method. Nevertheless, it requires a huge coil with various magnometers in order to read the weak magnetic signals generated on the brain. Moreover it is also necessary to perform the experiments on a completely magnetically shielded room, in order to isolate it from external magnetic sources, comprising the Earth's field. The advantages are the noninvasiveness, higher temporal resolution than the imaging techniques and higher spatial resolution than the EEG. Therefore, in spite of the constraint to a shielded room, it is a promising method and its viability on the BCI field has been studied by several groups [35, 36]. In addition, a new development regarding the equipment scale has been recently reported [37].

iv) Electroencephalogram (EEG)

EEG is the measurement technique still most used for BCI, due to its noninvasiveness and inexpensive experiments; which are characterized by high temporal resolution. Regarding the spatial resolution, there are problems resulting from the external placement of the electrodes, which lay on the scalp (figure 1.2). Nevertheless, as it will be exposed on the next section, various signal processing techniques grant the viability of such method through inverse methods, consisting of spatial filters to identify the sources of the signal obtained in each electrode.

Another problem concerns the attenuation of the signal registered by the electrodes, given by the great distance from their sources. To solve it, the EEG-based BCI experiments require a preparation phase based on the introduction of gel, which will be the interface between the scalp and the electrode, attenuating its high impedance. Considering a cap with approximately 70 electrodes, this process will probably take 30 minutes to be done. It may be tolerable when the application is performed on a laboratory for research purposes; but is out of question for practical contexts, e.g. for the identification of emergency braking situations when driving a car (section 1.1), since nobody would like to spend half an hour to prepare its cap before driving a car.

For this reason, efforts have been done towards the possibility of replacement for dry electrodes [38]. Recently, an encouraging comparison between gel and dry electrodes for spelling with a BCI based on P300 has been published, for that is reported a similar performance resulting from the usage of gold-coated pins in each electrode besides further signal amplification [39]. Still there are other challenges to be faced on the EEG-based BCI [40]. Some of them may be handled through machine learning techniques that will be exposed along this report, including the novel method proposed.

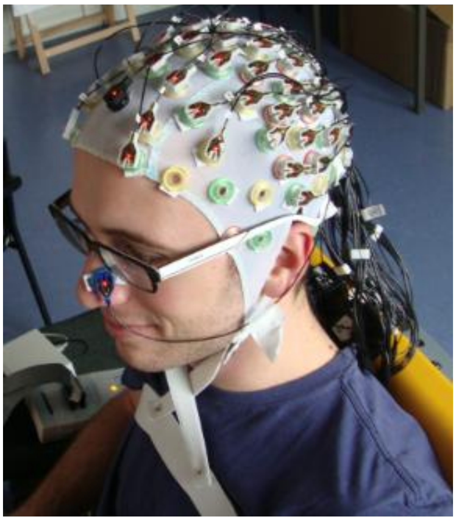


Figure 1.2: Cap with 62 electrodes positioned over the author's scalp for an EEG experiment.

The choice of what technique that one should use for a BCI approach as exposed above is neither unique nor linear. It depends mainly on the final purpose, i.e. the application one wants to achieve with such approach. Moreover, it may be a personal decision. The great amount of research reports focused on diverse measurement methods present on literature [20-41] reflects this point. Each researcher at a particular point may disagree with its colleagues in his preference [42] and it is not possible to tell which the best alternative is.

The present thesis' report will focus particularly on the EEG-based BCIs according to the group within which this work has been developed: the Berlin Brain-Computer Interface project.

1.3 Feature definition

After defining the method that one wants to use in order to measure the brain activity, it is important to determine which particular characteristics the searched information will contain, i. e. there must be something within the signal as an evidence of the corresponding activity, e.g. motor imagery performed by the subject. Such evidence, namely feature, can be divided into three classes for the BCI employment: event-related potential (ERP), slow cortical potential (SCP) and sensorimotor rhythm (SMR).

Event-Related Potentials (ERP)

As the name suggests, ERPs are potentials originated from some other event, which may be external, e.g. a visual stimulus; but also internal. Within the BCI context, the events of interest are those internal, because they symbolize the execution of the task by the subject. The ERPs are time-locked with the source event; and, due to the concurrent presence of diverse others potentials, they are hardly identified from one single trial. Therefore, this approach requires the average over many trials in order to increase the corresponding signal-to-noise ratio (SNR), since the resulting average of the background activity will gradually decrease, while the signal (feature) is always repeated. Nevertheless, averaging is not an option for BCI applications, given its necessity of single trial classification. One method to solve this problem is the usage of Independent Component Analysis (ICA) in order to demix the signals read by each electrode [43].

Some ERPs techniques studied on the BCI field are:

a) Steady State Visually Evoked Potentials (SSVEP), in which the generated signals are a response to light sources oscillating at different and constant frequencies. The response change is related to which image the subject is focusing on, identified by the frequency content on the EEG. [44, 45]
b) P300 is an evoked, positive potential that is recorded approximately 300ms after the stimulus is given. The stimuli are based on diverse images flashing randomly, and appearing only one at time. The subject's task is to attend one of these sources, and one expects the BCI to identify it. Hence, this approach may be used for virtual spellers, where the subject visually selects the letters. [46, 47]
c) Error-related potentials may be evaluated on the event-related responses of the subject, i.e. when the subject performs an erroneous trial, it will present two characterizing components, namely the error negativity and error positivity. Although the first component is also visible on correct trials, the last appears only on the wrong ones. Therefore the error-related potentials may be used as a further mechanism to evaluate and consequently improve the system performance. [48, 49]

Slow Cortical Potentials [50, 51]

Aiming to amyotrophic lateral sclerosis (ALS) patients, whose all motor neurons undergo progressive degeneration, slow cortical potentials (SCP) emerge as a possibility to control electronic spelling devices. In opposition to the P300 potential that is also used for virtual spellers, the SCPs result of a self-regulation of the negative and positive shifts on the EEG signal. Accordingly, within this context the selection of the letters must follow a binary pattern. The alphabet is initially divided into two groups and then the subject selects the group within the desired letter is present. Similarly, the select group is divided into two new groups of letters and the subject selects once again the wanted one with the EEG shift regulation. This process is repeated until the letter is finally identified. This approach has also the possibility of a "go back" option, which is presented when the subject rejects the current two possible letter sets. Therefore it offers the possibility to go back one level, increasing the performance of the letters selection by such approach.

Sensorimotor Rhythm

There are some particular bands of frequency of the signal obtained over the sensorimotor cortex that reflect, e.g. the subject's motor imagery, through a decrease or an increase of the corresponding band power. Such change on the band power is the response to a variation on synchronization of the neuronal populations measured by the electrodes. Depending on the subject's responses, which present variability among subjects, one may consider either the µ- (9 – 13Hz) or the ß-band (15 – 30Hz), since both undergo desynchronization effects, in order to classify the measured signal [52, 53].

Throughout this method, it is possible to observe a great discriminability (figure 1.2) between the rest- and the task-related signal obtained from the electrode on the cortex area that corresponds to the chosen limb, e.g. position CP3 for the right hand motor imagery and CP4 for the left. The discriminability is a result from the event-related synchronization (ERS) or desynchronization (ERD), which represent respectively an increase or decrease of the spectral power in a given frequency band. Both ERS and ERD are frequency-specific changes of the EEG activity.

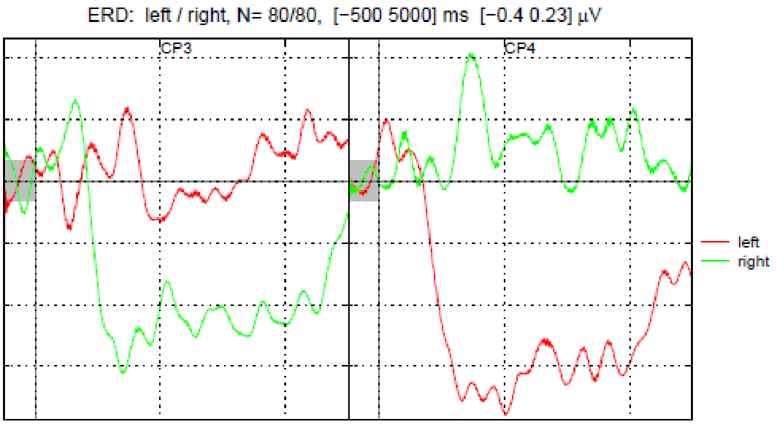


Figure 1.3: The event-related desynchronization (ERD) discriminability along time (ms) between left- and right-hand motor imagery on the electrodes CP3 (left positioned) and CP4 (right). Adapted from [54].

There are two prominent advantages of this method, namely the consequently high performance expected from the observed great discriminability; and the possibility to efficiently apply band-pass filters in order to remove a considerable part of the signal that is not important for this application. The application of band-pass filters, as well as further processing methods will be explored on the next section of this chapter.

Moreover, one may find in the literature [55] further approaches using nonlinear features, e.g. Singular Spectral Entropy, Temporal Asymmetry and Spectral Profile.

Due to the motor imagery context within which this thesis project is written, the report will be focused on the methods and techniques related to the features from the sensorimotor rhythms.

1.4 Signal processing

After the kind of feature that one wants to extract from the EEG signal has been defined, one needs to perform the extraction itself, which requires spatial and temporal signal processing in order to obtain the features correctly corresponding to the trials. In this section such processing techniques will be described. However, only those that were applied to obtain the data used for the experiments along this work are explained; since there is actually an extensive list of possibilities in this area, as for example the one reported after the BCI Meeting 2005 [56], which still does not represent all the methods current in usage.

Frequency Filters

Initially, the acquired signal in a given electrode holds a content of frequency greater than the necessary (considering the case of features based on sensorimotor rhythms) and therefore a band-pass filter could extract the elements of interest. It is a method often implemented as a pre-processing technique of the EEG signal on BCI experiments.

Digital filters are known to be stable since they do not depend on the tolerances of analog components. Moreover, there is the advantage on the flexibility, which is of great importance on the BCI context, since the chosen passing band may be different for each subject. It requires however an analog-to-digital convert, which is not a problem since most of the EEG systems currently available are actually digital. Accordingly it necessitates also an antialiasing filter in order to avoid distortions during the AC conversion.

The sampling rate, i.e. the frequency for the AC conversion, is reported at different values. It may range from 250Hz up to 1000Hz, as reported on the data sets of the IV BCI Competition [57]. Its choice depends on the application and the type of filter chosen. For example, it has been reported that, for autoregressive filters, a higher sampling rate requires a model with higher order in order to avoid distortions [58].

Examples of band-pass filters used for BCI are Kaiser window [59] and Butterworth filter [60], which is characterized by a flat response in the passing band.

Spatial Filters

Since the electrodes are located externally on the scalp, the recorded EEG signal also represents the activity generated from the surrounding area. This fact causes two problems to be solved. The first is related to the identification of the real sources of the signal. The second is associated with the frequency content. Section 1.3 has exposed the translation of the motor imagery from the µ- and the ß-band, which are present on the sensorimotor cortex. However, the α-rhythm, located on the parieto-occipital cortex, presents similar frequency content to the µ-rhythm. It would not be a problem if we were within the context of invasive methods of measurement, since the alfa rhythm is not located on the motor cortex, which is the target of measurement. But within the EEG, the mixture of surrounding sources urges for the application of spatial filters in order to correctly identify the features.

The spatial filters commonly used for BCI can be divided into two types, namely the predefined, with fixed parameters (e.g. Laplacian) filters and the data-derived filters, e.g. Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Common Spatial Pattern (CSP), whose parameters depend on the data presented to them.

i) Laplacian Filter

The Laplacian filter theoretically is based on calculating the second derivative of the instantaneous voltage for each electrode on the neighborhood of the current electrode of analysis. However, when applied to BCI, such voltage (*v)* is obtained through the approximation of subtraction of the *n* nearest electrodes:

$$v^{LAP} = v_i - \frac{1}{n}\sum_{j=1}^{n} v_j$$

The choice of *n* depends on the topology and the number of electrodes to be considered, i.e. how wide one wants to define the surrounding area of the electrode *i.* One may consider, e.g. the electrodes that are 3cm distant from the central one for a small area or 6cm to embrace the neighbors of the neighbors [61].

Advantages of Laplacian filter lie in the lack of constraint to a high number of recording channels and to an elevate number of samples to achieve its expected performance [62]. This dependence is present on e.g. CSP filters in order to avoid overfitting problems. However, as exposed before, these filters work with fixed parameters and therefore are not able to consider the particularities of the subject in order to improve its performance.

In the current work, although data with high number of trials (150 during calibration phase) and of channels (from 64 to 70) were available, Laplacian filters were applied as a first pre-processing approach in order to increase the signal-to-noise ratio (SNR) by eliminating background information.

ii) Common Spatial Pattern (CSP)

This data-derived method has as outputs filters and patterns. The filters represent the columns of the matrix that concurrently maximizes the variance of one class and minimizes the variance of the other when multiplied by the measured signals on the electrodes. Patterns, on the other hand, have the opposite effect: they are the rows of the matrix that can reproduce the measured signals from the filters-amplified brain activity. It may sound of no use to have the patterns, since its application would mean coming back to the non-processed signal. However, they might be very important when visualized graphically over the cortex, since they represent the actual distribution of the corresponding source over the scalp (figure 1.3).

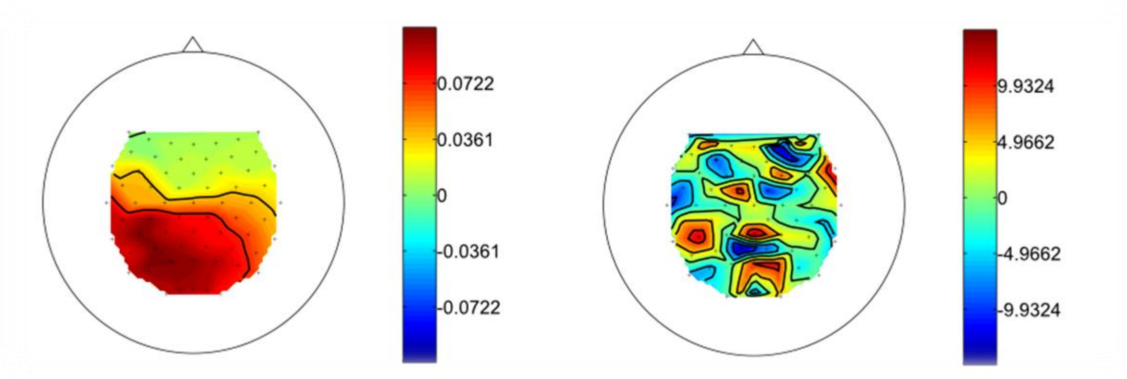


Figure 1.4: CSP pattern (left) and filter (right) scalp distribution for right-hand motor imagery. One could conclude that it is a right-hand imagery because of the pattern plot, where the strongest values are located mainly on the left area of the scalp. Positive and negative figures do not differ in this context, and therefore should be considered as having the same effect. All electrodes (channels) used for the experiment are represented by a corresponding value within the (de-)mixing matrix and translated into the plots above, which are continuous because of an interpolation process between the points where the electrodes are.

For an ERD-based BCI, as depicted on figure 1.2, the motor imagery of a limb corresponds to the decrease of the spectral power on the cortex, over its contralateral area; while, at the ipsilateral level, the power is kept idle without any significant change by the thought of motion. The CSP algorithm is focused on amplifying such discrimination of spectral powers observed on each of the sensorimotor areas, since the power spectral is obtained directly from the application of logarithm to the amplified variances of classes.

The basic mathematical theory behind the CSP algorithm consists of a generalized eigenvalue problem [63] in order to solve the following system of equations:

$$W^T.C^1.W = D$$

$$W^T.C^2.W = I - D$$

In which W and D are the variables of the system, matrices with values between [0,1] that one wants to find. I is the identity matrix; $C^1 \text{ and } C^2$ are the estimated covariance matrices corresponding to the first and second class. Usually in a BCI experiment within this context, the evaluation of the classifier is performed between two classes selected out of three (see section 3).

A detailed mathematical explanation of the CSP algorithm is found in [64], the first paper elucidating this approach for BCI. After this publication, further implementations on this algorithm have been suggested on literature. For example the Common Spatio-Spectral Pattern (CSSP) [65], which improves the information transfer rate and consequently the final BCI classification performance; and the stationary Common Spatial Patterns (sCSP) [66], which regularizes the CSP solution towards stationary subspaces in order to make it invariant to nonstationarities (responsible for performance decrease) present on the EEG signal.

1.5 Classification

Once the features have been extracted, i.e. after applying the logarithm to the trials whose variances have been maximized (first class) and minimized (second class) by the CSP filters application, they need to be classified as belonging to one or to the other class. It comes therefore to the next step on the BCI system, which is the classification block. But in order to have an efficient classifier, it has to be initially trained.

Within the EEG measurement context, the experiments are divided into two phases: calibration (offline), where the subject is asked to perform different tasks, for example to imagine the movement of left hand, right hand and feet; and the phase with feedback (online), where the subject has to once again perform some tasks, but with feedback, i.e. he has the chance to regularize his mental states in order to achieve the desired BCI output (see section 3 for a detailed explanation over the BBCI experiments).

The calibration phase is actually the one used for the classifier supervised training. For each trial performed by the subject, the classifier receives the feature extracted by its EEG signal through the signal processing methods previously explained, and the corresponding label for this feature. Accordingly the classifier learns which features, for the present subject, corresponds to each of the possible tasks. During the online phase, the resulting generalization capacity of the classifier will be evaluated.

For the current project, two types of classifiers have been used: Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM). LDA was applied to the purpose above explained, whereas SVM represents the novel method of pre-processing that will be suggested (section 4) in order to increase BCI performance.

As the name suggests, LDA is a linear approach for classification. It means that the separation of the n-dimensional space of trials is made through a linear hyperplane. For the current BCI chosen method, the classifier is used to discriminate two different classes. This technique consists in calculating the normal vector ***w*** and a threshold *b*, which define the direction and position of the hyperplane:

$$f(x) = sgn(\boldsymbol{w}.\boldsymbol{x} + b)$$

The function *f* above defines whether the present unlabeled BCI features, characterized by the vector ***x***, belong to the first or to the second class. The final class label is therefore given from the signum function, whose output for each feature might be +1 or -1 depending on the scalar product and the threshold value.

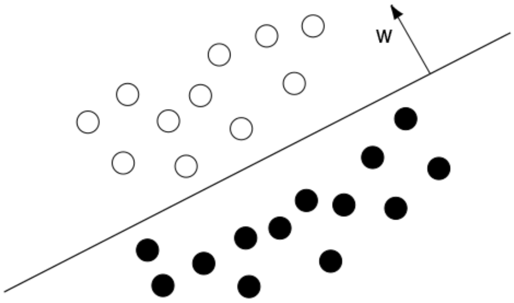


Figure 1.5: Classification boundary for a linear classifier defined by a hyperplane with normal vector ***w*** and the threshold *b.* The white circles represent the trials classified as the first class, e.g. those that resulted output +1 from the function *f(x)*, whereas the black circles represent the second class.

The figure 1.5 may be seen from two different points of view. The first is based on the online phase, where the hyperplane has been previously defined and therefore the *f(x)* is used to classify the trials collected. The other point is from the calibration phase, in which the inverse approach is done. The labels of the trials are known and one wants to find the optimal classification boundary, i.e. estimate the parameters ***w*** and *b*, that linearly separates the classes.

LDA computes (***w,*** *b*) maximizing the average margin, where margin is the minimal distance between a single trial and the created hyperplane. Firstly, it assumes that the trials are equally distributed into two classes as Gaussians with the same covariance when projected in the direction of ***w***, which is done to reduce the high features dimension.

The optimal classifier is the one resulting in the largest distance between the means of projected trials. Each mean corresponds to one class. Moreover, the variance of each class in the projected direction is also expected to be small, in order to guarantee the separability of the trials. (figure 1.6)

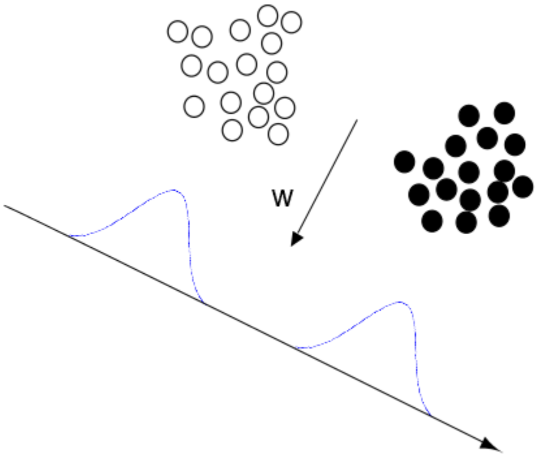


Figure 1.6: LDA projection of the two classes along the direction of the optimal vector **w**, resulting on corresponding Gaussians with maximal means and minimal variances, in order to guarantee the class separability desired for the classifier. The white circles represent the samples corresponding to one class, whereas the black circles correspond to the other class.

A detailed mathematical description of the Linear Discriminant Analysis can be found in [67], which is a book available on the Stanford University webpage (as it was checked on the 5th of June, 2012).

2. Outliers on BCI

2.1 Introduction

The effectiveness of a LDA classification is based on the discrimination of the trials that belong to each class. As explained in section 1.5, it is based on estimating the best direction to project the Gaussian distribution corresponding to each class. Accordingly, the optimal class separation is performed through the maximal mean and minimal variance of these distributions. For a pure task-related EEG data collection, it should not be a problem to obtain such optimal separation. However, the presence of outliers among the collected trials is commonly observed. Depending to the quantity and position of such outliers on the trials space, they can change drastically the final, i.e. after learning, LDA classifier properties (figure 2.1), resulting on a bad BCI performance during the feedback phase.

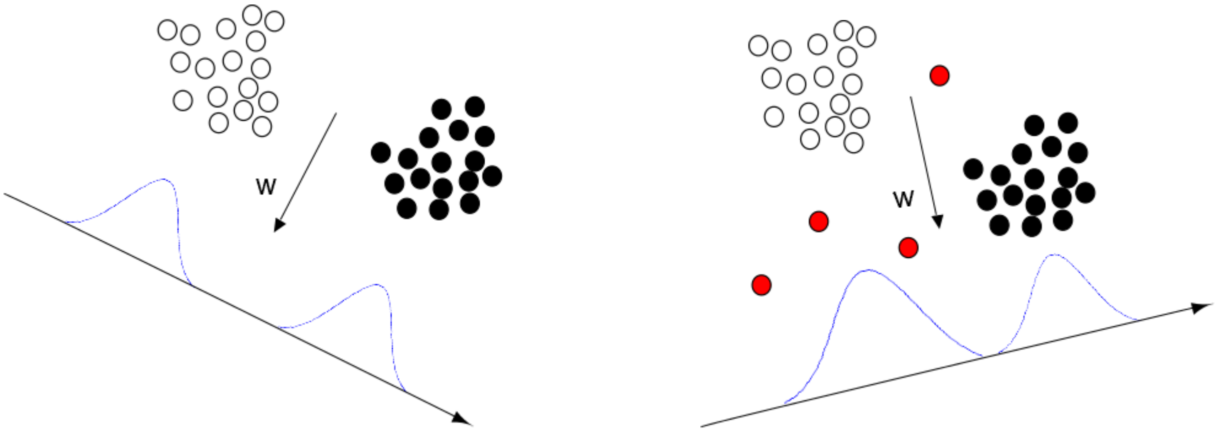


Figure 2.1: Influence of the presence of outliers among the samples. At left, the original distribution of the pure, task-related EEG samples. At right, four outliers (red circles) have been added. The outliers belong to the class represented by the white circles. Accordingly the optimal vector **w** direction changed in order to keep the maximal mean and minimal variance of the projected Gaussian, which resulted in a spatially approximation between the classes and consequently a decrease on the separability between them.

Therefore outliers on BCI are defined as samples that do not correspond to the actual aimed trials to be interpreted. In the context of the present work, outliers are any kind of trial whose characteristic differs from the actual ERD/ERS features from the EEG signal, what is translated into a notable decrease of the BCI performance.

Moreover, such outliers, besides representing a problem for the LDA classification process on a BCI system, can also strongly affect the calculation of the optimal CSP filters and patterns. It may occur since the CSP algorithm is based on the estimation of the covariance matrices representing the two classes. And the great sensitivity of the covariance matrix estimation has already been reported as correlated to the presence of outliers [68].

2.2 Outliers sources

For the EEG measurements, there are several sources of outliers, which can be divided into internal and external sources, i.e. artifacts generated respectively physiologically or not.

The external artifacts may be related to the interference of power-line, centered on 50 or 60Hz, depending on the country where the experiment is performed; detachment or elevated pressure of electrodes [69]; and changes in electrode impedance, caused by e.g. slight movements that change the area of contact.

However, the non-physiological artifacts are usually not a problem for BCI experiments. Concerning the influence of the noise from power-line, the related frequency is actually eliminated during the signal processing phase through frequency band-pass filters, as exposed on section 1.4. The other problems are avoided through an extra attention of the person who is running the experiment summed with the fact that the cap is firmly fixed over the subject's head and also tightly held around the chest (see figure 1.2).

Among the internal sources of outliers on EEG, those regarding BCI are reported as being mainly the oculogenic and myogenic potentials [70, 71]. The electrooculographic (EOG) activity is related to the movement of the eyes and the eyelids (e.g. an eyeblink), both resulting in an alteration of the electrical field to be registered by the closest electrodes, i.e. over the anterior head regions. Electromyographic (EMG) activity is subject of movement of the muscles, e.g. of the head, jaw, tongue and cranial muscles. Even though the patient is told to stand still, when he is confronted by difficult tasks, there is the possibility of non-intentional facial movement as an emotional reaction to them [72, 73].

EOG presents its spectral peak at frequencies below 4Hz and the EMG's is higher than 30Hz, which means both are out of the range of the bands of interest (mu and beta) that are approximately between 8-30Hz. Nevertheless, both activities are actually characterized by a wide frequency range [71], resulting on a direct noisy influence on the ERD/ERS features.

Although EOG and EMG are the most studied physiological artifacts on BCI, there are also other possible sources, such as the rhythm activity of the heart beats (Electrocardiography) and respiration movements; besides the possibility of sweat presence which could lead to a change on the interface electrodes-scalp impedance [74]. The last problem may be present e.g. after a common long-session BCI experiment.

Additionally, there are some outliers not reported from clinical EEG acquisitions, but in particular for BCI experiments. They occur when the subject does not manage to perform the current task on the calibration phase. It may happen for example due to lack of attention either when the subject is bored after a long repetition of tasks, or caused by simple distraction for several reasons. Usually when the experiment is running, other persons present in the room are advised neither to talk nor to make any kind of noise.

2.3 Artifacts handling

In order to avoid the problems presented on section 2.1, there must be some approach within the BCI techniques to eliminate the prejudicial effects of the outliers' presence. The first method to handle the artifacts is based on the instruction of the subjects before the experiment is performed. For example, it is asked not to move or close the eyes while a trial is being collected (see section 3 for experimental setup), in order to avoid the EOG artifacts. However, the previous teaching alone is not enough to have a set of data exempt of outliers. It is not possible either to avoid other involuntary sources such as facial and cranial muscles movement (EMG), or to control the respiratory movements and heart beats. Moreover, the subject attempt to control his eye's movements may be a new source of outlier, since it is another cognitive task to be performed [75, 76].

A second method, which may be used to complement the first one, is based on artifact samples removal. It can be performed through manual or automatic techniques. Manual ones depend on an expert's decision, while automatics are based on machine learning methods that run autonomously.

i) Manual methods

One alternative for manual technique is grounded on Independent Component Analysis (ICA) [77]. Similarly to the CSP technique (see section 1.4), ICA is a spatial filter method that attempts to separate the signals from different sources. However, it depends on the assumption that such sources are linearly mixed and independent of each other along the time, since this approach consists in separating the EEG signal into independent directions. After the signals have been separated, one may manually isolate the components produced by EOG artifact activity and then reconstruct the EEG with the clean data [78].

It has already been reported that IDA properly accomplishes to detect EOG artifacts on EEG data [79], nevertheless it was also demonstrated that its application to EEG data does not produce any improvement when considering artifacts related to muscle activities [80]. Moreover, given that it is a manual technique, depends on a further inspection of an expert, which represents a delay between calibration and online phase for the BCI experiments on laboratory and also the impossibility for daily life applications, in which all the processes between the subject's signal acquisition and the final BCI decision must be autonomous.

ii) Automated methods

There are several possibilities for automatic techniques on outlier detection. A first idea could be based on applying an amplitude threshold to the EOG and EMG signal, i.e. their acquisition would be needed in parallel to the EEG collection. Through this way, whenever an EOG/EMG sample exceeds the threshold, the corresponding EEG trial is automatically labeled (and removed, if this is the goal) as outlier. An intuitive drawback of this approach lies on the necessity of further signals acquisition, implying extra electrodes to be worn on the face, which would not feasible for out of laboratory usages. Moreover, it would also be restricted to the detection of outliers caused by EOG and EMG activity.

Mostly implemented algorithms are however based only on EEG signal acquisition. These methods can be further divided into new three categories, namely embedded, filter and wrapper methods.

Embedded techniques are characterized by an integrated process of trials selection, i.e. the process to take into account only the relevant samples and eliminating the outliers; with the classification process. An example of such approach is the C4.5 algorithm [81], in which the classifier is built upon the concept of decision tree whereas its learning is based on information entropy. In this case, the decision tree is gradually created using only the most relevant extracted features, symbolizing a feature selection. On applications for BCI, works based on filters or wrappers methods are most reported. Nevertheless, there are also attempts towards embedded techniques [82].

Filter methods, in opposition to the embedded ones, are apart from the classification process. It means that the feature selection based on them does not use any information about the classifier performance for its decision, i.e. the criterion used is inherent to the knowledge of the available features. In this direction there are attempts using for example Principal Component Analysis (PCA) to signal decomposition into frequency bands, followed by a threshold application to the variance, which translates the artifact activity [83]. Another filter method is presented on [84], where it has used a Relevant Dimension Estimation (RDE) [85], which is based in a kernel PCA for the signal decomposition. The work on [84] was particularly focused on the detection of noisy-samples that do not correspond to the actual task expected to be performed.

Although some performance improvement was reported on these works, the filter methods present the drawback of not considering the classification output related to the feature selection, which could be useful in order to further improve it. Wrapper methods [86] may be an optimal approach, since they work between the characteristics of embedded and filter approaches, i.e. they use the classification performance as a feedback to help on the final feature selection (figure 2.2).

The idea is to use the classifier in order to help in the selection of the best samples subset that will be used for the actual learning of the classifier on the calibration phase. This can be performed through a process called cross-validation. The training data set is divided into *n* smaller subsets and one may use *(n-1)* subsets as a hypothetical new training set and the *n-th* subset as the test to evaluate how the classifier would perform, i.e. its final performance, when using the current parameters for the samples subset selection. It is an important step since the samples selection algorithm may be extremely sensitive to such parameters.

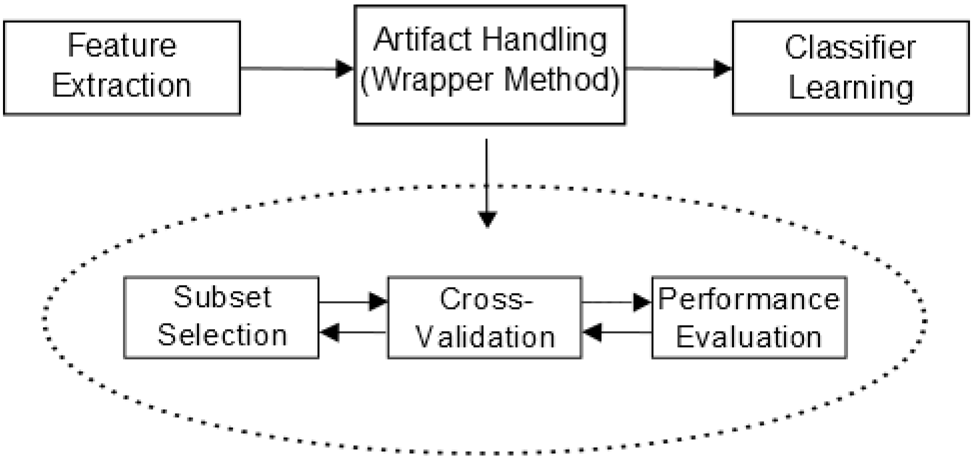


Figure 2.2: Wrapper method for feature subset selection. Such approach is used to choose which labeled features will be actually applied for the supervised learning of the classifier during the calibration phase. This artifact handling consists on using the own classifier for a performance evaluation of the possible subsets, depending on the wrapper algorithm parameters. Cross-validation divides the current subset into *n* smaller subsets, from which *(n-1)* are used for the learning and the *n-th* for the performance evaluation.

An example of wrapper method for feature extraction is the Genetic Algorithm (GA) [87]. It is inspired on the process of natural evolution, using similar properties such inheritance, mutation, selection and crossover. The selection of the features subset is based on a fitness function, which in this case represents the classifier performance. Recently some works [88, 89] have reported improvements on the BCI final classification when combined with such algorithms. Moreover, it has also been reported an attempt to combine a features subset selection with the reduction on the number of channels, both through GA [90].

The present project presents a novel wrapper method for artifacts handling based on Support Vector Machine (section 4). Moreover it compares (section 5) the performance results with a filter method available for the "Berlin Brain-Computer Interface" projects; and also with the original set of features.

Such filter method is based on removing the trials that present an excessive variance in at least 20% of the channels used for their collection, i.e. 14 channels for the current case. Although the CSP actually tries to increment the variance difference in order to have a better discriminability of two classes; an excessive variance, which is defined by a user-chosen threshold value, may actually illustrate a noise contamination of the corresponding trial.

Therefore, the trials, whose 20% of channels presented variance over the chosen threshold, were labeled as outliers. This method will be used for results comparison (section 5) and referred as "removetrials".

3. Data collection

The data applied on the current evaluation of the BCI classification performance improvement by 1-SVM as pre-processing method and outlier detection have been previously collected for past works within the Berlin Brain Computer Interface (BBCI) project. The original work report is found on [97].

It consists of EEG records during calibration and feedback phase, respectively corresponding to training and test data set, obtained from eighty BCI-naïve subjects, i.e. with no previous experience on BCI experiments. All the subjects had to fulfill the following requirements in order to be eligible for the experiments:

i) Full contractual capacity, i.e. legal ability to enter into a contract;
ii) German as mother tongue, because required psychological tests were performed in German;
iii) Absence of any neurological disease, since its presence could interfere on the final results.

The subjects that took part on the experiments were by that time from 18 until 41 years old. 41 of them (51.25%) were female. And 5% were left-handed. The last information is reported because there might be differences on the right/left motor imagery performed by a right- and by a left-handed subject.

In order to avoid variability on the data, each subject performed the complete session of experiments in the same day. Each session took from 5.5 to 6.5 hours; and the external subjects, i.e. those who were not members of the BBCI group, were paid 8 EUR per hour.

Although the experiments to collect the data for the original study were so long, not all BCI experiments are necessarily similar. It depends e.g. on the chosen configuration of electrodes and on the number of trials to be performed. For instance, the author has participated on another BCI experiment that lasted 2.5 hours.

However, the EEG data, to which the current work refers, was collected along with electromyogram (EMG) and electroculogram (EOG), which were also needed for the original project. It resulted in a cap with 128 electrodes, whose preparation, e.g. the gel insertion in order to attenuate the input impedance (figure 3.1), corresponded to more than one hour of the whole session.

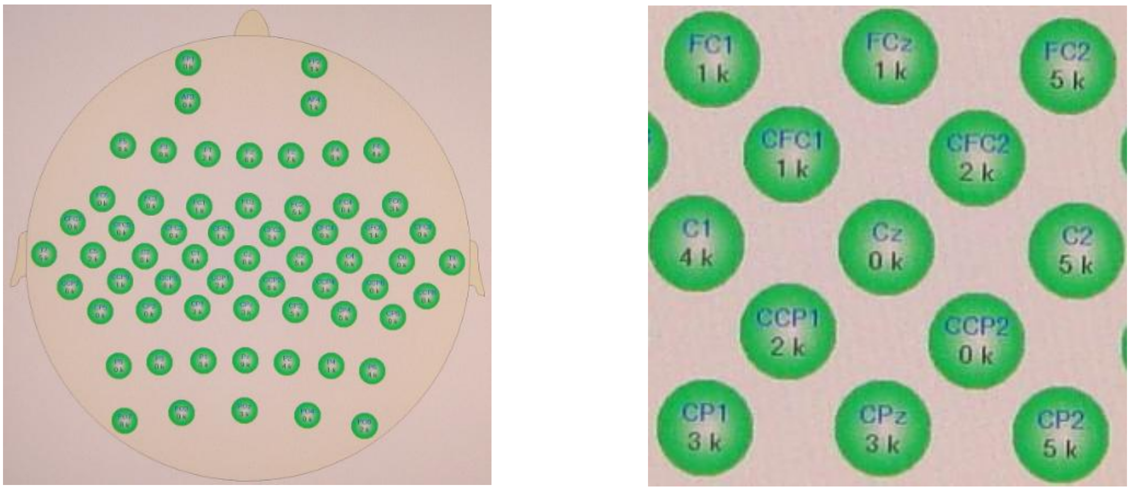


Figure 3.1: Projection of the electrodes and their corresponding actual impedances on a monitor screen. On left the whole scalp distribution is depicted; while on right, part of the same image zoomed in is showed. Each circle represents an electrode positioned on the subject's scalp. It is possible to control which is the minimum impedance desired for all the electrodes. Whenever an electrode's impedance reaches this value, it becomes green on the screen. It can also appear as yellow (close to the desired value) or red (far from it).

The EEG signal was collected from 68 electrodes positioned along the scalp of the subject. The technology is based on multichannel EEG amplifiers (BrainAmp DC by Brain Products, Munich, Germany) with 119Ag/AgCl electrodes. The signal was sampled at a frequency of 1000Hz and filter band-passed from 0.05Hz to 200Hz.

3.1 Measuring phases

Initially, a section focused on EEG artifact recording is executed. The tasks presented for the subject during this phase were to perform eye movements, eye blinking and maximum voluntary contraction of the hand. Subsequently the participant was asked to stay relaxed for 15s with the eyes open and 15s with the eyes closed, which was repeated five times to conclude this phase.

The second phase anteceding the calibration was focused on the observation of movements. The subject did not need to perform any particular task, rather only watch attentively video clips showing, from a first person perspective, the movements of left hand, right hand and feet; each one at a time.

The third phase is the calibration, in which the subject performs the motor imagery tasks without feedback. It is divided on four runs, each one of 75 trials collected. On the first run, the subject still realized the actual movement of the limb instead of the motor imagery, while on the other three runs, the participant was randomly asked to perform one of the three following motor imagery: either right-hand, left-hand or feet.

A trial was performed every 8s during a run. Each trial was constituted of a 2-second period of subject's fixation with a black cross in the center of the screen; followed by 4s of actual task performance which was conducted by an arrow, whose direction indicated the desired movement to be imagined; and finally 2s of blank screen that preceded the next task. For every 20 trials performed, there was a break of 15 seconds.

Moreover, between the motor imagery runs, the subject should take a virtual version of the d2-test [98] in German – an attention test that was part of the psychological tests previously mentioned on this section.

At the end of the calibration phase, 225 motor imagery trials have been therefore collected. These trials are uniformly distributed between the three classes: right-hand, left-hand and feet, i.e. 75 trials of each one.

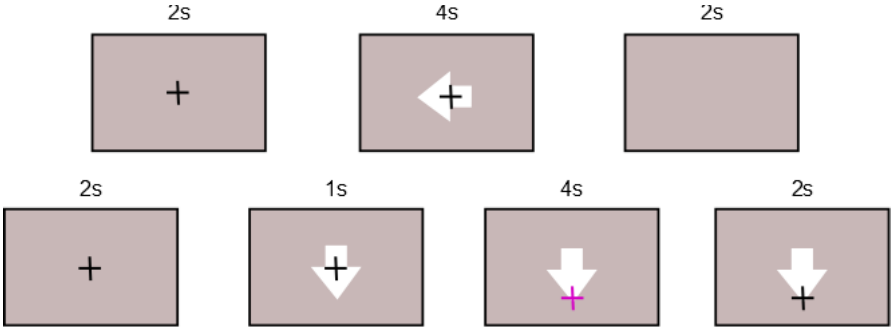


Figure 3.2: Trial collection on calibration (top) and feedback (bottom) phases along the time course. The calibration phase is characterized by a 2s preparation, 4s for task performance (in this case, the subject is asked to perform a left-arm motor imagery) and 2s preceding the next trial. The feedback phase begins in the same way, but during the task performance there is an actual feedback for the subject, who is able to see the classification output for the executed motor imagery. During the feedback frame, the cross turns purple and moves restrictedly along the screen according to the interpreted motor imagery. Depending on the classes evaluated, it has two directions of movements. For example, for left-arm versus feet, the cross can either moves down, as depicted, or left. On the last frame a correctly performed feet motor imagery is depicted, since the cross turned black, i.e. representing the end of task execution, on the desired position.

The fourth phase was performed with feedback, i.e. the subject could observe the classifier output that corresponded to its current task execution and therefore had the chance to correct it if it was different than the expected. It consisted of 3 runs, each one collecting 100 trials. However some subjects did not perform all of them because either of fatigue (three subjects) or lack of time (seven subjects).

The beginning of a trial collection is similar to the calibration phase, i.e. for 2s a black cross was fixed on the center of the screen and the subject had to prepare himself for the task performance. Afterwards, the arrow appears indicating the desired movement to be imagined. In this case however the task execution was during 5s instead of 4s. During the first second, the black cross was kept fixed; but, on the following 4s, it moved according to the motor imagery concurrently interpreted by the classifier, which characterized the feedback itself. Therefore the subject had the chance to control the cross for 4s and correct it whenever the classifier output was different than expected. Once this stage was finished, the cross was then fixed for 2s on its final position, preceding the next trial measurement. It resulted on a period of 9s for each trial.

There was also a break of 15 seconds after every 20 trials collected. Only during this break the subject could see the partial score of its previously performed tasks.

However three different movements are imagined during the calibration stage, on the feedback case there are only two possibilities of tasks, which correspond to the two classes selected after the processed results from the calibration, according to a cross-validation based estimation of the best classification capacity.

There are three possibilities of couples of classes (1+2; 1+3; 2+3) after the calibration phase and for each of these combinations, one applies cross-validation in order to evaluate the final performance of the classifier. Therefore, the couple of classes resulting on the minimum error-rate are taken for the feedback phase.

3.2 Early processing

The collected calibration data, besides undergoing the initial band-pass filter from 0.05Hz to 200Hz, is further processed specifically for each subject, according to the characteristics presented. The original heuristic algorithms are presented on the appendix of [63] (algorithms 1 and 2), and will be explained hereinafter.

i) Frequency filter

Initially the band-power of every trial collected on the calibration phase, for every channel used on the experiment, is taken into account and expressed in terms of decibels, i.e. the spectrogram is calculated. However, it is done only for the band from 5 to 35Hz, which is enough to incorporate the sensorimotor rhythms of interest (see section 1.3).

Subsequently frequency scores are estimated for each channel from a matrix of correlation coefficients *R* obtained from the covariance matrix *C* of the 3-D channel's spectrogram, i.e. the rows are the frequencies whereas the columns represent the trials.

$$R_c(f,i) = \frac{C_c(f,i)}{\sqrt{C_c(f,f)C_c(i,i)}}$$

The obtained scores are later summed along the channels, whose result is used to obtain the highest power corresponding to *fmax*. The subject's band is finally determined considering the minimum and maximum frequency values that correspond to 5% of the maximum band power.

Such values are found by decreasing and increasing, respectively, the value of *fmax* of 1 unit per step and comparing the power of the current frequency with the threshold of 5%. For example, if fmax = 20Hz, the first step will take into account 19Hz and 21Hz as candidates to the lower and upper bounds of the subject's frequency band to be considered for the next processing steps.

ii) Temporal filter

Although the calibration phase period designated for the task performance is fixed on four seconds, the actual period considered is also subject-dependent. The technique utilized is similar to the one for the frequency filter, while making use of correlation coefficients as scores. However, the variable used as input in order to obtain such scores is not the spectrogram anymore, but rather an envelope of the sample on instant *t* for a given trial and a given channel. Such envelope is calculated by Hilbert transform [99] and subsequently smoothed.

Analogously to the frequency filter, the maximum score here is also taken as reference in order to obtain the boundaries *t0, t1* of the period to be considered, whereas using the initial values both as *tmax* (instant *t* that corresponds to the maximum score). Nevertheless, the threshold to be respected in this case, in order to keep changing *t0* and *t1*, is expressed in terms of percentage of scores contained on [t0,t1] as following:

$$while \sum_{t_0}^{t_1} tscore(t) < threshold = 0.8 * \sum_{t} tscore(t)$$

Applying this technique for the calibration phase data obtained from the 80 subjects of the original study, the resulting period to be considered ranged from [480,2720]ms until [1700,4990]ms, with a median length of 2730ms and a median period ranging from 1000ms to 3890ms.

Both methods for earlier processing of data were considered in every test of this project, i.e. every result presented on the next section has, as determining factors, besides the particular situations and choices that will be exposed, also the temporal and frequency subject-specific filtering.

3.3 "Online" phase

So far the current section focused on earlier processing of the data generated during the calibration phase. The online phase, however, has to process the data concurrently to the subject's motor imagery in order to provide him the visual feedback. It means that all the steps leading to the final BCI classification should be constantly done, so as to have a reliable and effective feedback. On the original project that collected the data applied for this report's work every 40ms such processing steps were realized within a sliding window of 750ms. Therefore the feedback output, i.e. the purple cross depicted on figure 3.2, was refreshed 25 times per second while taking into account the trials recorded on the past 750ms from that moment.

The refresh of classification was performed for 4s, after that the cross turned to black and its current position determined whether that trial was a successful one. The final BCI performance simply counted in terms of percentage how many trials reached the desired task when the cross was fixed, independently of its positions during the previous 4s.

The range of final performances obtained by the original work was extreme vast, i.e. from 50% to 100%. And actually such results reflect one of the bases of comparison for the 1-SVM method proposed in this report, because the processing methods employed on the original work and on this project were almost the same. During the calibration phase, the CSP (Common Spatial Pattern) filters were calculated and then applied to the online stage without further adaptation; while the Linear Discriminant Analysis (LDA) was the classification technique chosen. However, on the current project, a further pre-processing step based on Laplacian spatial filters was used in order to reduce the background noise on the samples.

Another important difference is that the original work actually had the feedback, online phase; while the present work, based on a so called offline method, could not profit from it. It rather used the test data that are somehow conditioned to the previous experimental results.

Although an offline analysis is not ideal, it could be a first step to demonstrate the effectiveness of the proposed method for outlier identification on the BCI performance improvement. Moreover, it has been hypothesized that, if the novel method (presented on chapter 4) is able to enhance the classification of a test data set actually correlated to other feedback conditions, it would perform even better on real-time, online applications.

4. Support Vector Machine (SVM)

4.1 Motivation for BCI

As mentioned on the section 1.6, Support Vector Machine (SVM) is a method for classification of trials. The central concept is to find the hyperplane that maximizes the margin, i.e. the minimal distance between a single trial and the hyperplane chosen. This idea reflects the attempt to make the classifier robust for, e.g. the online phase classification on a BCI system. When the trials from the calibration phase are found too close to the hyperplane, a performance decrease is expected during the feedback phase, because probably there will be some trials classified as the second class while the subject had actually intended the first one. Therefore the margin indicates the discriminability level of the system.

Accordingly, SVM has also been reported as an option of classification method for BCI systems [91, 92]. For such scope, one generally uses the so-called two-class Support Vector Machine (2-SVM), i.e. the SVM distributes the trials into two, and only two, different classes. Therefore it is assumed that every sample available belongs to one of these two classes, which is a reasonable hypothesis since: i) the BCI systems usually work selecting the two most discriminative classes out of the three possible; ii) as it would be the classification step, one may expect that the outliers have been already removed or penalized, i.e. down-weighted with respect to the clear samples.

In this project, however, SVM has been applied as a novel method to identify artifacts on the calibration data set. After the EEG data are collected, Laplacian filters are applied as a first pre-processing approach in order to increase the signal-to-noise ratio (SNR) by eliminating background information. The output from these filters is the input for the one-class SVM (1-SVM) that will perform the outlier identification. After the bad trials have been removed or penalized (see section 4.3), the samples are further processed through the Common Spatial Pattern (CSP) method in order to obtain the labeled, spatially reduced trials to be used on the classification learning.

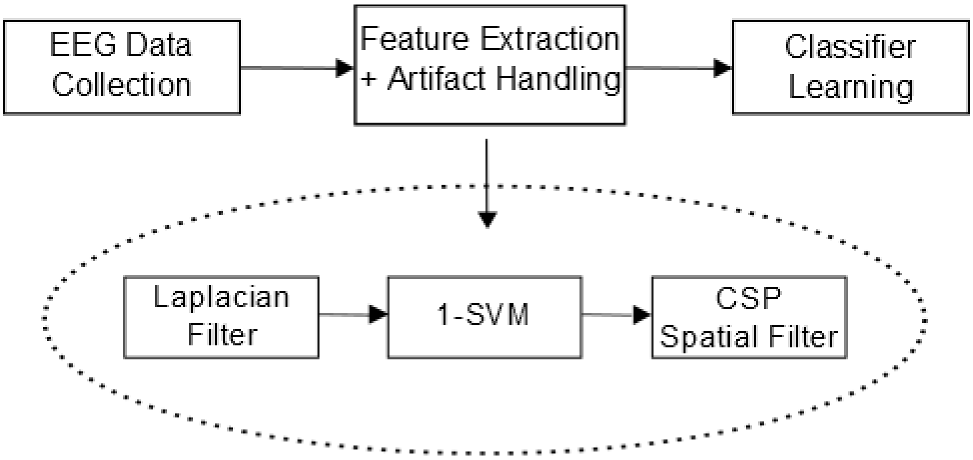


Figure 4.1: Scheme of the calibration phase proposed by the current project. After the EEG data have been collected – which includes the band-pass frequency filter (see section 3), there are three further processing steps prior to use the features for the supervised classifier learning. Namely, Laplacian filter in order to attenuate the background noise; 1-SVM to identify and remove/penalize the outliers; CSP spatial filters in order to reduce the data dimensions whereas amplifying the difference (through variance) of the classes.

The fundamental hypothesis is that the clean samples present similar spatial characteristics, even though they are distributed along 68 dimensions (number of channels used, as exposed on section 3). One could expect the outliers to be out of the subspace delimited by the proper trials and therefore use the 1-SVM in order to detect them. Note that the classification process is based upon the 2-SVM, while the current proposed method is founded on the one-class version. The basis for such choice is that the outliers are probably not concentrated on a single subspace. And accordingly there is no reason to separate the classes into two determined clusters, but rather identify which trials belong to the inliers cluster (one class) and which do not, i.e. outliers that can be found anywhere except in the subspace of the good, pure samples.

For this work, the SVM-KM toolbox [93] has been used with some further implementations in order to avoid high dimensionality problems, as explained on the next section along with the mathematical details.

4.2 How it works

Initially as a linear classification approach, the central idea of the Support Vector Machine is to separate trials belonging to *two* different classes through a linear hyperplane delimited by the following equation:

$$f(x) = \ sgn(\boldsymbol{w}.\boldsymbol{x} + b)$$

The vector **w** is normal to the optimal hyperplane chosen, which is characterized by resulting in the largest margin, i.e. the minimal distance between a single trial and the hyperplane. It is an optimization problem that can be translated into:

$$min_{\boldsymbol{w} \in R^N, b \in R} \quad \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$subject\ to\colon\ f(x_i).(\boldsymbol{w}.\boldsymbol{x_i} -\ b) \geq 1\,, (for\ i = 1, \ldots, N)$$

The name "Support Vector" comes from the idea that each $\boldsymbol{x}_i$ representing the *i-th* sample available is a constraint to solve the location and angle of the optimal hyperplane.

However, such hypothesized linear hyperplane is not always possible to be determined. There might be some mixed trials that preclude the trace of a linear solution separating the two classes (figure 4.2).

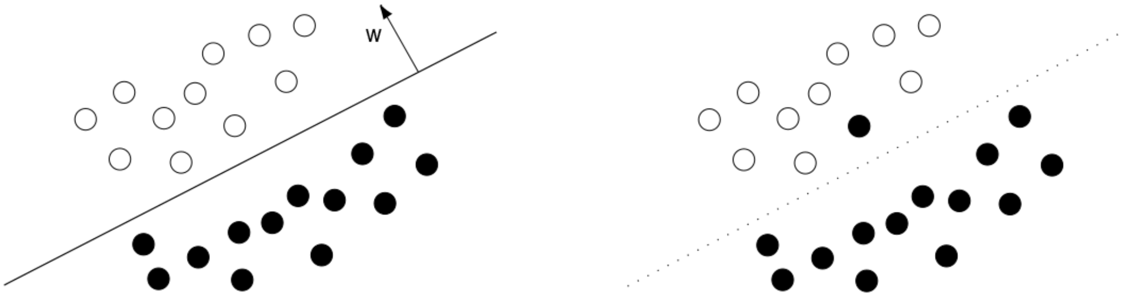


Figure 4.2: Comparison of linear separable (left) and non-separable (right) distributions of trials. Here, it is considered on two dimensions for a better illustration of the problem. At right, the presence of a single trial of the second class (black circles) close to the first (white circles) renders the impossibility of creating a linear hyperplane in order to correct separate all the trials from the two classes.

In order to solve this problem, there are two possible approaches. The first one is proposed on [94] and is called soft margin hyperplane, presenting the feasibility of creating a hyperplane to separate the training set with the minimal error and yet with the maximal distance to the nearest sample to the hyperplane. For that reason, this method introduces the so-called slack variables ξ_i and a regularization parameter *C*:

$$min_{\boldsymbol{w}\in R^n, b\in R, \xi\in R^N} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + \ C\sum_{i=1}^{N}\xi_i$$

$$subject\ to\colon\ y_i(\boldsymbol{w}.\boldsymbol{x_i} -\ b) \geq 1 -\ \xi_i\,, (for\ i = 1, \dots, N), \xi_i \geq 0$$

The second method, suggested on [95], is based on expanding the Support Vector Machine algorithm to a nonlinear classification through the application of a kernel function *K* that replaces the previous scalar products. The idea is to then generate the separating linear hyperplane in a transformed sample space *V*:

$$h:\ S \to V$$

$$K(x, y) = < h(x), h(y) >$$

The kernel function *K* may be represented by the inner product of the transformation vector *h*, as exposed above. It implies that *h* does not need to be explicitly defined. Some kernel functions are the following:

$$\text{Polynomial:} K(x, y) = (1 + < x, y >)^d$$

$$\text{Radial Basis:} K(x, y) =\ \exp\left(-\gamma\|x - y\|^2\right)$$

$$\text{Hyperbolic tangent:} K(x, y) =\ \tanh\left(k_1 < x, y > + k_2\right)$$

If the kernel used is a linear polynomial (d = 1), the resulting hyperplane on the original space *S* will also be linear. Otherwise, for higher polynomial dimensions or different functions, it will be a nonlinear separation.

So far the current section has focused on the *two* classes separation problem, while the work actually makes use of the one-class SVM. Nevertheless, the 1-SVM incorporates the both methods showed above, i.e. the use of slack variables and the space transformation through a kernel function as earlier exposed:

$$f(x) =\ sgn(\boldsymbol{w}.h(\boldsymbol{x_i}) -\ \rho)$$

$$min_{\boldsymbol{w}\in R^n, b\in R, \xi\in R^N} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{v.N}\sum_{i=1}^{N}\xi_i - \rho$$

$$subject\ to\colon\ \boldsymbol{w}.h(\boldsymbol{x_i}) \geq \rho -\ \xi_i\,, (for\ i = 1, \dots, N), \xi_i \geq 0$$

The variables ***w*** and ρ define the hyperplane on the transformed space *V*, while *h* is the function that leads to such space and is actually defined through its inner product, i.e. the kernel function *K.* There is a new parameter added into the 1-SVM equations, namely v (nu). It represents the probability, $v \in (0,1)$, of any sample from the input training set to belong to the class of clean trials. In other words, it is the probability of a sample not to be an outlier. Therefore such parameter controls the number of outliers that the 1-SVM algorithm will return as output.

For the current project, a step of 0.1 on the increment of nu has been initially employed, but further tests showed that 0.05 would be better in order not to miss changes of performances. It has been assumed that nu = 1.0 is not an option for this work, because it would mean that every single trial of the training set is actually an outlier. However, nu = 0 has been considered. Therefore for each subject, 20 values of nu with increment 0.05, $v \in [0, 0.95]$, have been considered.

Furthermore, the Gaussian (radial basis) kernel function has been used for the present work, because we hypothesize that the clean data, i.e. the samples that are not outliers, are distributed as a form of Gaussian in the original space S of dimension 68, which is the number of channels used for the EEG data collection.

$$\text{Radial Basis: } K(x, y) = \exp\left(-\gamma\|x - y\|^2\right)$$

One should determine the parameter γ depending on the input samples. Such parameter translates the width or spread of the kernel. When it is too small compared to the features, it will attenuate the samples difference and accordingly spoil the outliers' identification. Likewise, when too large, it results on a great, overestimated distance between the samples. In order to choose the optimal γ, the following simple code on Matlab was implemented:

```
n=size(train,1);
kern = zeros(n,n);
for i=1:n
    for j=1:n
        kern(i,j) = (train(i,:) - train(j,:))*(train(i,:) - train(j,:))';
    end
end
kerneloption=median(kern(:));
```

In which *train* represents the samples matrix [150 x 68], i.e. it consists of 150 vectors of samples, each one being of 68 dimensions; and the final parameter obtained - *kerneloption* - represents the inverse of γ.

4.3 Outliers: remove or penalize

After the outliers have been identified by the 1-SVM, there are two possible approaches to handle them, i.e. either remove them completely from the training set and therefore do not use them as input for the CSP algorithm or penalize them through application of certain weights.

Although the trials may have been correctly classified as outliers, it may be a better approach not to remove, but rather down-weight them. This idea is based on the fact that however an outlier sample presents different characteristics than the inliers, it can still contain some useful information for the next steps that lead to the final classification.

Therefore, in this work, both approaches, namely remove and penalize, have been considered. The outliers weighting is possible because the output of 1-SVM is actually a vector with positive and negative values, each one for a sample previously given as input. The positive values represent the samples that resulted to be inside the inlier subspace, whereas the negative values represent the outliers. Moreover, such values are proportional to the distance of the corresponding sample from the separating boundary. Thus one idea is to correlate weights to the outliers proportionally to the distances retrieved as output from the 1-SVM.

In particular sigmoid functions have been used for weighting the samples, because their attenuation is exponential with their input, which are the distances (d) of the trials to the 1-SVM boundary:

$$W(x) = \frac{1}{1+\ e^{-x}}\ , where\ x = k * d$$

Instead of applying the distance directly to the sigmoid function, a better approach is to use a multiplier in order to control the resulting spread of the function. Initially it was thought to evaluate fixed values of k for every nu and every subject. Values of different orders, e.g. k = 1, k = 10, k = 100, k = 1000, k = 10000, were compared. Afterwards, as the best results were centered on k = 1000, the performance was evaluated from k = 600 until 1400 with an increment step of 200. Finally it was found that the best improvements resulted when using k = 1200.

Subsequently it was thought that a better approach would be to estimate a value of k specifically for each subject. The constraint to it would be that a predetermined value of *d* should have a chosen output W(x).

For example, in order to guarantee W(x = k*d) = 0.01, i.e. a weight of 0.01 for the sample that presents distance *d* to the boundary created by the 1-SVM*,* one can obtain the corresponding value of *k* as following:

$$W(k.d) = \frac{1}{1+\ e^{-k*d}} = 0.01$$

$$\therefore k.d = -\ln(99) \cong\ -4.6$$

The predetermined value of *d* was thought to be obtained statistically. Given a vector *y* with the negative distances corresponding to the outliers, one can apply e.g. a 45% quantile, which roughly saying returns as output the value *d* such that 45% of the values present on the vector *y* are lower than the resulted *d.* The result is that applying the calculated *k*, one assures that 55% of the outliers will have, for this example, a weight between 0.01 and 0.50, while the other 45% will be weighted with values lower than 0.01.

This method permits a better control of the resulting sigmoid function spread and therefore was applied on the evaluation of the resulting performances. For each Matlab experiment performed, 8 different quantile functions were applied, plus a sigmoid function with *k* fixed on 1200, besides a binary function that gives output 1 for good trials and 0 for outliers, i.e. in order to completely remove them.

Moreover, five of these functions (4 quantile and the one with fixed k) were not only used to penalize the outliers, but also to give a weight to the results with positive distances, i.e. the supposed good samples. The idea is to make the weighting process more robust to outliers that may have been mislabeled as inliers because of their proximity to the boundary, besides to enhance the influence of the positive features that are located on the middle of the inliers subspace. Therefore, on the positive distances case, the more one sample is distant from the 1-SVM boundary, the better it is, and accordingly a higher weight is given to it.

5. Results

This section focuses on providing the comparable results obtained from the data set exposed on section 3. Firstly, a review of the data collection and processing methods employed on this work is presented below:

i) Multichannel EEG measurement from 68 electrodes along the subject's scalp;
ii) Signal is sampled at a frequency of 1000Hz and filter band-passed from 0.05Hz to 200Hz;
iii) 225 trials – 75 of each class (right-, left-hand, feet) – are collected during the calibration stage;
iv) The calibration data set undergoes both time and frequency subject-specific filtering;
v) Two out of three classes are selected following an estimation of best generalization capacity;
vi) The corresponding trials (150) are subjected to Laplacian filtering to remove background noise;
vii) 1-SVM is applied for the outliers identification, which are subsequently removed or penalized;
viii) Filters and Patterns are obtained from Common Spatial Pattern (CSP), which amplifies the differences between the two classes in terms of variances correlated to sensorimotor rhythms;
ix) Supervised learning of the Linear Discriminant Analysis (LDA) classifier with the CSP output;
x) The test data set – 300 trials obtained from the feedback phase – are spatially processed with the CSP filters created on (viii) and subsequently given as input to the LDA classifier;
xi) Classification output is compared to the test data set labels containing the expected results;
xii) Final BCI performance, i.e. the percentage of testing trials corrected classified, is calculated.

Along these steps, there were programming preferences to be elected and compared regarding the CSP. The first step in order to calculate the CSP filters and patterns is to estimate the covariance matrix, which can be performed through a regularization process called shrinkage so as to make the estimation stable.

The shrinkage method employed to the current project basically consists on adding one regularization parameters on the CSP's covariance matrix estimation. Such parameter shrinks the initial estimation towards the identity matrix. The reader may refer to [100] for a detailed, well-illustrated explanation of the shrinkage approach.

One could use shrinkage in order to make the CSP algorithm more robust to the presence of outliers [101]. Nevertheless, as it will be exposed, the 1-SVM can also be effective when combined with such covariance estimation option.

The second choice concerning CSP refers to the number of filters per class that it will generate as output. On section 1.4, an example of one filter per class was presented. But some projects prefer to apply three filters per class [102]. Such choice is not unique among BCI projects [102, 103, 104].

Since CSP is a method that also reduces the signal sources' spatial dimension, it may be argued that using three filters per class one could avoid excessive loss of information.

Therefore, the present section will focus on the analysis of the employment of CSP with three filters per class combined with the shrinkage covariance matrix estimation. The idea is to expose how the 1-SVM is efficient for outlier identification and for the subsequent performance improvement, even when the CSP choices that are expected to produce a better immunity to outliers are applied.

Besides the choices regarding CSP, the method proposed by the current project, which is represented by the step (vii), also presents different results according to choices to be elected. Such decisions are exposed, discussed and confronted along the following section.

5.1 SVM choices

In the section 4, the basic mathematical concepts of Support Vector Machine have been explained. However, there are still other points to be discussed and decided for its implementation, namely:

i) How can one choose the best nu (v) for 1-SVM on BCI outlier identification?

ii) Should one consider the samples from both classes as input for the 1-SVM?

5.1.1 Choice of nu (v)

The parameter nu represents the probability of any sample from the input training set to belong to the class of clean trials, i.e. the probability of a trial not to be an outlier. Accordingly, a nu choice close to 1.0 implies on an output with most of the samples labeled as outliers, while a near 0.0 value results on only a few outliers.

Therefore, although the 1-SVM is an unsupervised method to detect outliers, it requires as input a parameter that provides some information about the expected density of the artifacts present in the input set of samples.

A first thought towards this problem is the choice of a fixed nu as a threshold for every subject. One could try to hypothesize, for example statistically from previous results, the best probability value of a sample to be an outlier. This approach was tested, but showed to be worse than the cross-validation (table 5.1). It can be explained from the high variability among subjects when performing a BCI experiment [96].

The other idea is to implement a cross-validation (see section 2.3 for a review) that uses the training data in order to extract information about the current subject and to try to predict the probability (parameter *nu*) of the presence of outliers on the same training data set.

Although the CV method is not the best option for every subject; in average, it performs better than a fixed threshold. The mean error rates corresponding to each method are found on table 5.1.

CSP choices	Best nu	CV nu	nu = 0.20	nu = 0.45	nu = 0.70
Filters per class: 1 Shrinkage: No	22.442	**24.987**	26.675	27.737	28.667
Filter per class: 1 Shrinkage: Yes	23.179	**26.001**	26.775	26.771	27.421
Filter per class: 3 Shrinkage: Yes	22.100	25.879	25.375	**25.275**	27.592
Filter per class: 3 Shrinkage: No	22.612	**26.308**	26.871	27.137	29.312

Table 5.1: The average error rate of 40 subjects for cross-validation and fixed nu values. First column depicts the optimal case; while the second column, representing the cross-validation (CV) is the one presenting the lowest error rate (blue) for almost every case when comparing to the threshold approaches.

In this work, a 5-fold cross-validation was used, i.e. the training set (150 samples) was divided into five smaller subsets, each of them containing 30 samples. The algorithm evaluates the classifier performance for different values of nu using such subsets as both training and test sets through the following way:

i) The whole training data set is given as input to the 1-SVM with the current nu and the outliers are identified, i.e. the corresponding indexes are saved on a vector;
ii) Outliers that are present on a combination of 4 from the 5 are removed or penalized (down-weighted) during the covariance estimation for the Common Spatial Patterns (CSP);
iii) The final features resulting as output from the CSP are then employed to the supervised learning of the Linear Discriminant Analysis (LDA) classifier;
iv) Finally, the 5-th and remaining subset is given as input to the classifier, which will generate a final performance when comparing its results to the actual labels of these data.

The process explained above is repeated 5 times for each possible value of nu. The performances are saved. In the end, one calculates for each nu the mean of these performances; the output of the cross-validation approach is the value of nu that corresponds to the lowest error rate (1-performance) mean obtained.

The choice of 5 folds was empirically, i.e. when comparing the results with the 3-fold and the 10-fold versions, the 5-fold one presented better choices of nu. Such evaluation of which nu is actually the best is done through the computation of the final performance for every possible nu. It was possible because the evaluation process was completely offline (see section 3.3) and therefore one did not need to run different EEG experiments for each nu choice.

The cross-validation used along this work actually presents a slight difference from the one commonly used. It is particularly on the step (i) described above, since here all the training data set was given as input for the 1-SVM, when usually one would provide only the 4 of 5 subsets as input.

It was however hypothesized that such change would provide better results, because it keeps the same outlier labels – and weights, in the case of penalized samples – during the cross-validation and during the actual classification process. Otherwise, there could be the case that, for one combination of subsets, all the outliers would be actually on the remaining subset that is used for validation; accordingly it would result on a spoiled classification performance, interfering on the final nu choice.

5.1.2 Input samples

Before employing the 1-SVM for outliers' identification, one should decide what to give as input to it. Such choice concerns whether all the training samples should be applied together, or rather separated into the tasks classes, during the outliers labeling.

One important argument against providing the classes altogether is based on the possibility of an outlier for the first class not to be identified as so; because it is found on the subspace where the samples from the second class are. Therefore, the 1-SVM would identify the trials with characteristics that differ from the two classes united, and not between them (figure 5.1).

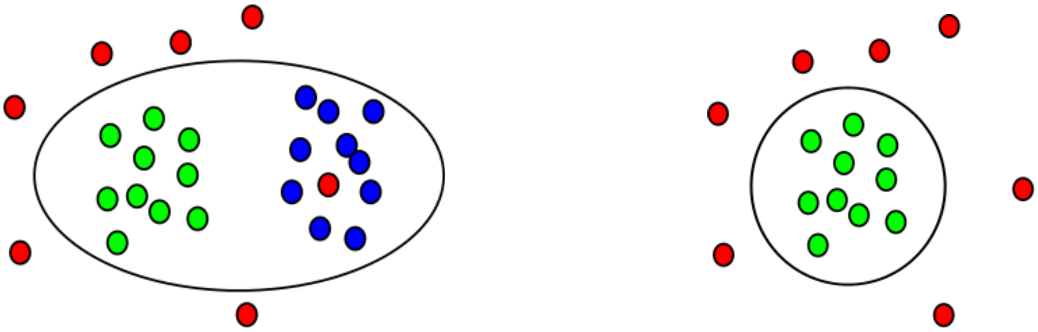


Figure 5.1: Comparison of 1-SVM boundaries for two- (left) and one-class (right) samples input. Green circles represent the first class and blue the second class. Red circles are the outliers of the first class. On left, there is one outlier which is actually inside of the 1-SVM boundary because it could not have been detected, since it is among the second class samples. However, on right, when one applies just one class as input, the 1-SVM is therefore able to create a boundary that identifies correctly all the outliers.

Nevertheless, for all the experiments ran along this project, both approaches have been considered. The reason is that, in the first attempts to compare the one- and two-class input, the results are not as different as expected. Rather, the employment of all the trials for the 1-SVM resulted on a better final classification performance of the test data in a great part of the cases (table 5.2).

	Both classes (best nu)	One class (best nu)	Both classes (CV nu)	One class (CV nu)
Filters per class: 1 Shrinkage: No	**-5.6583**	-4.8375	**-3.1125**	-2.8042
Filter per class: 1 Shrinkage: Yes	**-5.0333**	-4.5167	-2.6042	**-2.7500**
Filter per class: 3 Shrinkage: No	**-6.0750**	-5.4292	-2.3792	**-2.4958**
Filter per class: 3 Shrinkage: Yes	**-4.5167**	-3.9583	**-1.1375**	-0.6542

Table 5.2: Mean of the error-rate reduction for every possibility of CSP choice. The more the value is negative, the more the average performance improved. The first two columns (green) confront the 1-SVM input for the best nu possible, while the last two (orange) confront it for the nu chosen by cross-validation (CV). The best value of error rate reduction, for each couple of columns comparison, is depicted in blue.

It has been hypothesized that such results were caused by a particular choice made for the algorithm. When separating the classes for the SVM, the same nu was always used for both classes in order to evaluate the resulting performance. The problem is that there might be the case in which one class presents much more outliers than the other class; and if both are evaluated with the same nu for the outliers identification, it will consequently generate mislabeled outliers as output in every nu choice.

On the other hand, the decision of using the same nu when separating the classes was grounded on another hypothesis, which is the possibility of a drawback on the cross-validation efficiency, since the nu chosen from the cross-validation hardly reaches the optimal performance (section 5.1.1).

In some subjects, a final value of nu close to the optimal one is observed, while in others it resulted to be really different. Therefore electing two different values of nu through the cross-validation approach could result on a worse final performance and has not been considered on this work.

5.2 Penalizing the trials

5.2.1 Functions to handle outliers

As exposed on section 4.3, ten different functions were employed to the current project.

The first 8 functions were quantile-based, i.e. they considered the subject's specific calibration trial data set in order to select the best multiplier to apply on the sigmoid, which could be utilized either to only penalize the trials labeled as outliers (4 functions), or to also down-weight the inliers according to the distance to the boundary created by the 1-SVM (4 functions).

Accordingly, there were couples of functions with similar configurations differing on whether the inliers were also down-weighted. For example, the first function has the following configuration: the samples classified as inliers were equally weighted 1.0; and the outliers were penalized such that 55% had weights between 0.01 and 0.50, while 45% received weights below 0.01. The second function has the same weights for outliers, while the samples classified as inliers were also weighted such that 55% had weights between 0.50 had 0.99 and 45% of them had weights higher than 0.99.

These couples of functions differ between themselves on the percentage of outliers and inliers trials that would respectively present weights lower than 0.01 and weights higher than 0.99. Such percentage could be 45%, 35%, 25%, 15%. For example, the third function gives to 65% of the outliers weights between 0.01 and 0.50 and the other 35% receives weights below 0.01. While the first function, as exposed above, takes respectively 55% and 45% for the same weights. Analogously, the fourth function maintains the outlier's weights of the third function, and also apply weights higher than 0.99 to the reduced percentage of 35% instead of 45% (second function). The couple fifth-sixth takes therefore 25%, and the seventh-eighth 15%.

Finally, the ninth function employs a fixed multiplier (k = 1200) spreading factor for the sigmoid function, which is applied to both outliers and inliers. And the tenth function is the binary one, i.e. outliers are practically removed (weights = 10^-25), while all the inliers are kept with weights 1.0.

The Appendix A of the current report elucidates the Matlab code that refers to the 10 functions above described in order to provide to the reader a better explanation of how such functions actual work.

5.2.2 Evaluation methods

In order to evaluate the excellence of a function, three different mean-based methods were employed. Such methods confronted the performance results for four different artifact handling choices, namely:

v) **CV**: 1-SVM + nu chosen by cross-validation
vi) **opt**: 1-SVM + optimal, theoretical nu
vii) **remov**: removing outliers labeled through a filter method (section 2.3)
viii) **orig**: considering all the trials, without any outlier detection

The methods created to confront the subjects' (*N = 40)* performances (p), in terms of error rate (1-p), are:

4) $\sum_{i=1}^{N} \frac{(CV_i - opt_i)}{N}$

5) $\sum_{i=1}^{N} \frac{(opt_i - \min(remov_i, orig_i))}{N}$

6) $\sum_{i=1}^{N} \frac{(CV_i - \min(remov_i, orig_i))}{N}$

Accordingly, the first function measures how far the error rates obtained by the cross-validation method are from the optimal ones: therefore, the more its output is close to zero, the better it is.

The second function compares the optimal performances, i.e. the minimum error rate, with the minimum value between the removing trials approach (iii) and the original one (iv). The most negative value depicts a great decrease of the error rate.

Finally, the third function works exactly the same as the second one, but comparing the error rates from the cross-validation method (i) instead of the theoretical one (ii).

5.2.3 Evaluation results

For the evaluation of the functions performance, the following variables were interchanged:

a) Number of CSP filters (one or three filters per class);
b) Method to estimate the covariance matrix for the CSP (presence or absence of shrinkage);
c) Set of samples as input to the 1-SVM (only one or both classes).

The choices regarding CSP may vary from work to work, and therefore it had been decided to consider both cases on items (a) and (b) when evaluating the efficiency of the 1-SVM for outlier identification. However, as explained on the beginning of the current chapter, it was decided to focus the current analysis only on the case of three filters per class with shrinkage covariance matrix estimation, since it is expected to result in the highest immunity to the presence of outliers. Accordingly, the 1-SVM efficiency in this particular case would represent a great potential for the other combinations as well.

On section 5.1.2, the choice regarding (c), i.e. whether one should give as input the two classes or only one at a time, was discussed. And it was exposed how the 1-SVM approach taking both classes performed better. Therefore, the "one class" case will not be considered along this chapter anymore.

Function Index	Evaluation Method 1	Evaluation Method 2	Evaluation Method 3
#1	3.3792	-3.0750	0.3042
#2	3.1958	-2.9042	0.2917
#3	3.0875	-2.9917	0.0958
#4	3.2875	-2.7000	0.5875
#5	3.6167	-2.9583	0.6583
#6	2.6042	-2.5708	0.0333
#7	2.9167	-2.9708	-0.0542
#8	2.7417	-2.6667	0.0750
#9	2.9875	-2.9542	0.0333
#10	**3.7792**	**-3.3708**	0.4083

Table 5.3: Function evaluation for shrinkage covariance matrix, CSP with three filters per class and both classes as 1-SVM input. The result in blue represents the best performance for the optimal nu (method 2), while in red is depicted the worst result for method 1. Both correspond to the binary function (#10).

On table 5.3, the election of the function to be emphasized in blue was based on the best output of the method (2), i.e. the most negative value. The reason for such decision is that this method would actually represent the real potential of the 1-SVM, whereas it evaluates the error rate reduction for the best nu.

However, as exposed on the same table, the binary function presents the worst result for the evaluation method 1; which symbolizes the average distance between the best, theoretical nu and the one elected by the cross-validation. It means that, for a real-time experiment, in which the best value for nu is unavailable, the binary function, i.e. the completely remove of the trials labeled as outliers by the 1-SVM, would actually result in one of the worst approaches to be applied.

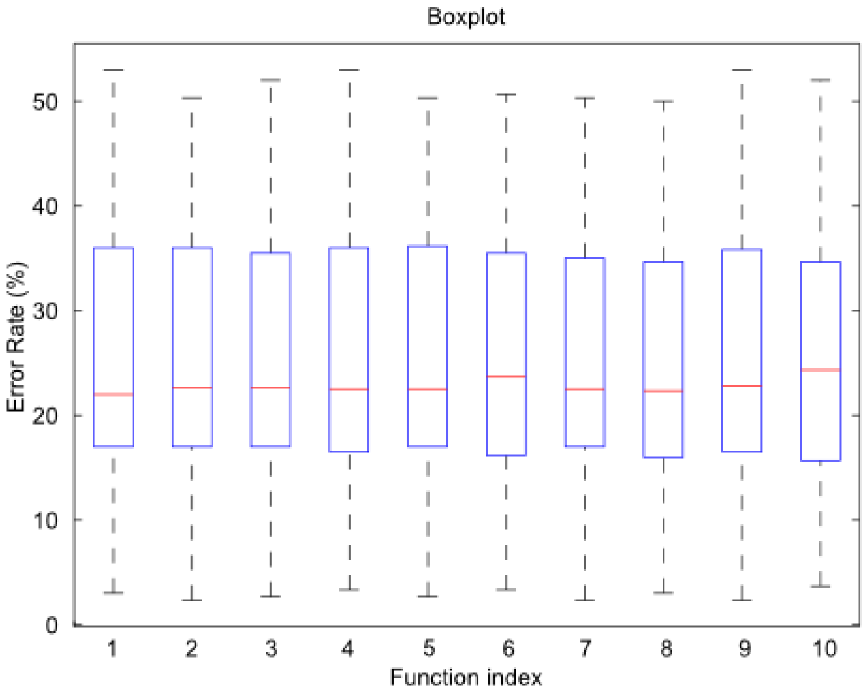


Figure 5.2: Boxplot comparing the error rates generated after applying 1-SVM with cross-validation for the 10 functions. The mark in red for each box is the median, while the blue edges of the box represent the 25th and 75th percentiles. The whiskers prolong until the most extreme error rates, i.e. the highest and lowest.

The figure 5.2, whereas depicting the median of error rates for every function applied with cross-validation, reinforces the fact that the binary function actually presents the worst average performance for the forty subjects. It can be conclude from the position of the red mark that symbolizes the median of the 10th box, which is higher than the others, i.e. it presents a higher error rate and accordingly a worse performance.

Furthermore, the same figure depicts how the other nine functions presented a similar median of error rate. It can be explained from the fact that such weighting functions were also applied during the cross-validation for the nu selection. Therefore, the effect of the nu choice, i.e. the election of the probability of a sample from the training data set to be an outlier, is actually influenced by the function chosen to penalize the trials. And they are mutually compensated until reach an optimal point. For example, the employment of a function that applies weights closer to zero could result on a cross-validation election of a lower nu, i.e. less samples would be labeled as outliers in order to compensate the stronger penalizing weights.

Therefore, for the following performance comparison, the function #1 is employed, since no particular function presented a special final performance in order to be elected as the best one to be analyzed.

This function works as following: the samples classified as inliers were equally weighted 1.0; and the outliers were penalized such that 55% had weights between 0.01 and 0.50, while 45% received weights below 0.01. The weighting function is a sigmoid and its outputs are proportional to the distance from the boundary created by the 1-SVM according to the nu elected (either the theoretical or the CV one).

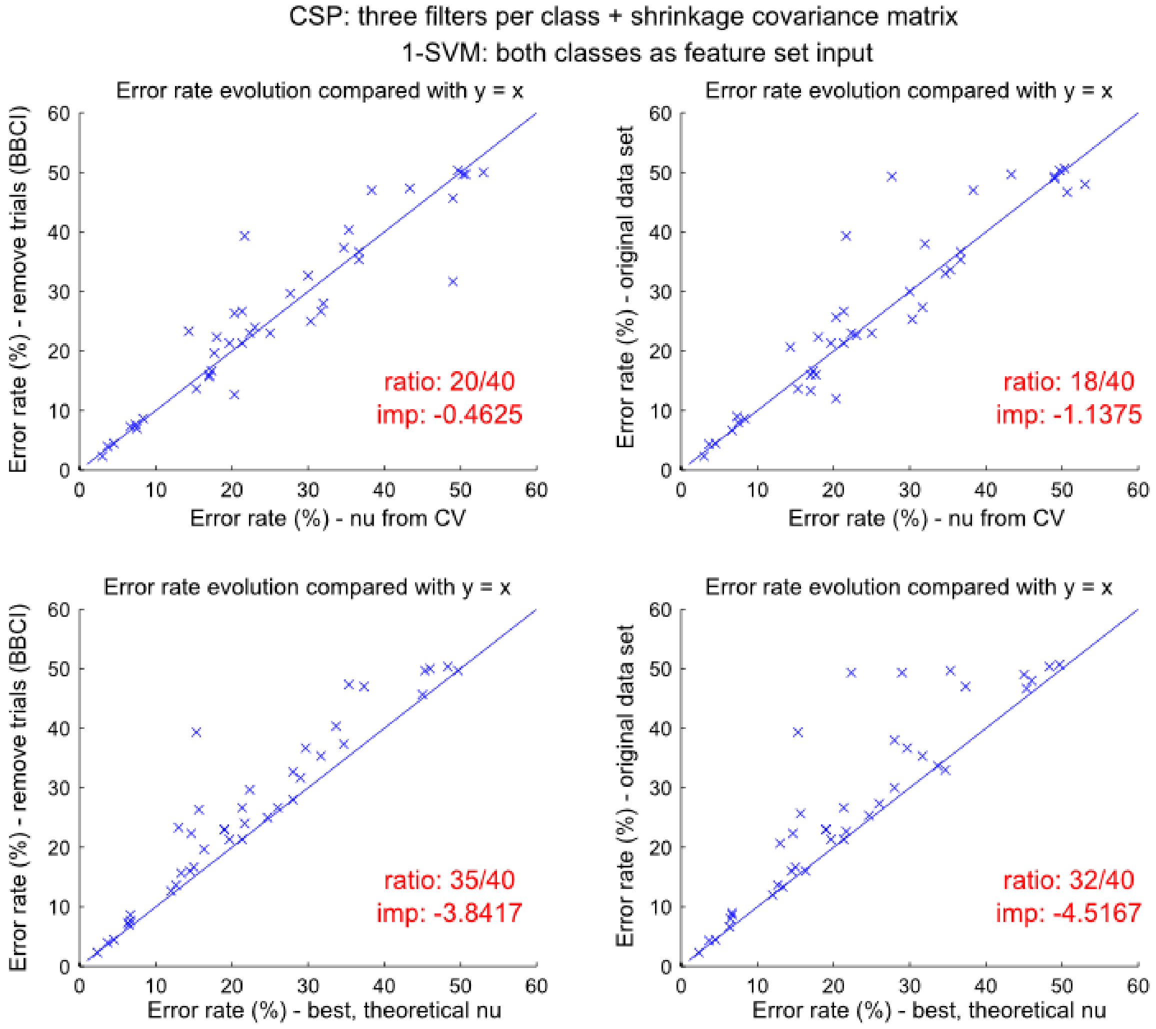


Figure 5.3: Performance comparison for shrinkage covariance matrix, CSP with three filters per class and both classes as 1-SVM input. Each cross on a graph represents a subject, and its spatial location follows the registered error rate for the method of reference (y-axis) compared with the proposed method (x-axis). For the graphs on the left side, the y-axis stands for a filter method for artifact handling (see section 2.3). Each graph presents a straight line $y = x$ that represents a visual threshold illustrating which subjects actually presented an improvement on the performance, i.e. the crosses located above this line.

The values in red numerically represent the improvement observed on the presence of 1-SVM compared to the current method of reference. “Ratio” represents the number of subjects that strictly presented a better performance; while “imp”, which stands as a diminutive for “improvement”, shows the attenuation on the error rate in terms of mean of all subjects.

5.3 Discussion

Initially, the present section discussed the choices regarding the 1-SVM. It exposed how the cross-validation had showed to be the most efficient method in order to elect the nu parameter (probability of a trial to be an outlier) for the 1-SVM (table 5.1). Besides, it also presented a comparison of error rate reduction for one or both classes as input for the SVM (table 5.2), in which the both classes approach performed better.

Subsequently, following the SVM choices above exposed, the proposed functions to handle the outliers labeled by the 1-SVM were analyzed. One notices that the binary function had performed better according to the method 2, which evaluates the performance improvement for the theoretical nu. It could indicate a contradiction of the theory, on section 5.1.2, that outliers can contain some useful information for the next steps leading to the final classification.

However, the functions' comparison also elucidates that the binary function presents a great difference between the final performances obtained from the optimal nu and the cross-validation one (table 5.3, figure 5.2). Therefore it has been hypothesized that the 1-SVM may be able, on an ideal case that cannot be practically reproduced, to identify the outliers to be fully removed; but for the cross-validation nu selection, it could be a safer alternative to penalize the trials and therefore retain their information for a better classification.

Along the performance graphs exposed on figure 5.3, it is possible to conclude that the 1-SVM has the potential to reduce the error rate caused by the final classification on a BCI-based system. The 4th graph, i.e. the one comparing the original data set error rate with the employment of the optimal nu for the 1-SVM, clearly elucidates how the proposed method could effectively increment the performance. It is translated on the number of subjects that presented a strict improvement: 32/40 (80%).

Moreover, from a further visual inspection of the same graphs, one sees that the other 20% of subjects resulted on a practically equal performance. It is another positive mark for the 1-SVM potential; since its ideal employment, i.e. with the optimal nu, could guarantee at least the stabilization of the current performance for every single subject, besides offering an 80% probability of reducing the error rate, according to the current study.

A similar result can be found when comparing the error rate reduction produced by the filter method of removing trials previously available for the BBCI group (section 2.3) with the 1-SVM method proposed by the current project (figure 5.3, first graph on bottom). No subject in such comparison presented decrease of performance. Rather, 87.5% of the subjects (35/40) resulted on a strict performance improvement.

However, one can also notice that, when applying cross-validation, the ratio of subjects presenting an absolute performance increment compared to the original data set (figure 5.3, second graph on top) reduces to 45%; moreover, a subject could present up to 10% reduction on performance while employing the 1-SVM for outlier detection.

Nevertheless, even though the current case applies three filters per class with shrinkage covariance matrix estimation, which was expected to represent the best immunity against outliers; the 1-SVM method for outlier handling combined with cross-validation as nu selection was still able to present a reduction on the error rate of approximately 20% for two particular subjects: an expressive improvement for a real case.

The next chapter will expose further analyses regarding one of these two subjects.

6. Further analyses

6.1 Subject election

For this section, the subject labeled as VPjh_08_05_27 among the dataset available for the project is taken into consideration. Such decision is based on the fact that this subject presented a considerable error rate improvement – a reduction of 17,66% – after the 1-SVM application within a combination of CSP conditions from which one could expect a higher immunity to the presence of outliers, i.e. a change not so significant. The CSP algorithm was performed with three filters per class and applying the shrinkage covariance matrix estimation. Moreover, the current improvement depicted was for the cross-validation method for the *nu* selection; which, although not optimal, was able to guarantee the performance improvement exposed.

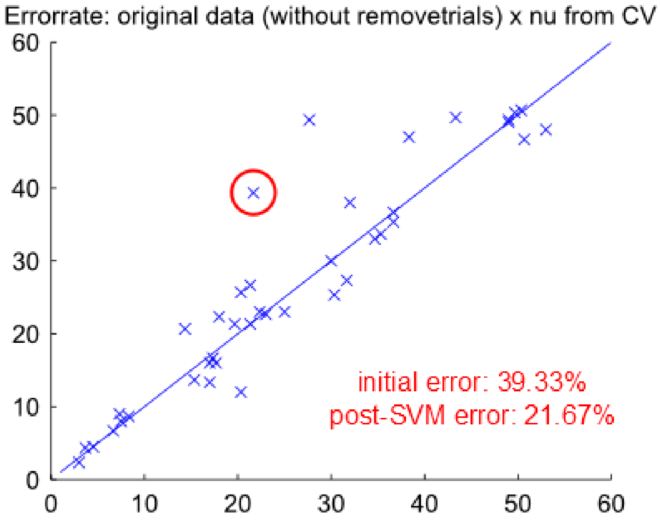


Figure 6.1: Performance characteristics of the subject chosen (red circle) for further results' analyses. The graph depicts the error rate reduction after the application of 1-SVM. The comparison is performed within the context of shrinkage covariance matrix, CSP with three filters per class and both classes as 1-SVM input.

Nevertheless, one could ask why not to consider instead the other subject depicted on figure 6.1, whose performance increment was even better (initial error 49.33%; post-SVM error 27.67%) than the chosen one. However, this subject alternative presents an initial performance that can be approximated to the completely random performance case (50%). Therefore, even though such error rate reduction may seem at a first impression a good sign, it was hypothesized that it could also result in an unreliable analysis.

6.2 Varying the weights

In an attempt to understand better what happens during the outliers processing, a method to individually analyze the influence of each trial from the calibration data set on the final performance is employed. It consists of taking a weight configuration as starting point, e.g. all trials equally weighted 1.0; and then analyze how the error rate changes according to the weight variation of a single trial at a time. More precisely, this section will expose four particular cases, namely:

a) All trials equally weighted 1.0, i.e. without the application of 1-SVM
b) The same case as (a), but for no shrinkage on the CSP covariance matrix estimation
c) Weights proportional to the 1-SVM output with the optimal, theoretical nu ($nu_{opt} = 0.50$)
d) Weights proportional to the 1-SVM output with $nu = nu_{opt} - 0.05$

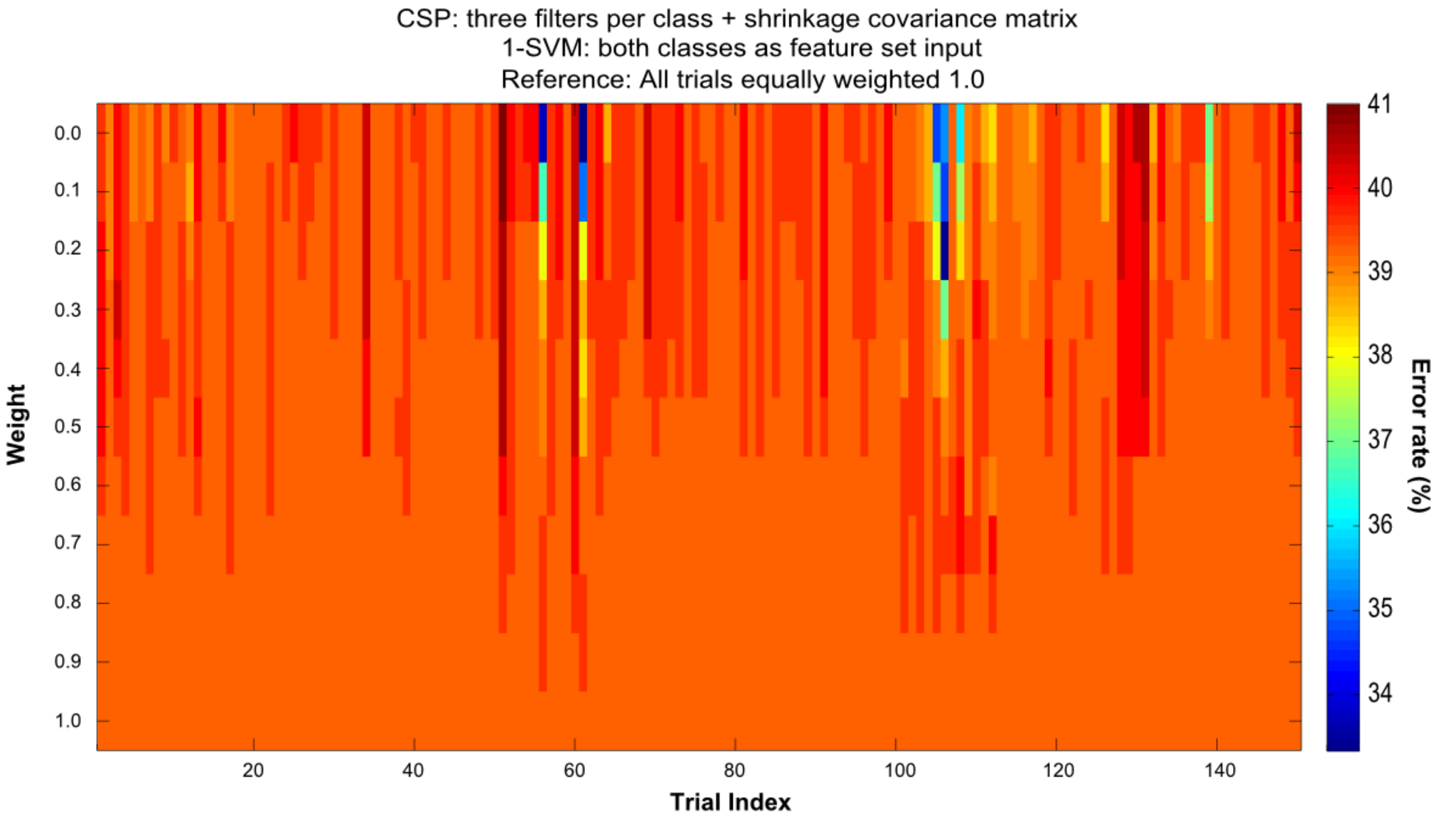


Figure 6.2: Error rate variance following the weight's change of each trial on the calibration data set. The chosen configuration for analysis is the CSP with three filters per class combined with shrinkage covariance matrix estimation, besides employing both classes as 1-SVM input. The axis *x* stands for the trials (150 in total); while the *y* represents the weights, from 1.0 until 0.0 with a step of 0.1. Those are the two variables for the output, i.e. the error rate, which is depicted with different colors according to the colorbar on the right. For the analysis, all trials were initially weighted as 1.0, as if there was no SVM information. It is translated on the lowest row of the graph, corresponding to the weight 1.0, in which the color is uniform for every trial (x axis), i.e. the performance is the same (error rate = 39.33%).

As one could expect, for the combination of CSP with three filters per class and shrinkage covariance matrix estimation, the resulting performance span depicted on figure 6.2 is not large. Nevertheless, one can see that changing only one particular trial can lead to a considerable change on the final performance. For instance when removing the trial #61, the error rate is 5% reduced. Moreover, it also illustrates how some trials are important for the calibration set, for example the absence of the trial #51 results on a 2% worse error rate.

Another important point to be examined on the graph regards the performance change along a fixed column, i.e. for a given sample. One could suppose that, if a trial is a real outlier, the more such trial is down-weighted, the more the corresponding error rate would be reduced. However, that is not true. Instead, there are several columns on the graph depicted above showing that there may be an optimal weight for the sample taken into account, after which the performance becomes worse once again. For example the trial #106 that presents the lowest error-rate, translated into the color dark blue, when it is weighted as 0.2; and when its weight becomes either 0.1 or 0.0, the error rate increases. This fact could reinforce the assumption to utilize penalizing weights instead of completely removing a trial that was labeled as outlier.

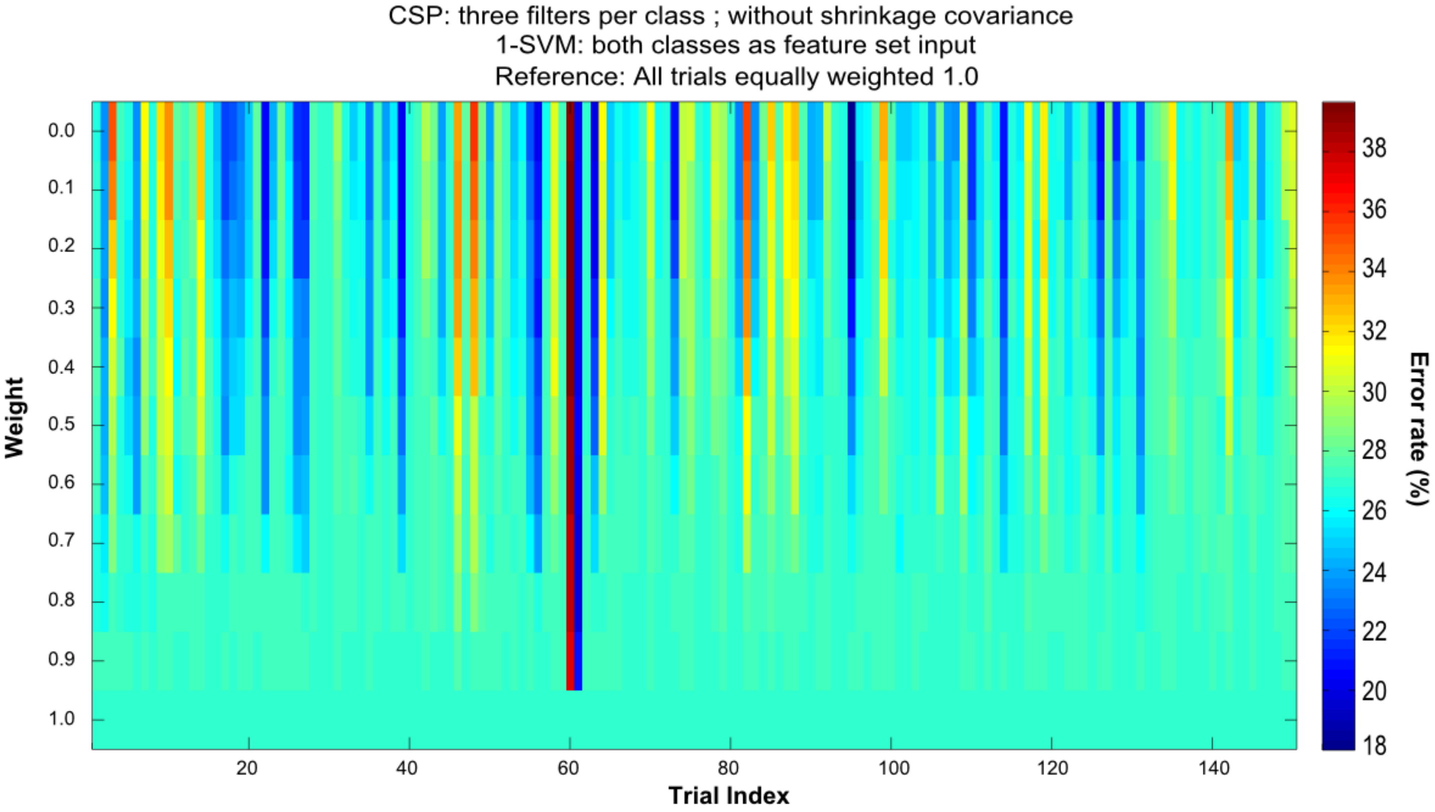


Figure 6.3: Error rate variance following the weight's change of each trial on the calibration data set. The chosen configuration for analysis is the CSP with three filters per class without shrinkage covariance matrix estimation, besides employing both classes as 1-SVM input. The axis *x* stands for the trials (150 in total); while the *y* represents the weights, from 1.0 until 0.0 with a step of 0.1. Those are the two variables for the output, i.e. the error rate, which is depicted with different colors according to the colorbar on the right. For the analysis, all trials were initially weighted as 1.0, as if there was no SVM information. It is translated on the lowest row of the graph, corresponding to the weight 1.0, in which the color is uniform for every trial (axis *x*), i.e. the performance is the same (error rate = 27%).

When comparing the figure 6.3, which is without shrinkage covariance matrix, with the previous one that employs such method; there are some particular, important resulting points to be noted and discussed. Initially, one can see how the distribution of colors on the graph has changed. It illustrates the consequent higher sensitivity to small changes on the data set given as input to the covariance matrix estimation. Such fact is also translated into the span of error rate represented on the colorbar. While with shrinkage matrix, the limits of error rate observed were 34% and 41%; in the current case, it became between 18% and 38%.

Such span also illustrates another resulting difference between these methods, namely the error rate for the reference case, in which all trials are equally weighted as 1.0. While in the last case it was 39.33%, now it resulted on 27%. It could be an indication that the employment of shrinkage covariance matrix not always is the best alternative to avoid the interference of outliers.

Finally, another interesting observed result from the visual inspection of this graph is the actual linear performance change along the weights. On the previous case (figure 6.2), it was showed how a trial could have an optimal weight different than zero, for example 0.2. In this case, however, either the best or the worst performances were present exactly in the case in which the corresponding trial was removed (depicted on the top row of the graph, which corresponds to weight 0.0).

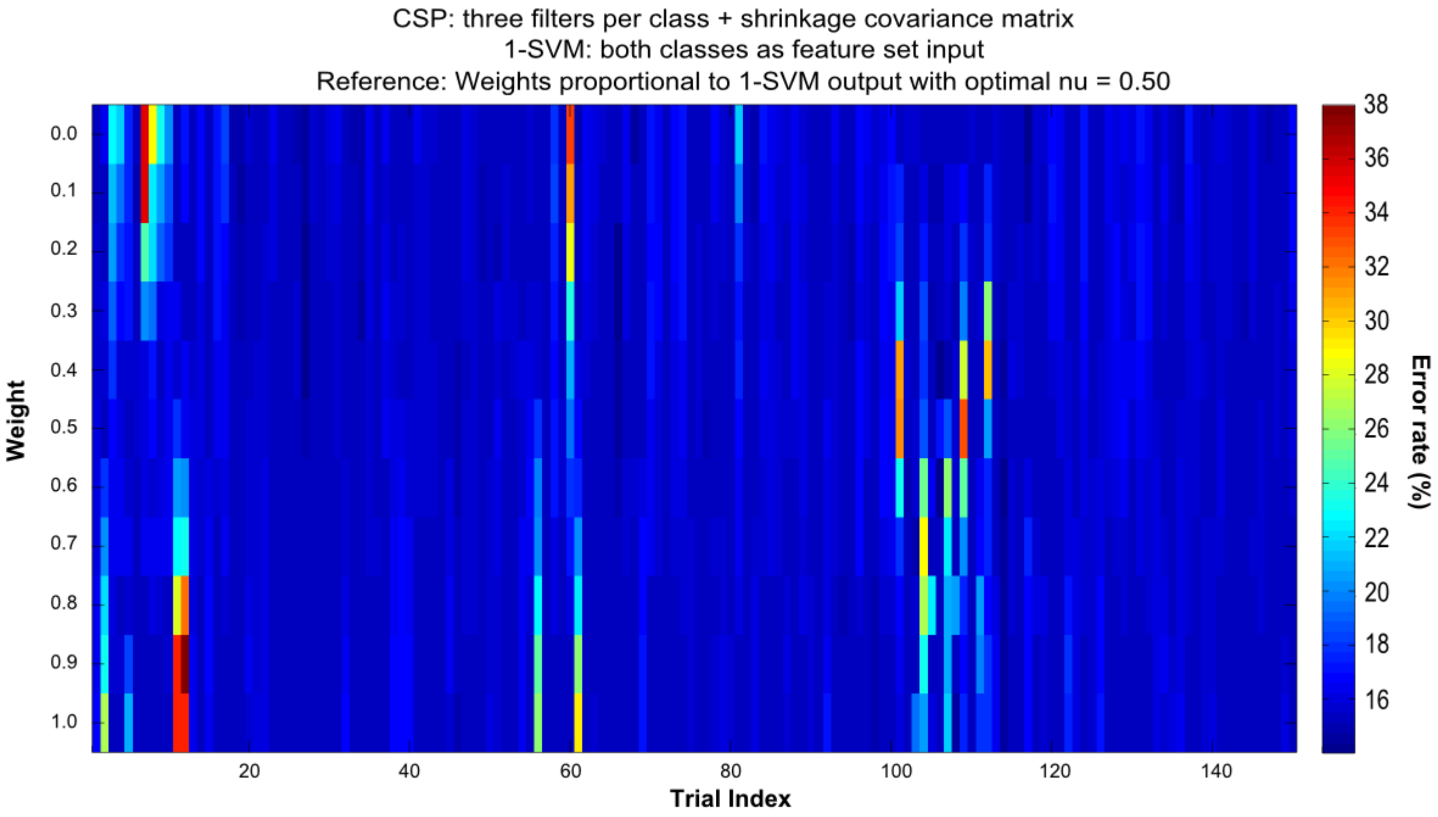


Figure 6.4: Error rate variance following the weight's change of each trial on the calibration data set. The chosen configuration for analysis is the CSP with three filters per class with shrinkage covariance matrix estimation, besides employing both classes as 1-SVM input. The axis *x* stands for the trials (150 in total); while the *y* represents the weights, from 1.0 until 0.0 with a step of 0.1. Those are the two variables for the output, i.e. the error rate, which is depicted with different colors according to the colorbar on the right. For the analysis, all trials were initially weighted according to the 1-SVM output for the optimal, theoretical *nu* observed for this particular subject and function. The corresponding error rate is 15.33%.

The expected stability for a shrinkage covariance estimation is conserved on the graph exposed above while concerning the whole set of trials, i.e. when comparing with the figure 6.3, it presents a notable uniformity of color along the graph that elucidates a higher constancy. Nevertheless, one can also notice on the figure above how the performance span, when varying the weight of only a single trial at a time, can be elevated in this case. It is translated into the error rate from 15% to 38%.

In particular there are two trials which when completely removed lead to an error rate higher than 35%, i.e. a reduction of 20% on the performance for the absence of a single trial that therefore can be classified as an indispensable inlier for the calibration set. In the other hand, there are two outliers previously penalized that, when taking again into account with weight either 0.9 or 1.0, also cause an analogous performance reduction. Both cases are depicted by red and dark red colors on the graph.

And similarly to the phenomenon of the optimal weight explained for the figure 6.2, in which such weight could present some value different than 1.0 or 0.0, in the current figure the same occurs; but related to the worst weight, i.e. the one that results on the worst performance. It happens in particular for the trial #110, which reaches its maximum error rate when down-weighted by 0.5. And it is also visible for #101 and #113.

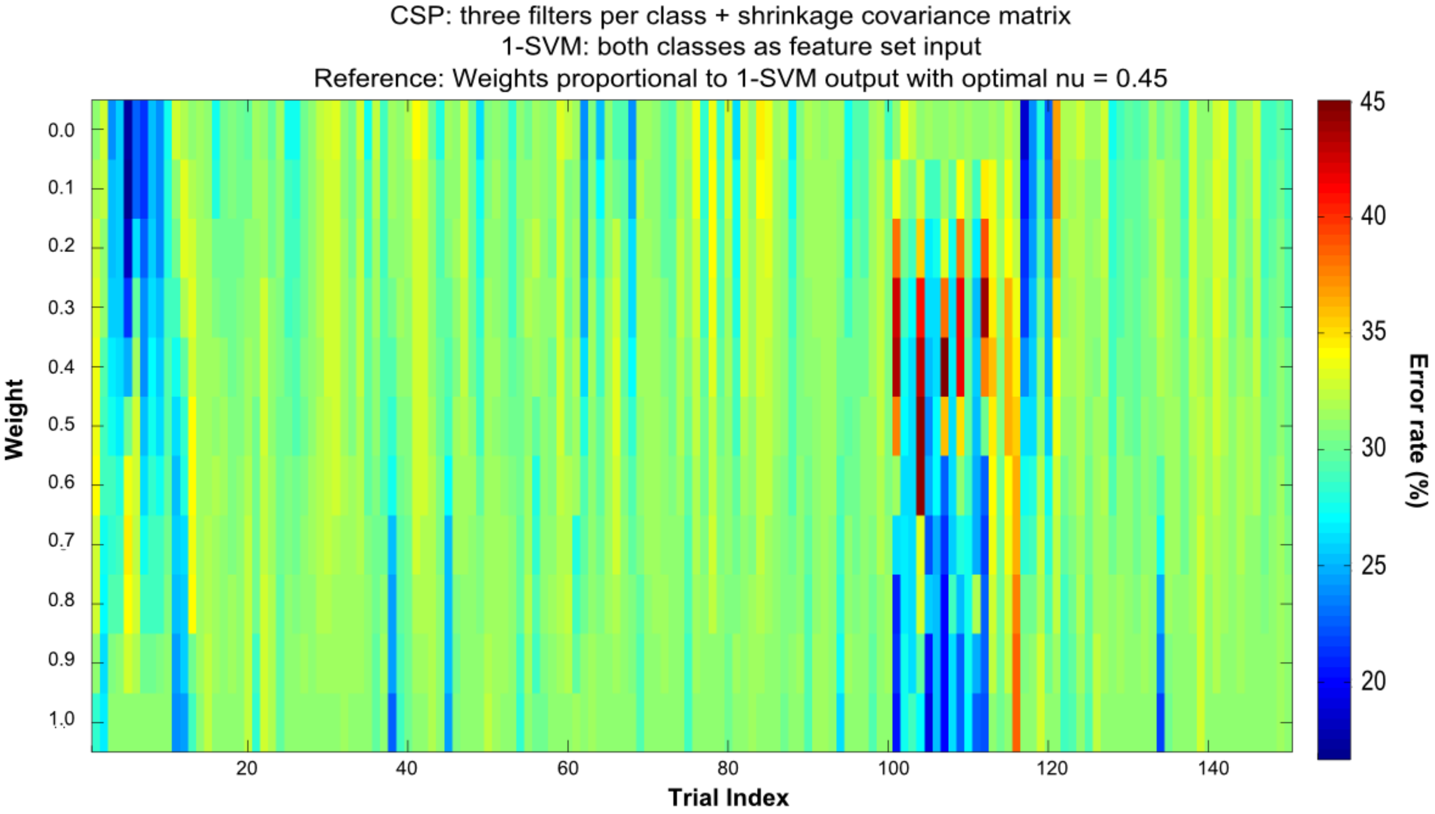


Figure 6.5: Error rate variance following the weight's change of each trial on the calibration data set. The chosen configuration for analysis is the CSP with three filters per class with shrinkage covariance matrix estimation, besides employing both classes as 1-SVM input. The axis *x* stands for the trials (150 in total); while the *y* represents the weights, from 1.0 until 0.0 with a step of 0.1. Those are the two variables for the output, i.e. the error rate, which is depicted with different colors according to the colorbar on the right. For the analysis, all trials were initially weighted according to the 1-SVM output for $nu = nu_{opt} - 0.05 = 0.45$.

The resulting error rate in this case is 31%, while the error rate corresponding to only a *nu*'s step different was 15.33%, i.e. already reaching the optimal one. This fact illustrates a large difficult when applying 1-SVM for outliers identification, namely the *nu* selection and its consequent reproduced sensitivity on the final BCI performance. In the current work, the proposed method to face this decision was the cross-validation, which showed to perform better than a fixed threshold (section 4.3.1), but still not optimal and sometimes considerable far from it, as depicted on the performance graphs (section 5.2).

Therefore it was decided to analyze better the influence of some trials that are related to a considerable performance change as depicted by the darkest colors, either blue or red, on the figure above. In particular the trials depicted by dark blue, illustrating a great performance improvement, will be taken into account.

Initially the outliers, i.e. the ones resulting on a better performance when completely removed (weight 0.0), namely the trials #5 and #117. Subsequently, the previously penalized inliers (trials #101, #105 and #107) will be relocated to the data set, i.e. employing weight 1.0 to then, according to the optimal performance depicted on figure 6.5. Finally, both cases will be together analyzed, i.e. removing the outliers and including the inliers, in order to evaluate what is the resulting performance. The results are exposed through tables on the next page.

	Initial	Removing #5	Removing #117	Removing both
Weight #005	1.0	0.0	1.0	0.0
Weight #117	1.0	1.0	0.0	0.0
Error rate	**31.00%**	**17.00%**	**18.67%**	**14.67%**

Table 6.1: Analyze of error rate evolution according to the removal of trials suggested from figure 6.5. The first column depicts the initial weights of those trials and the corresponding error rate. Second and third columns represent the removal of only one of them, resulting on a considerable increment of performance. Forth and last column remove both trials together, supposing that the previously observed improvements could be supplementary. Indeed, the final performance resulted to be even better (error rate of 14.67%).

	Initial	Including #101	Including #105	Including #107	Including all
Weight #101	0.0001	1.0	0.0001	0.0001	1.0
Weight #105	0.0102	0.0102	1.0	0.0102	1.0
Weight #107	0.0016	0.0016	0.0016	1.0	1.0
Error rate	**31.00%**	**20.33%**	**19.00%**	**20.00%**	**16.67%**

Table 6.2: Analyze of error rate evolution according to the inclusion of trials suggested from figure 6.5. The first column depicts the initial weights of those trials and the corresponding error rate. Second, third and forth columns represent the inclusion of only one of them, resulting on a considerable increment of performance. Fifth and last column include all the trials together, supposing that the previously observed improvements could be supplementary. The final error rate resulted to be indeed better (16.67%).

	Initial	Removing outliers	Including inliers	Both methods
Weight #005	1.0	0.0	1.0	0.0
Weight #117	1.0	0.0	1.0	0.0
Weight #101	0.0001	0.0001	1.0	1.0
Weight #105	0.0102	0.0102	1.0	1.0
Weight #107	0.0016	0.0016	1.0	1.0
Error rate	**31.00%**	**14.67%**	**16.67%**	**30.33%**

Table 6.3: Analyze of error rate evolution according to the removal and inclusion of trials suggested from figure 6.5. The first column depicts the initial weights of those trials and the corresponding error rate. Second column depicts the complete removal of both trials #005 and #117, while on third column trials #101, #105 and #107 are included with maximum weight (1.0). Forth and last column employs both methods together, supposing that the previously observed improvements could be supplementary. However, the methods combined resulted on a worse performance (30.33%), almost as bad as the initial.

While one could expect an additional property of improvements on performance, as actually illustrated on tables 6.1 and 6.2; the last table, however, depicts that it is not always a possible approach to select the samples' weights. It has been hypothesized that the problem lies on the assumption that the trials could have the same impact on the performance for different configurations of reference. In other words, e.g. after removing the trials #005 and #117, the varying weights figure has inherently changed, since its reference has been modified, i.e. the weights for each trial are not the same. Therefore the ideal weight correction for each trial seems to intrinsically depend on the current configuration; which confers a high sensitivity to the 1-SVM approach for outlier identification and to the weighting function's employment.

6.3 CSP Filters angles

In order to understand better the sensitivity previously exposed, the current section focus on the analysis of how the CSP filters change when removing or adding some trials; in particular the trials used on the last section. Namely, the removal of #005, #117; and the inclusion of #101, #105, #107.

Such analysis is initially performed with the Matlab "subspace" function, which basically returns as output the angle between the subspaces defined by the columns of the two matrices given as input. Each of these matrices corresponds has [68x6] dimensions, since 68 channels were applied to collect the data (section 3) and each column is a different filter (configuration of 3 filters per class; two classes selected out of three).

	Removing #005 and #117	Including #101, #105 and #107
Original Reference (nu = 0.45)	**21.896**	**77.215**

Table 6.4: Angle comparison (in degrees) of subspaces defined for three configurations by the CSP filters. Namely, the original data set of reference (nu = 0.45); the one regarding removal of trials #005 and #117; and for the inclusion of trials #101, #105 and #107.

The angles depicted on table 6.4 illustrate how the CSP filters changed after the removal and inclusion of respectively two and three trials. In particular for the case of inclusion, the final angle was extremely great (77.215 degrees); which depicts a great distance between the subspaces.

For a more detailed illustration, the table 6.5 below elucidates the angles between each filter at a time.

	Angle: original vs. removal	Angle: original vs. inclusion
1st filter	74.582	85.273
2nd filter	52.674	53.486
3rd filter	19.539	85.830
4th filter	8.805	5.943
5th filter	14.026	10.665
6th filter	2.904	7.457

Table 6.5: Angle comparison (in degrees) of each CSP filter between original set (nu = 0.45) and either trials removal (#005 and #117) or inclusion (#101, #105 and #107).

One can notice from the values exposed above on table 6.5, that the most expressive difference for both cases belongs to first three filters. When employing CSP with three filters per class, it divides the filters with indices 1-3 for the first class, while 4-6 are given to the second class. For the particular subject, such classes were respectively left arm and feet. Therefore, the results show that the left arm class was most affected after either the removal or the inclusion of data.

#005	#117	#101	#105	#107
Feet	Left	Left	Feet	Left

Table 6.6: Depiction of the class (left arm or feet) to which each trial either removed or included belongs. The trials used for the removing analyses are depicted in blue, while those for the including are in green.

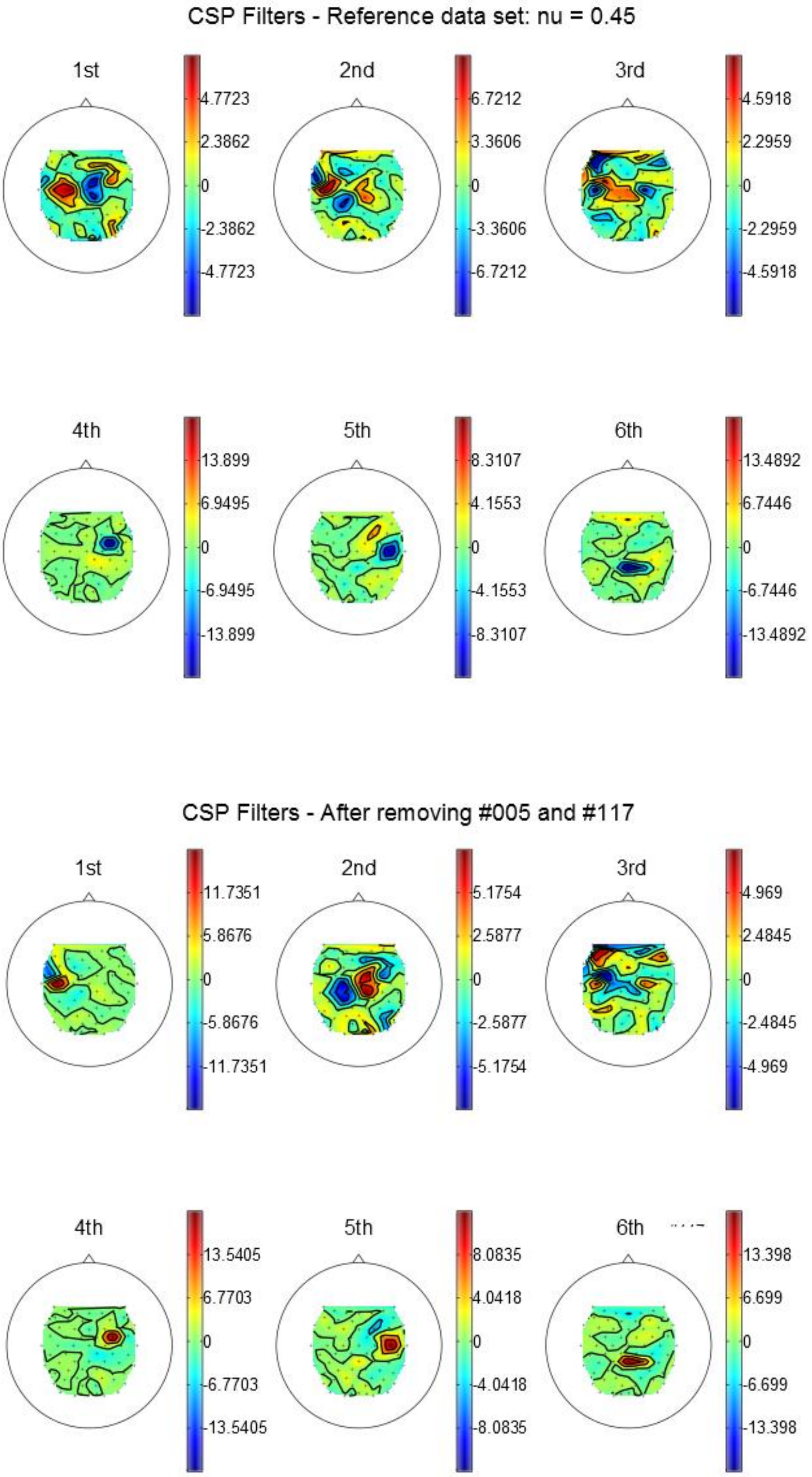
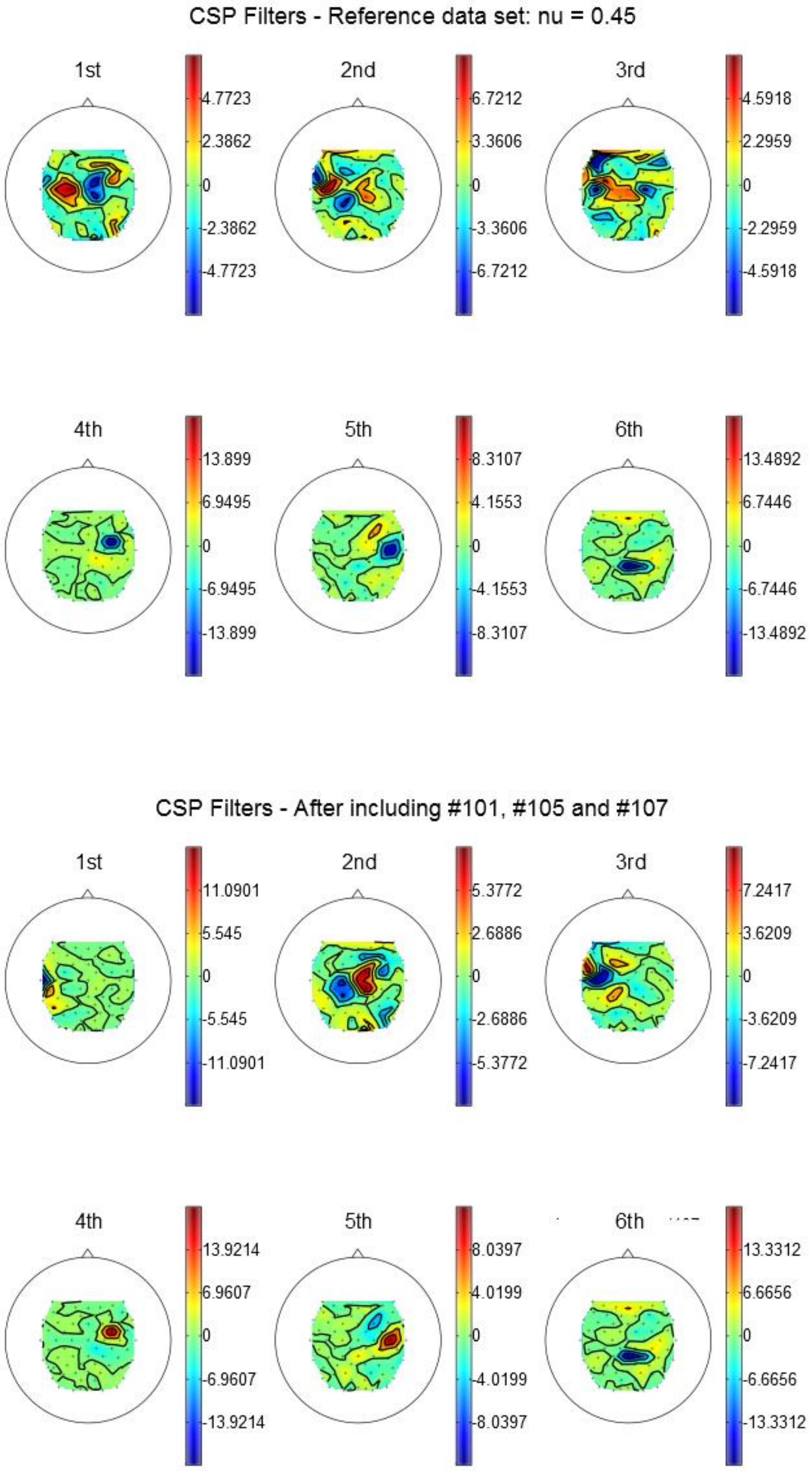


Figure 6.6: Comparison of CSP filters plots for reference data set (nu = 0.45) and the subsequent removal of trials #005 and #117. The most affected filters topologies are those regarding to the first class (left arm), i.e. the filters with indices from 1 to 3. Besides the change on topology, in particular the first filter also presented a considerable variance on the amplitude's limits (initially 4.7723, afterwards 11.7531).



CSP Filters - Reference data set: nu = 0.45

CSP Filters - After including #101, #105 and #107

Figure 6.7: Comparison of CSP filters plots for reference data set (nu = 0.45) and the subsequent inclusion of trials #101, #105 and #107. The most affected filters topologies are those regarding the first class (left arm), i.e. the filters with indices from 1 to 3. Besides the change on topology, in particular the first filter also presented a considerable variance on the amplitude’s limits (initially 4.7723, afterwards 11.0901).

6.4 Review and final considerations

This section presented further analyses for a particular subject, labeled as VPjh_08_05_27 on the data set, which had presented a reduction of 17.66% on its error rate after the 1-SVM application. It is a considerable improvement since the CSP algorithm was with 3 filters per class and shrinkage for the covariance matrix estimation, from which one could expect a higher immunity to the presence of outliers.

Therefore, in order to evaluate the influence of a single trial on the final performance, graphs correlating the error rate with the weight for a given trial were exposed on section 6.2 for different sets of reference. Some important facts were observed and discussed from the results exposed, namely:

a) The best achievable performance does not necessarily lies on the extreme weights (0.0 or 1.0), but can rather be found for intermediate weights, for example weight = 0.2 (figure 6.2);
b) The absence of shrinkage on the covariance estimation resulted, as expected on a greater variance of performance, translated on the elevated transition of colors along the graph (figure 6.3);
c) Only one step (0.05) change on the value of *nu* may result on a great lost of performance (e.g. 15%, as when comparing figures 6.4 and 6.5), which illustrates a high sensitivity on the 1-SVM output and accordingly an obstacle for the cross-validation to achieve an efficient decision;
d) Actually one single trial, depending on the reference set, can considerably influence on the final BCI performance, whereas 15% of both improvement and reduction were registered on figure 6.5;
e) The combination of removal and inclusion of trials that separated had improved the performance (tables 6.1 and 6.2) may lead to an actual increment of the error rate. It depicts the inherently dependence to the set of reference that the efficient weighting process present.

In order to better understand the point (e), section 6.3 exposed analyses regarding the change on position of the subspaces defined by the CSP filters. It was evaluated while either removing two trials (#005, #117) or including three trials (#101, #105, #107); both methods were based on the performance improvement depicted on tables 6.1 and 6.2.

In particular, table 6.5 showed how the filters regarding the first class (left arm) had underwent a great subspace change when either removing or including trials, which was translated into the angle related to the original set. The angle for the first filter specifically was very high (74 and 85 degrees).

It is therefore hypothesized that such considerable change on the subspace is responsible for the worse performance depicted on table 6.3. The problem may lie on the fact that the CSP filters obtained from the calibration data set are afterwards applied to the training data set. If such filters already undergo a great change for both cases, i.e. the removal of trials #005, #117 and the inclusion of #101, #105, #107; when combining such methods the resulting filters may actually not be efficient anymore and interfere on the final classification.

Finally, table 6.6 depicts the correspondence between each trial and the class (left arm or feet) performed. Although one could expect, for example on the case of trials removing, that both trials should belong to the left arm, since these filters were highly affected after their removal; the trial #005 is actually related to the feet. It is however hypothesized that while the left arm trial (#117) was deciding for the filter definition, the trial regarding the feet not. Nevertheless, #005 alone could improve the performance in 14% (table 6.1). It suggests that classifier learning and CSP filters determination are not always correlated, although both can interfere on the final BCI performance and consequently improve it.

7. Conclusion

The current work proposed a novel wrapper method for the identification of outliers on the EEG signal that is employed as non-invasive method for Brain-Computer Interface applications. In particular, it is combined with other processing methods such as band-pass frequency filter, Laplacian filter, Common Spatial Pattern.

The initial idea was to label the outliers from the calibration data set and either remove or penalize them before the covariance matrix estimation for the generation of CSP filter; and before the LDA classifier learning, which may be high affected by such outlier trials, as exposed on figure 2.1.

For that purpose, data from forty BCI-naïve subjects, i.e. with no previous experience on BCI experiments, were used. All the subjects had no neurological disease, since its presence could interfere on the final results. The data collection consisted of two different phases: calibration (150 trials) and feedback (300).

Actually the calibration phase had initially 225 trials, 75 belonging to each of the three possible classes (right arm, left arm, feet). However, only two classes were selected according to a cross-validation based estimation of the best generalization capacity, i.e. the two classes which would supposedly result on a more robust classification process on the feedback phase.

Another point regarding the data collection concerns the online phase. The present work could not profit from the feedback advantages and the consequent possibility of task correction by the subject, since the data had been previously collected for another project.

Nevertheless, although an offline analysis is not ideal, it was hypothesized to be a first step to demonstrate the effectiveness of the proposed method for outlier identification on the BCI performance improvement. Moreover, it has also been hypothesized that, if the method proposed is able to enhance the classification of a test data set actually correlated to other feedback conditions, it would perform even better on real-time, online applications.

The method presented on the current project for the outlier identification was the one-class SVM (1-SVM). A similar approach for outlier detection had been reported for the identification of intrusions on a network [105]; however, before this project, it had not been yet evaluated for Brain-Computer Interfaces.

The 1-SVM creates a boundary between the trials classified as inliers and outliers. The topology of such boundary, which in the current case presents 68 dimensions – each dimension represents one channel applied for the EEG data collection, depends on the choice of *nu*, which is a user-defined parameter that represents the probability of a given sample to be an outlier.

Section 5.1.1 exposed the reasons and results that led to the employment of the cross-validation instead of some fixed threshold value in order to elect the *nu* value. However it could not reach the best values of nu possible, the cross-validation performed better than the threshold-based approaches.

This method is based on a performance estimation performed from only the calibration data set, dividing it in five subsets. Four of them (120 trials) are applied as 1-SVM input for each value of nu. The trials labeled as outliers are penalized and the resulting set of trials is employed to the classifier learning. The last subset of calibration trials (30) is subsequently used to evaluate the classification performance, i.e. they are given to the previously trained LDA classifier, which attempts to correctly identify to which class each trial belongs. It is possible since all the data employed are already labeled, since it is the calibration data set (section 3).

Another choice concerning the 1-SVM regards which trials present as its input (section 5.1.2), i.e. whether to present both classes (all the calibration data set) or only one class at a time. One important argument to present the classes separately is based on the possibility of an outlier for the first class not to be identified as so; because it is found on the subspace where the samples from the second class are (figure 5.1).

Nevertheless, table 5.2 showed that presenting both classes together resulted better for the current work. It was hypothesized that it resulted from the employment of the same nu for both classes, even when they were given separately to the 1-SVM. The problem is that there might be the case in which one class presents much more outliers than the other class; and if both are evaluated with the same nu for the outliers identification, it will consequently generate mislabeled outliers as output in every nu choice.

On the other hand, such decision was grounded on another hypothesis, which is the possibility of a drawback on the cross-validation efficiency, since the nu chosen hardly reaches the optimal performance. In some subjects, a final value of nu close to the optimal one is observed, while in others it resulted to be really different. Therefore electing two different values of nu through the cross-validation approach could result on a worse final performance and has not been considered on this work.

After the outliers have been labeled by the 1-SVM, one should decide how to handle them, i.e. one could either completely remove or penalize them with weights proportional to the distance to the boundary created by the 1-SVM. Section 4.3 presented the idea to employ a sigmoid function to penalize the trials instead of removing them, based on the hypothesis that every trial may contain characteristics somehow important for the classification learning, even though they are real outliers.

On section 5.2, the results performance improvement for the ten different functions (Appendix A) applied on the project work were presented. The binary function, i.e. the function that completely removed the outliers, presented a great difference between the final performances obtained from the best, theoretical nu and the cross-validation one (table 5.3, figure 5.2).

Therefore it has been hypothesized that the 1-SVM may be able, on an ideal case that cannot be practically reproduced, to identify the outliers to be fully removed; but for the cross-validation nu selection, it could be a safer alternative to penalize the trials and therefore retain their information for a better classification.

The final evaluation of error rate reduction depicted on figure 5.3 considered the 1-SVM choices exposed above, namely the employment of cross-validation for nu election; the application of the whole calibration data set as its input; and the usage of a penalizing function to handle the outlier, rather than completely remove them.

Moreover, some constraints regarding the Common Spatial Pattern (CSP) have already been done. For the covariance matrix estimation, a regularization process called shrinkage [100] was applied so as to make the estimation stable and more robust to the presence of outliers. The second choice concerning CSP refers to the number of filters per class that it will generate as output. Three filters per class were applied for the current project in order to avoid excessive loss of information.

The idea was to expose how the 1-SVM is efficient for outlier identification and for the according error rate reduction, even when the CSP choices that are expected to produce a better immunity to outliers are used.

The 1-SVM results were confronted with two cases: i) the complete absence of any outlier handling process before the covariance estimation for the CSP; ii) removing some trials that were labeled as outlier by a filter method previously developed within the Berlin Brain Computer Interface project (see section 2.3).

Along the performance graphs exposed on figure 5.3, it is possible to conclude that the 1-SVM has the potential to reduce the error rate caused by the final classification on a BCI-based system. The 4th graph, i.e. the one comparing the original data set error rate with the employment of the optimal nu for the 1-SVM, clearly elucidates how the proposed method could effectively increment the performance. It is translated on the number of subjects that presented a strict improvement: 32/40 (80%).

Moreover, from a further visual inspection of the same graphs, one sees that the other 20% of subjects resulted on a practically equal performance. It is another positive mark for the 1-SVM potential; since its ideal employment, i.e. with the optimal nu, could guarantee at least the stabilization of the current performance for every single subject, besides offering an 80% probability of reducing the error rate, according to the current study.

A similar result can be found when comparing the error rate reduction produced by the filter method of removing trials previously available for the BBCI group (section 2.3) with the 1-SVM method proposed by the current project (figure 5.3, first graph on bottom). No subject in such comparison presented decrease of performance. Rather, 87.5% of the subjects (35/40) resulted on a strict performance improvement.

However, one can also notice that, when applying cross-validation, the ratio of subjects presenting an absolute performance increment compared to the original data set (figure 5.3, second graph on top) reduces to 45%; moreover, a subject could present up to 10% reduction on performance while employing the 1-SVM for outlier detection.

Nevertheless, even though the current case applies three filters per class with shrinkage covariance matrix estimation, which was expected to represent the best immunity against outliers; the 1-SVM method for outlier handling combined with cross-validation as nu selection was still able to present a reduction on the error rate of approximately 20% for two particular subjects: an expressive improvement for a real case.

Further analyses for the subject labeled as VPjh_08_05_27 on the BBCI data base are performed and discussed. Such subject is one of the two pointed on the last paragraph that had presented a considerable performance improvement after the employment of 1-SVM with cross-validation.

In order to evaluate the influence of a single trial on the final performance, graphs correlating the error rate with the weight for a given trial were exposed on section 6.2 for different sets of reference. Some important facts were observed and discussed from the results exposed, namely:

a) The best achievable performance does not necessarily lies on the extreme weights (0.0 or 1.0), but can rather be found for intermediate weights, for example weight = 0.2 (figure 6.2);
b) The absence of shrinkage on the covariance estimation resulted, as expected on a greater variance of performance, translated on the elevated transition of colors along the graph (figure 6.3);
c) Only one step (0.05) change on the value of *nu* may result on a great lost of performance (e.g. 15%, as when comparing figures 6.4 and 6.5), which illustrates a high sensitivity on the 1-SVM output and accordingly an obstacle for the cross-validation to achieve an efficient decision;
d) Actually one single trial, depending on the reference set, can considerably influence on the final BCI performance, whereas 15% of both improvement and reduction were registered on figure 6.5;
e) The combination of removal and inclusion of trials that separated had improved the performance (tables 6.1 and 6.2) may lead to an actual increment of the error rate. It depicts the inherently dependence to the set of reference that the efficient weighting process present.

In order to better understand the point (e), section 6.3 exposed analyses regarding the change on position of the subspaces defined by the CSP filters. It was evaluated while either removing two trials (#005, #117) or including three trials (#101, #105, #107); both methods were based on the performance improvement depicted on tables 6.1 and 6.2.

In particular, table 6.5 showed how the filters regarding the first class (left arm) had underwent a great subspace change when either removing or including trials, which was translated into the angle related to the original set. The angle for the first filter specifically was very high (74 and 85 degrees).

It is therefore hypothesized that such considerable change on the subspace is responsible for the worse performance depicted on table 6.3. The problem may lie on the fact that the CSP filters obtained from the calibration data set are afterwards applied to the test (feedback) data set.

If such filters already undergo a great change for both cases, i.e. the removal of trials #005, #117 and the inclusion of #101, #105, #107; when combining such methods the resulting filters may actually not be efficient anymore and interfere on the final classification.

Finally, table 6.6 depicted the correspondence between each trial and task (left arm or feet) performed. Although one could expect, for example on the case of trials removing, that both trials should belong to the left arm, since these filters were highly affected after their removal; the trial #005 is actually related to the feet.

It was however hypothesized that while the left arm trial (#117) was deciding for the filter definition, the trial regarding the feet not. Nevertheless, #005 alone could improve the performance in 14% (table 6.1). It suggests that classifier learning and CSP filters determination are not always correlated, although both can interfere on the final BCI performance and consequently improve it.

To sum up everything previously exposed, the current thesis report exposed how the 1-SVM can be efficient for the outlier detection and how it can be translated into the final performance of a BCI-based system. It is an expressive result, since the 1-SVM had not been previously employed for BCIs for it.

The cross-validation method may not be ideal and present in an average case between 40 subjects a worse performance than the ideal one, which is however not reproducible. One possible future approach to improve the proposed method could be a new technique in order to elect the nu to be used for the 1-SVM.

Appendix A – Matlab code: functions to handle outliers

```
ypred = zeros(data,1); %data = 75 (only one class) or 150 (both classes)
x = onesvm(svm, nu); %1-SVM outlier identification
indneg = find(x<0); %trials classified as outliers
indpos = find(x>0); %inliers

if(f==1) %outliers -> sigmoid ; mult 45% (outliers)
            sigmult = -4.6/(quantile(x(indneg),[.45]));
    for j=1:size(ypred,1)
        if(x(j) > 0)
            ypred(j) = 1;
        else
            ypred(j) = 1./(1 + exp(-1*abs(sigmult)*x(j))); %sigmoid
        end
    end
end

if(f==2) %all data -> sigmoid ; mult 45% (outliers) / 55% (inliers)
               sigmultn = -4.6/(quantile(x(indneg),[.45]));
               sigmultp = -4.6/(quantile(x(indpos),[.55]));
    for j=1:size(ypred,1)
        if(x(j) > 0)
            ypred(j) = 1./(1 + exp(-1*abs(sigmultp)*x(j)));
        else

            ypred(j) = 1./(1 + exp(-1*abs(sigmultn)*x(j))); %sigmoid
        end
    end
end

if(f==3) %outliers -> sigmoid ; mult 35% (outliers)
            sigmult = -4.6/(quantile(x(indneg),[.35]));
    for j=1:size(ypred,1)
        if(x(j) > 0)
            ypred(j) = 1;
        else

            ypred(j) = 1./(1 + exp(-1*abs(sigmult)*x(j))); %sigmoid
        end
    end
end

if(f==4) %all data -> sigmoid ; mult 35% (outliers) / 65% (inliers)
               sigmultn = -4.6/(quantile(x(indneg),[.35]));
               sigmultp = -4.6/(quantile(x(indpos),[.65]));
    for j=1:size(ypred,1)
        if(x(j) > 0)
            ypred(j) = 1./(1 + exp(-1*abs(sigmultp)*x(j)));
        else

            ypred(j) = 1./(1 + exp(-1*abs(sigmultn)*x(j))); %sigmoid
        end
    end
end

if(f==5) %outliers -> sigmoid ; mult 25% (outliers)
            sigmult = -4.6/(quantile(x(indneg),[.25]));
    for j=1:size(ypred,1)
        if(x(j) > 0)
            ypred(j) = 1;
        else %x(j)<0
            ypred(j) = 1./(1 + exp(-1*abs(sigmult)*x(j))); %sigmoid
        end
    end
end

if(f==6) %all data -> sigmoid ; mult 25% (outliers) / 75% (inliers)
               sigmultn = -4.6/(quantile(x(indneg),[.25]));
               sigmultp = -4.6/(quantile(x(indpos),[.75]));
    for j=1:size(ypred,1)
        if(x(j) > 0)
            ypred(j) = 1./(1 + exp(-1*abs(sigmultp)*x(j)));
        else

            ypred(j) = 1./(1 + exp(-1*abs(sigmultn)*x(j))); %sigmoid
        end
    end
end

if(f==7) %outliers -> sigmoid ; mult 15% (outliers)
            sigmult = -4.6/(quantile(x(indneg),[.15]));
    for j=1:size(ypred,1)
        if(x(j) > 0)
            ypred(j) = 1;
        else

            ypred(j) = 1./(1 + exp(-1*abs(sigmult)*x(j))); %sigmoid
        end
    end
end

if(f==8) %all data -> sigmoid ; mult 15% (outliers) / 85% (inliers)
               sigmultn = -4.6/(quantile(x(indneg),[.15]));
               sigmultp = -4.6/(quantile(x(indpos),[.85]));
    for j=1:size(ypred,1)
        if(x(j) > 0)
            ypred(j) = 1./(1 + exp(-1*abs(sigmultp)*x(j)));
        else

            ypred(j) = 1./(1 + exp(-1*abs(sigmultn)*x(j))); %sigmoid
        end
    end
end

if(f==9) %all data -> sigmoid ; mult k=1200 (constant)
   sigmult = 1200;
   ypred(:) = 1./(1 + exp(-1*sigmult*x)); %sigmoid
end

if(f==10) %binary (weights = 10^-25 or 1.0)
   for j=1:size(ypred,1)
       if(x(j) < 0)
           ypred(j) = 0.0000000000000000000000001;
       else
           ypred(j) = 1;
       end
   end
end
```

Bibliography

[1] Schwartz, A. B.; Tracy Cui, X.; Weber, D. J; Moran, D. W.; , “Brain-Controlled Interfaces: Movement Restoration with Neural Prosthetics”, *Neuron*, vol. 52, no. 1, pp. 205-220, ISSN 0896-6273, 2006. doi:10.1016/j.neuron.2006.09.019

[2] Muller-Putz, G. R.; Pfurtscheller, G.; , "Control of an Electrical Prosthesis With an SSVEP-Based BCI," *Biomedical Engineering, IEEE Transactions on* , vol. 55, no. 1, pp. 361-364, Jan. 2008 doi: 10.1109/TBME.2007.897815

[3] Hochberg, L. R.; Bacher, D.; Jarosiewicz, B.; Masse, N. Y.; Simeral, J. D.; Vogel, J.; Haddadin, S.; Liu, J.; Cash, S. S.; Smagt, P.; Donoghue, J. P.; , “Reach and grasp by people with tetraplegia using a neurally controlled robotic arm,” *Nature*, vol. 485, pp. 372-375, 2012. doi: 10.1038/nature11076

[4] Iturrate, I.; Antelis, J.; Kübler, A.; Minguez, J.; ,“Non-invasive brain actuated wheelchair based on a p300 neurophysiological protocol and automated navigation,” *IEEE Trans. Robot.,* vol. 25, no. 3, pp. 614-627, 2009.

[5] Millán, J.d.R.; Galan, F.; Vanhooydonck, D.; Lew, E.; Philips, J.; Nuttin, M.; , "Asynchronous non-invasive brain-actuated control of an intelligent wheelchair," *Engineering in Medicine and Biology Society, 31st Annual International Conference of the IEEE* , pp.3361-3364, 2009. doi: 10.1109/IEMBS.2009.5332828

[6] Leeb, R.; Friedman, D.; Müller-Putz, G. R.; Scherer, R.; Slater, M.; Pfurtscheller, G.; , “Self-Paced (Asynchronous) BCI Control of a Wheelchair in Virtual Environments: A Case Study with a Tetraplegic,” *Computational Intelligence and Neuroscience*, vol. 2007, Article ID 79642, 8 pages, 2007. doi:10.1155/2007/79642

[7] Blankertz, B.; Dornhege, G.; Krauledat, M.; Schröder, M.; Williamson, J.; Murray-Smith, R.; Müller, K.R.; , “The Berlin Brain-Computer Interface presents the novel mental typewriter Hex-o-Spell,” *Proceedings of the 3rd International Brain-Computer Interface Workshop and Training Course*, Verlag der TU Graz, 2006.

[8] Sellers, E. W.; Krusienski, D. J.; McFarland, D. J.; Vaughan, T. M.; Wolpaw, J. R.; , “A P300 event-related potential brain–computer interface (BCI): The effects of matrix size and inter stimulus interval on performance,” *Biological Psychology*, vol. 73, no. 3, pp. 242-252, ISSN 0301-0511, 2006. doi:10.1016/j.biopsycho.2006.04.007

[9] Bensch, M.; Karim, A. A.; Mellinger, J.; Hinterberger, T.; Tangermann, M.; Bogdan, M.; Rosenstiel, W.; Birbaumer, N.; , “Nessi: an EEG-controlled web browser for severely paralyzed patients,” *Comput. Intell. Neurosci.,* vol. 2007, no. 6, Article ID 71863, 2007. doi:10.1155/2007/71863

[10] Mugler, E. M.; Ruf, C. A.; Halder, S.; Bensch, M.; Kübler, A.; , “Design and implementation of a P300-based brain-computer interface for controlling an internet browser,” *IEEE Trans. Neural Syst. Rehabil. Eng.,* vol. 18, no. 6, pp. 599-609, 2010. doi:10.1109/TNSRE.2010.2068059

[11] Miranda, E. R.; , “Brain-computer music interface for composition and performance”, *Int. J. Dis. Human Dev.,* vol. 5, no. 2, pp. 119-125, 2006. doi: 10.1515/IJDHD.2006.5.2.119

[12] Leeb, R.; Lee, F.; Keinrath, C.; Scherer, R.; Bischof, H.; Pfurtscheller, G.; , “Brain-computer communication: motivation, aim, and impact of exploring a virtual apartment,” *IEEE Trans. Neural Syst. Rehabil. Eng.,* vol. 15, no. 4, pp. 473-482, 2007. doi: 10.1109/TNSRE.2007.906956

[13] Krepki, R.; Blankertz, B.; Curio, G.; Müller, K.-R.; , “The Berlin brain-computer-interface (BBCI): towards a new communication channel for online control in gaming applications,” *J. Multimedia Tools Appl.*, vol. 33, no. 1, pp. 73-90, 2007. doi: 10.1007/s11042-006-0094-3

[14] Nijholt, A.; Tan, D.; Allison, B.; Millán, J.d.R.; Graimann, B.; Jackson, M. M.; , “Brain-computer interfaces for human-computer interactions and games,” *Proceedings of ACM CHI 2008,* Florence.

[15] Miranda, E. R.; Magee, W.; Wilson, J. J.; Eaton, J.; Palaniappan, R.; , “Brain-computer music interfacing (BCMI): from basic research to the real world of special needs", *Music and Medicine*, vol. 3, no. 3, pp. 134-140, 2011. doi: 10.1177/1943862111399290

[16] Cutrell, E.; Tan, D.; , “BCI for passive input in HCI,” *ACM CHI’08 Conference on Human Factors in Computing Systems Workshop on Brain–Computer Interfaces for HCI and Games*, 2008.

[17] Nijholt, A.; , “BCI for games: ‘a state of the art’ survey,” *Proceedings of the 8th International Conference on Entertainment Computing,* Paris*,* pp. 225-228, 2009. doi: 10.1007/978-3-540-89222-9_29

[18] Doyle, T. E.; Kucerovsky, Z.; Ieta, A.; , “Affective state control for neuroprothetics,” *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pp. 1248-1251, Aug. 30 2006-Sept. 3 2006. doi: 10.1109/IEMBS.2006.260531

[19] Haufe, S.; Treder, M. S.; Gugler, M. F.; Sagebaum, M.; Curio, G.; Blankertz, B.; , “EEG potentials predict upcoming emergency brakings during simulated driving,” *J. Neural Eng.,* vol. 8, no. 5, 056001, 2011. doi:10.1088/1741-2560/8/5/056001

[20] Nicolelis, M. A.; Dimitrov, D.; Carmena, J. M.; Crist, R.; Lehew, G.; Kralik, J. D.; Steven, P. W.; , “Chronic, multisite, multielectrode recordings in macaque monkeys,” *Proc. Natl. Acad. Sci. USA*, vol. 100, pp. 11041–11046, 2003. doi: 10.1073/pnas.1934665100

[21] Carmena, J. M.; Lebedev, M. A.; Crist, R. E.; O’Doherty, J. E.; Santucci, D. M.;Dimitrov, D. F.; Patil, P. G.; Henriquez, C. S.; Nicolelis, M. A.; , “Learning to control a brain-machine interface for reaching and grasping by primates,” *Public Library of Science Biology*, E42, 2003. doi: 10.1371/journal.pbio.0000042

[22] Ganguly, K.; Carmena, J. M.; , “Emergence of a stable cortical map for neuroprosthetic control,” *PloS Biol.*, vol. 7, no. 1–3, e1000153, 2009. doi: 10.1371/journal.pbio.1000153

[23] Leuthardt, E. C.; Schalk, G.; Wolpaw, J. R.; Ojemann, J. G.; Moran, D. W.; , “A brain-computer interface using electrocorticographic signals in humans,” *J. Neural Eng.,* vol. 1,. no. 2, 63, 2004. doi:10.1088/1741-2560/1/2/001

[24] Lal, T. N.; Hinterberger, T.; Widman, G.; Schröder, M.; Hill, J.; Rosenstiel, W.; Elger, C. E.; Schölkopf, B.; Birbaumer, N.; , “Methods towards invasive human brain computer interfaces,” *Advances in Neural Information Processing System,* vol. 17, pp. 737-744, 2005.

[25] Shenoy, P.; Miller, K. J.; Ojemann, J. G.; Rao, R. P. N.; , “Generalized features for electrocorticographic BCIs,” *IEEE Trans. on Biomedical Engineering,* vol. 55, no. 1, pp. 273-280

[26] Schalk, G.; Miller, K. J.; Anderson, N. R.; Wilson, J. A.; Smyth, M. D.; Ojemann, J. G.; Moran, D. W.; Wolpaw, J. R.; Leuthardt, E. C.; , “Two-dimensional movement control using electrocorticographic signals in humans,” *J. Neural Eng.,* vol. 5, pp. 75-84, 2008. doi:10.1088/1741-2560/5/1/008

[27] Nair, D. G.; Purcott, K. L.; Fuchs, A.; Steinberg, F.; Kelso, J. A.; , “Cortical and cerebellar activity of the human brain during imagined and executed unimanual and bimanual action sequences: a functional MRI study,” *Brain Research: Cognitive Brain Research,* vol. 15, no. 3, pp. 250-260, 2003.

[28] Weiskopf, N.; Mathiak, K.; Bock, S. W.; Scharnowski, F.; Veit, R.; Grodd, W.; Goebel, R.; Birbaumer, N.; , “Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI),” *IEEE Trans. Biomed. Eng.,* vol. 51, no. 6, pp. 966-970, 2004. doi: 10.1109/TBME.2004.827063

[29] Haynes, J.; Rees, G.; , “Decoding mental states from brain activity in humans,” *Nature Reviews Neuroscience,* vol. 7, pp. 523-534, 2006. doi:10.1038/nrn1931

[30] Sitaram, R.; Weiskopf, N.; Caria, A.; Veit, R.; Erb, M.; Birbaumer, N.; , “fMRI brain-computer interfaces: a tutorial on methods and applications,” *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 95-106, 2008.

[31] Laitinen, L.; , “Neuromagnetic sensorimotor signals in brain computer interfaces,” , *Master Thesis*, Helsinki Institute of Technology, Helsinki (2003).

[32] Coyle, S.; Ward, T.; Markham, C.; McDarby, G.; , “On the suitability of near-infrared (NIR) systems for next-generation brain–computer interfaces,” *Physiol. Meas.,* vol. 25, pp. 815-822, 2004.

[33] Power, S. D.; Kushki, A.; Chau, T.; , “Towards a system-paced near-infrared spectroscopy brain-computer interface: differentiating prefrontal activity due to mental arithmetic and mental singing from the no-control state,” *J. Neural Eng.,* vol. 8, no. 6, 2011. doi:10.1088/1741-2560/8/6/066004

[34] Girouard, A.; Solovey, E. T.; Jacob, R. J. K.; , “Designing a passive brain computer interface using real time classification of functional near-infrared spectroscopy,” *Int. J. Autonomous and Adaptive Communication Systems,* vol. 6, no. 1, pp. 26-44, 2013.

[35] Mellinger, J.; Schalk, G.; Braun, C.; Preissl, H.; Rosenstiel, W.; Birbaumer, N.; Kübler, A.; , “An MEG-based brain-computer interface (BCI),” *Neuroimage,* vol. 36, no. 3, pp. 581-593, 2007. doi:10.1016/j.neuroimage.2007.03.019

[36] Buch, E.; Weber, C.; Cohen, L. G.; Braun, C.; Dimyan, M.; Ard, T.; Mellinger, J.; Caria, A.; Soekadar, S.; Birbaumer, N.; , “Think to move: a neuromagnetic BCI system for chronic stroke,” *Stroke*, vol. 39, pp. 910-917, 2008. doi:10.1161/STROKEAHA.107.505313

[37] Sander, T. H.; Preusser, J.; Mhaskar, R.; Kitching, J.; Trahms, L.; Knappe, S.; , “Magnetoencephalography with a chip-scale atomic magnetometer,” *Biomedical Optical Express,* vol. 3, no. 5, pp. 981-990, 2012. doi:10.1364/BOE.3.000981

[38] Zander, T. O.; Lehne, M.; Ihme, K.; Jatzev, S.; Correia, J.; Kothe, C.; Picht, B.; Nijboer, F.; , “A dry EEG-system for scientific research and brain–computer interfaces,” *Front. Neurosci.,* vol. 5, no. 53, 2011. doi:10.3389/fnins.2011.00053

[39] Guger, C.; Krausz, G.; Allison, B. Z.; Edlinger, G.; , “Comparison of dry and gel based electrodes for P300 brain-computer interfaces,” *Fron. Neurosci.,* vol. 6, no. 60, 2012. doi: 10.3389/fnins.2012.00060

[40] Popescu, F.; Blankertz, B.; Müller, K.-R.; , “Computational challenges for noninvasive brain computer interfaces,” *IEEE Intelligent Systems,* vol. 23, no. 3, pp. 78-79, 2008.

[41] Birbaumer, N.; , “Breaking the silence: brain-computer interfaces (BCI) for communication and motor control,” *Psychophysiology,* vol. 43, pp. 517-532, 2006. doi: 10.1111/j.1469-8986.2006.00456.x

[42] Millán, J. del R.; Carmena, J.; , “Invasive or noninvasive: understanding brain-machine interface technology,” *IEEE Engineering in Medicine and Biology Magazine* vol. 29, no. 1, pp. 16-22, 2010. doi:10.1109/MEMB.2009.935475

[43] Makeig, S.; Onton, J.; , “ERP features and EEG dynamics: an ICA perspective,” *Oxford Handbook of Event-Related Potential Components (Luck, S.; Kappenman, E.)*, Oxford University Press, 2009.

[44] Friman, O.; Lüth, T.; Volosyak, I.; Gräser, A.; , “Spelling with steady-state visual evoked potentials,” *3rd International IEEE/EMBS Conference on Neural Engineering, 2007,* pp. 354-357, 2007. doi:0.1109/CNE.2007.369683

[45] Allison, B. Z.; McFarland D. J.; Schalk, G.; Zheng, S. D.; Jackson, M. M.; Wolpaw, J. R.; , “Towards an independent brain-computer interface using steady state visual evoked potentials,” *Clin. Neurophysiol.*, vol. 119, no. 2, 2008

[46] Hoffmann, U.; Vesin, J.-M.; Ebrahimi, T.; Diserens, K.; , “An efficient P300-based brain-computer interface for disabled subjects,” *Journal of Neuroscience Methods,* vol. 167, no. 1, pp. 115-125, 2008. doi:10.1016/j.jneumeth.2007.03.005

[47] Nijboer, F.; Sellers, E. W.; Mellinger, J.; Jordan, M. A.; Matuz, T.; Furdea, A.; Halder, S.; Mochty, U.; Krusienski, D. J.; Vaughan, T. M.; Wolpaw, J. R.; Birbaumer, N.; Kübler, A.; , “A P300-based brain–computer interface for people with amyotrophic lateral sclerosis,” *Clin. Neurophysiol.,* vol. 119, no. 8, pp. 1909-1916, 2008. doi:10.1016/j.clinph.2008.03.034

[48] Buttfield, A.; Ferrez, P. W.; Millán, J.d.R.; , “Towards a robust BCI: error potentials and online learning,” *IEEE Trans. on Neural Systems and Rehab. Eng.,* vol. 14, no. 2, pp. 164-168, 2006. doi:10.1109/TNSRE.2006.875555

[49] Chavarriaga, R.; Millán, J.d.R.; , “Learning from EEG error-related potentials in noninvasive brain-computer interfaces,” *IEEE Trans. on Neural Systems and Rehab. Eng.,* vol. 18, no. 4, pp. 381-388, 2010. doi:10.1109/TNSRE.2010.2053387

[50] Birbaumer, N.; Ghanayim, N.; Hinterberger, T.; Iversen, I.; Kotchoubey, B.; Kübler, A.; Perelmounter, J.; Taub, E.; Flor, H.; , “A spelling device for the paralysed,” *Nature*, vol. 398, pp. 297–298, 1999. doi:10.1038/18581

[51] Hinterberger, T.; Schmidt, S.; Neumann, N.; Mellinger, J.; Blankertz, B.; Curio, G.; Birbaumer, N.; , “Brain-computer communication and slow cortical potentials,” *IEEE Trans. Biomed. Eng.,* vol. 51, no. 6, pp. 1011-8, 2004.

[52] Krusienski, D. J.; Schalk, G.; McFarland D. J.; Wolpaw, J. R.; , “A mu-rhythm matched filter for continuous control of brain-computer interfaces,” *IEEE Trans. Biomed. Eng.,* vol. 54, no. 2, pp. 273-280, 2007. doi:10.1109/TBME.2006.886661

[53] Bai, O.; Lin, P.; Vorbach, S.; Floeter, M. K.; Hattori, N.; Hallett, M.; , “A high performance sensorimotor beta rhythm-based brain–computer interface associated with human natural motor behavior,” *J. Neural Eng.,* vol. 5, pp. 24-35, 2008. doi:10.1088/1741-2560/5/1/003

[54] Krauledat, M.; , “Analysis of nonstationarities in EEG signals for improving brain-computer interface performance,” , *PhD Thesis*, *Teschnische Universität Berlin*, 2008.

[55] Woon, W. L.; Cichocki, A.; , “Novel features for brain-computer interfaces,” *Computational Intelligence and Neuroscience,* vol. 2007, ID 82827, 7 pages, 2007. doi:10.1155/2007/82827

[56] McFarland, D. J.; Anderson, C. W.; Müller, K-.R.; Schlögl, A.; Krusienski, D. J.; , “BCI meeting 2005 – workshop on signal processing: feature extraction and translation,” *IEEE Trans. on Neural Systems and Rehab. Eng.,* vol. 14, no. 2, pp. 135-138, 2006. doi: 10.1109/TNSRE.2006.875637

[57] Tangermann, M.; Müller, K-.R.; Aertsen, A.; Birbaumer, N.; Braun, C.; Brunner, C.; Leeb, R.; Mehring, C.; Miller, K. J.; Mueller-Putz, G.; Nolte, G.; Pfurtscheller, G.; Preissl, H.; Schalk, G.; Schlögt, A.; Vidaurre, C.; Waldert, S.; Blankertz, B.; , “Review of the BCI Competition IV,” *Frontiers of Neuroscience,* vol. 6, no. 55, 2012. doi: 10.3389/fnins.2012.00055

[58] McFarland, D. J.; Wolpaw, J. R.; , “Sensorimotor rhythm-based brain-computer interface (BCI): model order selection for autoregressive spectral analysis,” *J. Neural Eng.,* vol. 5, no. 2, pp. 155-162, 2008. doi:10.1088/1741-2560/5/2/006

[59] Müller, K-.R.; Krauledat, M.; Dornhege, G.; Curio, G.; Blankertz, B.; , “Machine learning techniques for brain-computer interfaces,” *Biomedical Engineering,* vol. 49, no. 1, pp. 11-22, 2004. doi:10.1.1.161.2435

[60] Li, Y.; Gao, X.; Liu, H.; , “Classification of single-trial electroencephalogram during finger movement,” *IEEE Trans. on Biomed. Eng.,* vol. 51, no. 6, pp. 1019-1025, 2004. doi:10.1109/TBME.2004.826688

[61] McFarland, D. J.; McCane, L. M.; David, S. V.; Wolpaw, J. R.; , “Spatial filter selection for EEG-based communication,” *Electroencephalography and clinical Neurophysiology,* vol. 103, no. 3, pp. 386-394, 1997. doi:10.1016/S0013-4694(97)00022-2

[62] Sannelli, C.; Vidaurre, C.; Müller, K.-R.; Blankertz, B.; , “CSP patches: an ensemble of optimized spatial filters. An evaluation study,” *J. Neural Eng.*, vol. 8, 025012, 2011. doi:10.1088/1741-2560/8/2/025012

[63] Blankertz, B.; Tomioka, R.; Lemm, S.; Kawanabe, M.; Müller, K.-R.; , “Optimizing spatial filters for robust EEG single-trial analysis,” *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41-56, 2008. doi:10.1109/MSP.2008.4408441

[64] Ramoser, H.; Müller-Gerking, J.; Pfutrscheller, G.; , “Optimal spatial filtering of single trial EEG during imagined hand movement,” IEEE Trans on Rehab Eng, vol. 8, no. 4, pp. 441-446, 2000. doi:10.1109/86.895946

[65] Lemm, S.; Blankertz, B.; Curio, G.; Müller, K.-R.; , “Spatio-spectral filters for improving classification of single trial EEG,” IEEE Trans Biomed Eng, vol. 52, no. 9, pp. 1541-1548, 2005. doi:10.1109/TBME.2005.851521

[66] Samek, W.; Vidaurre, C.; Müller, K.-R.; Kawanabe, M.; , “Stationary common spatial patterns for brain-computer interfaces,” *J. Neural Eng,* vol. 9, no. 2, 026013 (14pp), 2012. doi:10.1088/1741-2560/9/2/026013

[67] Hastie, T.; Tibshirani, R.; Friedman, J.; , “The elements of statistical learning: data mining, inference and prediction,” *Springer Series in Statistics,* February 2009.

[68] Visuri, S.; Koivunen, V.; Oja, H.; , “Sign and rank covariance matrices,” *Journal of Statistical Planning* and Inference, vol. 91, no. 2, pp. 557-575, 2000. doi:10.1016/S0378-3758(00)00199-3

[69] Savelainen, A.; , “An introduction to EEG artifacts,” *Mat-2.4108 Independent research projects in applied mathematics*, 2010.

[70] McFarland, D. J.; Sarnacki, W. A.; Vaughan, T. M.; Wolpaw, J. R.; , “Brain-computer interface (BCI) operation: signal and noise during early training sessions,” *Clinical Neurophysiology*, vol. 116, no. 1, pp. 56-62, 2005. doi:10.1016/j.clinph.2004.07.004

[71] Fatourechi, M.; Bashashati, A.; Ward, R. K.; Birch, G. E.; , “EMG and EOG artifacts in brain computer interfaces systems: a survey,” *Clinical Neurophysiology*, vol. 118, no. 3, pp. 480-494, 2007. doi:10.1016/j.clinph.2006.10.019

[72] Cohen, B. H.; Davidson, R. J.; Senulis, J. A.; Saron, C. D.; Weisman, D. R.; , “Muscle tension patterns during auditory attention,” Biological psychology, vol. 33, no. 2-3, pp. 133-156, 1992. doi:10.1016/0301-0511(92)90028-S

[73] Waterink, W.; van Boxtel, A.; , “Facial and jaw-elevator EMG activity in relation to changes in performance level during a sustained information processing task,” *Biological psychology,* vol. 37, no. 3, pp. 183-198, 1994. doi:10.1016/0301-0511(94)90001-9

[74] Barlow, J. S.; , “Artifact processing (rejection and minimization) in EEG data processing,” *Handbook of Electroencephalography and Clinical Neurophysiology*, Amsterdam: Elsevier, vol. 2, pp. 15-62, 1986.

[75] Gratton, G.; , “Dealing with artifacts: the EOG contamination of the event-related brain potential,” *Behavior Research Methods, Instruments & Computers,* vol. 30, no. 1, pp. 44-53, 1998. doi:10.3758/BF03209415

[76] Ochoa, C. J.; Polich, J.; , “P300 and blink instructions,” *Clinical Neurophysiology,* vol. 111, no. 1, pp. 93-98, 2000. doi:10.1016/S1388-2457(99)00209-6

[77] Hyvärinen, A.; Karhunen, J.; Oja, E.; , “Independent Component Analysis,” *John Wiley & Sons*, 2001.

[78] Jung, T. P.; Makeig, S.; Humphries, C.; Lee, T. W.; McKeown, M. J.; Iragui, V.; Sejnowski, T. J.; , “Removing electroencephalographic artifacts by blind source separation,” *Psychophysiology*, vol. 37, no. 2, pp. 163-178, 2000. doi:10.1111/1469-8986.3720163

[79] Jung, T.-P.; Makeig, S.; Westerfield, M.; Townsend, J.; Courchesne, E.; Sejnowski, T. J.; , “Removal of eye activity artifacts from visual event-related potentials in normal and clinical subjects,” *Clinical* Neurophysiology, vol. 111, no. 10, 2000. doi:10.1016/S1388-2457(00)00386-2

[80] Delorme, A.; Sejnowski, T. J.; Makeig, S.; , “Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis,” *NeuroImage,* vol. 34, no. 4, pp. 1443-1449, 2007. doi:10.1016/j.neuroimage.2006.11.004

[81] Quinlan, J. R.; , “C4.5: Programs for machine learning,” *Kaufmann*, San Francisco, 1993.

[82] Tangermann, M.; , “Feature selection for brain-computer interfaces,” Dissertation, *Universität Tübingen,* 2007.

[83] Schachinger, D.; Schindler, K.; Kluge, T.; , "Automatic reduction of artifacts in EEG-signals," *15th* International Conference on Digital Signal Processing, 2007. doi:10.1109/ICDSP.2007.4288539ISBN

[84] Sannelli, C.; Braun, M.; Müller, K.-R.; , "Improving BCI performance by task-related trial pruning," Neural Networks, vol. 22, pp. 1295-1304, 2009. doi:10.1016/j.neunet.2009.08.006

[85] Braun, M. L.; Buhmann, J. M.; Müller, K.-R.; , "On relevant dimensions in kernel feature spaces," *Journal of Machine Learning Research,* vol. 9, pp. 1875-1908, 2008.

[86] Kohavi, R.; John, G. H.; , "Wrappers for feature subset selection," *Artificial Intelligence,* no. 97, no. 1-2, pp. 273-324, 1997. doi:10.1016/S0004-3702(97)00043-X

[87] Jarmulak, J.; Craw, S.; , "Genetic algorithms for feature selection and weighting," *In Proceedings of the IJCAI'99 workshop on Automating the Construction of Case Based Reasoners,* pp. 28-33, 1999.

[88] Dal Seno, B.; Matteucci, M.; Mainardi, L. T.; , "A genetic algorithm for automatic feature extraction in P300 detection," *International Joint Conference on Neural Networks 2008 (IJCNN 2008),* pp. 3145-3152.

[89] Corralejo, R.; Hornero, R.; Álvarez, D.; , "Feature selection using a genetic algorithm in a motor imagery-based brain computer interface," *Engineering in Medicine and Biology Society, 2011 Annual International Conference of the IEEE*, pp. 7703-7706, 2011. doi:10.1109/IEMBS.2011.6091898

[90] Ghanbari, A. A.; Broumandnia, A.; Navidi, H.; Ahmadi, A.; , "Brain computer interface with genetic algorithm," *International Journal of Information and Technology Research,* vol. 2, no. 1, pp. 79-86, 2012.

[91] Thulasidas, M.; Guan, C.; Wu, J.; , "Robust classification of EEG signals for brain-computer interface," *IEEE Transactions on Neural Systems and Rehab Eng*, vol. 14, no. 1, pp. 24-29, 2006. doi:10.1109/TNSRE.2005.862695

[92] Li, Y.; Guan, C.; , "A semi-supervised SVM learning algorithm for joint feature extraction and classification in brain computer interfaces," *Engineering in Medicine and Biology Society, 2006 Annual* International Conference of the IEEE, pp. 2570-2573, 2006. doi:10.1109/IEMBS.2006.260327

[93] Canu, S.; Grandvalet, Y.; Guigue, V.; Rakotomamonjy, A.; "SVM and kernel methods matlab toolbox," *Perception Systèmes et Information, INSA de Rouen, Rouen, France,* 2005.

[94] Cortes, C.; Vapnik, V. N.; , "Support-Vector Networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995. doi:10.1007/BF00994018

[95] Boser, B. E.; Guyon, I. M.; Vapnik, V. N.; , "A training algorithm for optimal margin classifiers," *COLT '92 Proceedings of the fifth annual workshop on Computational learning theory,* pp. 144-152, 1992. doi:10.1145/130385.130401

[96] Kübler, A.; Neumann, N.; Wilhelm, B.; Hinterberger, T.; Birbaumer, N.; , "Predictability of brain-computer communication," *Journal of Psychophysiology,* vol. 18, no. 2-3, pp. 121-129, 2004. doi:10.1027/0269-8803.18.2–3.121

[97] Blankertz, B.; Sannelli, C.; Halder, S.; Hammer, E. M.; Kübler, A.; Müller, K-.R.; Curio, G.; Dickhaus, T.; , "Neurophysiological predictor of SMR-based BCI performance," *Neuroimage*, vol. 51, no. 4, pp. 1303-1309, 2010. doi:10.1016/j.neuroimage.2010.03.022

[98] Brickenkamp, R.; Zillmer, E.; , “d2 test of attention,” *Hogrefe & Huber*, Göttingen, Germany, 1998.

[99] Bracewell, R. N.; , “The fourier transform and its applications,” *3rd ed. New York: McGraw-Hill*, 1999.

[100] Kwan, Clarence C. Y.; "An introduction to shrinkage estimation of the covariance matrix: a pedagogic illustration," *Spreadsheets in Education (eJSiE),* vol. 4, no. 3, article 6, 2011. http://epublications.bond.edu.au/ejsie/vol4/iss3/6

[101] Lotte, F.; Guan, C.; , “Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms,” *IEEE Trans on Biomedical Engineering,* vol. 58, no. 2, pp. 355-362, 2011. doi:10.1109/TBME.2010.2082539

[102] Blankertz, B.; Dornhege, G.; Krauledat, M.; Müller, K-.R.; Curio, G.; , “The non-invasive Berlin brain-computer interface: fast acquisition of effective performance in untrained subjects,” *Neuroimage*, vol. 37, no. 2, pp. 539-550, 2007. doi:10.1016/j.neuroimage.2007.01.051

[103] Devlaminck, D.; Wyns, B.; Santens, P.; Otte, G.; , “A multitask learning approach to spatial filtering in BCI,” *International Journal of Bioelectromagnetism,* vol. 13, no. 1, pp. 29-31, 2011.

[104] Sannelli, C.; Vidaurre, C.; Müller, K.-R.; Blankertz, B.; , “Common spatial patterns patches: online evaluation on BCI-naïve users,” *Conf Proc IEEE Eng Med Biol Soc,* 2012.

[105] Laskov, P.; Schäfer, C.; Kotenko, I.; and Müller, K.-R; , “Intrusion detection in unlabeled data with quarter-sphere support vector machines,” *Praxis der Informationsverarbeitung und Kommunikation*, vol. 27, pp. 228-236, 2004.