



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR
INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

MÁSTER EN INGENIERÍA DE
TELECOMUNICACIÓN

Extracción y clasificación de características aplicados a señales electroencefalográficas en sistemas Brain Computer Interface basados en potenciales evocados P300

Autor:

Pablo Núñez Novo

Tutor:

Dr. D. Roberto Hornero Sánchez

Valladolid, 14 de Diciembre de 2015

TÍTULO: **Extracción y clasificación de características aplicados a señales electroencefalográficas en sistemas Brain Computer Interface basados en potenciales evocados P300**

AUTOR: **Pablo Núñez Novo**

TUTOR: **Dr. D. Roberto Hornero Sánchez**

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

Tribunal

PRESIDENTE: **Dr. D. Jesús Poza Crespo**

VOCAL: **Dra. D^a. María García Gadañón**

SECRETARIA: **Dr. D. Carlos Gómez Peña**

FECHA: **14 de Diciembre de 2015**

CALIFICACIÓN:

Resumen del TFM

Los sistemas *Brain Computer Interface* (BCI) permiten enviar mensajes y órdenes al mundo externo sin necesidad de realizar ninguna actividad muscular. Estos programas pueden habilitar a personas con discapacidades neuromusculares severas como la esclerosis lateral amiotrófica, ataque cerebral y lesiones de la médula espinal para comunicarse y utilizar programas y equipos de diversos tipos que mejoran su calidad de vida. Para conseguir este objetivo, se usan una serie de señales electroencefalográficas registradas en el cuero cabelludo en las cuáles está codificada la intención del usuario. En este estudio nos centramos en una de ellas: los potenciales evocados P300.

El objetivo de este proyecto es el estudio de algoritmos de extracción y clasificación de características en BCI basados en potenciales evocados P300, con el propósito de determinar los parámetros óptimos de todos los métodos utilizados para obtener la máxima precisión en la clasificación. Como métodos de extracción de características se proponen un sub-muestreo y extracción de características mediante transformada wavelet. A su vez se desarrolla un algoritmo de eliminación y selección de canales. En cuanto a la clasificación de características, se utilizan tres métodos: discriminante lineal de Fisher (FLDA), *Spectral Regression Kernel Discriminant Analysis* (SRKDA) y un conjunto de máquinas de vectores de soporte (SVM). Estos métodos se prueban sobre el conjunto de datos II de la III Competición BCI proporcionados por el centro Wadsworth (NYS *Department of Health*), con y sin eliminación de canales. Los resultados se comparan con los del algoritmo ganador de esa competición, el cual consistía en un conjunto de 17 SVM con selección de canales y con otros métodos del estado del arte que usaron la misma base de datos, así como los resultados obtenidos en un estudio preliminar de métodos de clasificación de características.

Los resultados muestran que dos de los métodos propuestos (FLDA y SRKDA), con sub-muestreo y aplicación de transformada wavelet adaptados a cada sujeto, obtienen una precisión (99.5 % y 99 %) mayor que la del método ganador (96.5 %). Comparados con otros métodos del estado del arte, estos dos algoritmos obtienen unos resultados excelentes que los colocan en primer y segundo puesto de la clasificación global.

Palabras clave

Brain Computer Interface (BCI), electroencefalograma (EEG), potenciales evocados P300, conjunto de datos II de la III Competición BCI, transformada wavelet (WT), algoritmo de selección de canales, extracción de características, discriminante lineal de Fisher (FLDA), *Spectral Regression Kernel Discriminant Analysis* (SRKDA), máquinas de vectores de soporte (SVM).

Abstract

Brain Computer Interface (BCI) systems allow people to send messages and commands to the external world without requiring any muscle activity. These programs may enable people with severe neuromuscular disorders, such as amyotrophic lateral sclerosis, brainstem stroke and spinal cord injury to communicate and operate programs and equipment of various types that improve their quality of life. To achieve this, a variety of electrophysiological signals recorded from the scalp in which the user's intentions are coded are used. In this study we focus on one of these: P300 evoked potentials.

The main purpose of this paper is conducting a study on feature extraction and classification algorithms in P300-based BCIs, in order to determine the optimal parameters of each of the methods used, necessary to achieve maximum accuracy in classification. The proposed feature extraction methods are sub-sampling and feature extraction by means of wavelet transform. A channel elimination and selection algorithm is also developed. Regarding feature classification, three different approaches, Fisher's Linear Discriminant (FLDA), Spectral Regression Kernel Discriminant Analysis (SRKDA) and an ensemble of Support Vector Machines (SVM) are considered. These algorithms are tested on dataset II of the III BCI Competition (2003) provided by Wadsworth Center (NYS Department of Health), with and without channel elimination. The results are compared with those of the winning algorithm, which consists of an ensemble of 17 SVM with channel selection, as well as and with other state of the art methods which use the same database, and the results of a preliminar study that focuses on feature classification algorithms.

The results show that two of our proposed methods (SRKDA and FLDA), with sub-sampling and application of wavelet transform adapted to each subject, achieve a higher accuracy (99.5 % and 99 %) than the method that won the competition (96.5 %). Compared to other state of the art methods, these two algorithms provide excellent results that place them on first and second place of the global classification.

Keywords

Brain Computer Interface (BCI), electroencephalogram (EEG), P300 evoked potentials, BCI competition III dataset II, wavelet transform (WT), channel selection algorithm, feature extraction, Fisher's Linear Discriminant (FLDA), Spectral Regression Kernel Discriminant Analysis (SRKDA), Support Vector Machines (SVM).

Agradecimientos

Al Grupo de Ingeniería Biomédica de la Universidad de Valladolid, en especial a Roberto Hornero por confiar en mí para realizar este proyecto.



Índice general

Lista de acrónimos	1
1. Introducción	3
1.1. Señales Biomédicas	3
1.2. Electroencefalograma (EEG)	3
1.3. Aspectos anatómicos y fisiológicos del EEG	5
1.4. Potenciales evocados	7
1.5. Brain Computer Interface	7
1.6. Objetivos del Trabajo Fin de Máster	8
1.7. Estructura del Trabajo Fin de Máster	9
2. Sistemas BCI basados en potenciales evocados P300	11
2.1. Introducción a los BCI	11
2.2. Paradigmas de control BCI	11
2.3. Señales de control en los BCI	12
2.3.1. Potenciales corticales lentos (SCP)	13
2.3.2. Ritmos sensoriomotores	13
2.3.3. Potenciales evocados visuales (VEP)	14
2.3.4. Potenciales evocados P300	15
2.4. Procesado de señales BCI	16
2.4.1. Adquisición de la señal	16
2.4.2. Procesado de la señal	16
2.4.3. Aplicación	17
2.5. Estado del arte en el campo de los potenciales evocados P300	17
2.5.1. Métodos de adquisición de señal	17
2.5.2. Paradigmas de presentación de estímulos	18
2.5.3. Algoritmos de extracción y clasificación	19
2.5.3.1. III Competición BCI (2004)	25
2.5.3.2. Conjunto de datos de la competición	27
2.5.4. Aplicaciones	29
2.6. Método ganador del conjunto de datos II de la III Competición BCI	32
2.6.1. Preprocesado y extracción de características	32
2.6.2. Métodos de clasificación	32

3. Métodos de extracción de características	35
3.1. Introducción	35
3.2. ¿Qué es una wavelet?	35
3.3. Motivación para el análisis Wavelet de señales neuroeléctricas	37
3.4. Transformada Wavelet	38
3.4.1. Análisis de señales no estacionarias	38
3.4.2. Transformada de Fourier de Tiempo Reducido	39
3.4.3. Transformada Wavelet Continua: Análisis Multiresolución	41
3.4.4. El caso discreto	42
3.4.5. Pirámide Multiresolución	42
3.4.6. Codificación de Sub-bandas	43
3.4.7. Transformada Wavelet Discreta	44
3.5. Análisis wavelet de señales neuroeléctricas	45
3.6. Sub-muestreo óptimo	47
4. Métodos de clasificación de características	49
4.1. Introducción	49
4.2. Clasificación propuesta	49
4.2.1. Discriminante lineal de Fisher	49
4.2.2. Spectral Regression Kernel Discriminant Analysis (SRKDA)	51
4.2.3. Conjunto de máquinas de vectores de soporte (SVM)	54
5. Metodología del estudio	59
5.1. Introducción	59
5.2. Preprocesado	59
5.3. Primer estudio: métodos de clasificación de características	60
5.3.1. Método de obtención de parámetros óptimos	60
5.3.2. Validación del método propuesto	61
5.4. Segundo estudio: métodos de extracción de características	61
5.4.1. Primera etapa: evaluación del sub-muestreo óptimo	61
5.4.2. Segunda etapa: selección de la Wavelet Madre óptima	62
5.4.3. Tercera etapa: clasificación con los parámetros óptimos	62
5.4.4. Cuarta etapa: algoritmo de eliminación de canales	62
5.4.5. Algoritmo de eliminación y selección de canales	63
5.4.5.1. Algoritmo usado por los ganadores de la competición	63
5.4.5.2. Algoritmo usado en este estudio	64
6. Resultados	67
6.1. Introducción	67
6.2. Primer estudio	67
6.2.1. Parámetros óptimos de los clasificadores	67
6.2.1.1. LDA	67
6.2.1.2. SRKDA	67
6.2.1.3. Conjunto de SVM	72
6.2.2. Resultados de los clasificadores	73
6.2.2.1. LDA	74

6.2.2.2.	SRKDA	74
6.2.2.3.	Conjunto de SVM	74
6.3.	Segundo estudio	75
6.3.1.	Primera etapa: evaluación del sub-muestreo óptimo	75
6.3.1.1.	LDA	75
6.3.1.2.	Conjunto de 2 SRKDA	76
6.3.1.3.	Conjunto de 19 SVM	76
6.3.2.	Segunda etapa: selección de la Wavelet Madre óptima	76
6.3.2.1.	LDA	76
6.3.2.2.	Conjunto de 2 SRKDA	76
6.3.2.3.	Conjunto de 19 SVM	86
6.3.3.	Tercera etapa: clasificación con los parámetros óptimos	86
6.3.4.	Cuarta etapa: algoritmo de eliminación de canales	90
7.	Discusión de los resultados	95
7.1.	Discusión de los resultados del sub-muestreo	95
7.2.	Discusión de los resultados de la Wavelet Madre óptima	95
7.3.	Discusión de los resultados de la clasificación	95
7.3.1.	Discusión de los resultados con eliminación de canales	98
7.4.	Comparación con otros métodos	101
7.5.	Limitaciones de este estudio	102
8.	Conclusiones y líneas futuras	105
8.1.	Conclusiones	105
8.2.	Líneas futuras	107
	Referencias	108
	Anexos	114
A.	Listado del código desarrollado en Matlab	115

Índice de figuras

1.1.	Ejemplo de electroencefalograma (EEG).	4
1.2.	Esquema de disposición de los electrodos en el sistema internacional 10-20	5
1.3.	Dibujo esquemático de la morfología neuronal y glial.	6
1.4.	Potencial evocado visual (VEP) registrado en el cuero cabelludo occipital	8
2.1.	SCP en función de la intención de mover un cursor hacia arriba (negativos) o hacia abajo (positivos)	13
2.2.	Ritmo sensoriomotor μ usado por un sujeto para el desplazamiento de un cursor hacia arriba o hacia abajo	14
2.3.	Potencial evocado P300. El asociado al objetivo presenta un pico positivo de voltaje	15
2.4.	Esquema de las etapas de tratamiento de una señal en un BCI	16
2.5.	Matriz de caracteres mostrada al usuario	26
2.6.	Asignación de valores de la variable StimulusCode a diferentes intensificaciones de filas o columnas	27
2.7.	Sistema BCI para el control de una silla de ruedas.	30
3.1.	Wavelet B-spline. La fila de arriba muestra una wavelet en el dominio del tiempo en tres escalas; la fila de abajo el espectro en frecuencia de la wavelet en las tres escalas. La traslación en tiempo se muestra en escala media por la figura de línea discontinua.	36
3.2.	Algunos ejemplos de formas de wavelet.	36
3.3.	Plano tiempo-frecuencia correspondiente a la Transformada de Fourier de Tiempo Reducido. Se puede ver como una sucesión de Transformadas de Fourier de un segmento inventanado de la señal (bandas verticales) o como un análisis basado en un banco de filtros (bandas horizontales).	40
3.4.	División del dominio frecuencial para STFT y transformada Wavelet	41
3.5.	Esquema piramidal. Obtención de una aproximación paso-bajo submuestreada y recuperación de la señal.	43
3.6.	Codificación Sub-banda.	44
3.7.	Diagrama de bloques de la Transformada Wavelet Discreta implementada con filtros discretos y sub-muestreo por dos.	45
3.8.	Coefficientes wavelet de un ERP para wavelets a gran y pequeña escala como función de una traslación en tiempo.	46
4.1.	Comparativa entre LDA y FLDA.	51

4.2.	Hiperplano óptimo de separación de dos clases linealmente separables. . .	55
4.3.	Tipos de errores en la clasificación. 1) Patrón dentro de la región de separación pero correctamente clasificado. 2) Patrón incorrectamente clasificado	56
6.1.	Área bajo la curva ROC para un barrido del parámetro de regularización del LDA. El valor óptimo es de 13001. Sujeto A.	68
6.2.	Área bajo la curva ROC para un barrido del parámetro de regularización del LDA. El valor óptimo es de 5001. Sujeto B.	68
6.3.	Curva ROC para el parámetro de regularización óptimo del Sujeto A. LDA.	69
6.4.	Curva ROC para el parámetro de regularización óptimo del Sujeto B. LDA.	69
6.5.	Área bajo la curva ROC para un barrido del parámetro t del SRKDA. El valor óptimo es de 1000. Sujeto A.	70
6.6.	Área bajo la curva ROC para un barrido del parámetro t del SRKDA. El valor óptimo es de 900. Sujeto B.	70
6.7.	Curva ROC para el parámetro t óptimo del Sujeto A. SRKDA.	71
6.8.	Curva ROC para el parámetro t óptimo del Sujeto B. SRKDA.	71
6.9.	Área bajo la curva ROC para un barrido del parámetro C del conjunto de SVM. El valor óptimo de C es 30. Sujeto A.	72
6.10.	Área bajo la curva ROC para un barrido del parámetro C del conjunto de SVM. El valor óptimo de C es 30. Sujeto B.	73
6.11.	Comparativa porcentaje de aciertos de caracteres en función del método de clasificación y el número de iluminaciones utilizadas para la misma. .	75
6.12.	Representación gráfica de los valores obtenidos para el Sujeto A con los tres clasificadores y extracción de características mediante sub-muestreo y Transformada Wavelet.	88
6.13.	Representación gráfica de los valores obtenidos para el Sujeto B con los tres clasificadores y extracción de características mediante sub-muestreo y Transformada Wavelet.	89
6.14.	Representación gráfica de los valores obtenidos para los valores medios del Sujeto A y B con los tres clasificadores y extracción de características mediante sub-muestreo óptimo y Transformada Wavelet con sus respectivas Wavelets Madre.	90
6.15.	Representación gráfica de los valores obtenidos para el sujeto A con LDA, con y sin la aplicación del algoritmo de eliminación de canales.	92
6.16.	Representación gráfica de los valores obtenidos para el sujeto A con SVM, con y sin la aplicación del algoritmo de eliminación de canales.	93
6.17.	Representación gráfica de los valores obtenidos para el sujeto B con el conjunto de SVM, con y sin la aplicación del algoritmo de eliminación de canales.	94
7.1.	Canales descartados por el algoritmo de eliminación de canales para el sujeto A y el clasificador LDA.	99
7.2.	Canales descartados por el algoritmo de eliminación de canales para el sujeto A y el clasificador SVM.	99

7.3. Canales descartados por el algoritmo de eliminación de canales para el sujeto B y el clasificador conjunto de SVM.	100
7.4. Histograma topológico de la clasificación de los canales en el método ganador de la competición, para el Sujeto A y el Sujeto B	100

Índice de tablas

2.1. Resultados para diferentes paradigmas de clasificación, extracción y selección.	25
2.2. Resultados de los tres primeros puestos de la competición BCI 2003	28
2.3. Precisión de algoritmos que usan la base de datos de la III Competición BCI.	29
6.1. Resultados para 15 iluminaciones sobre el conjunto de datos de entrenamiento en función del número de SVM.	73
6.2. Resultados obtenidos mediante la clasificación usando LDA para distinto número de iluminaciones.	74
6.3. Resultados obtenidos mediante la clasificación usando SRKDA para distinto número de iluminaciones.	74
6.4. Resultados obtenidos mediante la clasificación usando un conjunto de 19 SVM para distinto número de iluminaciones.	74
6.5. Resultados obtenidos para el Sujeto A con el clasificador LDA para cada sub-muestreo	77
6.6. Resultados obtenidos para el Sujeto B con el clasificador LDA para cada sub-muestreo	77
6.7. Resultados obtenidos para el Sujeto A con el conjunto de 2 SRKDA como clasificador para cada sub-muestreo	78
6.8. Resultados obtenidos para el Sujeto B con el conjunto de 2 SRKDA como clasificador para cada sub-muestreo	78
6.9. Resultados obtenidos para el Sujeto A con el conjunto de 19 SVM como clasificador para cada sub-muestreo	79
6.10. Resultados obtenidos para el Sujeto B con el conjunto de 19 SVM como clasificador para cada sub-muestreo	79
6.11. Resultados obtenidos para el Sujeto A con el clasificador LDA para cada Wavelet madre y 10 muestras por canal e iluminación	80
6.12. Resultados obtenidos para el Sujeto B con el clasificador LDA para cada Wavelet madre y 8 muestras por canal e iluminación	81
6.13. Resultados obtenidos para el Sujeto A con el conjunto de 2 SRKDA como clasificador para cada Wavelet madre y 10 muestras por canal e iluminación	82
6.14. Resultados obtenidos para el Sujeto B con el conjunto de 2 SRKDA como clasificador para cada Wavelet madre y 8 muestras por canal e iluminación	83
6.15. Resultados obtenidos para el Sujeto A con el conjunto de 19 SVM como clasificador para cada Wavelet madre y 10 muestras por canal e iluminación	84

6.16. Resultados obtenidos para el Sujeto B con el conjunto de 19 SVM como clasificador para cada Wavelet madre y 8 muestras por canal e iluminación	85
6.17. Resultados obtenidos para el Sujeto A con los tres clasificadores con submuestreo y extracción de características mediante Transformada Wavelet óptimos para cada clasificador	86
6.18. Resultados obtenidos para el Sujeto B con los tres clasificadores con submuestreo y extracción de características mediante Transformada Wavelet óptimos para cada clasificador	87
6.19. Resultados medios obtenidos para ambos sujetos con los tres clasificadores	87
6.20. Resultados obtenidos para el Sujeto A con los tres clasificadores para 10 muestras por canal e iluminación, extracción de características mediante Transformada Wavelet y eliminación de canales	91
6.21. Resultados obtenidos para el Sujeto B con los tres clasificadores para 8 muestras por canal e iluminación, extracción de características mediante Transformada Wavelet y eliminación de canales	91
7.1. Comparación de resultados para los distintos métodos de clasificación de este estudio	96
7.2. Comparación de resultados para los distintos métodos de clasificación del estudio previo	96
7.3. Comparativa de los resultados con el método ganador.	101
7.4. Comparación de resultados con otros métodos del estado del arte ordenados por porcentaje de aciertos para 15 iluminaciones.	102

Lista de acrónimos

AAR	Adaptive Autoregressive Model
ALS	Amyotrophic Lateral Sclerosis
BOLD	Blood oxygen level dependent
BCI	Brain Computer Interface
CBP	Checkerboard Paradigm
CNN	Convolutional Neural Networks
CSP	Common Spatial Pattern
ECG	Electrocardiograma
EEG	Electroencefalograma
EG	Electrograma
EMG	Electromiograma
ENG	Electroneurograma
EOG	Electrooculograma
ERD	Event-Related Desynchronization
ERG	Electroretinograma
ErrP	Error Evoked Potential
FLDA	Fisher's Linear Discriminant Analysis
fMRI	Functional Magnetic Resonance Imaging
IB	Ingeniería Biomédica
JADE	Joint Approximate Diagonalization of Eigenmatrices
KDA	Kernel Discriminant Analysis
LDA	Linear Discriminant Analysis

LWT	Lifting Wavelet Transform
MEG	Magnetoencefalograma
MICS	Mutual Information Channel Selection
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristic
RCP	Row-Column Paradigm
SCP	Single Cell Paradigm
SCP	Slow Cortical Potential
SMR	Sensorimotor Rhythm
SOBI	Second Order Blind Identification
SRKDA	Spectral Regression Kernel Discriminant Analysis
SSVEP	Steady State Visually Evoked Potential
SVM	Support Vector Machine
VEP	Visually Evoked Potential
WT	Wavelet Transform

Capítulo 1

Introducción

1.1. Señales Biomédicas

La ingeniería biomédica (IB) es la aplicación de los principios de la tecnología al campo de la medicina y biología. Este campo combina el diseño y las capacidades de resolución de problemas de la ingeniería con las ciencias médicas para avanzar en el tratamiento de enfermedades, incluyendo diagnóstico, monitorización y terapia. Es un campo de estudio relativamente reciente y la investigación y el desarrollo son por tanto de una gran importancia en el mismo.

Las señales biomédicas son observaciones de procesos fisiológicos en organismos, como por ejemplo ritmos neuronales o cardíacos, imágenes de órganos, etc. El procesamiento de estas señales consiste en el análisis de las mismas para obtener información que pueda ayudar a tomar decisiones en diversos campos de la medicina [1].

Estas señales se pueden clasificar a su vez según su descripción matemática en cuatro tipos: deterministas, estocásticas, fractales y caóticas [2], y según su origen en eléctricas, acústicas, mecánicas, biomagnéticas, ópticas, bioquímicas, impedancias e imágenes. Las señales de tipo estocástico son aquellas para las cuales es imposible predecir su valor futuro incluso conociendo valores pasados debido a la presencia de componentes aleatorios. Normalmente las señales biomédicas son de este tipo por diversas razones, tales como el ruido generado por los instrumentos de medición o por la propia naturaleza aleatoria de las mismas.

Una parte importante de las señales biomédicas son las de tipo bioeléctrico, las cuales pueden ser debidas a actividad espontánea o a como respuesta a un estímulo. Entre las que se originan por actividad espontánea destacan el electroencefalograma (EEG)(Figura 1.1), el electrocardiograma (ECG), el electrograma (EG), el electromiograma (EMG), el electroneurograma (ENG), el electrooculograma (EOG) y el electroretinograma (ERG).

En este trabajo nos centraremos en el EEG, que describimos a continuación.

1.2. Electroencefalograma (EEG)

El EEG consiste en el registro de la actividad eléctrica en el cuero cabelludo producida por las neuronas del encéfalo mediante la medición de las fluctuaciones de voltaje que

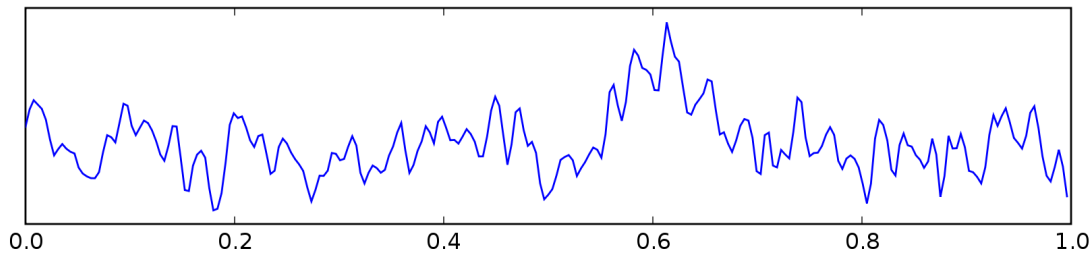


Figura 1.1: Ejemplo de electroencefalograma (EEG).

resultan de las corrientes iónicas entre las neuronas del cerebro [3]. Se usa activamente en la investigación en el campo de la neurociencia. Algunas de sus aplicaciones en el ámbito clínico son la distinción entre ataques epilépticos y otros episodios como síncope o migraña y entre encefalopatías y estados psiquiátricos como la catatonia.

En un EEG convencional, el registro se obtiene colocando una serie de electrodos en el cuero cabelludo mediante un gel conductor para reducir la impedancia. Los nombres y la posición de los electrodos están determinados por el sistema internacional 10-20 [4] para la mayoría de aplicaciones de investigación. Dicho sistema fue desarrollado con el objetivo de estandarizar la reproducibilidad de tal forma que los estudios sobre un sujeto pudieran ser comparados a lo largo del tiempo y con otros sujetos. Cada electrodo tiene una letra para identificar el lóbulo cerebral: Frontal (F), Temporal (T), Parietal (P) y Occipital (O), y un número para identificar el hemisferio. Los nodos de referencia usan la letra A. Los electrodos situados en el hemisferio izquierdo tienen números impares y los situados en el derecho, pares. El número z se usa para los electrodos localizados en el eje central. La Figura 1.2 muestra el esquema de electrodos del sistema 10-20.

Las señales recogidas por estos electrodos tienen unas amplitudes del orden de los μV y están comprendidas en el espectro entre 0.5 y 100 Hz.

Los EEG suelen estar afectados por señales eléctricas de origen no cerebral denominadas artefactos. La amplitud de dichos artefactos puede llegar a ser grande en comparación con la amplitud de las señales corticales de interés. Algunos de los artefactos biológicos más comunes son los inducidos por el ojo (parpadeos, movimientos del globo ocular), cardíacos y musculares.

Los usos habituales del EEG son el diagnóstico de la epilepsia, enfermedades cerebrales como el Alzheimer, encefalopatías, infecciones, tumores, coma y trastornos del sueño [5]. Sin embargo, el uso del EEG como método de diagnóstico de tumores ha decrecido con el advenimiento de técnicas de imagen médica de alta resolución como la tomografía computerizada y la resonancia magnética. El uso del EEG no se limita al diagnóstico de enfermedades. Una aplicación de gran importancia es la detección de potenciales cerebrales, como los potenciales evocados (visuales, P300) y ritmos sensorio-motores entre otros [6]. El registro de estas señales se usa para multitud de aplicaciones, pero la de mayor importancia es permitir la comunicación e interacción de personas con discapacidades motoras graves con el exterior. [7]

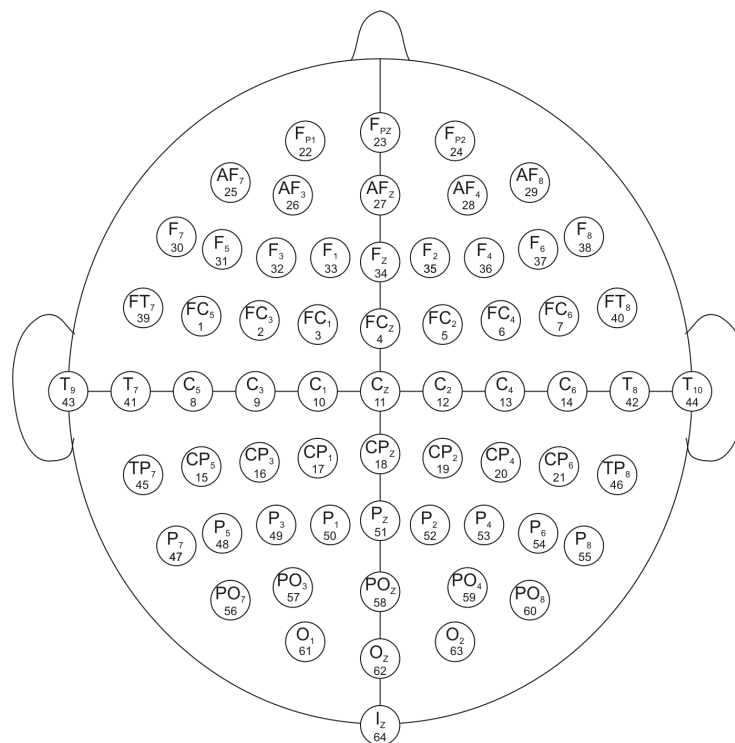


Figura 1.2: Esquema de disposición de los electrodos en el sistema internacional 10-20. [8].

1.3. Aspectos anatómicos y fisiológicos del EEG

El electroencefalógrafo clínico correla funciones, disfunciones y enfermedades del sistema nervioso central (CNS) con ciertos patrones de EEG de forma empírica. Este método es útil y valioso para la práctica clínica. Es de utilidad conocer los procesos elementales que están detrás de las señales EEG para evitar los diagnósticos erróneos y mejorar las interpretaciones [9]. Este trabajo se centra en el estudio de las señales P300, las cuales forman parte del conjunto de potenciales evocados introducidos a continuación.

El sistema nervioso central consiste esencialmente en células nerviosas o neuronas y células gliales, estando estas últimas localizadas entre las neuronas. Como se muestra en la figura 1.3, muchas prolongaciones surgen del cuerpo (soma) de la célula nerviosa. Estas prolongaciones se pueden dividir en dos tipos de acuerdo a su función. La mayoría de las prolongaciones son dendritas que tienen numerosas ramificaciones. Las células tienen además un axón que se puede dividir en múltiples colaterales. Dicho axón habilita el contacto con otras células nerviosas u otros órganos. En el caso de las conexiones interneuronales, el contacto consiste en sinapsis que cubren un gran número de dendritas, soma y conos del axón. Las células gliales se encuentran entre los soma, dendritas y axones de las células nerviosas. Suelen tener muchas prolongaciones que hacen contacto con soma y prolongaciones de otras células nerviosas [9].

Las neuronas son las responsables de la actividad eléctrica del cerebro. Estas células

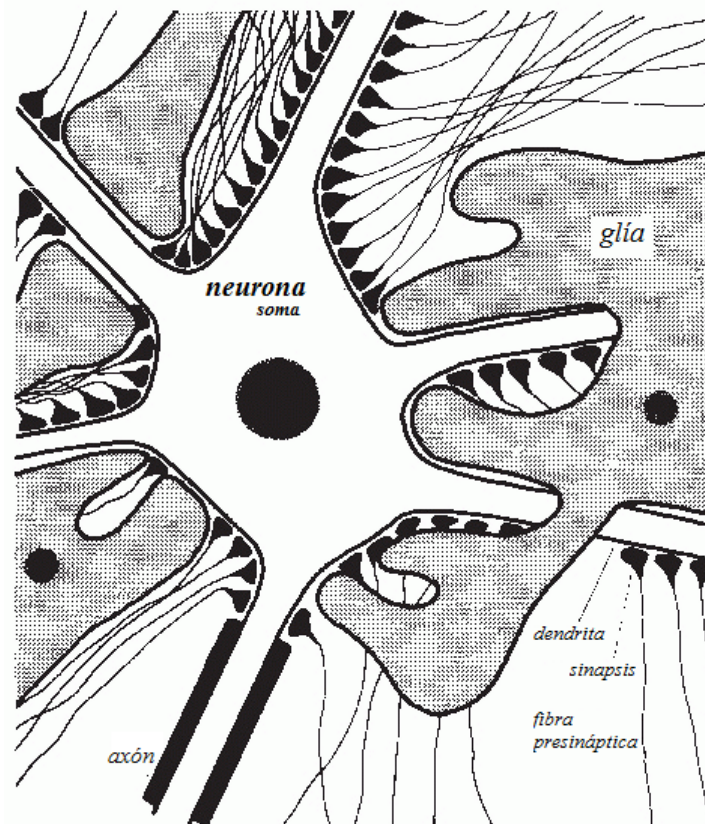


Figura 1.3: Dibujo esquemático de la morfología neuronal y glial.

están cargadas eléctricamente o polarizadas por las proteínas de transporte de las membranas, que bombean iones a través de las mismas. Las neuronas están constantemente intercambiando iones con el entorno extracelular, por ejemplo para propagar potenciales de acción. Los iones de carga similar se repelen entre sí, y cuando muchos iones son expulsados de muchas neuronas al mismo tiempo, expulsan a sus vecinos, los cuales hacen lo mismo con los suyos, y así sucesivamente. Este proceso se conoce como conducción volumétrica.

El potencial eléctrico que se registra en un EEG nunca es el de una neurona individual, ya que éste es demasiado pequeño para ser detectado. La actividad EEG, por tanto, siempre refleja la suma de la actividad sincrónica de miles o millones de neuronas con orientación espacial similar. Si las células no tienen una orientación espacial similar, sus iones no se alinean y no crean ondas.

La actividad EEG en el cuero cabelludo muestra oscilaciones en muchas frecuencias. Muchas de estas oscilaciones tienen rangos de frecuencia característicos y distribuciones espaciales, además de estar asociadas a distintos estados del funcionamiento cerebral (por ejemplo, despertarse y las etapas del sueño). Algunas de las redes neuronales detrás de estas oscilaciones son conocidas y estudiadas, pero muchas otras no [9].

1.4. Potenciales evocados

Los potenciales evocados son potenciales eléctricos registrados del sistema nervioso humano que aparecen tras la presentación de un estímulo, lo que los distingue de los potenciales espontáneos.

La amplitud de los potenciales evocados suele ser pequeña, en un rango de entre un microvoltio a unos pocos microvoltios. Debido a esta baja amplitud suele ser necesario el promediado de estimulaciones repetidas del mismo estímulo, ya que de esta forma se puede eliminar el ruido, de naturaleza aleatoria y por tanto diferente en cada estimulación [10]. En función del tipo de estímulo que los provoca (véase [6]), algunos de los potenciales evocados se pueden clasificar en:

- Potenciales evocados visuales: cambios en el potencial producidos en la corteza visual tras una estimulación luminosa. Su uso principal es el estudio del nervio óptico.
- Potenciales evocados auditivos: estos potenciales son generados en la cóclea y siguen un camino que pasa por el nervio coclear hasta llegar al córtex cerebral. Son evocados por estímulos auditivos.
- Potenciales evocados somatosensoriales: se usan para comprobar el correcto funcionamiento de la médula espinal. Se registran estimulando nervios periféricos como el tibial, normalmente mediante estímulos eléctricos.

En la Figura 1.4 se puede observar un ejemplo de potencial evocado visual.

La información que aportan los potenciales evocados es utilizada, mediante los sistemas que se introducen a continuación, para el desarrollo y diseño de aplicaciones orientadas a mejorar la vida de las personas discapacitadas.

1.5. Brain Computer Interface

Brain Computer Interface (BCI) es un sistema de comunicación mediante el cual mensajes u órdenes que un individuo envía al mundo exterior no pasan por los caminos habituales de salida del cerebro de los nervios periféricos y músculos. Se pueden considerar, por tanto, como sistemas que permiten al cerebro interactuar con el medio sin hacer uso de dichos nervios y músculos. Los BCI se pueden clasificar en dos clases: dependientes e independientes [6].

Los BCI dependientes son aquellos que se basan en actividad cerebral eléctrica provocada mientras que los independientes son aquellos que se basan en señales que aparecen de forma espontánea. Un ejemplo de los primeros son los BCI basados en potenciales evocados visuales (VEP) en los que el usuario mira una letra concreta de una matriz y la señal generada depende de la dirección de la mirada. Entre los segundos destacan los BCI basados en potenciales evocados P300, los cuales aparecen independientemente de la intención del usuario.

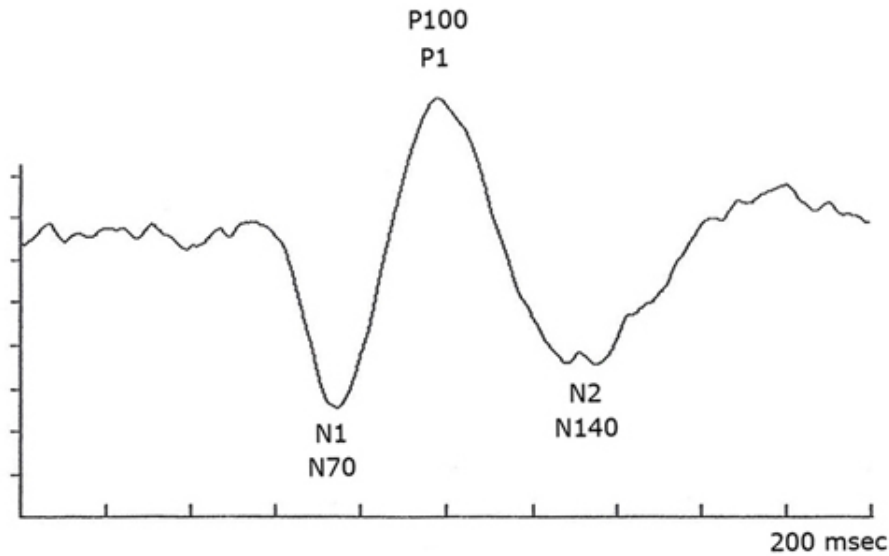


Figura 1.4: Potencial evocado visual (VEP) registrado en el cuero cabelludo occipital [11].

Los BCI basados en EEG traducen la intención del usuario, la cual se encuentra codificada en la actividad cerebral, en acciones de distinto tipo.

Tras ser registradas, las señales se muestrean y digitalizan. Dichas señales son entonces procesadas en dos etapas: extracción de características y clasificación de características. La extracción consiste en la selección de la información más relevante de las señales para que en la posterior etapa de clasificación sea más sencillo discriminar las mismas en clases mediante métodos estadísticos.

Uno de los principales usos de los BCI es el servir como sistemas de comunicación y control para personas con discapacidades motoras severas como lesiones de la médula espinal o esclerosis lateral amiotrófica. Mediante el BCI estos usuarios pueden expresar sus deseos a los cuidadores o incluso operar sistemas procesadores de texto o sillas de ruedas.

1.6. Objetivos del Trabajo Fin de Máster

El objetivo principal de este trabajo es realizar un estudio comparativo de métodos de extracción y clasificación de características, así como encontrar los parámetros óptimos de los mismos para llegar a la máxima precisión en la clasificación posible. Los métodos se aplican sobre las señales BCI que constituyen el conjunto de datos de la III Competición BCI (2004) proporcionado por el Centro Wadsworth (*NYS Department of Health*) [12]. El conjunto de datos sobre el que se trabaja está proporcionado por los organizadores de la competición y se explica con detalle en el capítulo 2. Para alcanzar este objetivo general se deben cumplir los siguientes objetivos específicos:

1. Estudiar los sistemas BCI, sus paradigmas de control, señales asociadas y métodos de tratamiento de las mismas, con hincapié en los P300.

2. Realizar un repaso del estado del arte en el campo de los P300 en los ámbitos de adquisición de señales, presentación de estímulos, extracción de características, clasificación de las mismas y aplicaciones.
3. Estudiar en profundidad la Transformada Wavelet como método de extracción de características, haciendo hincapié en su aplicación a señales neuroeléctricas.
4. Estudiar y aplicar los tres algoritmos de clasificación usados en este estudio : *Linear Discriminant Analysis* (LDA), *Spectral Regression kernel Discriminant Analysis* (SRKDA) y *Support Vector Machines* (SVM).
5. Obtener los parámetros óptimos de los tres algoritmos de clasificación mediante validación cruzada usando la herramienta Matlab[®] y usando como criterio el área bajo la curva ROC (Receiver Operating Characteristic).
6. Obtener el sub-muestreo y Wavelet Madre óptimos para cada uno de los sujetos de la base de datos y clasificadores por separado y aplicar la transformada wavelet como método de extracción de características de forma previa a su clasificación.
7. Aplicar los métodos de extracción y clasificación a las señales de la competición para obtener la separación de las mismas en dos clases: objetivo y no objetivo. La programación y aplicación de los algoritmos se lleva a cabo mediante Matlab[®].
8. Usar la clasificación obtenida para determinar los caracteres objetivo que buscaban los usuarios y establecer el porcentaje de aciertos.
9. Desarrollar y aplicar un algoritmo de selección y eliminación de los canales que menos información de interés aporten a la clasificación. Observar las posibles mejoras en la clasificación que aporte.
10. Realizar una discusión de los resultados y compararlos con los de otros estudios.
11. Extraer conclusiones y determinar posibles líneas futuras de acción.

1.7. Estructura del Trabajo Fin de Máster

Este trabajo se divide en 9 capítulos: Introducción, Sistemas BCI basados en potenciales evocados P300, Métodos de clasificación de características, Extracción de características mediante Transformada Wavelet, Metodología del estudio, Resultados del estudio previo, Resultados del estudio actual, Discusión de los resultados y Conclusiones y líneas futuras.

En el capítulo de introducción se establece un primer contacto con las señales biomédicas con especial hincapié en el EEG y se introduce el concepto de potencial evocado. A su vez se define el concepto de BCI, su finalidad y tipos. Finalmente se explican los objetivos y la estructura del trabajo.

El segundo capítulo introduce en profundidad los BCI basados en potenciales evocados P300, así como un pequeño repaso de otros tipos de señales BCI. También se explican

las etapas de procesado de las señales. Finalmente se hace un repaso del estado del arte del campo de la investigación sobre P300, se introducen las competiciones BCI y se explica los objetivos de la misma y la base de datos con la que se trabaja en este proyecto.

En el tercer capítulo se explican el método de clasificación de características usado por los ganadores de la III Competición BCI; después se explican los tres métodos de clasificación utilizados y cómo se obtuvieron los parámetros óptimos de cada uno de ellos.

El capítulo cuatro introduce el concepto de Transformada Wavelet, haciendo hincapié en su aplicación a señales neuroeléctricas y sus propiedades como método de extracción de características

En el quinto capítulo se detalla la metodología que se va a seguir en este trabajo, así como el algoritmo de selección y eliminación de canales desarrollado, comparándolo con el de los ganadores de la III Competición BCI.

El siguiente capítulo detalla los resultados obtenidos en el estudio previo al aplicar los métodos anteriores a las señales de la competición, con gráficas y tablas de interés.

El capítulo séptimo se centra en los resultados del estudio actual, explicándolos con tablas y figuras etapa a etapa con el fin de obtener una visión completa de los mismos y observar las diferencias entre los sujetos del estudio.

A continuación, en el capítulo octavo se comparan los resultados con los del método ganador y con los de otros métodos actuales que usan la misma base de datos, tanto con aplicación del algoritmo de eliminación de canales como sin él.

En el último capítulo se exponen las conclusiones a las que se ha llegado y las limitaciones del estudio. Tras ello se exponen posibles líneas de investigación futuras que podrían abrirse a partir de dichas limitaciones.

Capítulo 2

Sistemas BCI basados en potenciales evocados P300

2.1. Introducción a los BCI

Un gran número de enfermedades pueden impedir el correcto funcionamiento de los canales neuromusculares por los que el cerebro se comunica y controla su entorno externo, como la esclerosis lateral amiotrófica (ALS), ataques cerebrales, lesiones de la médula espinal, parálisis cerebral y otras enfermedades [6]. Se calcula que estas enfermedades afectan a cerca de dos millones de personas solo en los Estados Unidos [6]. Las personas que estén afectadas de forma más grave pueden perder todo movimiento voluntario de los músculos, incapacitándolos para cualquier tipo de comunicación.

Los sistemas BCI están diseñados para restaurar las funciones sensibles de estas personas, permitiendo que puedan comunicarse e interactuar con el exterior de diversas formas [7].

A lo largo de los últimos 20 años se ha producido un gran incremento en la investigación sobre BCI para comunicación y control. Además un mejor entendimiento del funcionamiento del cerebro, mejores equipos y mayor conocimiento de las necesidades de las personas con discapacidad han ayudado al avance de la misma.

En este capítulo se explicarán los aspectos principales de los sistemas BCI: las señales de control posibles, con hincapié en los potenciales evocados P300, las etapas de procesamiento en los sistemas BCI basados en P300 debido a que son en los que nos centramos en este trabajo y un repaso por el estado del arte en la investigación, incluyendo la descripción de la III Competición BCI (2004) y su base de datos, que se usará en el estudio posterior. Dicho repaso por el estado del arte incluye a su vez apartados dedicados a aplicaciones novedosas de los sistemas BCI P300.

2.2. Paradigmas de control BCI

Existen dos aproximaciones al control de los sistemas BCI, aunque casi todos los sistemas usan una mezcla de ambas [7]:

- (1) aprender a controlar voluntariamente la actividad cerebral mediante aprendizaje y entrenamiento tras el cual el sujeto controla voluntariamente una actividad cerebral específica y
- (2) aprendizaje de máquina (*Machine Learning*), en el cual el que aprende no es el sujeto sino el propio algoritmo.

El primer paradigma (*Biofeedback Approach*) es un procedimiento que mediante retroalimentación de un parámetro aparentemente autónomo intenta adquirir un control voluntario sobre dicho parámetro. Los sujetos reciben información visual, auditiva o táctil sobre su actividad cardiovascular, temperatura, conductancia de la piel, actividad eléctrica cerebral (EEG), etc. y se les pide que incrementen o disminuyan la actividad de interés, lo que eventualmente lleva a un control de dicha actividad.

El segundo paradigma (*Machine Learning Approach*) consiste en unos algoritmos que se adaptan individualmente al usuario que realiza la tarea. Los algoritmos de aprendizaje requieren ejemplos a partir de los cuales puedan inferir la estructura estadística subyacente de un estado cerebral. Por ello se pide a los sujetos que produzcan un estado cerebral repetidamente en una sesión de calibración.

En la práctica los BCI no van a basarse solamente en uno de estos dos paradigmas, aun así no está claro como unirlos en sincronía. Es importante hacer notar, además, que el 20 % de la población no es capaz de clasificar los patrones de activación cerebral, independientemente de la aproximación al entrenamiento [7].

2.3. Señales de control en los BCI

Los BCI actuales difieren en cuanto a cómo se registra la actividad cerebral, cómo se entrena a los sujetos, cómo las señales se transforman en comandos y qué aplicación se provee al usuario. Existen dos métodos principales de registro: invasivos y no invasivos [7].

Los métodos invasivos miden la actividad cerebral en la corteza cerebral o debajo de la misma. Estos métodos tienen muchas ventajas en cuanto a la calidad de las señales y sus dimensiones, pero por otra parte requieren cirugía y tienen problemas potenciales de estabilidad de los implantes e infecciones a largo plazo, por lo que para que se decida por uso estos deben ser sustancialmente mejores que los no invasivos.

La actividad eléctrica cerebral se registra de forma no invasiva mediante el EEG, el magnetoencefalograma (MEG), nivel dependiente de oxígeno en la sangre (blood oxygen level dependent BOLD), o las imágenes de resonancia magnética funcional (fMRI) entre otros. El sistema más usado es EEG debido a su bajo coste en relación a otros métodos, así como su sencillez.

En un BCI típico los participantes son presentados con estímulos o se les pide que realicen actividades mentales específicas mientras la actividad eléctrica cerebral se registra mediante EEG. Algunas características del EEG son reguladas por el usuario (potenciales corticales lentos o SCP, ritmos sensoriomotores o SMR) mientras que otras aparecen

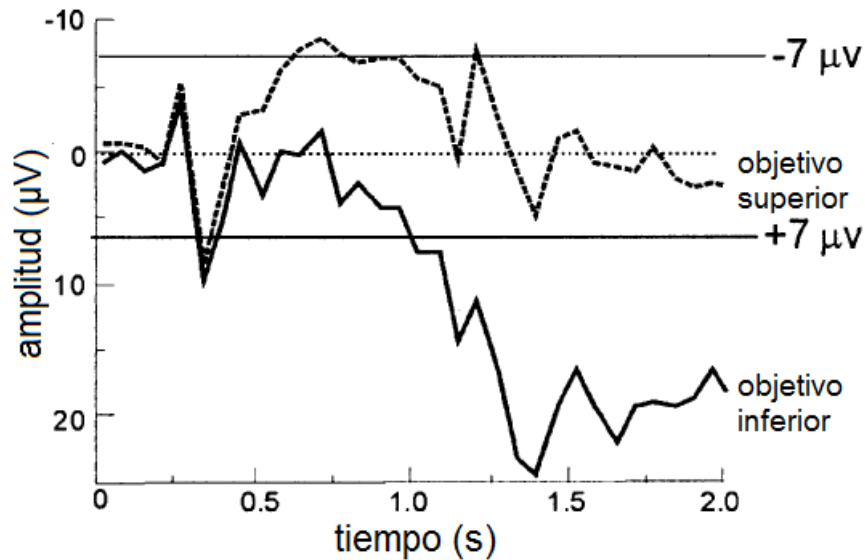


Figura 2.1: SCP en función de la intención de mover un cursor hacia arriba (negativos) o hacia abajo (positivos) [6].

por estimulación visual, táctil o auditiva (potenciales evocados visuales (VEP), potenciales evocados P300). A continuación describimos estas señales.

2.3.1. Potenciales corticales lentos (SCP)

Los potenciales corticales lentos (SCP) son una de las características del EEG de menor frecuencia. Consisten en cambios de voltaje que ocurren entre 0.5 y 10 segundos. Los negativos se asocian con el movimiento y otras funciones relacionadas con la activación cortical, mientras que los positivos están relacionados con activación cortical reducida [6]. En la figura 2.1 se pueden observar SCP en función de la intención de un usuario.

Los SCP se registran por electrodos en el vertex y con referencia en los mastoides. Los SCP se extraen mediante filtrado y se realimentan al usuario mediante una pantalla que muestra una elección en la parte de arriba y una en la de abajo. La selección dura 4 segundos: durante los dos primeros el sistema mide el nivel de voltaje inicial del usuario y durante los dos siguientes el usuario selecciona una de las dos aumentando o disminuyendo el voltaje por una cantidad fijada [6]. El voltaje se muestra como un cursor vertical.

2.3.2. Ritmos sensoriomotores

En las personas despiertas, las zonas corticales primarias a menudo presentan actividad cerebral EEG en el rango de los 8-12 Hz cuando no están realizando actividad sensorial ni motora. Esta actividad se conoce como ritmos μ cuando se centran en el córtex somatosensorial o motor y ritmos α cuando se centran en el córtex visual [6].

La actividad μ agrupa una serie de ritmos que se distinguen por localización, frecuencia y relación con salidas motoras o entradas sensoriales concurrentes. Estos ritmos μ se

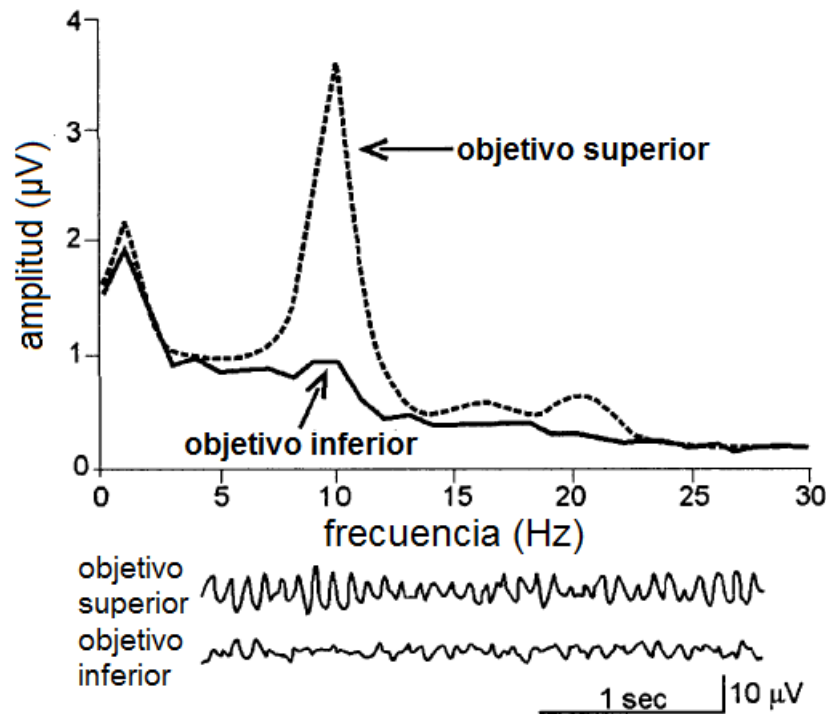


Figura 2.2: Ritmo sensoriomotor μ usado por un sujeto para el desplazamiento de un cursor hacia arriba o hacia abajo [6].

asocian normalmente con los ritmos β en el rango de 18-26 Hz. Algunos de estos ritmos son armónicos de los β , pero otros son separables.

El interés por estos ritmos radica en el hecho de que disminuyen cuando hay movimiento o se prepara el movimiento: este fenómeno se conoce como “desincronización con el evento relacionado o *event-related desynchronization* (ERD)”. Lo opuesto ocurre después del movimiento o con la relajación. Lo más relevante es, sin embargo, que no es necesario el movimiento en sí para que aparezca un ERD, basta con la imagen motora en el pensamiento. En la figura 2.2 se observa un ejemplo de ritmo sensoriomotor.

2.3.3. Potenciales evocados visuales (VEP)

Tras una estimulación visual, se pueden registrar potenciales evocados en el córtex visual en el lóbulo occipital (electrodos 01, 02, 0z en el sistema internacional 10-20 [4]). Un potencial evocado visual se hace estacionario (*steady-state visually evoked potential* o SSVEP) cuando la tasa de los estímulos es superior a 6 Hz [13]. Cuando un sujeto fija su mirada en un objetivo parpadeante, la amplitud del SSVEP aumenta en la frecuencia fundamental del objetivo y sus armónicos de segundo y tercer orden. La resolución en frecuencia de los SSVEP es de 0.2 Hz y el ancho de banda en el que se pueden detectar con fiabilidad es entre 6 y 24 Hz.

Los SSVEP requieren la atención y mirada pero no el entrenamiento del usuario, ya que la respuesta cortical aparece por estimulación visual. Para que aparezcan SSVEP,

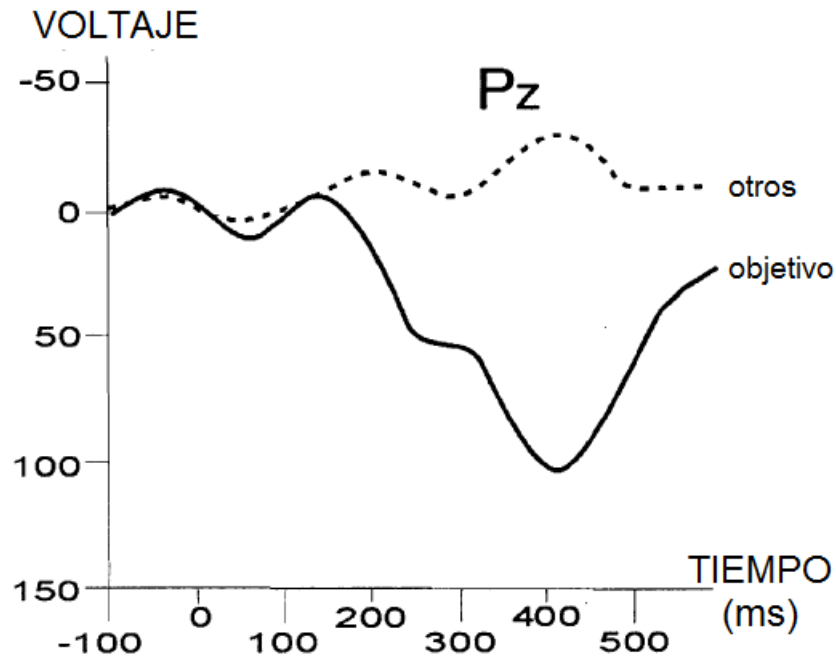


Figura 2.3: Potencial evocado P300. El asociado al objetivo presenta un pico positivo de voltaje [6].

se presentan objetivos con diferentes frecuencias de iluminación en un monitor, con un número de objetivos que varía entre 4 y 48 [7]. Se suelen obtener precisiones de clasificación del 90 % hacia arriba con tasas binarias de 18.37-27.29 bits/min [14]. La desventaja de estos sistemas es que dependen de la capacidad del usuario de mantener la mirada, lo que los hace incompatibles con pacientes con movimiento restringido en el ojo.

2.3.4. Potenciales evocados P300

Los potenciales evocados P300 son picos de voltaje positivos que aparecen en el córtex parietal como respuesta a un estímulo auditivo, visual o somatosensorial infrecuente o significativo a los 300 ms aproximadamente [6]. Aparece normalmente cuando se pide a los usuarios que atiendan a un estímulo infrecuente entre un flujo de estímulos frecuentes [7]. La amplitud del P300 varía en función de características de la tarea como la facilidad de discriminar entre de los estímulos estándar y objetivo, volumen de los tonos en caso de estímulos auditivos, probabilidad del estímulo objetivo, el orden de la secuencia de estímulos y la posición de los electrodos [15]. En la figura 2.3 se muestra un ejemplo de P300.

La aplicación típica que se realiza con este tipo de señales es la selección de letras. Para ello se muestra al usuario una matriz de 6×6 letras y números. En un intento se van iluminando aleatoriamente las filas y las columnas de forma que, tras doce iluminaciones, cada fila y columna se ha iluminado una vez. Para el elemento que el usuario quiere seleccionar se habrán producido dos P300, uno al iluminarse la fila en que se encuentra



Figura 2.4: Esquema de las etapas de tratamiento de una señal en un BCI [17].

y otro al iluminarse su columna. Se promedia entonces la respuesta para cada elemento de la matriz y el potencial que presente un pico de mayor amplitud será el que el usuario tenía como objetivo [16].

2.4. Procesado de señales BCI

Los sistemas BCI constan de 3 etapas a la hora de tratar la señal: adquisición, procesado y aplicación del usuario. A su vez el procesado se divide en dos partes: extracción de características y traducción o clasificación de las mismas. En la figura 2.4 se puede ver este esquema [17]. A continuación se describen estas etapas.

2.4.1. Adquisición de la señal

La señal de entrada se registra típicamente mediante un EEG en el cuero cabelludo. La señal es adquirida mediante electrodos según el estándar 10-20 [4], amplificada, filtrada y digitalizada para su uso mediante programas informáticos. Es importante además almacenar las señales de forma digital para su posterior análisis *offline*, para poder establecer qué métodos de extracción y clasificación de características son más adecuados. En nuestro estudio trabajamos con un BCI no invasivo (registro sobre el cuero cabelludo). Es importante amplificar las señales porque su amplitud es muy baja, del orden de los μV .

2.4.2. Procesado de la señal

a) Extracción de características

Las señales digitalizadas son sometidas a una serie de procedimientos de extracción de características, como filtrado espacial o análisis espectral. Este análisis extrae las características que, en caso de que el procedimiento sea factible, codifican los mensajes u órdenes del usuario. Los BCI pueden usar señales que están en el dominio del tiempo (amplitud potenciales evocados) o en el de la frecuencia (amplitudes ritmos μ y β).

Como el objetivo es conocer la intención del usuario, debe haber una correlación entre los registros electrofisiológicos y la acción que se pretende realizar. La extracción de características debe maximizar la relación señal a ruido, ya que la señal estará afectada por el ruido del sistema nervioso central.

b) Traducción o clasificación de características

En esta etapa se transforman las características extraídas de la señal en señales de control que servirán para controlar un dispositivo. Se lleva a cabo mediante un clasificador que, en función de cómo sea la característica en cada intento, le asocia una selección u otra [6]. El objetivo es que la selección atribuida sea la misma selección que el usuario pretendía realizar.

Los algoritmos lineales son en general más robustos que los no lineales. Los algoritmos lineales asumen que los datos se pueden separar de manera lineal. Para ello dividen el espacio en hiperplanos de separación que generan dos regiones, cada una de las cuales corresponde a una selección distinta. Los algoritmos no lineales, por otra parte, tienen en cuenta más parámetros por lo que la carga computacional es mayor y no son adecuados para una gran cantidad de datos [18].

Es conveniente aplicar inicialmente una regularización de los datos para limitar la influencia de *outliers* y ruido y reducir la complejidad del clasificador. Por otra parte, si se trata de un algoritmo adaptativo, como en el caso de los usados para clasificar P300, es necesario que se adapte a las características de la señal del usuario. Para ello se entrena a los clasificadores en sesiones de entrenamiento en los que las clases son conocidas. Es posible realizar este ajuste de manera *offline*.

2.4.3. Aplicación

En esta etapa, una vez detectadas las características y traducidas en señales de control, se usan las mismas para realizar acciones concretas en aplicaciones. Algunos ejemplos sería seleccionar una letra de la matriz o mover un cursor en una pantalla de arriba abajo.

2.5. Estado del arte en el campo de los potenciales evocados P300

En esta sección se hace un repaso del estado actual, así como de los últimos avances en el campo de la investigación y desarrollo de BCI basados en potenciales evocados P300. Asimismo se hace una comparación de resultados entre diferentes algoritmos de clasificación.

Los campos que se estudian serán los métodos de adquisición de señal, paradigmas de presentación de estímulos, aplicaciones y algoritmos de extracción y clasificación, con un especial énfasis en el último campo. Asimismo se explica el procedimiento de la III Competición BCI (2004) y se hace un estudio comparativo de algoritmos de clasificación y extracción que hagan uso de la base de datos de la misma.

2.5.1. Métodos de adquisición de señal

Tradicionalmente, una de las principales trabas a la hora de realizar un BCI que pueda ser aplicado a situaciones de la vida cotidiana es el engorroso proceso preparatorio,

que incluye preparación de la piel, aplicación del gel conductor y posicionamiento de electrodos, entre otras cosas.

Además del método tradicional de adquisición consistente en electrodos y gel aplicado sobre el cuero cabelludo existe actualmente un casco desarrollado por la empresa EMOTIV que cuenta con la ventaja de que no necesita gel y el consiguiente limpiado de la cabeza del usuario. Además cuenta con electrodos semi-rígidos que permiten un posicionamiento más rápido aunque menos preciso.

Se realizó un estudio comparativo por parte de Mayaud *et al.* del rendimiento del casco EMOTIV con respecto a tres tipos de electrodos (disco, aguja, y cuadrado) en un mismo equipo y sobre diez usuarios [19]. Los resultados obtenidos sugieren que no hay diferencia en cuanto a rendimiento entre el casco y los electrodos de disco, aunque se encontró una cierta tendencia a una caída de rendimiento en el casco EMOTIV tras dos horas de utilización, además de una mayor incomodidad empezando al mismo tiempo que dicha caída. De cualquier forma, es un avance esperanzador a la hora de establecer sistemas de uso común (no estrictamente de investigación). Sin embargo, este sistema también ha recibido críticas por tener un rendimiento muy inferior al de sistemas médicos de mayor coste, y en algunos casos se cuestiona que realmente registre señales EEG, sugiriéndose que en realidad se trata de artefactos musculares u oculares.

2.5.2. Paradigmas de presentación de estímulos

Los sistemas BCI-P300 se basan en el paradigma de la aparición de un estímulo esperado entre muchos no esperados. A día de hoy existen tres paradigmas visuales principales: el original de la fila/columna (RCP), donde las filas y las columnas de una matriz visual parpadean en orden aleatorio mientras el usuario espera a que aparezca su elección. Otro es el de la celda única (SCP), donde los elementos de la matriz parpadean de forma aleatoria (con el que se obtienen potenciales más grandes a costa de velocidad y precisión) y finalmente el paradigma del tablero de damas (CBP), donde grupos de elementos de la matriz parpadean de forma cuasi-aleatoria en lugar de filas y columnas.

Se ha propuesto una variación al sistema de la fila/columna por parte de Akram *et al.* [20] que consiste en integrar un mecanismo de sugerencia de palabras mediante una búsqueda en un diccionario y una selección final de la palabra en función de las sugerencias cuando el número de las mismas se ha reducido a un número determinado. Este sistema reduce el tiempo de entrada de las palabras de forma considerable, si bien solo es útil para aplicaciones de escritura.

Un estudio realizado previamente por Taha Ahi *et al.* [21]) sugiere un posicionamiento alternativo del alfabeto en la matriz bajo el supuesto de que la mayoría de los pares objetivo-error están en la misma fila o columna.

El paradigma del tablero de damas ha sido mejorado recientemente por sus creadores, Townsend *et al.* [22], mediante la propuesta del paradigma de los cinco parpadeos (FFP), con el que se consiguió una tasa de transmisión de información muy superior a la obtenida con el CBP. Básicamente en este paradigma cada elemento de la matriz parpadea cinco veces en lugar de dos como en el CBP en cada secuencia; de esta forma al haber

más candidatos es más fácil seleccionar el deseado por el usuario y se requieren menos secuencias.

Además de los paradigmas visuales también existen paradigmas táctiles y auditivos, los cuales se pueden usar con personas cuya discapacidad sea demasiado severa y no puedan ver correctamente. En el campo de los paradigmas auditivos se propuso un sistema desarrollado por Guo *et al.* [23] para seleccionar un objetivo entre una secuencia aleatoria de dígitos con el que se obtuvo una precisión del 85 %, mayor que lo que habitualmente se consigue en este tipo de paradigmas.

Una forma novedosa de enviar los estímulos auditivos propuesta por Aminaka *et al.* [24] es a través de las apófisis mastoides de sujeto, las cuales fueron estimuladas por un transductor que emitía señales sinusoidales en tonos que se debían reconocer por patrones.

2.5.3. Algoritmos de extracción y clasificación

Uno de los mayores retos a la hora de optimizar el rendimiento de los BCI P300 es mejorar la detección de los potenciales evocados por los estímulos elegidos. Este proceso consiste en la extracción de las características que representan mejor las intenciones del usuario y la clasificación de las mismas en una salida adecuada por un algoritmo.

Se describen a continuación trabajos de investigación relacionados con los métodos de extracción y selección de características aplicados a los BCI basados en potenciales P300 junto con una comparativa entre los resultados obtenidos por los mismos, con especial hincapié en aquellos que hagan uso de la base de datos de III Competición BCI (2004).

a) Extracción

Elsawy *et al.* [25] usaron en su estudio un filtro espacial de media común como extractor de características. Además se estudiaron los resultados para 8 canales y los 64 canales totales para comparar el rendimiento. El conjunto de datos usados para este estudio fue el de la III Competición BCI (2004).

El estudio de Wang *et al.* [26] aplicó transformada de ondícula o wavelet a las señales de la III Competición BCI (2004), en concreto a un subconjunto de 10 canales, obteniéndose una precisión del 89.5 %.

He *et al.* [27] propusieron un método de extracción de características basado en autoregresión multiresolución en de series de tiempo multicanal (MAMTS) y clasificación basada en SVM, llegándose a una precisión del 94.74 % para señales de registro propio por parte de los autores.

En un estudio realizado sobre la base de datos II de la III Competición BCI (2004), Li *et al.* [28] propusieron la combinación de filtro de mediana como método de extracción de características, y Discriminante Lineal Bayesiano (BLDA) como clasificador de características. Esta combinación obtuvo una precisión del 90 % de forma computacionalmente eficiente. En este mismo estudio se combinó BLDA con un análisis basado en wavelet y se obtuvo una precisión media algo inferior (no especificada) pero de forma menos computacionalmente eficiente.

El filtro basado en subespacios utilizado por Shahriari y Erfanian [29] es un filtro espacial diseñado para maximizar el ratio entre las señales sincronizadas con el estímulo objetivo y las señales sincronizadas con el estímulo no objetivo. Los resultados fueron adecuados aunque no especialmente destacados (89.53 % de precisión).

Un algoritmo que combina Wavelets para extracción de características y una combinación de FLDA combinados, aplicado por Salvaris *et al.* [30], dio resultados parecidos a los de los ganadores de la III Competición BCI (95 %) pero con un coste computacional muy inferior para la selección de canal.

Sakamoto y Aono [31] proponen un método de extracción de características basado en el diezmado adaptativo con FLDA como clasificador, obteniendo una precisión media del 96.5 %. Este método de diezmado se basa en dividir las señales base en intervalos no uniformes y obtener la media de cada intervalo para aproximar los datos originales con muestras reducidas. Dichos intervalos se determinan mediante entrenamiento de forma previa.

Mirghasemi *et al.* [32] propusieron en su estudio sobre la base de datos II de la III Competición BCI un filtrado paso banda como método de extracción de características, obteniéndose una precisión del 96.77 %.

Onishi *et al.* [33] realizaron un estudio en el que proponen un método de extracción de características basado en descomposición tensorial. Este método se diferencia de los tradicionales en que extrae simultáneamente información espacial y temporal dominante, mientras que normalmente se extrae información temporal exclusivamente (aunque algunos estudios realizan selección de canal mediante algoritmos de eliminación recursiva [12]). El método realiza una tensorización mediante expansión de datos para luego reducir la dimensionalidad. En este caso se obtuvo una tasa de aciertos del 92 %

Çelik y Anca [43] aplicaron aproximación multiresolución como método de extracción de características a la base de datos II de la III Competición BCI, obteniendo los coeficientes correspondientes. Mediante un clasificador de tipo FLDA se obtuvo una precisión del 91 % mediante canales predefinidos y del 94.5 % mediante una transformación wavelet previa.

Li *et al.* [28] estudiaron la aplicación del filtro de mediana como método de extracción de características, acompañándolo de un clasificador consistente en un LDA Bayesiano. La base de datos fue el conjunto de datos II de la III Competición BCI (2004). Los resultados de este estudio fueron del 90 % de aciertos. La transformada de ondícula o transformada wavelet es una herramienta de análisis de señales que se puede usar para maximizar la precisión del clasificador P300 gracias al conocimiento de la localización espacial y frecuencial de estos potenciales (González *et al.* [42]). Se usó el algoritmo *Lifting Wavelet Transform* (LWT) como método de extracción de características y se obtuvieron resultados similares a otros métodos (un 67.5 %), pero podrían mejorar con futuros refinamientos: por ejemplo se podrían probar otras topologías con pasos y coeficientes de filtro adicionales.

b) Selección

Elsawy *et al.* [25] realizaron un estudio sobre la aplicación de PCA (Principal Component Analysis) a un clasificador P300 basado en un conjunto de clasificadores lineales (LDA, LDA de Fisher y SVM lineal). Tras ello se aplicó PCA a los vectores de características concatenados (todos los segmentos de datos para todos los canales, de un solo carácter se concatenan) y se les aplicó PCA. Se estudiaron dos casos diferentes: con y sin diezmado previo a la aplicación de PCA. Los mejores resultados se obtuvieron para 64 canales, PCA con diezmado previo y un conjunto de FLDA's (91.5 %).

Un estudio realizado por Wang *et al.* [26] usó como método de selección de características ICA (Independent Component Analysis). Como clasificador se usó un conjunto de SVM. Se aplicó ICA y un proceso complejo basado en transformada wavelet como extracción de características y computación de la distancia de Fisher entre señales con y sin P300. Se combinaron las características óptimas en tiempo y frecuencia. Se obtuvo una precisión del 85 % para el sujeto A con una reducción de datos del 62.5 % y del 94 % para el sujeto B con una reducción de datos del 64.3 % (en ambos casos respecto a las señales sin procesar), lo que hace un total de precisión del 89.5 %.

Los algoritmos genéticos se pueden usar para obtener aspectos relevantes de una señal de forma automática. Dal Seno y Matteucci [34] aplicaron un algoritmo genético propio al conjunto de datos IIB de la II Competición BCI (2004) y a otro propio, clasificando mediante un clasificador logístico. Se usaron 10 canales de los 64. En función de distintos parámetros del algoritmo genético se obtuvieron distintos resultados. Sin embargo, en el estudio no se especifica la precisión obtenida sobre este conjunto de datos, sino solo sobre el propio, llegando al 90 % para un sujeto de los 7, aunque los resultados son muy variables entre los mismos.

Recientemente se llevó a cabo un estudio comparativo de tres métodos de extracción de características por parte de Amini *et al.* [35]. A continuación se describen los métodos estudiados:

- La “segmentación inteligente” consiste en segmentar en intervalos no uniformes de los datos del EEG adquiridos en los electrodos y calcular la media de cada segmento. La idea detrás de esta estrategia se basa en la suposición de que un submuestreo efectivo debería proporcionar una distinción clara entre clases (objetivo y no objetivo).
- El CSP (Common Spatial Pattern) es una técnica de filtrado espacial que encuentra las direcciones de discriminación óptima entre dos clases mediante la varianza: maximiza la varianza de una condición y al mismo tiempo minimiza la de la otra. También es posible combinar los dos métodos anteriores (primero CSP y luego reducción de muestras con segmentación inteligente) obteniéndose una precisión del 85.20 %. Combinando características de los grupos anteriores se llegó al 97.40 % de aciertos.
- Turnip y Widyotriamo [36] realizaron un estudio sobre una base de datos propia en la que combinaron un FLDA como clasificador con tres métodos de selección

de características: *Adaptive Autoregressive Model* (AAR), *Joint Approximate Diagonalization of Eigenmatrices* (JADE) y *Second Order Blind Identification* (SOBI). Los resultados de precisión obtenidos para cada uno de los tres métodos respectivamente fueron 96.16 %, 94 % y 93.36 %.

En el estudio de Onishi et al. [33] se empleó como método de selección de características un algoritmo de reducción de la dimensionalidad basado en High Order Discriminant Analysis (HODA). Se aplicó este algoritmo a la base de datos II de la III Competición BCI con LDA como clasificador, llegando a un porcentaje de aciertos del 92 %.

El trabajo realizado por Liu et al. [37], el cual obtuvo el tercer puesto en el conjunto de datos II de la III Competición BCI (2004) se basó en la selección de características basada en PCA y clasificación mediante el criterio del peso T (T-Weight). Es importante destacar que este trabajo realizó la media de las 15 iluminaciones como reducción de características. El porcentaje de acierto obtenido fue del 90 %.

c) Clasificación

El Discriminante Linear de Fisher (LDA) es un mecanismo de clasificación cuyo propósito es crear un vector discriminante que separe dos clases lo mejor posible. Usando un LDA típico como clasificador en el estudio de Amini *et al.* [35] se llegó a la conclusión de que la segmentación inteligente producía los mejores resultados (95.05 % de precisión). La combinación de las mejores características de los métodos anteriores dio como resultado una precisión del 97.4 %.

Un método novedoso de clasificación de P300 es el de las redes neuronales convolucionales (CNN), aplicado por Cecotti y Gräser [38], la topología de las cuales se adapta a la detección de potenciales P300 en el dominio del tiempo. Las CNN son redes neuronales (modelos computacionales basados en el sistema nervioso central de animales) normalmente usadas para reconocimiento de imágenes. Consisten en múltiples capas de pequeñas colecciones de neuronas que miran a pequeñas porciones de la imagen de entrada, las cuales luego se superponen para obtener una representación mejor de la imagen original, de esta forma toleran traslaciones de la imagen. El hecho de que estas redes se usen para tareas de clasificación las hace ideales para los P300, especialmente porque las señales EEG son muy variables en el tiempo y respecto a las personas. Los resultados obtenidos fueron buenos para un número pequeño de electrodos (el 95.5 %, algo inferior a la precisión del método ganador de la tercera competición BCI).

Mediante un LDA Bayesiano, una extensión del LDA de Fisher que efectúa una regresión de forma Bayesiana de forma computacionalmente eficiente, aplicada por Li *et al.* [28], se obtuvo una precisión media del 90 %.

Recientemente están cobrando una especial importancia los sistemas de basados en información mutua (MICS). Shahriari y Erfanian [39] propusieron un método de selección de canal que hace uso de MICS, seleccionando de forma iterativa un canal con la máxima dependencia respecto a la clase objetivo y mínima dependencia con los canales seleccionados previamente. De esta forma se alcanzó una precisión media del 96.9 %.

Los intentos de optimizar el sistema de deletreador P300 en general no tienen en cuenta los conocimientos existentes sobre el lenguaje, es decir, tratan las selecciones de caracteres como elementos independientes elegidos de un grupo sin información previa. Sin embargo, se puede usar la información sobre el lenguaje, como hicieron Speier *et al.* [40] para crear suposiciones previas sobre los caracteres a elegir. Modelando el deletreador P300 como un modelo oculto de Markov o HMM (proceso de Markov de parámetros desconocidos que se deducen a partir de los observables) se supone el deletreado como un proceso secuencial donde la selección de cada carácter (cada estado) está influenciada por las anteriores selecciones, y se deduce la secuencia óptima de caracteres (los estados no visibles) usando el algoritmo de Viterbi. Los resultados obtenidos mejoraron notablemente el rendimiento del sistema de comunicación en cuanto la velocidad de introducción de caracteres y la ITR (32.33), aunque el porcentaje de precisión fue del 88.34 %.

Los potenciales evocados por errores (ErrP) en un EEG son generados por la percepción del sujeto de un error. Se estudió la posibilidad de usar estos potenciales evocados para mejorar el rendimiento del deletreador P300 por parte de Combaz *et al.* [41], mediante la eliminación de falsos positivos: la teoría es que cuando el usuario se diera cuenta de que el sistema no ha reconocido correctamente el carácter deseado aparecería un ErrP, lo cual se usaría para cancelar la selección. La desventaja es que se requiere un largo entrenamiento para que este sistema funcione correctamente y el porcentaje de acierto (90.55 %) no es especialmente destacable.

En un estudio de Mirghasemi *et al.* [32] se ha llegado a la conclusión de que el preprocesamiento de datos, la extracción de características y la clasificación no deberían ser procesos aislados. En este estudio se demostró que para cada algoritmo de clasificación existe otro algoritmo de preprocesamiento adecuado, llegándose a una precisión del 96.77 % con una combinación de filtrado paso banda, Principal Component Analysis (PCA) y FLD.

En la tabla 2.1 se presenta una comparativa de resultados.

Artículo	Nº	Extracción	Selección	Clasificación	Base de Datos	Precisión
Elsawy <i>et al.</i> (2013) [25]	2	Diezmado	PCA	Conjunto de FLDA	Competición BCI III	91.5 %
Wang <i>et al.</i> (2014) [26]	2	Transformada wavelet	ICA	Conjunto de SVM	Competición BCI III	89.5 %
Dal Seno y Matteucci (2008) [34]	7	Ninguno	Algoritmo genético	Clasificador logístico	Propia	90 %
He <i>et al.</i> (2010) [27]	1	MAMTS	Ninguno	SVM	Propia	94.74 %
Li <i>et al.</i> (2012) [28]	2	Filtro de mediana	Ninguno	BLDA	Competición BCI III	90 %

Artículo	Nº	Extracción	Selección	Clasificación	Base de Datos	Precisión
Amini <i>et al.</i> (2012) [35]	2	Segmentación inteligente	Ninguno	FLDA	Competición BCI III	95.05 %
Amini <i>et al.</i> (2012) [35]	2	Common Spatial Patern (CSP)	Ninguno	FLDA	Competición BCI III	82.45 %
Amini <i>et al.</i> (2012) [35]	2	Segmentación inteligente + CSP	Ninguno	FLDA	Competición BCI III	85.20 %
Amini <i>et al.</i> (2012) [35]	2	Combinación de características anteriores	Ninguno	FLDA	Competición BCI III	97.40 %
Shahriari y Erfanian (2013) [29]	5	Filtros basados en subespacios	Ninguno	SVM	Propia	89.53 %
Combaz <i>et al.</i> (2011) [41]	9	Filtro Butterworth y submuestreo	Ninguno	SVM	Propia	90.55 %
Cecotti y Gräser (2011) [38]	2	Submuestreo, filtrado y normalización	Ninguno	Redes neuronales convolucionales	Competición BCI III	95.5 %
Turnip y Widyotriatmo (2013) [36]	8	Ninguno	SOBI	FLDA	Propia	93.36 %
Turnip y Widyotriatmo (2013) [36]	8	Ninguno	JADE	FLDA	Propia	94 %
Turnip y Widyotriatmo (2013) [36]	8	Ninguno	AAR	FLDA	Propia	96.16 %
Li <i>et al.</i> (2012) [28]	2	Filtro de mediana	Ninguno	LDA Bayesiano	Competición BCI III	90 %
Speier <i>et al.</i> (2013) [40]	15	Ninguno	Ninguno	Modelo oculto de Markov	Propia	88.34 %
González <i>et al.</i> (2013) [42]	2	Lifting Wavelet Transform	Ninguno	SVM + Particle Swarm Optimization	Competición BCI III	67.50 %

Artículo	Nº	Extracción	Selección	Clasificación	Base de Datos	Precisión
Shahriari y Erfanian (2011) [39]	2	Ninguno	Información mutua	SVM con núcleo lineal	Competición BCI III	96.90 %
Mirghasemi <i>et al.</i> (2006) [32]	2	Filtrado paso banda	PCA	FLDA	Competición BCI III	96.77 %
Salvaris <i>et al.</i> (2009) [30]	2	Transformada Wavelet	Ninguno	Conjunto de FLDs	Competición BCI III	95 %
Sakamoto y Aono (2009) [31]	2	Supervised Adaptive Downsampling	Ninguno	FLDA	Competición BCI III	96.50 %
Onishi <i>et al.</i> (2012) [33]	2	Tensorización mediante expansión de datos	HODA	LDA	Competición BCI III	92 %
Çelik y Anca (2010) [43]	2	Transformada wavelet + Aproximación multiresolución	Ninguno	FLDA	Competición BCI III	94.50 %
El Dabbagh y Fakhr (2011) [44]	2	Filtro Chebyshev orden 8	Ninguno	SVM con pesos	Competición BCI III	97 %
Tomioaka y Haufe (2008) [45]	2	Filtrado paso bajo	Ninguno	Regularización canales con norma euclídea columnas	Competición BCI III	96.50 %
Liu <i>et al.</i> (2005) [37]	2	Promediado 15 iluminaciones	PCA	Criterio <i>T-Weight</i>	Competición BCI III	90 %

Tabla 2.1: Resultados para diferentes paradigmas de clasificación, extracción y selección.

2.5.3.1. III Competición BCI (2004)

Las competiciones BCI son eventos periódicos realizados desde 2001 cuyo objetivo es validar métodos de procesamiento de señal y de clasificación para BCI. Se provee a los participantes de una serie de sets de datos registradas de usuarios en sesiones de entrenamiento

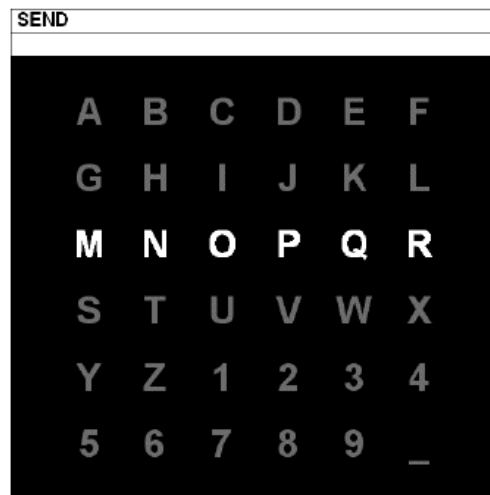


Figura 2.5: Matriz de caracteres mostrada al usuario [8].

y test, cada uno de los cuales tiene objetivos específicos.

En el caso que nos interesa (conjunto de datos II de la III Competición BCI, 2004), las señales de dicho conjunto eran un registro completo de potenciales evocados P300 obtenidos mediante el sistema BCI2000 para un total de dos sujetos. El objetivo era predecir el carácter correcto que el usuario deseaba seleccionar entre los del tablero tradicional bajo el paradigma de la fila/columna de Farwell y Donchin [35]. Este algoritmo consiste en que al usuario se le presenta una matriz 6×6 de caracteres (ver Figura 2.5). El usuario debía fijarse en los caracteres de una palabra fijada por los investigadores. Todas las filas y columnas de la matriz se iluminaron de forma sucesiva y aleatoria con una frecuencia de 5.7 Hz. Dos de cada 12 intensificaciones de filas o columnas contenían el carácter deseado; en estos casos las respuestas evocadas por los estímulos son diferentes de las que no los contienen.

Estas señales fueron tratadas de la siguiente forma: se digitalizaron con una frecuencia de muestreo de 240 Hz y a continuación se les aplicó un filtrado paso banda de 0.1-60 Hz. Los canales son los habituales del sistema internacional 10-20 [4]. Para la captura se usó el software BCI2000 [8]. Cada sujeto realizó 5 sesiones, consistentes en diferentes secuencias. En cada secuencia el sujeto centraba su atención en una serie de caracteres.

El display evolucionaba de la siguiente forma: primero se mostraba la matriz (Figura 2.5) durante 2.5 segundos (en este tiempo todos los caracteres tenían la misma intensidad). Tras ello, cada fila y columna de la matriz se iluminaban de forma aleatoria durante 100 ms (12 estímulos: 6 filas y 6 columnas). Tras cada intensificación la matriz no se iluminaba durante 75 ms. Cada set de 12 intensificaciones se repetía 15 veces para cada selección de carácter, es decir, cada combinación de fila y columna se intensificaba 15 veces haciendo un total de 180 intensificaciones de cada selección de carácter. Al pasar a la siguiente selección de carácter la matriz no se iluminaba durante 2.5 s para informar al usuario de que tenía que centrarse en el siguiente carácter de la palabra.



Figura 2.6: Asignación de valores de la variable StimulusCode a diferentes intensificaciones de filas o columnas [8].

2.5.3.2. Conjunto de datos de la competición

Para la competición estos datos se convirtieron en 4 ficheros Matlab[®] (*.mat), un entrenamiento (85 caracteres) y un test (100 caracteres) para cada uno de los dos sujetos A y B. Los datos se guardaron en *precisión simple*. Para cada *.mat, la señal EEG (obtenida mediante el casco estándar) se organiza en una gran matriz (*Signal*). Es importante saber que para evitar que los participantes en la competición adivinaran las palabras, los caracteres buscados fueron desordenados en los sets. El resto de variables son las siguientes:

Flashing: 1 cuando la fila/columna se intensificaba, 0 si no.

StimulusCode: 0 cuando no se intensifica ninguna fila/columna.

1... 6 para columnas intensificadas de izquierda a derecha.

7... 12 para filas intensificadas de arriba abajo (Figura 2.6).

StimulusType: 0 cuando ninguna fila/columna se intensifica o la fila/columna intensificada no contiene el carácter deseado.

1 cuando la fila/columna contiene el carácter deseado.

El propósito de esta variable es proveer de un acceso fácil a las etiquetas en los sets de entrenamiento para separar las respuestas que contenían el carácter deseado de las que no.

TargetChar: La etiqueta correcta para cada carácter buscado en los datos de entrenamiento.

Las dos últimas variables lógicamente solo están disponibles en las matrices de entrenamiento.

El algoritmo ganador fue el que consiguió un mayor número de aciertos para el conjunto de datos de test. Los tres primeros clasificados se muestran en la tabla 2.2.

Autor	Método	% de aciertos (15 ilumin.)	% de aciertos (5 ilumin.)
1. A. Rakotomamonjy	SVM lineal con selección de canales específicos para cada sujeto	96.50 %	73.50 %
2. L. Yandong	SVM lineal con selección de canales específicos para cada sujeto	90.50 %	55.00 %
3. Z. Zongtan	PCA (Principal Component Analysis)	90.00 %	59.50 %

Tabla 2.2: Resultados de los tres primeros puestos de la competición BCI 2003 [46].

El método ganador fue el propuesto por Rakotomamonjy y Guigue. Dicho método está basado en un conjunto de SVM lineales diferente para cada usuario, usando para construirlos los datos de entrenamiento proporcionados y acompañado de un proceso de selección de canales que se adapta a las diferentes respuestas de cada usuario [12].

En la tabla 2.3 se recogen los resultados, ordenados por porcentaje de aciertos, de una serie de algoritmos que usan el conjunto de datos II de la III competición BCI.

Autor	Método	% de aciertos
1. Amini <i>et al.</i> (2012) [35]	Segmentación inteligente, CSP + LDA de Fisher (Combinación de características)	97.40 %
2. El Dabbagh y Fakhr (2011) [44]	Filtro Chebychev orden 8 + Conjunto de SVMs son pesos	97 %
3. Shahriari y Erfanian (2011) [39]	Información mutua + SVM con kernel lineal	96.90 %
4. Mirghasemi <i>et al.</i> (2006) [32]	Filtrado paso banda + PCA y FLD	96.77 %
5. Rakotomamonjy y Guigue (2008) [12]	Filtrado paso banda + SVM lineal con selección de canales específicos para cada sujeto	96.50 %
6. Sakamoto y Aono (2009) [31]	Supervised Adaptive Downsampling + FDA	96.50 %
7. Tomioka y Haufe (2008) [45]	FPB + Regularización de canales mediante la norma euclídea de las columnas	96.50 %
8. Cecotti y Gräser (2011) [38]	Submuestreo, filtrado y normalización + CNN	95.50 %

	Autor	Método	% de aciertos
9.	Amini <i>et al.</i> (2012) [35]	Segmentación inteligente, LDA de Fisher	95.05 %
10.	Salvaris <i>et al.</i> (2009) [30]	Transformada Wavelet + Conjunto de FLDA	95.00 %
11.	Çelik y Anca (2010) [43]	Aproximación multiresolución + FLD	94.50 %
12.	Onishi <i>et al.</i> (2012) [33]	Tensorización mediante expansión de datos + LDA	92 %
13.	Elsawy <i>et al.</i> (2013) [25]	Diezmado + PCA + Conjunto de FLDA	91.5 %
14.	Li <i>et al.</i> (2012) [49]	Filtro de mediana + LDA Bayesiano	90.00 %
15.	Wang <i>et al.</i> (2014) [26]	Algoritmo genético + Clasificador logístico	89.5 %
16.	Amini <i>et al.</i> (2012) [35]	Segmentación inteligente, CSP + LDA de Fisher	85.20 %
17.	Amini <i>et al.</i> (2012) [35]	CSP + LDA de Fisher	82.45 %

Tabla 2.3: Precisión de algoritmos que usan la base de datos de la III Competición BCI.

2.5.4. Aplicaciones

Se propone en este apartado un repaso por algunas aplicaciones actuales de sistemas BCI basados en P300, con el propósito de ofrecer una visión global de las posibilidades que ofrecen estos sistemas incluso más allá de la ayuda a discapacitados.

Uno de los principales problemas a la hora de controlar vehículos motorizados por BCI basado en P300 es la baja tasa de información y el esfuerzo que implica usarlo de forma continuada. Para evitar estas dificultades, Rebsamen *et al.* [47] desarrollaron una silla de ruedas en la cual se selecciona un destino entre una serie de localizaciones prefijadas de forma que ésta se desplaza por caminos virtuales predeterminados de forma segura, permitiendo al usuario parar en cualquier momento. Esta forma de abordar el problema supone una clara reducción del esfuerzo mental por parte del usuario, que solamente tiene que prestar atención a la pantalla de selección una vez, en lugar de tener que concentrarse de forma continuada, como sería el caso si el sistema estuviera basado en controles clásicos (izquierda, derecha, adelante).

Para poder parar la silla de ruedas en cualquier momento los autores proponen dos

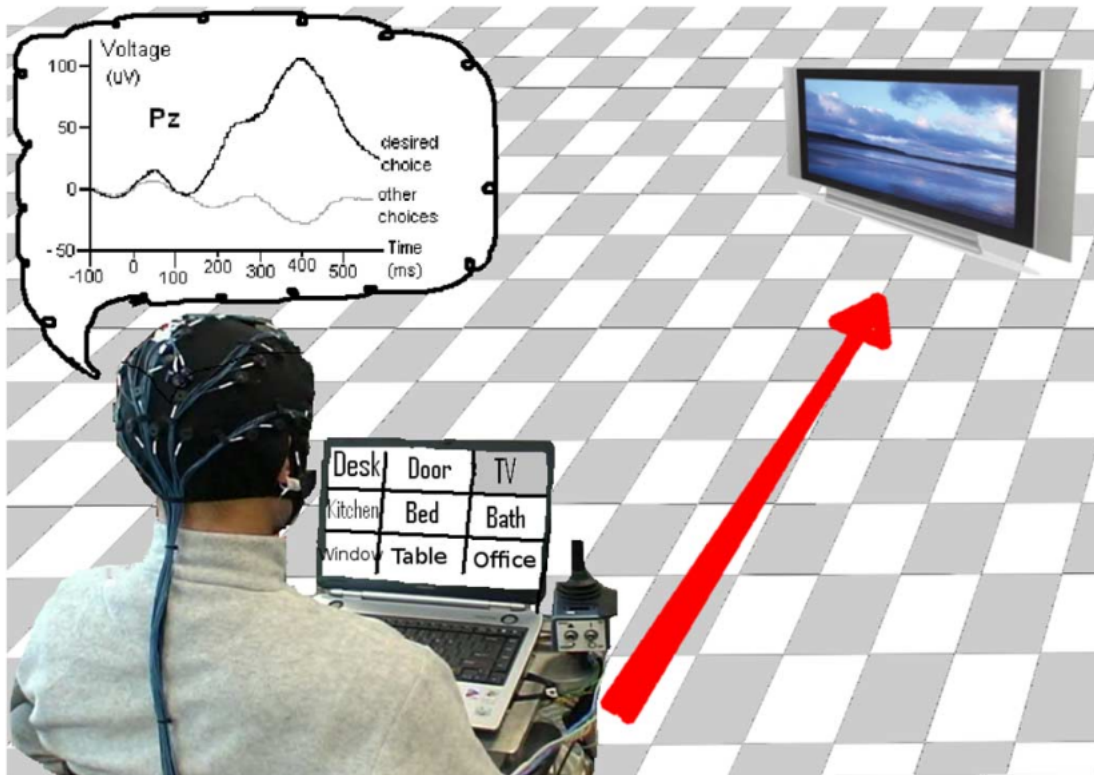


Figura 2.7: Sistema BCI para el control de una silla de ruedas: el usuario selecciona el destino (en este caso la televisión) usando el BCI P300 y la silla de ruedas sigue un camino guiado hasta el mismo. En el monitor del ordenador aparece el menú donde los elementos se iluminan aleatoriamente. Si el usuario centra su atención en un elemento, gracias a los potenciales P300 se seleccionará como destino dicho lugar [47].

BCIs más rápidos: uno basado en un algoritmo P300 rápido con solo un elemento y otro basado en ritmos μ y β . La figura 2.7 muestra un esquema simplificado de este sistema.

En los últimos tiempos están empezando a aparecer sistemas BCI híbridos, como el que combina P300 con potenciales relacionados con desincronizaciones (ERD) (Riechmann *et al.* [48]). Este sistema propone controlar un cursor 2D usando la combinación de ambos potenciales, de una dimensión cada uno, alcanzándose dicho objetivo satisfactoriamente.

Otro sistema BCI híbrido propuesto por Li *et al.* [49] combina P300 con ritmos sensoriomotores para la misma aplicación, de forma que se puedan controlar las dos direcciones del espacio bidimensional de forma simultánea e independiente. Además el sistema permite movimiento en direcciones arbitrarias del cursor, a diferencia del tradicional movimiento discreto, de forma que los P300 controlan los movimientos verticales y los ritmos sensoriomotores determinan la velocidad y dirección del movimiento horizontal.

El desarrollo de la robótica en combinación con los sistemas BCI permite avances como el presentado por Yun *et al.* [50]. En este trabajo se expone un sistema BCI para el control de un robot humanoide. Debido a que los BCI basados en P300 son uno de los

sistemas de este tipo que menos fatiga causan a los usuarios por uso continuado es un candidato ideal para este objetivo. El sistema se basa en comandos de control avanzados que se complementan con los propios sensores del robot, los cuales le permiten explorar sin peligro de colisión, así como sistemas de reposicionamiento en caso de caída.

Tradicionalmente la investigación en torno a la aplicación de los P300 se ha centrado exclusivamente en los usuarios con discapacidad motora. Por esta razón resultan interesantes estudios recientes de Rosenfeld *et al.* [51] en los que se propone usar estos potenciales para otro propósito, como es el de revelar información oculta. Donchin y Farwell ya vieron el potencial de los P300 como medio para detectar información oculta en 1991 [52]. En el estudio propuesto se realizaron dos experimentos, el primero de los cuáles era una simulación de un crimen con veinte voluntarios. El segundo experimento contaba solamente con cuatro sujetos culpables de pequeñas transgresiones. En ambos experimentos se les mostraba a los sujetos tres tipos de estímulos: (1) elementos sonda que solo el culpable y gente familiarizada con el crimen reconocerían; (2) elementos irrelevantes que no estaban relacionados con el crimen pero eran de la misma categoría que la sonda; (3) elementos objetivo que no tenían relación con el crimen pero a los que el sujeto tenía que responder según las instrucciones que se le proporcionaban. A los sujetos se les dijo que presionaran un botón de sí a los objetivos y un botón de no al resto de estímulos.

Los sujetos del primer experimento habían participado en escenarios simulados de crímenes de espionaje en los que se pasaron maletas a agentes en operaciones con nombres concretos. Los detalles de estas actividades generaban seis categorías de estímulos, por ejemplo el nombre de la operación. Para cada categoría, el nombre real podía ser por ejemplo operación “burro”. El resto de los seis estímulos serían otros nombres de animales incluyendo la sonda, cuatro irrelevantes y un nombre objetivo. Las seis categorías, con seis estímulos cada una, daban un total de 36 elementos que se presentaban de forma aleatoria dos veces por bloque. Tras cada bloque, los estímulos se volvían a presentar en otro orden aleatorio y se presentaban de nuevo para un total de cuatro bloques. El crimen simulado se cometía el día antes de las pruebas P300. De los veinte casos de culpabilidad se llegó a decisiones correctas en todos menos dos, que no se pudieron clasificar de manera no ambigua como inocente o culpable. A pesar del éxito de esta prueba hay que tener en cuenta que los sujetos habían sido entrenados para recordar los detalles de sus crímenes, lo que no tiene mucha validez de campo (no se puede entrenar a un sujeto en los detalles de un crimen que está negando). De los sujetos inocentes, solo 17 de los 20 fueron correctamente clasificados, mientras que el resto fueron clasificados como “indeterminados”, lo que significa que el 12.5 % de los casos quedaron sin clasificar [52].

El segundo experimento solo contaba con cuatro sujetos voluntarios, los cuales eran todos culpables de pequeños crímenes en el campus universitario. Los detalles de sus crímenes se detectaron bien con P300, pero teniendo en cuenta que ya habían admitido previamente sus delitos probablemente ya se habían “entrenado” con investigadores del campus, profesores, padres, etc. lo que lleva a la misma situación previa y dificulta la extracción de conclusiones [52]. A pesar de todo, la perspectiva de usar los P300 para revelar información oculta es de gran interés para la investigación.

2.6. Metodología del método ganador del conjunto de datos II de la III Competición BCI

El objetivo de la competición es proporcionar un método que prediga la secuencia de caracteres del conjunto de datos de test de los dos sujetos mediante un algoritmo que puede ser entrenado mediante el conjunto de datos de entrenamiento. El método ganador fue el que obtuvo un mayor porcentaje de aciertos para las 15 iluminaciones de filas y columnas [8].

2.6.1. Preprocesado y extracción de características

En primer lugar, para cada canal, se extraen todas las muestras del conjunto de datos entre 0 y 667 ms tras el inicio de una intensificación. Los potenciales evocados P300 aparecen alrededor de los 300 ms tras la aparición de un estímulo. Por esta razón se considera que esta ventana es suficientemente grande para capturar todas las características temporales para una clasificación eficiente.

Tras ello, cada señal extraída se filtra con un filtro paso banda de Chebyshev de Tipo I y orden 8 con frecuencias de corte de 0.1 y 10 Hz. Finalmente se diezman las señales con un factor de 10, teniendo entonces 14 muestras para cada canal, y se concatenan dichas muestras para los 64 canales. Obtenemos de esta forma una matriz de trabajo de 896 columnas (14 muestras \times 64 canales) y 15300 filas para las señales de entrenamiento (12 columnas/filas \times 15 repeticiones \times 85 caracteres de entrenamiento) cuyas etiquetas son 1 o -1 en función de si la iluminación era del carácter deseado o no.

2.6.2. Métodos de clasificación

A. Conjunto de máquinas de vectores de soporte (SVM)

Cuando se usa un sistema basado en un conjunto de clasificadores, cada clasificador tiene su propio conjunto de datos de entrenamiento. Es necesario, por tanto, subdividir las 15300 muestras del conjunto de entrenamiento en subconjuntos.

Las particiones usadas se construyen de la siguiente forma: cada carácter se corresponde con 180 señales post-estímulo (12 columnas/filas \times 15 repeticiones). Cada partición de entrenamiento se compone de señales asociadas a 5 caracteres, lo que hace un total de 17 particiones ($17 \times 5 = 85$ caracteres) denominadas A_1, A_2, \dots, A_{17} para el sujeto A y B_1, B_2, \dots, B_{17} para el sujeto B . Cada partición se compone de 900 señales de entrenamiento de dimensión 896.

En esta estrategia se ha diseñado un sistema de clasificación múltiple para cada sujeto. Cada clasificador individual del sistema es una SVM entrenada en una de las 17 particiones. Se usó SVM debido a que es un buen método de reconocimiento de patrones para problemas de grandes dimensiones. Cada entrenamiento de una SVM incluye un procedimiento de selección de modelo para establecer el parámetro de regularización C . La función de decisión para una SVM entrenada en la partición P_k viene dada por la

expresión:

$$f_k(\mathbf{x}) = \sum_{i \in P_k} y_i \alpha_i^{(k)} \langle \mathbf{x}, \mathbf{x}_i \rangle + b^{(k)}, \quad (2.1)$$

donde $\{\alpha_i^{(k)}\}$, $b^{(k)}$ son parámetros obtenidos tras el entrenamiento de las SVM y los \mathbf{x}_i vectores de post-estímulo con etiquetas $y_i = \{1, -1\}$.

B. Esquema global de clasificación

Las salidas de estos clasificadores tienen que ser fusionadas de tal forma que se obtenga un solo carácter predicho. De esta forma se transforman los resultados de todos los clasificadores en un solo resultado dentro de las 36 clases posibles (caracteres de la matriz).

El método de clasificación se basa en $K = 17$ clasificadores que han sido clasificados en uno de los subconjuntos de entrenamiento. Cada clasificador asigna una puntuación $f_k(\mathbf{x}_{r|c})$, $k = 1, \dots, 17$, a cada vector de post-estímulo $\mathbf{x}_{r|c}$ asociado a una fila r o columna c .

Después de un número de secuencias J se considera que la fila o columna más probable es la que maximiza la puntuación:

$$S_{r|c}(\mathbf{x}) = \frac{1}{J} \frac{1}{K} \sum_{j=1}^J \sum_{k=1}^K f_k(\mathbf{x}_{r|c}^{(j)}), \quad (2.2)$$

donde $\mathbf{x}_{r|c}^{(j)}$ es el post-estímulo asociado a cada columna o fila para la partición j y $S_{r|c}$ es la puntuación de cada columna o fila.

Si desarrollamos esta expresión obtenemos:

$$S_{r|c}(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^K \sum_{i \in P_k} y_i \alpha_i^{(k)} \frac{1}{J} \sum_{j=1}^J \langle \mathbf{x}_{r|c}^{(j)}, \mathbf{x}_i \rangle + b^{(k)}, \quad (2.3)$$

donde se puede observar que se realiza un doble promediado. El primer promedio se calcula en el espacio de los datos. A medida que las secuencias van aumentando se va calculando el promedio de las filas y columnas. El segundo promedio se calcula en el espacio del clasificador.

C. Procedimiento de selección de canales

Las señales que ofrece la competición en los 64 canales están registradas mediante el gorro estándar del sistema 10-20 [4]. En muchos estudios se emplean únicamente parte de estos canales, aquellos donde se detecta mejor el potencial evocado P300, que normalmente es en la zona centro-parietal.

El método ganador incluye un algoritmo de selección de canales basado en un proceso de reducción de los mismos. En este algoritmo el rendimiento de los clasificadores se evalúa mediante la puntuación:

$$C_{cs} = \frac{t_p}{t_p + f_p + f_n}, \quad (2.4)$$

donde t_p es el número de verdaderos positivos, f_p es el número de falsos positivos y f_n es el número de falsos negativos. El valor de C_{cs} se evalúa después de cada post-estímulo.

El procedimiento que se sigue es el siguiente: se entrena el SVM lineal con las 896 características proporcionadas por los 64 canales; cada canal se elimina temporalmente, por lo que las 14 muestras correspondientes a ese canal se eliminan y se evalúa el valor de C_{cs} . Este procedimiento se realiza para los 64 canales. Aquellos canales cuya eliminación proporciona los valores más altos de C_{cs} , son aquellos que no aportan información relevante al modelo. A continuación, se volvería a proceder de la misma forma pero con un canal menos. Para que el algoritmo sea más rápido, en vez de eliminar canal a canal, se eliminan 4 canales a la vez (aquellos 4 canales cuyo eliminación dé como resultado las 4 mayores puntuaciones de C_{cs}). De esta forma, se eliminan primero los canales que proporcionan menos información. La limitación que se le puede encontrar a este procedimiento es que los canales eliminados no vuelven a ser tenidos en cuenta en el mismo.

Capítulo 3

Métodos de extracción de características

3.1. Introducción

En este capítulo se describe el método de extracción de características utilizado en este estudio: la Transformada Wavelet, que es acompañada de un proceso de selección de sub-muestreo óptimo de forma complementaria.

3.2. ¿Qué es una wavelet?

Una wavelet en el dominio temporal es una función simple de amplitud oscilante en el tiempo, al igual que los senos y cosenos del análisis de Fourier. La diferencia con estas señales es que el seno y el coseno se encuentran localizados en frecuencia y extendidos de forma infinita en el tiempo, mientras que las wavelets están localizadas tanto en tiempo como en frecuencia. Tienen amplitudes grandes y fluctuantes durante un periodo de tiempo restringido y muy bajas o nulas fuera de ese periodo de tiempo, es decir, la mayoría de su energía se restringe a un dominio específico de tiempo. Además, las wavelets están limitadas en banda y por tanto se componen de un rango limitado de frecuencias [53].

Las wavelets se pueden escalar en tiempo estirándolas o comprimiéndolas y se pueden mover (traslación) a otras posiciones temporales sin cambiar su forma básica. El *escalado* y la *traslación* son los dos parámetros básicos de las representaciones wavelet. El conjunto de todas las wavelets escaladas y trasladadas de una forma básica se conoce como *familia wavelet*. Hay un número infinito de wavelets en una familia wavelet. Como se muestra en la figura 3.1, estirar una wavelet a una escala mayor hace que esté menos localizada en el tiempo, su espectro se desplaza a frecuencias más bajas y se concentra más sobre un ancho de banda menor. De la misma forma, comprimir una wavelet a una escala menor hace que esté más localizada en tiempo y su espectro se desplaza a frecuencias mayores, con mayor ancho de banda.

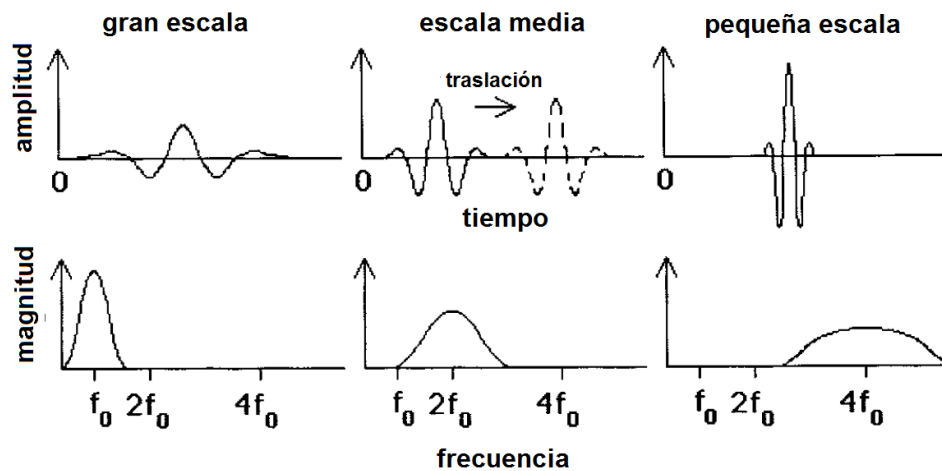


Figura 3.1: Wavelet B-spline. La fila de arriba muestra una wavelet en el dominio del tiempo en tres escalas; la fila de abajo el espectro en frecuencia de la wavelet en las tres escalas. La traslación en tiempo se muestra en escala media por la figura de línea discontinua.

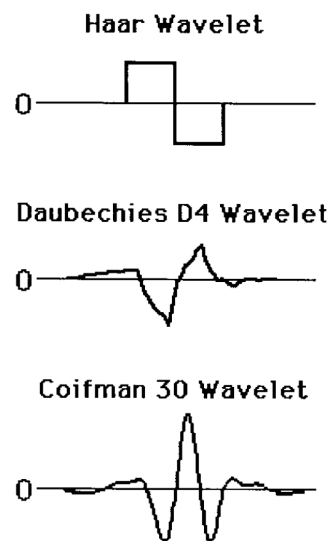


Figura 3.2: Algunos ejemplos de formas de wavelet.

Las wavelets pueden tomar una gran variedad de formas. Algunas wavelets usadas comúnmente, como la Haar y la Daubechies D4 (Figura 3.2) no se parecen a la mayoría de las señales biológicas, mientras que otras, como la Coifman 30 (Figura 3.2) o la B-spline (Figura 3.1) tienen formas más naturalmente biológicas.

Debido a que las wavelets pueden tomar muchas formas diferentes, es posible elegir wavelets que se acerquen a las formas de onda de señales cerebrales reales. De forma general, las wavelets elegidas adecuadamente modelarán de forma cercana las propiedades espectrales y temporales de ciertas componentes de las señales neuroeléctricas.

3.3. Motivación para el análisis Wavelet de señales neuroeléctricas

Antes de presentar los conceptos técnicos detrás del análisis wavelet de señales neuroeléctricas es interesante describir algunas de las aplicaciones del análisis wavelet y su capacidad para capturar la estructura detallada de las ondas neuroeléctricas.

Las representaciones wavelet dan medidas precisas de cuándo y hasta qué punto ocurren los eventos transitorios en una onda neuroeléctrica y cuándo y cómo cambia el contenido en frecuencia de las mismas en el tiempo o la frecuencia. Una vez computada, la representación wavelet de una onda neuroeléctrica es invertible, lo que significa que la forma de onda original se puede reconstruir a partir de un conjunto de coeficientes de análisis que capturan toda la información en tiempo o frecuencia de la misma. Esto lleva a una serie de aplicaciones importantes, algunas de ellas descritas a continuación [53].

- **Filtrado de ruido.** Realizar un análisis wavelet de una onda neuroeléctrica como un EEG es equivalente a descomponer la onda en piezas, cada una de las cuales captura una banda de frecuencias única en una región pequeña de tiempo o espacio. Estas piezas se miden por un conjunto de coeficientes conocidos como coeficientes wavelet. Es posible, por tanto, un filtrado de ruido preciso mediante la atenuación o cambio a cero de los coeficientes asociados a ruido antes de reconstruir la señal neuroeléctrica.
- **Compresión de señales neuroeléctricas.** El archivado o monitorización a largo plazo de señales EEG, especialmente aquellas registradas mediante sistemas de alta densidad de electrodos requiere una compresión de datos eficiente. Por lo general, la cantidad de compresión que se puede conseguir suele estar en oposición a la precisión de la misma. Las técnicas de compresión wavelet mejoran la compresión de datos con poca pérdida de información en la señal. La velocidad de compresión también es importante. Las descomposiciones wavelet son computacionalmente muy eficientes, lo que permite optimizar la transformación a formatos comprimidos en tiempo real.
- **Detección de picos y transitorios.** Una representación wavelet tiene la propiedad de que su resolución temporal o espacial mejora a medida que la escala del evento neuroeléctrico decrece. La escala se refiere a la duración de un evento en el dominio del tiempo o la distancia topográfica del cuero cabelludo cubierta por un evento en el dominio espacial. Esta propiedad hace que la representación wavelet sea ideal para detectar el tiempo de ocurrencia y la localización de eventos transitorios a pequeña escala como los picos epileptiformes.
- **Detección de componentes y eventos.** El análisis wavelet de ERPs basado en modelos de estimación estadística ofrece un método mejorado para descomponer ERPs en componentes funcionalmente específicos. Estos modelos ofrecen un control preciso sobre la selectividad en frecuencia de la descomposición, resultando en una identificación precisa de componentes, incluso cuando estos componentes se superpongan sustancialmente en tiempo y frecuencia. Además, existen wavelets de

una infinidad de formas. Las formas de las wavelets se pueden seleccionar o ser diseñadas para coincidir con las formas de componentes de un ERP. Estas wavelets son patrones excelentes para detectar y separar aquellos componentes y eventos de la forma de onda de fondo. Una vez separados, los coeficientes wavelet dan una medida directa de las características de estos componentes, incluyendo sus amplitudes, latencias y duración.

- Análisis en tiempo y espacio de ondas neuroeléctricas.** Las wavelets son buenas herramientas para observar la estructura de las ondas EEG y ERP a lo largo del rango completo de magnificación de detalle. Las técnicas wavelet permiten a un usuario expandir formas de onda neuroeléctricas en muchos formatos para representación a escala temporal y espacial. Un formato es similar al espectrograma en tiempo-frecuencia, otro es un conjunto de ondas componente (en tiempo) o imágenes (en espacio), similar a la salida de un banco de filtros discretos pasa-banda. Otro es un conjunto de coeficientes que representan información a cada escala de forma separada y se pueden utilizar para análisis cuantitativo. Las representaciones a escala temporal y espacial permiten al usuario buscar eventos funcionalmente significativos en escalas específicas u observar relaciones espacio-temporales en escalas específicas. Seleccionando diferentes wavelets y fijando ciertos parámetros de transformación, las representaciones a escala temporal y espacial se pueden personalizar para definir el rango de escala del análisis, la resolución espacial y temporal, la cantidad de información redundante entre escalas distintas, las características frecuenciales de componentes específicos, etc. La posibilidad de estudiar de forma precisa el comportamiento dinámico básico a pequeña escala de los campos eléctricos del cerebro es potencialmente muy útil para los neurocientíficos y médicos. La estructura a pequeña escala disponible en las ondas registradas en el cuero cabelludo contiene información neuroeléctrica detallada sobre los procesos subyacentes. El análisis wavelet proporciona un método de aislamiento de estos procesos para su estudio.

3.4. Transformada Wavelet

3.4.1. Análisis de señales no estacionarias

El objetivo del análisis de señales es extraer información relevante de una señal mediante una transformación de la misma. Denominamos señales estacionarias a aquellas señales cuyas propiedades no evolucionan temporalmente. Para tales señales, la transformación estacionaria “natural” es la Transformada de Fourier [54]:

$$X(f) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt \quad (3.1)$$

Los coeficientes de análisis $X(f)$ definen la noción de *frecuencia global* en una señal y se obtienen como producto interno de la señal con funciones base de tipo seno de duración infinita. Como resultado, el análisis de Fourier funciona bien si $x(t)$ se compone de unos pocos componentes estacionarios. Sin embargo, cualquier cambio abrupto en el tiempo

en una señal no estacionaria $x(t)$ se expande a lo largo de todo el eje de frecuencias en $X(f)$. Por tanto, un análisis adaptado a señales *no estacionarias* requiere más que la transformada de Fourier.

El método habitual es introducir dependencia temporal en el análisis de Fourier, pero preservando la linealidad. La idea es introducir un parámetro de “frecuencia local” (local en el tiempo) de tal forma que la Transformada de Fourier “local” observe a la señal a través de una ventana sobre la cual la señal es aproximadamente estacionaria. Otra forma equivalente es modificar las funciones seno base a otras que estén más concentradas en el tiempo pero menos en frecuencia.

3.4.2. Transformada de Fourier de Tiempo Reducido

La “frecuencia instantánea” se ha concebido a menudo como una manera de introducir dependencia frecuencial en el tiempo. Si la señal no es de banda estrecha, la frecuencia instantánea tiene de media distintas componentes espectrales en el tiempo. Para ser precisos en temporalmente se necesita por tanto una representación frecuencial en dos dimensiones $S(t, f)$ de la señal $x(t)$ compuesta de características espectrales dependientes del tiempo, definiéndose la frecuencia local f mediante una definición apropiada de $S(t, f)$. Esta representación es similar a la de una partitura, que también muestra “frecuencias” en el tiempo.

La transformada de Fourier se adaptó para definir $S(t, f)$ de la forma que se expone a continuación. Considérese una señal $x(t)$ y asúmase que es estacionaria cuando se observa a través de una ventana $g(t)$ de extensión finita y centrada en la localización temporal τ . La transformada de Fourier de la señal enventanada $x(t)g^*(t - \tau)$ da la Transformada de Fourier de Tiempo Reducido (*Short-Time Fourier Transform, STFT*)

$$STFT(\tau, f) = \int_{-\infty}^{+\infty} x(t)g^*(t - \tau)e^{-j2\pi ft} dt \quad (3.2)$$

que asigna la señal a una función de dos dimensiones en el plano tiempo-frecuencia (τ, f) .

La figura 3.3 muestra bandas verticales en el plano tiempo-frecuencia, ilustrando el concepto de “enventanado” de la señal en la STFT.

A partir de esta interpretación dual se puede ver un inconveniente relacionado con la resolución temporal y frecuencial. Consideremos la habilidad de la STFT de distinguir entre dos sinusoides puras. Dada una función de ventana $g(t)$ y su transformada de Fourier $G(f)$, se define su ancho de banda como

$$\Delta f^2 = \frac{\int f^2 |G(f)|^2 df}{|G(f)|^2 df} \quad (3.3)$$

donde el denominador es la energía de $g(t)$. Dos sinusoides se discriminarán solo si están separadas al menos Δf . De esta forma la resolución en frecuencia del análisis STFT viene dado por Δf . De forma similar, el espacamiento en tiempo viene dado por Δt

$$\Delta t^2 = \frac{\int t^2 |g(t)|^2 dt}{|g(t)|^2 dt} \quad (3.4)$$

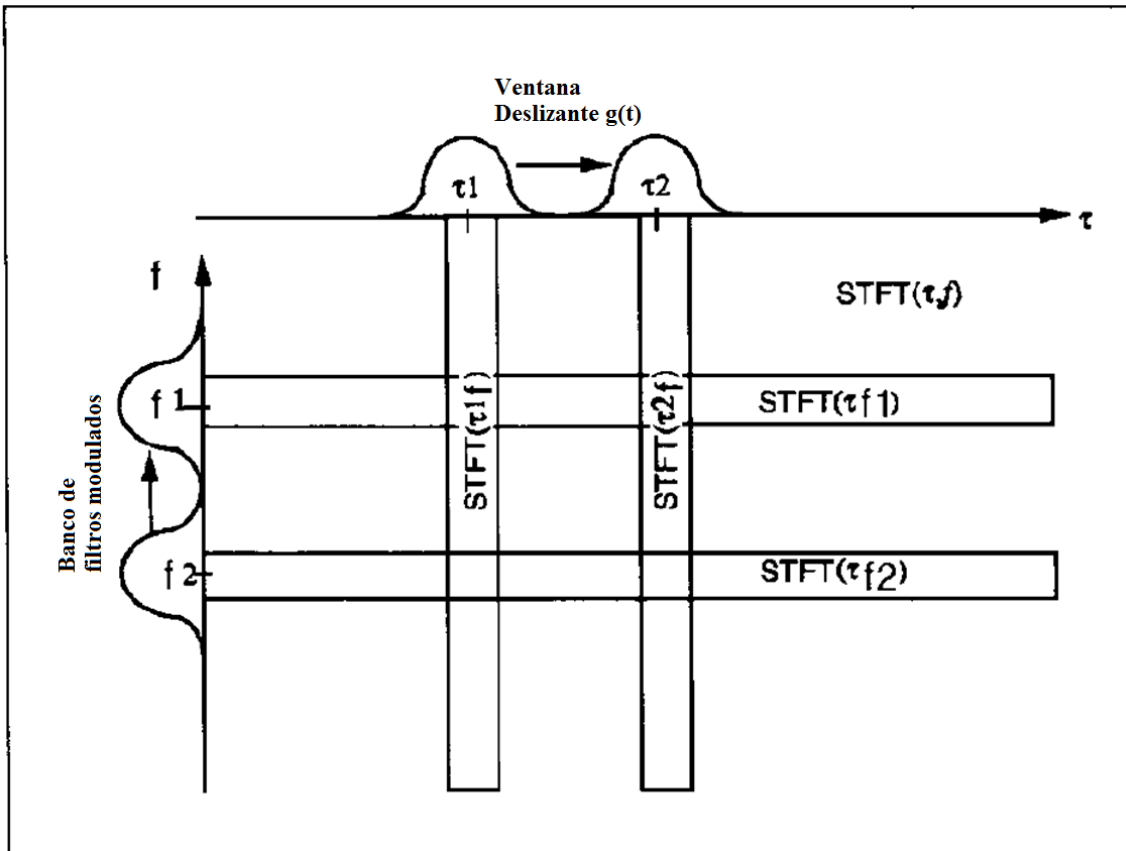


Figura 3.3: Plano tiempo-frecuencia correspondiente a la Transformada de Fourier de Tiempo Reducido. Se puede ver como una sucesión de Transformadas de Fourier de un segmento enventanado de la señal (bandas verticales) o como un análisis basado en un banco de filtros (bandas horizontales).

donde el denominador de nuevo es la energía de $g(t)$. Dos pulsos en el tiempo se pueden discriminar solo si están separados al menos Δt .

La resolución en tiempo y frecuencia no puede ser arbitrariamente pequeña, ya que su producto tiene que cumplir la condición:

$$\Delta t \Delta f \geq \frac{1}{4\pi} \quad (3.5)$$

Esto se conoce como principio de incertidumbre o desigualdad de Heisenberg y quiere decir que para aumentar la resolución de tiempo hay que reducir la de frecuencia y viceversa. A menudo se usan ventanas Gaussianas ya que cumplen esta condición con igualdad.

Más importante aún es que una vez que se ha elegido una ventana para la STFT, la resolución tiempo-frecuencia es *fija* para todo el plano tiempo-frecuencia (ya que se usa la misma ventana para todas las frecuencias).

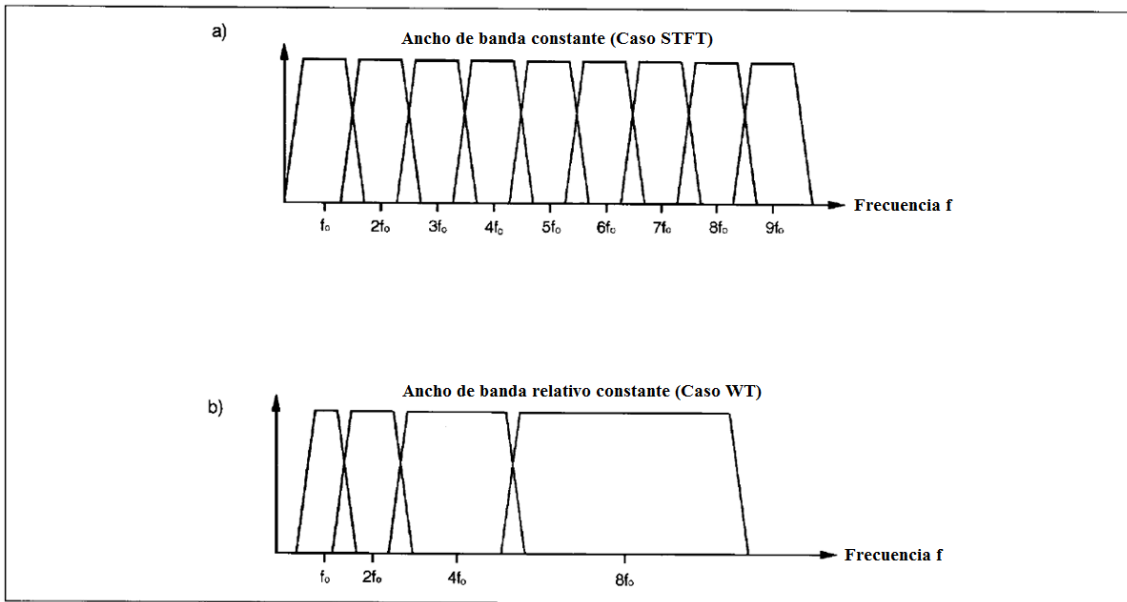


Figura 3.4: División del dominio frecuencial para (a) STFT y (b) transformada Wavelet.

3.4.3. Transformada Wavelet Continua: Análisis Multiresolución

Una forma de superar la limitación de frecuencia de la STFT es variar la resolución de Δt y Δf en el plano tiempo-frecuencia para obtener un análisis multiresolución. Intuitivamente, cuando el análisis se ve como un banco de filtros, la resolución temporal debe aumentar con la frecuencia central de los filtros de análisis, Se impone por tanto que Δf es proporcional a f o

$$\frac{\Delta f}{f} = c \quad (3.6)$$

donde c es una constante. El banco de filtros se compone de filtros paso-banda con ancho de banda relativo constante. Se puede ver también como que los filtros están espaciados de forma logarítmica. En la figura 3.4 se observa este comportamiento.

Si se satisface 3.6 se observa que Δf y Δt cambian con la frecuencia central del filtro de análisis. De esta forma la resolución temporal se hace arbitrariamente grande a altas frecuencias, mientras que la resolución frecuencial lo hace a bajas frecuencias. Este análisis, como es lógico, funciona mejor para señales con componentes de duración corta a altas frecuencias y de duración larga a bajas frecuencias.

La Transformada Wavelet Continua (*Continuous Wavelet Transform, CWT*) sigue las ideas anteriores añadiendo una simplificación: todas las respuestas al impulso del banco de filtros se definen como versiones escaladas del mismo prototipo $h(t)$ (Wavelet Madre). Por ejemplo:

$$h_a(t) = \frac{1}{\sqrt{|a|}} h\left(\frac{t}{a}\right)$$

donde a es el *factor de escala* (la constante $1/\sqrt{|a|}$ se usa para normalizar la energía).

Esto resulta en la definición de CWT:

$$CWT_x(\tau, a) = \frac{1}{\sqrt{|a|}} \int x(t) h^* \left(\frac{t - \tau}{a} \right) dt \quad (3.7)$$

3.4.4. El caso discreto

Para el caso en tiempo discreto, se desarrollaron dos métodos de forma independiente: Codificación de Sub-bandas y Codificación Piramidal o Análisis Multiresolución. Es importante repasar los conceptos de *escala* y *resolución*. La escala está relacionada con el tamaño de la señal, mientras que la resolución lo está con la cantidad de detalle presente en la señal. El parámetro *escala* en el análisis wavelet discreto se entiende de esta forma: para grandes escalas, las wavelet dilatadas muestran “visiones globales” de una señal *sub-muestreada*, mientras que para escalas pequeñas, las wavelets contraídas analizan pequeños detalles en la señal.

3.4.5. Pirámide Multiresolución

Dada una secuencia original $x(n) \in \mathbf{Z}$, se deriva una señal de baja resolución mediante un filtrado paso-bajo con una respuesta al impulso $g(n)$. Según la regla de Nyquist, podemos sub-muestrear por un factor dos, doblando de esta forma la escala en el análisis. Esto resulta en la señal $y(n)$ dada por

$$y(n) = \sum_{k=-\infty}^{k=+\infty} g(k) x(2n - k) \quad (3.8)$$

El cambio en la resolución se obtiene mediante el filtro paso-bajo (pérdida de detalle en alta frecuencia). El cambio en la escala se debe al sub-muestreo por dos.

Ahora intentamos encontrar una aproximación $a(n)$ de la señal original a partir de $x(n)$. Para ello primero se sobremuestra $y(n)$ por un factor dos. Después, la señal sobremuestreada $y'(n)$ se interpola con un filtro de respuesta al impulso $g'(n)$ para obtener la aproximación $a(n)$.

$$a(n) = \sum_{k=-\infty}^{k=+\infty} g'(k) y'(n - k) \quad (3.9)$$

En general, $a(n)$ no va a ser igual que $x(n)$. La diferencia entre $a(n)$ y $x(n)$ se denomina $d(n)$, de forma que $d(n) = x(n) - a(n)$. Es evidente, por tanto, que $x(n)$ se puede reconstruir sumando $d(n)$ y $a(n)$. El proceso completo se muestra en la figura 3.5.

En el caso de que el filtro paso-bajo sea perfecto y de media banda (respuesta en frecuencia igual a 1 entre $-\pi/2$ y $\pi/2$), $d(n)$ contiene exactamente las frecuencias por encima de $\pi/2$ de $x(n)$, y por ello $d(n)$ se puede submuestrear por un factor dos sin perder información.

La separación de la señal original $x(n)$ en una aproximación $a(n)$ más detalle adicional contenido en $d(n)$ es conceptualmente importante. Este método se denomina Análisis

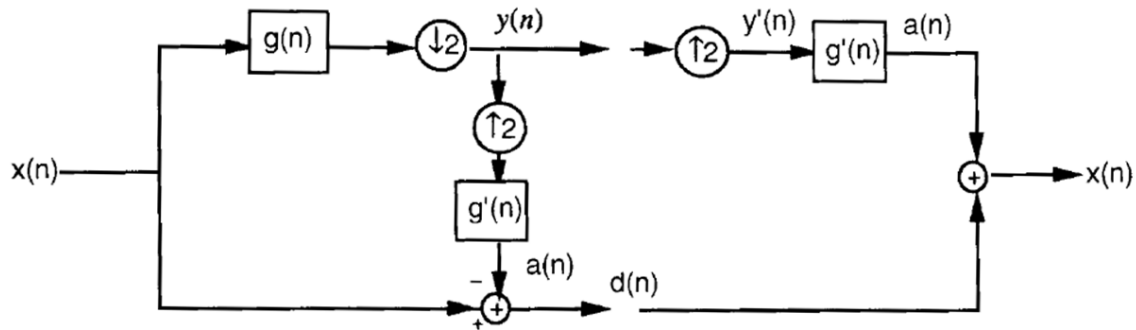


Figura 3.5: Esquema piramidal. Obtención de una aproximación paso-bajo submuestreada y recuperación de la señal.

Multirresolución debido al cambio en la resolución que implica (el filtrado paso-bajo seguido del sub-muestreo por dos produce una señal con la mitad de resolución y el doble de escala).

Este proceso se puede iterar en $y(n)$, creando una jerarquía de señales a menor resolución a escalas más bajas cada vez más cortas, lo que lleva al otro nombre por el que se conoce a este algoritmo (Pirámide Multirresolución).

3.4.6. Codificación de Sub-bandas

El anterior sistema genera una serie de muestras redundantes: una etapa de una descomposición piramidal lleva a una señal de baja resolución a mitad de tasa de muestreo y una señal obtenida por diferencias a tasa completa, es decir un aumento de las muestras del 50%. Este sobre-muestreo se puede evitar si los filtros $g(n)$ y $g'(n)$ cumplen determinadas condiciones.

En la Codificación de Sub-bandas, que se popularizó originalmente para compresión de voz, no aparece tal redundancia. Las aproximaciones paso-bajo sub-muestreadas se obtienen de la misma forma que en el caso anterior, pero, en lugar de una señal obtenida por diferencia, el “detalle adicional” se computa como una versión paso-alto de $x(n)$ (usando un filtro con una respuesta $h(n)$), seguido de sub-muestreo por dos. Intuitivamente, está claro que el “detalle adicional” a la aproximación paso-bajo tiene que ser la señal paso-alto, y también que si $g(n)$ es un filtro paso-bajo de media banda ideal, entonces un filtro paso-alto ideal de media banda $h(n)$ llevará a una representación perfecta de la señal original en dos conjuntos sub-muestreados.

Esto es exactamente un paso de una descomposición wavelet usando filtros $(sen(x))/x$, dado que la señal original está dividida en una aproximación paso-bajo al doble de escala y una señal de detalle añadido (también al doble de escala). Usando estos filtros ideales, la transformada discreta es igual que la continua.

Pero no es necesario usar filtros ideales (y por tanto, poco prácticos), y aun así $x(n)$ se puede recuperar de sus dos versiones submuestreadas y filtradas, que llamaremos $y_0(n)$

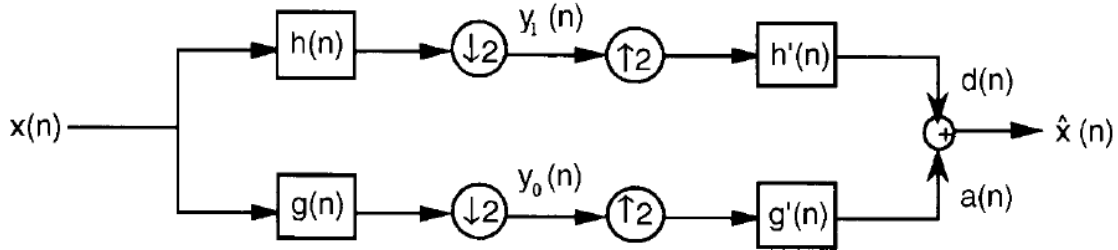


Figura 3.6: Codificación Sub-banda. Se obtienen dos aproximaciones sub-muestreadas, una correspondiente a frecuencias bajas y otra a altas. La señal reconstruida se obtiene re-interpolando las aproximaciones y sumándolas. Los filtros de la izquierda son un banco de análisis, mientras que los de la derecha lo son de síntesis.

e $y_1(n)$. Para ello, ambas son sobremuestreadas y filtradas por $g'(n)$ y $h'(n)$, respectivamente, y finalmente sumadas de forma conjunta, como se muestra en la figura 3.6. A diferencia del caso piramidal, la señal reconstruida $\hat{x}(n)$ no es idéntica a $x(n)$ a no ser que los filtros cumplan ciertas condiciones. Dichos filtros se dice que tienen la propiedad de reconstrucción perfecta.

El caso más sencillo de analizar es cuando los filtros de análisis y síntesis son idénticos (con tiempo invertido debido a la convolución). En dicho caso se puede demostrar que el análisis/síntesis sub-banda se corresponde con una descomposición de la señal en una base ortonormal, seguida de una reconstrucción equivalente a sumar las proyecciones ortogonales. Esta descomposición ortonormal se conoce como banco de filtros “sin pérdidas”.

3.4.7. Transformada Wavelet Discreta

Previamente se ha mostrado como descomponer una secuencia $x(n)$ en dos subsecuencias a mitad de tasa de muestreo o mitad de resolución mediante filtros ortogonales. Este proceso se puede iterar en cada una de las sub-secuencias. Para obtener una resolución frecuencial más fina a frecuencias más bajas (como en el caso de la transformada wavelet continua) se itera el sistema en la banda de baja frecuencia únicamente. De esta forma, una iteración del sistema en la primera banda de baja frecuencia crea otra banda de baja frecuencia que se corresponde con el primer cuarto del espectro de frecuencia. Cada iteración posterior reduce a la mitad la anchura de la banda de baja frecuencia, pero debido al sub-muestreo por dos su resolución también se reduce a la mitad. En cada iteración, la porción de banda de alta frecuencia se corresponde con la diferencia entre la anterior banda de baja frecuencia y la actual, es decir, una banda intermedia. En la figura 3.7 se observa este esquema.

Una característica importante de este algoritmo es que su complejidad es relativamente baja. De hecho, de forma independiente del número de etapas del árbol, la complejidad es lineal con el número de muestras de entrada, con un factor constante que depende de la longitud del filtro. La complejidad satisface la desigualdad:

$$C_{total} < 2C_0$$

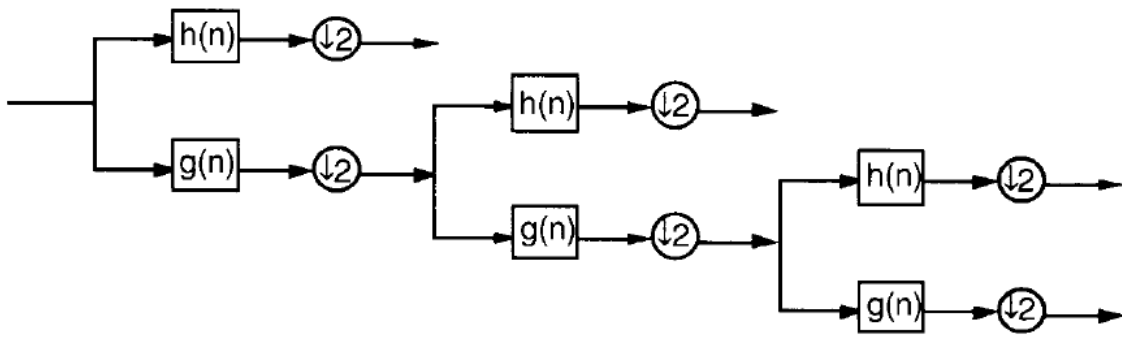


Figura 3.7: Diagrama de bloques de la Transformada Wavelet Discreta implementada con filtros discretos y sub-muestreo por dos.

donde C_0 es el número de operaciones por muestra de entrada que realiza una etapa del algoritmo.

Las salidas de cada iteración de la Transformada Wavelet Discreta se conocen como *función de detalle* (la que se obtiene de la convolución con el filtro paso-bajo $g(n)$) y *señal de baja resolución* (obtenida de la convolución con el filtro paso-alto $h(n)$).

3.5. Análisis wavelet de señales neuroeléctricas

De la misma forma que las señales neuroeléctricas se pueden descomponer mediante STFT en un conjunto de componentes de Fourier dentro de ventanas sucesivas de tiempo, también se pueden descomponer mediante la transformada wavelet (WT) en un conjunto de versiones escaladas en tiempo y frecuencia de la misma wavelet básica (familia wavelet). Tal y como la señal original se puede reconstruir a partir de sus coeficientes de Fourier sumando esos componentes en las proporciones y desfase adecuados, la onda neuroeléctrica original se puede reformar también. En ambos casos, la señal se divide en un conjunto de piezas más pequeño que captura la información de forma selectiva sobre la estructura de la forma de onda en rangos específicos de tiempo y frecuencia.

Computacionalmente, la WT usa wavelets para descomponer las ondas neuroeléctricas de la misma forma que la STFT usa ondas seno y coseno enventanadas. La figura 3.8 muestra una ilustración simplificada de la WT de un ERP para dos de las muchas escalas posibles de estructura de onda: gran escala y pequeña escala. A gran escala, la wavelet se alinea con el principio de la onda neuroeléctrica y se calcula la correlación de la forma de la wavelet con la forma de la onda neuroeléctrica en ese punto. Esta correlación es conocida como coeficiente wavelet. La misma wavelet se traslada una pequeña cantidad a una posición posterior en tiempo, llevando una porción ligeramente diferente de la onda neuroeléctrica a la “vista” de la wavelet, y se calcula un nuevo coeficiente wavelet. Este proceso continua hasta que se agotan las posibles traslaciones y se han calculado los coeficientes.

La figura 3.8 muestra este proceso. En dicha figura la wavelet a gran escala se ha congelado en el tiempo en una posición con alta correlación con la forma local del ERP.

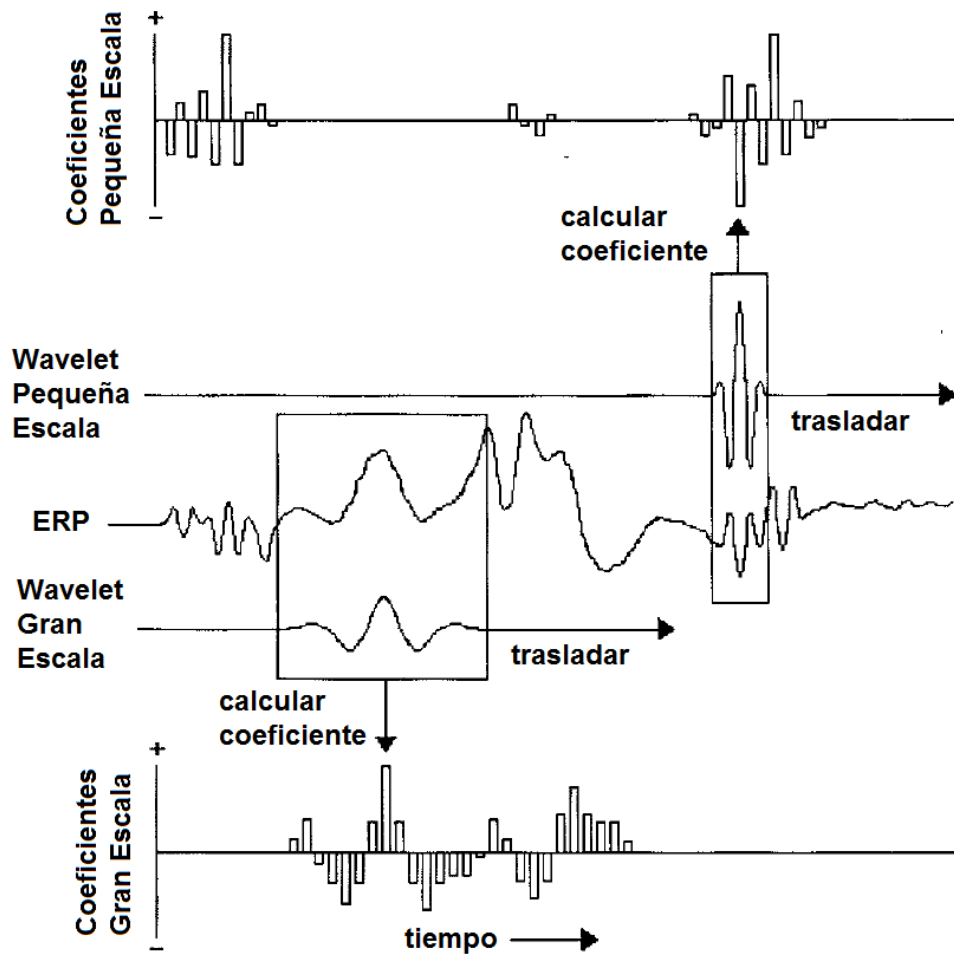


Figura 3.8: Coeficientes wavelet de un ERP para wavelets a gran y pequeña escala como función de una traslación en tiempo. Cada coeficiente se corresponde con una posición trasladada de la wavelet y su magnitud refleja la correlación entre la forma de la wavelet y la de la porción del ERP que abarca la wavelet.

Su correspondiente coeficiente wavelet a gran escala es grande y positivo como es de esperar. A pequeña escala el proceso es similar, con la única diferencia de que la wavelet se contrae en tiempo para que la “vista” sea un rango distinto de fluctuaciones wavelet. Esto se muestra en la parte superior de la figura 3.8, donde se ha congelado la wavelet en una posición con alta correlación con la forma de onda, pero con polaridad invertida entre wavelet y señal, por lo que el coeficiente es grande y negativo.

La secuencia de coeficientes wavelet en cada escala como función del tiempo de traslación generalmente revela donde ocurren los eventos de onda en esa escala. Por ejemplo, observando los coeficientes a pequeña escala en la figura 3.8, la estructura a pequeña escala (alta frecuencia) del ERP parece que se desarrolla principalmente en la porción inicial y en la del final. De forma opuesta, los coeficientes a gran escala (baja frecuencia) muestran que la estructura del ERP es más prominente a lo largo de un alto rango en la mitad de la onda.

3.6. Sub-muestreo óptimo

Otro método de extracción de características es el sub-muestreo de las señales con el objetivo de reducir la dimensionalidad de las mismas y encontrar un punto de diezmado óptimo que haga que se obtengan mejores resultados en la clasificación, además de reducir el tiempo de computación de los clasificadores. Este método de extracción de características se usa en uno de los estudios desarrollados. El caso concreto de la base de datos II de la III Competición BCI se describe en la sección correspondiente.

Capítulo 4

Métodos de clasificación de características

4.1. Introducción

En este capítulo se describen los métodos de clasificación de características utilizados en este estudio. Los resultados de la clasificación se comparan con los obtenidos por el método ganador, propuesto por Rakotomamonjy y Guigue [12], y otros métodos del estado del arte. El objetivo final es superar el porcentaje de clasificación del método ganador y de los métodos con mayor tasa de aciertos en el campo de investigación actual.

4.2. Clasificación propuesta

A continuación pasamos a describir a nivel teórico los tres métodos de clasificación usados en este estudio. Los métodos en cuestión son el *discriminante lineal de Fisher* (FLDA), *Spectral Regression Kernel Discriminant Analysis* (SRKDA) y máquinas de vectores de soporte (SVM).

4.2.1. Discriminante lineal de Fisher

Un discriminante es una función que coge un vector de entrada \mathbf{x} y le asigna a una de K clases. Los discriminantes lineales son aquellos cuya superficie de decisión es un hiperplano [55]. En nuestro caso el problema se limita a 2 clases y por tanto nos centraremos en el caso $K = 2$.

Una forma de ver el problema de clasificación lineal es en términos de la reducción de la dimensionalidad. El FLDA coge un vector de entrada x de D dimensiones y lo proyecta en una sola dimensión mediante:

$$y = \mathbf{w}^T \mathbf{x}, \quad (4.1)$$

donde \mathbf{w} es un vector de pesos. En general, la proyección a una dimensión da lugar a una pérdida considerable de información, y las clases que están muy separadas en el espacio original de D dimensiones pueden estar muy superpuestas en una sola. De todas formas,

ajustando los componentes del vector \mathbf{w} podemos seleccionar una proyección que maximiza la separación entre clases. Se considera un problema en el cual hay N_1 puntos de la clase C_1 y N_2 puntos de la clase C_2 de tal forma que los vectores media de las dos clases vienen dados por:

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in C_1} \mathbf{x}_n, \quad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in C_2} \mathbf{x}_n. \quad (4.2)$$

La forma más simple de separar las clases cuando se proyectan en \mathbf{w} es la separación de la media de las clases proyectadas. De esta forma se elige \mathbf{w} para maximizar:

$$m_2 - m_1 = \mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1), \quad (4.3)$$

donde la media de los datos proyectados de la clase C_k viene dada por:

$$m_k = \mathbf{w}^T \mathbf{m}_k. \quad (4.4)$$

Esta expresión se puede hacer arbitrariamente grande únicamente cambiando la magnitud de \mathbf{w} . Para solucionar este problema se podría hacer que \mathbf{w} tuviera longitud unidad, de tal forma que $\sum_i w_i^2 = 1$. Usando un multiplicador de Lagrange para resolver el problema de maximización con restricciones se obtiene que $\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$. El problema de esta aproximación se puede ver en la figura 4.1. En ella se puede observar que dos clases que están muy separadas en el espacio original de dos dimensiones (x_1, x_2) tienen una superposición considerable al ser proyectadas sobre la línea que une sus medias. Este efecto aparece por las covarianzas de las distribuciones de las clases, las cuales distan mucho de ser diagonales. La idea del discriminante lineal de Fisher es maximizar una función que dará una gran separación entre las medias de las clases proyectadas a la vez que una varianza pequeña dentro de las propias clases, minimizando de esta forma la superposición entre las mismas.

La varianza de las clases de los datos transformados por (4.1) viene dada por:

$$s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2, \quad (4.5)$$

donde $y_n = \mathbf{w}^T \mathbf{x}_n$. Se define la varianza total de las clases para el conjunto de datos como $s_1^2 + s_2^2$. El criterio de Fisher se define como el cociente entre la varianza entre clases y la varianza total de las clases y viene dado por:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)}{s_1^2 + s_2^2}. \quad (4.6)$$

Se puede reescribir el criterio de Fisher mediante (4.1), (4.4) y (4.5) de la forma:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}, \quad (4.7)$$

donde S_B es la matriz de covarianza entre clases y viene dada por:

$$S_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T \quad (4.8)$$

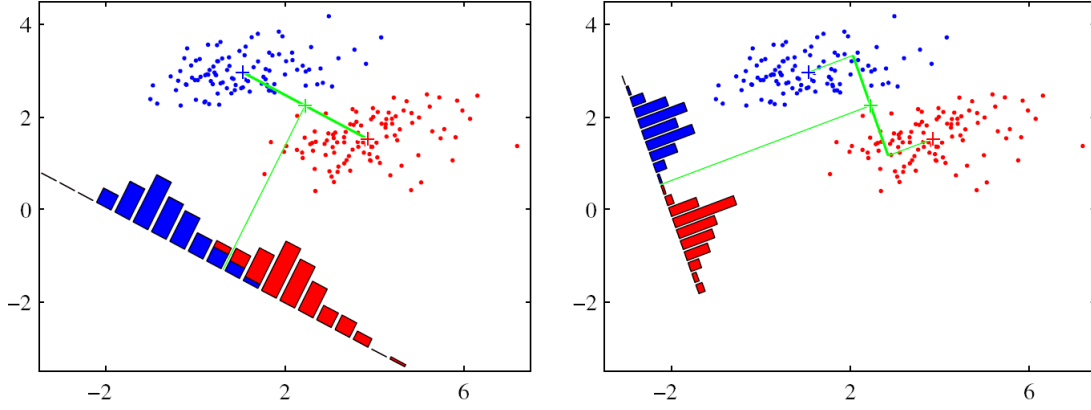


Figura 4.1: En el gráfico de la izquierda se observan las muestras de las dos clases (en rojo y azul) junto con los histogramas que resultan de la proyección sobre la línea que une las medias de las dos clases. Se puede ver una gran superposición entre las clases en el espacio proyectado. El gráfico de la derecha corresponde a la proyección basada en el discriminante lineal de Fisher, con una mejor separación entre clases [55].

y S_W es la matriz de covarianza total de las clases, dada por:

$$S_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T. \quad (4.9)$$

Derivando (4.7) respecto de \mathbf{w} se obtiene que $J(\mathbf{w})$ se maximiza cuando

$$(\mathbf{w}^T S_B \mathbf{w}) S_W \mathbf{w} = (\mathbf{w}^T S_W \mathbf{w}) S_B \mathbf{w}. \quad (4.10)$$

De (4.8) se puede ver que $S_B \mathbf{w}$ siempre está en la dirección de $(\mathbf{m}_2 - \mathbf{m}_1)$. Además, al no importar la magnitud de \mathbf{w} , solo su dirección, se pueden eliminar los factores de escala $(\mathbf{w}^T S_B \mathbf{w})$ y $(\mathbf{w}^T S_W \mathbf{w})$. Multiplicando ambos lados de (4.10) por S_W^{-1} se obtiene:

$$\mathbf{w} \propto S_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1). \quad (4.11)$$

Esta ecuación se conoce como el discriminante lineal de Fisher (FLDA), aunque estrictamente no es un discriminante sino una dirección de proyección de los datos a una dimensión. De todas formas los datos proyectados se pueden usar para construir un discriminante eligiendo un umbral y_0 tal que se clasifique un punto como perteneciente a C_1 si $y(x) \geq y_0$ y a C_2 si no. Una regla habitual es que \mathbf{x} pertenece a la clase i si:

$$|\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{m}_i| = \min_j |\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{m}_j|. \quad (4.12)$$

4.2.2. Spectral Regression Kernel Discriminant Analysis (SRKDA)

El algoritmo *kernel discriminant analysis* (KDA) fue propuesto como una extensión del LDA para el caso no lineal. El KDA es un clasificador que, de forma similar al LDA, busca direcciones que mejoran la separación entre clases. La diferencia está en que el

KDA considera el problema en el espacio de características \mathcal{J} inducido por una aplicación no lineal $\Phi: \mathbb{R}^N \rightarrow \mathcal{J}$, donde N es el número de características [56].

La base del KDA es aplicar los vectores de características en un espacio de dimensión alta donde los problemas complejos de clasificación sean más fácilmente separables. La función objetivo del KDA para encontrar las funciones óptimas de proyección \mathbf{v}_{opt} es la siguiente:

$$\mathbf{v}_{opt} = \operatorname{argmáx} \frac{\mathbf{v}^T S_B^\Phi \mathbf{v}}{\mathbf{v}^T S_W^\Phi \mathbf{v}}, \quad (4.13)$$

donde $\mathbf{v} \in \mathcal{J}$, S_B^Φ y S_W^Φ son las matrices de dispersión o *scatter* (estimador de la matriz de covarianza de la distribución normal multivariante) entre clases en \mathcal{J} . Es decir:

$$S_B^\Phi = \sum_{k=1}^C M_k \left(\boldsymbol{\mu}_\Phi^{(k)} - \boldsymbol{\mu}_\Phi \right) \left(\boldsymbol{\mu}_\Phi^{(k)} - \boldsymbol{\mu}_\Phi \right)^T, \quad (4.14)$$

$$S_W^\Phi = \sum_{k=1}^C \left[\sum_{i=1}^{M_k} \left(\Phi(\mathbf{x}_i^{(k)}) - \boldsymbol{\mu}_\Phi^{(k)} \right) \left(\Phi(\mathbf{x}_i^{(k)}) - \boldsymbol{\mu}_\Phi^{(k)} \right)^T \right], \quad (4.15)$$

donde C es el número de clases, $\boldsymbol{\mu}_\Phi^{(k)}$ y $\boldsymbol{\mu}_\Phi$ son los centroides de la clase k y la media global respectivamente en el espacio de características, y M_k es el número de vectores de características en la clase k .

Se puede probar que el problema de maximización planteado se puede resolver eficientemente mediante el truco del núcleo [57]. Para una función de aplicación Φ el producto interno $\langle \cdot, \cdot \rangle$ se puede definir en \mathcal{J} lo que proporciona el llamado *reproducing kernel Hilbert space* (RKHS)

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle = K(\mathbf{x}, \mathbf{y}),$$

donde $K(\mathbf{x}, \mathbf{y})$ es una función semidefinida positiva del núcleo. Entonces, a partir de la teoría de los núcleos reproducibles, sabemos que cualquier solución $\mathbf{v}_{opt} \in \mathcal{J}$ tiene que estar dentro del espacio de las muestras de entrenamiento en \mathcal{J} . Tienen que existir coeficientes α_i tales que:

$$\mathbf{v}_{opt} = \sum_{i=1}^M \alpha_i \Phi(\mathbf{x}_i), \quad (4.16)$$

donde M es el número de puntos de datos de entrenamiento.

Sea $\boldsymbol{\alpha}_{opt} = [\alpha_1, \alpha_2, \dots, \alpha_M]$, entonces se puede probar que la ecuación (4.13) es equivalente a:

$$\boldsymbol{\alpha}_{opt} = \operatorname{argmáx}_{\boldsymbol{\alpha}} \frac{\boldsymbol{\alpha}^T K V K \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T K K \boldsymbol{\alpha}}, \quad (4.17)$$

donde K es la matriz núcleo $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ y la matriz V se define como:

$$V_{ij} = \begin{cases} \frac{1}{M_k} & \text{si } \mathbf{x}_i \text{ y } \mathbf{x}_j \text{ están en la misma clase (k).} \\ 0 & \text{si } \mathbf{x}_i \text{ y } \mathbf{x}_j \text{ no están en la misma clase (k).} \end{cases} \quad (4.18)$$

El problema de maximización descrito se corresponde con el siguiente problema de descomposición de autovalores:

$$KVK\alpha = \lambda K\alpha. \quad (4.19)$$

Cada autovector α_{opt} da la proyección de un nuevo patrón de test \hat{x} en v en el espacio de características. Para un nuevo vector de datos de ejemplo \hat{x} tenemos:

$$\Theta(\hat{x}, \alpha_{opt}) = \langle v, \Phi(\hat{x}) \rangle = \sum_{i=1}^M \alpha_i K(x_i, \hat{x}). \quad (4.20)$$

Finalmente, \hat{x} es clasificada en base a la distancia euclídea a la proyección de la media de cada clase:

$$\hat{l} = \underset{k}{\operatorname{argmín}} \left\| \langle v, \Phi(\hat{x}) \rangle - \langle v, \mu_{\Phi}^{(k)} \rangle \right\|. \quad (4.21)$$

El algoritmo SRKDA es una mejora de KDA que encuentra la proyección óptima mediante una regresión. La gran ventaja de este algoritmo es que facilita una computación más eficiente ya que no se necesita computación de autovectores implicada en la resolución de la ecuación (4.17). Para obtener el SRKDA, Cai *et al.* [58] probaron el siguiente resultado:

Sea y el autovector del problema de autovalores

$$V y = \lambda y \quad (4.22)$$

con autovalor λ . Si $K\alpha = y$, entonces α es el autovector del problema de autovalores de la ecuación (4.19) con el mismo autovalor λ .

Según este teorema, las funciones de proyección se pueden obtener en dos pasos:

- (1) resolver el problema de autovalores en la ecuación (4.22) para obtener y ;
- (2) encontrar α_{opt} solución de $K\alpha = y$.

La solución al problema de autovalores en el primero de estos pasos se puede encontrar de manera trivial explotando la estructura especial de V . Sin pérdida de generalidad, se puede asumir que los puntos de los datos de entrenamiento están ordenados de acuerdo a sus etiquetas, con lo que la matriz V tiene una estructura diagonal por bloques:

$$V = \begin{pmatrix} V^{(1)} & 0 & \dots & 0 \\ 0 & V^{(2)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & V^{(C)} \end{pmatrix}, \quad (4.23)$$

donde $V^{(k)}$, $k \in \{1, \dots, C\}$, es una matriz $M_k \times M_k$ con todos los elementos iguales a $1/M_k$. Se puede observar que los autovalores de $V^{(k)}$ son 0 y 1 y que solo tiene un autovector $e^{(k)} = [1, 1, \dots, 1]^T \in \mathbb{R}^{M_k}$, asociado con el autovalor 1. Dada la estructura diagonal por bloques de V , los autovalores y autovectores son la unión de los autovalores

y autovectores de sus bloques (rellenando apropiadamente con ceros). De esta forma V tiene C autovectores de V con el mismo autovalor 1. Estos autovectores son:

$$\overline{\mathbf{y}}_k = \left[\underbrace{0, \dots, 0}_{\sum_{i=1}^{k-1} M_i}, \underbrace{1, \dots, 1}_{M_k}, \underbrace{0, \dots, 0}_{\sum_{i=k+1}^C M_i} \right]^T, \quad k = 1, \dots, C. \quad (4.24)$$

Dado que todos estos autovectores están asociados al autovalor 1 de V , podemos tomar otros vectores ortogonales C cualesquiera en el espacio generado por $\{\overline{\mathbf{y}}_k\}_{k=1}^C$ y definirlos como nuestros C autovectores. El vector de unos está incluido de forma natural en el espacio. Este vector no tiene ninguna utilidad puesto que la correspondiente función proyectiva llevará todas las muestras al mismo punto. Por esa razón este vector se elige como el primer autovector de V y los restantes autovectores se encuentran mediante el algoritmo de Gram-Schmidt. Tras ello el vector de unos se puede eliminar dejando exactamente $C - 1$ autovectores de V , $\{\mathbf{y}_k\}_{k=1}^C$.

En el segundo paso se obtiene un α_{opt} para cada autovector de V resolviendo el correspondiente sistema de ecuaciones lineales $K\alpha = \mathbf{y}$. La matriz núcleo K es semidefinida positiva. Cuando K es singular, el sistema puede no tener solución o infinitas soluciones. Entonces, una manera posible de obtener un estimador aproximado es adoptar la técnica de regularización: $(K + \delta I)\alpha_{opt,k} = \mathbf{y}_k$, donde I es la matriz identidad y $\delta > 0$ es el parámetro de regularización. Dado que ahora la matriz $K + \delta I$ es definida positiva, la factorización de Cholesky se puede usar para computar eficientemente la solución. Finalmente, tras los dos pasos, los nuevos patrones son clasificados proyectando el vector de características con las $C - 1$ funciones proyectivas $\{\alpha_{opt,k}\}_{k=1}^C$ de la misma forma que el KDA.

4.2.3. Conjunto de máquinas de vectores de soporte (SVM)

Las máquinas de vector soporte se suelen emplear tanto para clasificación como para regresión no lineal. La idea principal es construir un hiperplano de separación de tal manera que el margen de separación entre clases sea máximo. A este hiperplano se le denomina hiperplano óptimo [59].

Para la construcción del hiperplano se supone un conjunto de entrenamiento dado por $\{\mathbf{x}_i, d_i\}_{i=1}^N$, donde \mathbf{x}_i representa el valor de entrada y d_i el correspondiente valor objetivo. La distancia entre un hiperplano y el punto de los datos más cercano a éste se denomina margen de separación y se denota por ρ . La frontera de decisión viene especificada por la ecuación:

$$\mathbf{w}^T \mathbf{x} - b = 0, \quad (4.25)$$

donde \mathbf{x} es el vector de entrada, \mathbf{w} es el vector de pesos ajustable y b es el sesgo. Si el objetivo conseguir es el hiperplano óptimo, la función discriminante vendrá dada por la siguiente expresión:

$$g(\mathbf{x}) = \mathbf{w}_0^T \mathbf{x} - b_0. \quad (4.26)$$

Aquellos puntos $\{\mathbf{x}_i, d_i\}$ que se encuentran más cercanos al hiperplano óptimo y cuyo margen de separación ρ_0 es máximo, se les denomina vectores soporte y su clasificación es la más complicada. En la figura 4.2 se observa un ejemplo de estos conceptos.

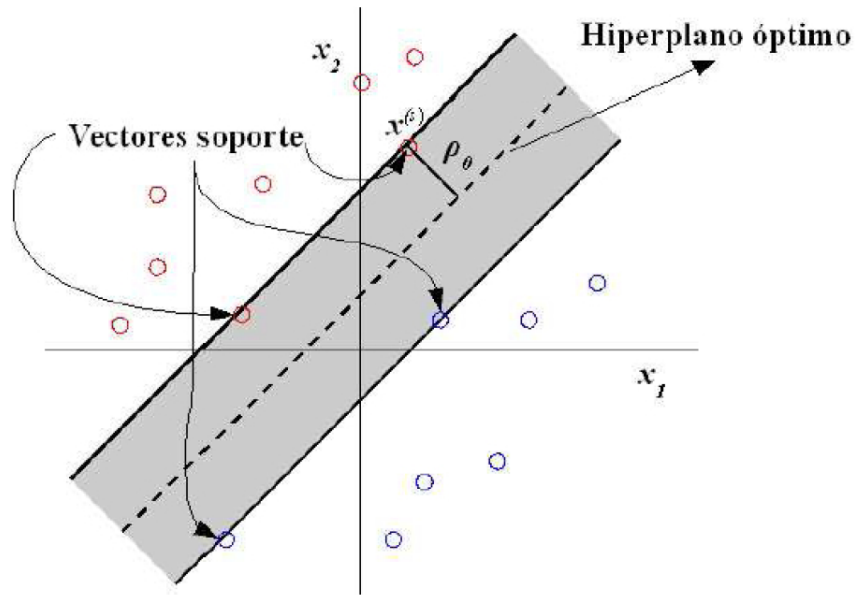


Figura 4.2: Hiperplano óptimo de separación de dos clases linealmente separables. Los círculos rojos representan los patrones de una clase y los azules de la otra [60].

Para la construcción del hiperplano óptimo se suele emplear como método los multiplicadores de Lagrange. El problema de optimización puede ser planteado de la siguiente forma:

Dado los datos de entrenamiento $\{\mathbf{x}_i, d_i\}_{i=1}^N$, encontrar los valores óptimos del vector de pesos \mathbf{w} y del sesgo b para que satisfaga el criterio:

$$d_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \text{para } i = 1, 2, \dots, N \quad (4.27)$$

y el vector de pesos \mathbf{w} que minimice la función de coste:

$$\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}. \quad (4.28)$$

Para resolver este problema se emplea el método de los multiplicadores de Lagrange:

Dado los datos de entrenamiento $\{\mathbf{x}_i, d_i\}_{i=1}^N$, encontrar el multiplicador de Lagrange que maximice la función objetivo:

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j \mathbf{x}_i^T \mathbf{x}_j \quad (4.29)$$

sujeto a las restricciones

$$\sum_{i=1}^N \alpha_i d_i = 0, \quad (4.30)$$

$$\alpha_i \geq 0 \quad \text{para } i = 1, 2, \dots, N. \quad (4.31)$$

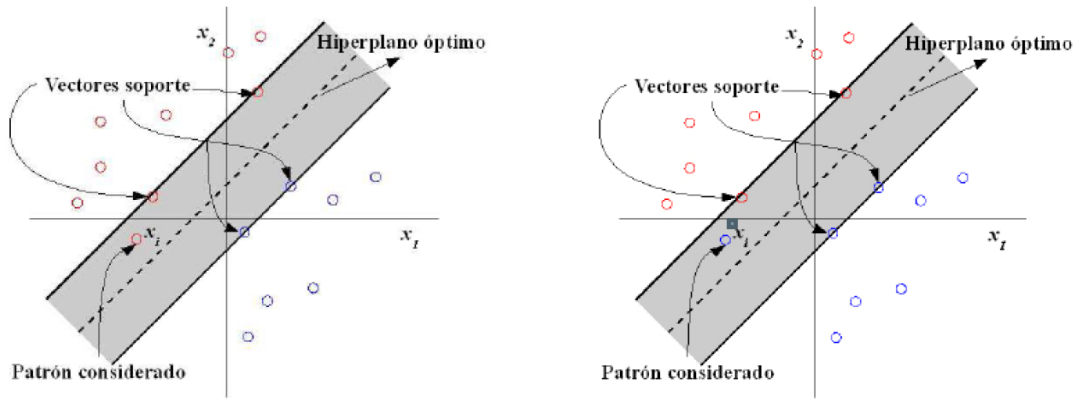


Figura 4.3: Tipos de errores en la clasificación. 1) Patrón dentro de la región de separación pero correctamente clasificado. 2) Patrón incorrectamente clasificado [60].

A partir de los multiplicadores de Lagrange óptimos $\alpha_{0,i}$, d_i , \mathbf{x}_i se calculan los parámetros del hiperplano óptimo [59].

$$\mathbf{w}_0 = \sum_{i=1}^N \alpha_{0,i} d_i \mathbf{x}_i, \quad (4.32)$$

$$b_0 = 1 - \mathbf{w}_0^T \mathbf{x}^{(s)} \quad \text{para } d^{(s)} = 1. \quad (4.33)$$

Cuando las clases no se pueden separar linealmente no se puede construir el hiperplano sin que se produzcan errores en la clasificación de algunos de los patrones. Para este caso, el hiperplano óptimo será aquel que minimice la probabilidad de cometer ese error. Se introduce un nuevo parámetro ξ_i que mide la diferencia desde el patrón considerado al hiperplano óptimo.

El error puede estar definido de dos formas:

- (1) que uno de los datos caiga en la zona de decisión pero en el lado correcto del hiperplano. En este caso la clasificación es correcta y $0 \leq \xi_i \leq 1$;
- (2) que uno de los datos caiga en el lado incorrecto de la frontera de decisión, en cuyo caso $\xi_i > 1$.

Se pueden observar ambos errores en la figura 4.3.

El hiperplano óptimo cuando las variables no son linealmente separables se encuentra minimizando la siguiente expresión:

$$\Phi(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \quad (4.34)$$

donde C es un parámetro escalar y positivo determinado de forma experimental que controla la complejidad de la máquina de vector soporte y el número de patrones que se han clasificado incorrectamente.

El problema de optimización se puede resolver empleando, de nuevo, el método de multiplicadores de Lagrange cambiando la segunda de las restricciones (4.31) de la siguiente forma:

$$0 \geq \alpha_i \geq C \quad \text{para } i = 1, 2, \dots, N. \quad (4.35)$$

Si $C \rightarrow \infty$ estamos ante un problema de clasificación linealmente separable.

Cuando se emplean las máquinas de vector soporte para reconocimiento de patrones el problema varía ligeramente. Como se ha presentado previamente el espacio de entrada se aplica a un espacio de dimensión elevada y una vez en este plano se construye el hiperplano óptimo de separación. Este hiperplano será una combinación lineal de los vectores del espacio de dimensión mayor, no de los del espacio original [59].

El problema de optimización será el siguiente:

Dado el conjunto de entrenamiento $\{\mathbf{x}_i, d_i\}_{i=1}^N$, encontrar el multiplicador de Lagrange $\{\alpha_i\}_{i=1}^N$ que maximice la función objetivo:

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (4.36)$$

sujeto a las restricciones

$$\sum_{i=1}^N \alpha_i d_i = 0, \quad (4.37)$$

$$0 \leq \alpha_i \leq C \quad \text{para } i = 1, 2, \dots, N. \quad (4.38)$$

La ecuación (4.36) difiere de (4.29) en la expresión $K(\mathbf{x}_i, \mathbf{x}_j)$ que es lo que se conoce como núcleo del producto interno. Existen diferentes tipos de núcleos: polinómicos, gaussianos, exponenciales, etc. En nuestro estudio usamos un núcleo de tipo gaussiano. La expresión que lo define es [59]:

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right), \quad (4.39)$$

donde σ especifica la anchura de las funciones gaussianas, \mathbf{x}_i es su centro y \mathbf{x} es el patrón de entrada. Se suele identificar $1/(2\sigma^2) = \gamma$.

Para obtener los resultados de la clasificación mediante un conjunto de SVM entrenados en subconjuntos del conjunto de datos de entrenamiento, de forma totalmente análoga al método ganador.

Capítulo 5

Metodología del estudio

5.1. Introducción

El objetivo principal de este trabajo consiste en aplicar diferentes métodos de extracción (transformada wavelet) y clasificación de características al conjunto de datos II de la III Competición BCI [8].

Se realizan un total de dos estudios: en el primero se evalúan únicamente métodos de clasificación de características, mientras que en el segundo se introducen métodos de extracción de las mismas, así como un proceso de selección de sub-muestreo óptimo.

5.2. Preprocesado

En nuestro estudio las señales de entrenamiento se guardan en sendos ficheros .mat cuyo contenido incluye las siguientes variables:

<i>code</i>	Matriz 85×180 correspondiente a la fila columna/iluminada en cada muestra de los 85 caracteres.
<i>target</i>	Cadena de 85 elementos en los que se almacena el carácter objetivo del sujeto en cada prueba.
<i>class</i>	Matriz 85×180 que contiene la clase (1 o -1, que se corresponden con objetivo o no objetivo) que indica si la iluminación correspondía con un carácter objetivo o no.
<i>features</i>	Matriz $85 \times 180 \times 896$ que almacena para cada carácter (85) y cada iluminación (180) las 896 muestras que corresponden a la concatenación de los 64 canales.
<i>name</i>	Cadena con el nombre del sujeto.
<i>matrix</i>	Matriz de caracteres del paradigma usado en la competición.

Las señales de test son similares pero con 100 caracteres en lugar de 85. Las matrices *class* y *features* para este segundo tipo fueron construidas con conocimientos a posteriori

de la cadena de caracteres que se buscaba en la competición, con el fin de poder establecer el porcentaje de aciertos tras la clasificación.

5.3. Primer estudio: métodos de clasificación de características

En el primer estudio se realiza una comparativa entre los tres métodos de clasificación de características descritos en el capítulo previo. Para ello se obtienen primero los parámetros óptimos de los mismos mediante validación cruzada, tras lo cual se procede a realizar la clasificación y validarla según el modelo de la III Competición BCI.

5.3.1. Método de obtención de parámetros óptimos

Una característica común que tienen todos los clasificadores utilizados es el hecho de que poseen parámetros configurables. Para obtener los valores óptimos de dichos parámetros en nuestro estudio se usó validación cruzada. La validación cruzada es una técnica de validación de modelos para estimar cómo de preciso será un modelo predictivo en la práctica. El objetivo de la validación cruzada es definir un conjunto de datos para probar el modelo en la fase de entrenamiento y de este modo dar una visión de cómo el modelo actuará con un conjunto de datos independiente.

Una ronda de validación cruzada incluye partir una muestra de datos en subconjuntos complementarios, realizando el análisis en un subconjunto (conjunto de entrenamiento) y validando el análisis en otro subconjunto (conjunto de test). Para reducir la variabilidad se realizan múltiples rondas de validación cruzada usando diferentes particiones y se hace la media de los resultados de la validación.

En una validación cruzada de k iteraciones, la muestra original se subdivide en k particiones de igual tamaño. De esos k subconjuntos, uno se usa como conjunto de validación y los restantes como conjunto de entrenamiento. El proceso se repite k veces, usando cada uno de los subconjuntos una vez como conjunto de validación. Se realiza la media de los resultados para obtener una estimación. Para este estudio usamos la validación cruzada de 10 iteraciones (*10-fold cross-validation*). Para obtener los parámetros óptimos de los clasificadores realizamos validación cruzada para distintos valores de los mismos con el conjunto de datos de entrenamiento (85 caracteres). Tras el proceso se seleccionan como óptimos aquellos que nos proporcionen mayor área bajo la curva ROC al clasificar. Los parámetros configurables de los clasificadores y sus barridos son los siguientes:

- LDA:** parámetro de regularización “ReguAlpha” (desde 1 hasta 50000 con salto 1000);
- SRKDA:** parámetro de regularización “ReguAlpha” (desde 0.05 hasta 0.1 con salto 0.01) y parámetro t (desde 800 hasta 2200 con salto 100);
- SVM:** parámetro C , con valores [1 5 10 15 20 25 30].

La razón por la que se barren estos valores es porque se ha demostrado en la práctica que los valores óptimos suelen encontrarse dentro de este rango [59].

5.3.2. Validación del método propuesto

Una vez obtenidos los parámetros óptimos para cada clasificador, se entrenan mediante el conjunto de datos de entrenamiento. Tras ello, se aplican los modelos al conjunto de datos de test y se obtiene la probabilidad de que la fila o columna para cada iluminación fuera de clase 1 (objetivo) o -1 (no objetivo). Cada carácter tiene asignada una fila y una columna de la matriz. Para la decisión final se escoge aquel carácter cuya media de probabilidad de ser clase 1 para la fila y la columna que le corresponden sea mayor. El proceso se repite con los 100 caracteres y se calcula el porcentaje de aciertos comparando con los caracteres correctos.

Es importante destacar que los sujetos A y B por lo general tienen distintos parámetros óptimos para los clasificadores.

5.4. Segundo estudio: métodos de extracción de características

En este segundo apartado se procede a mostrar el procedimiento que se va a seguir en un segundo estudio. El objetivo es aplicar dos métodos de extracción de características: sub-muestreo (para optimizar la banda de frecuencia y el número de muestras por iluminación) y transformada wavelet. Finalmente se explica el algoritmo de eliminación de canales implementado y se compara con el desarrollado por los ganadores de la III Competición BCI.

5.4.1. Primera etapa: evaluación del sub-muestreo óptimo

En la primera etapa de este estudio se determina el número de muestras y la banda de frecuencia óptima para cada uno de los sujetos de la base de datos II de la III Competición BCI.

Las señales del conjunto de datos de la III Competición BCI originalmente contaban con 160 muestras por canal. Dicho número es excesivamente elevado para que la clasificación sea viable en tiempo real, por lo que hasta ahora se ha trabajado con un conjunto filtrado y sub-muestreado. La frecuencia de muestreo del conjunto de datos era de 240 Hz. Las señales se filtraban paso banda entre 0.1 y 20 Hz y se diezmaban por un factor que venía dado por

$$\text{Factor de diezmado} = \frac{\text{Frecuencia de muestreo}}{\text{Frecuencia de corte superior}} \quad (5.1)$$

Por tanto, el factor original de diezmado era de 12. Al haber 160 muestras por canal e iluminación originalmente, al final se tienen 160/12 muestras, redondeando hacia arriba 14 muestras por canal e iluminación.

En esta parte del estudio se va a estudiar el efecto de cambiar la frecuencia de corte y, por tanto, el número de muestras por canal e iluminación. Tras obtener unas nuevas señales de entrenamiento y test de esta forma, se procede a su clasificación y se comparan

resultados. El objetivo es determinar si el número de muestras utilizado previamente (y la banda de frecuencias asociada) no es el óptimo y podría mejorar. Para el estudio se van a utilizar frecuencias de corte superior de 6, 9, 12, 15, 18, 20, 24, 27, 30 y 33 Hz, con los que se obtienen respectivamente 4, 6, 8, 10, 12, 14, 16, 18, 20 y 22 muestras por canal.

Se determina el sub-muestreo óptimo para cada sujeto y clasificador por separado, siendo los clasificadores LDA, un conjunto de 2 SRKDA y un conjunto de 19 SVM (se verá que este es el número óptimo obtenido en los resultados del primer estudio). Se selecciona como sub-muestreo óptimo para cada sujeto el que arroje mejores resultados para este clasificador, cuyos parámetros óptimos ya se determinaron previamente para cada sujeto. Se realiza la clasificación para 5 y 15 muestras, ya que es lo que se pedía en la III Competición BCI. Como valor principal se toma el resultado para 15 iluminaciones, en caso de empate se considera óptimo aquel que tenga mejores resultados para 5 iluminaciones.

5.4.2. Segunda etapa: selección de la Wavelet Madre óptima

En la segunda etapa del estudio se determina la Wavelet Madre óptima para cada sujeto una vez determinado el sub-muestreo para cada uno de ellos. En concreto, se realiza la Transformada Wavelet Discreta con las Wavelet madre: 'haar', 'db1', 'db2', 'db3', 'db4', 'db5', 'db6', 'db7', 'db8', 'db9', 'db10', 'sym2', 'sym3', 'sym4', 'sym5', 'sym6', 'sym7', 'sym8', 'coif1', 'coif2', 'coif3', 'coif4' y 'coif5' y se determina cuál es la óptima para usar como método de extracción de características para cada sujeto (o bien si en ningún caso se mejora la clasificación). El criterio seguido es el mismo que en el caso anterior: como valor principal se toma el resultado para 15 iluminaciones, en caso de empate se considera óptimo aquel que tenga mejores resultados para 5 iluminaciones. De nuevo, para cada sujeto y clasificador se obtiene un resultado óptimo.

5.4.3. Tercera etapa: clasificación con los parámetros óptimos

En la tercera etapa del estudio se procede a la clasificación de las bases de datos correspondientes a ambos sujetos con los sub-muestreos y Wavelets Madre fijados anteriormente. En esta etapa se usan como clasificadores LDA, un conjunto de 2 SRKDA y un conjunto de 19 SVM. Además, en este caso la clasificación se realiza para todas las iluminaciones entre 5 y 15, con el propósito de ver la evolución del rendimiento de los clasificadores de forma más detallada. El objetivo final es determinar el clasificador óptimo para cada sujeto, de forma que una vez más se seguirá el criterio de comparar los resultados: el que mejores resultados tenga para 15 iluminaciones se considerará como óptimo, en caso de empate el que mejores resultados tenga para 5 iluminaciones de entre los empatados.

5.4.4. Cuarta etapa: algoritmo de eliminación de canales

En esta cuarta etapa del estudio se procede a aplicar el algoritmo de eliminación de canales que se describe en la siguiente sección, antes de la clasificación. El objetivo es

descartar aquellos canales que no aporten información de interés para la clasificación y cuya eliminación se espera que mejore la misma. El algoritmo se aplica sobre las señales de entrenamiento una vez que se ha aplicado la Transformación Wavelet sobre ellas. Este algoritmo se aplica para cada clasificador por separado, ya que puede que los canales redundantes para uno no lo sean para los demás. Los parámetros de diezmado que se utilizan, así como la Wavelet Madre, son los obtenidos en las etapas anteriores.

5.4.5. Algoritmo de eliminación y selección de canales

Las señales registradas correspondientes al conjunto de datos II la III Competición BCI se obtuvieron de acuerdo al Sistema Internacional 10-20, ya descrito en un apartado previo, el cual se compone de 64 electrodos. Por tanto, las señales de las que se disponen cuentan con 64 canales que se corresponden con señales registradas en distintos puntos del cuero cabelludo de cada sujeto. El método ganador de la competición (Rakotomamonjy y Guigue, 2008 [12]) aplica un algoritmo recursivo de eliminación de canales con el objetivo tanto de reducir las dimensiones de las señales como de descartar aquellos canales que no aporten información valiosa a la clasificación. A continuación se describe el algoritmo usado por los ganadores y tras esto el que se ha aplicado en este estudio. El algoritmo se desarrolló en base al indicado por los ganadores de la competición con las correspondientes variaciones.

5.4.5.1. Algoritmo usado por los ganadores de la competición

El rendimiento de un clasificador se evalúa habitualmente en función de la puntuación

$$C_{cs} = \frac{t_p}{t_p + f_p + f_n}, \quad (5.2)$$

donde t_p , f_p , f_n son respectivamente el número de verdaderos positivos, falsos positivos y falsos negativos del conjunto de datos de validación tras la clasificación. Es importante destacar que para la selección de canales el rendimiento de los clasificadores se evalúa para una sola señal (clasificación binaria) y no el rendimiento del reconocimiento de caracteres. Esto quiere decir que esta puntuación se calcula de la siguiente forma: una vez clasificadas las iluminaciones del conjunto de datos de entrenamiento de forma individual, si la señal se correspondía con una iluminación deseada (1) y se clasificó como tal (1), entonces es verdadero positivo, si no era iluminación deseada (0) y se clasificó como iluminación (1), entonces es falso positivo, y si se correspondía con una iluminación deseada (1) y se clasificó como no iluminación (0), entonces es falso negativo.

En este caso la puntuación C_{cs} no tiene en cuenta el número de verdaderos negativos. Esto es importante para conjuntos de datos no balanceados, como es el caso, ya que esta omisión da puntuaciones positivas a los casos positivos, que son muchos menos que los negativos. Interesa más valorar los casos de verdaderos positivos porque son muchos menos que los negativos.

El algoritmo usado sigue el siguiente procedimiento. En primer lugar se entrena el clasificador (en este caso era un conjunto de SVM lineales) con las características de todos los canales del conjunto de datos de entrenamiento. Se evalúa el rendimiento del

clasificador de acuerdo a C_{cs} . A continuación, cada canal individual se retira temporalmente (por ejemplo, suponemos que se retira el canal j), es decir, todas las características correspondientes a ese canal se eliminan y se evalúa la puntuación $C_{cs}^{(-j)}$ (puntuación cuando el canal j se retira). El canal cuya eliminación maximice $C_{cs}^{(-j)}$ se retira definitivamente. Este procedimiento se continúa hasta que todos los canales han sido eliminados, quedando entonces éstos ordenados de mejor a peor. Para aumentar la velocidad de este procedimiento, se decidió que se eliminaran los canales de forma arbitraria de cuatro en cuatro [12]. Este procedimiento de eliminación de canales es también una forma de ordenar los canales de acuerdo a su puntuación C_{cs} , siendo el primer canal eliminado el menos importante y el último en ser eliminado el más importante. A continuación se indica el pseudocódigo del algoritmo.

Algoritmo 1 Algoritmo de eliminación recursiva de canales de la competición

Inicialización: RANKED = 0; CHANNEL = [1, . . . , Número de canales]

while CHANNEL no está vacío **do**

for all canales en CHANNEL **do**

 Eliminar temporalmente canal j en CHANNEL

 Entrenar una SVM lineal con los canales restantes

 Calcular puntuación $C_{cs}^{(-j)}$

end for

 RANKCHAN = $argmax_i C_{cs}^{(-j)}$

 Variable Rank: RANKED = [RANKCHAN RANKED]

 Eliminar variable RANKCHAN de la lista de variables CHANNEL

end while

5.4.5.2. Algoritmo usado en este estudio

El algoritmo desarrollado en este estudio parte de premisas similares pero cuenta con diferencias importantes respecto al descrito en el apartado anterior. Este algoritmo usa también la puntuación C_{cs} como criterio. La principal diferencia, aparte naturalmente de que acepta más clasificadores además de SVM, es que su objetivo no es ordenar los canales de más a menos importantes, sino eliminar aquellos canales que no aporten información útil a la clasificación. El funcionamiento del algoritmo se describe a continuación.

En primer lugar se entrena el clasificador con las características de todos los canales del conjunto de datos de entrenamiento y se evalúa el rendimiento del clasificador de acuerdo a C_{cs} . Esta evaluación se hace mediante validación cruzada en 10 partes. Se divide el conjunto de entrenamiento en 10 partes. En primer lugar se usa una parte como conjunto de datos de test y las 9 restantes como conjunto de datos de entrenamiento. Se calcula C_{cs} con estos resultados. Este proceso se repite otras 9 veces usando en cada caso como conjunto de test una parte y como conjunto de entrenamiento las 9 restantes. La puntuación C_{cs} final es la media de las 10 calculadas.

Una vez calculado C_{cs} con todos los canales se calcula de nuevo, siguiendo el mismo procedimiento, pero eliminando el primer canal. Si la puntuación C_{cs} es superior a la de todos los canales, este canal se añade a la lista de canales a eliminar (pero no se elimina aún). Se realiza este procedimiento para todos los canales. Una vez realizado para todos se procede a eliminar definitivamente aquellos que han sido añadidos a la lista de canales a eliminar.

Una vez eliminados estos canales, se vuelve a realizar otra ronda con el mismo criterio. El algoritmo acaba en el momento en que no se elimina ningún canal en una ronda completa. A continuación se indica el pseudocódigo del algoritmo. Se omite indicar el proceso de validación cruzada pero se sobreentiende que se incluye al indicar el proceso de calcular la puntuación C_{cs} y $C_{cs}^{(-j)}$.

Algoritmo 2 Algoritmo de eliminación recursiva de canales del estudio

Inicialización: CANALES = [1, ..., Número de canales]

RATIOTOTAL = calcular C_{cs} con todos los canales

FLAG = 1

while FLAG == 1

CANALELIMINAR = 0

for all canales en CANALES **do**

Eliminar temporalmente canal j en CANALES

Entrenar un clasificador con los canales restantes

Calcular puntuación $C_{cs}^{(-j)}$

if $C_{cs}^{(-j)} > C_{cs}$

CANALELIMINAR = CANALES[j]

CANALELIMINAR se añade a CANALESELIMINADOS

```
end if  
end for  
if CANALESELIMINAR == 0  
FLAG = 0  
else  
Eliminar CANALESELIMINADOS de CANALES  
end if  
end while
```

Capítulo 6

Resultados

6.1. Introducción

En este capítulo se muestran los resultados obtenidos en cada una de las etapas de los estudios ya explicados en el capítulo dedicado a la metodología. En cada sección se muestran los resultados obtenidos en el correspondiente estudio.

6.2. Primer estudio

En esta sección se presentan los resultados del primero de los estudios, el cuál sirve de base para realizar una comparativa con el segundo, que es en el que se centra este trabajo.

6.2.1. Parámetros óptimos de los clasificadores

Tras realizar el proceso de selección de parámetros óptimos descrito previamente se obtuvieron los siguientes resultados.

6.2.1.1. LDA

El área bajo la curva ROC que se obtuvo en el proceso de validación cruzada se puede observar en la figura 6.1 para el sujeto A y en la figura 6.2 para el sujeto B. A su vez, en las figuras 6.3 y 6.4 se pueden ver las curvas ROC para el valor óptimo de cada sujeto. El valor óptimo para el parámetro de regularización fue de 13001 para el sujeto A, con un área bajo la curva ROC de 0.7834 y de 5001 para el B, con un área bajo la curva ROC de 0.8711.

6.2.1.2. SRKDA

El área bajo la curva ROC que se obtuvo en el proceso de validación cruzada se puede observar en la figura 6.5 para el sujeto A y en la figura 6.6 para el sujeto B. A su vez, en las figuras 6.7 y 6.8 se pueden ver las curvas ROC para los valores óptimos de cada sujeto. El valor óptimo para el parámetro de regularización fue de 0.05 para ambos sujetos. El

parámetro t fue de 1000 para el sujeto A, con un área bajo la curva ROC de 0.7868 y de 900 para el B, con un área bajo la curva ROC de 0.8491.

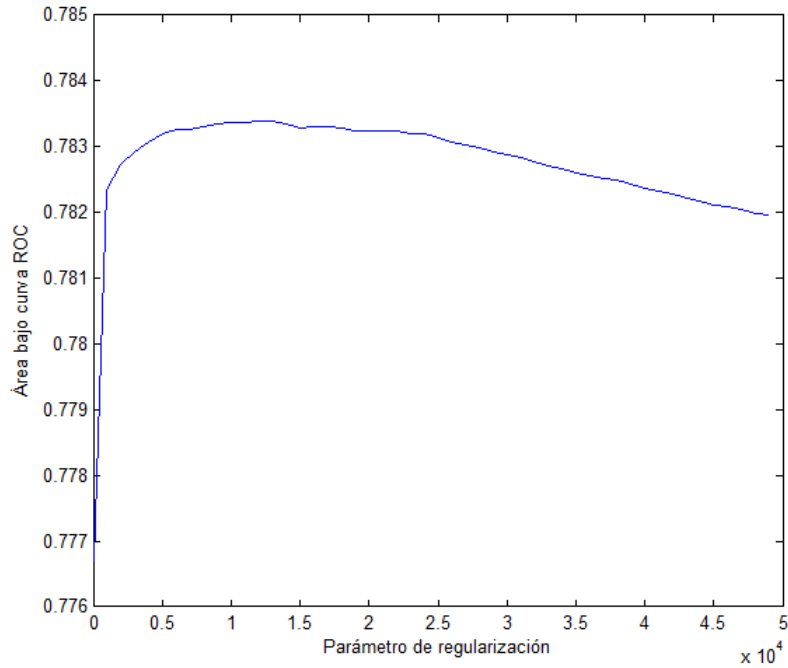


Figura 6.1: Área bajo la curva ROC para un barrido del parámetro de regularización del LDA. El valor óptimo es de 13001. Sujeto A.

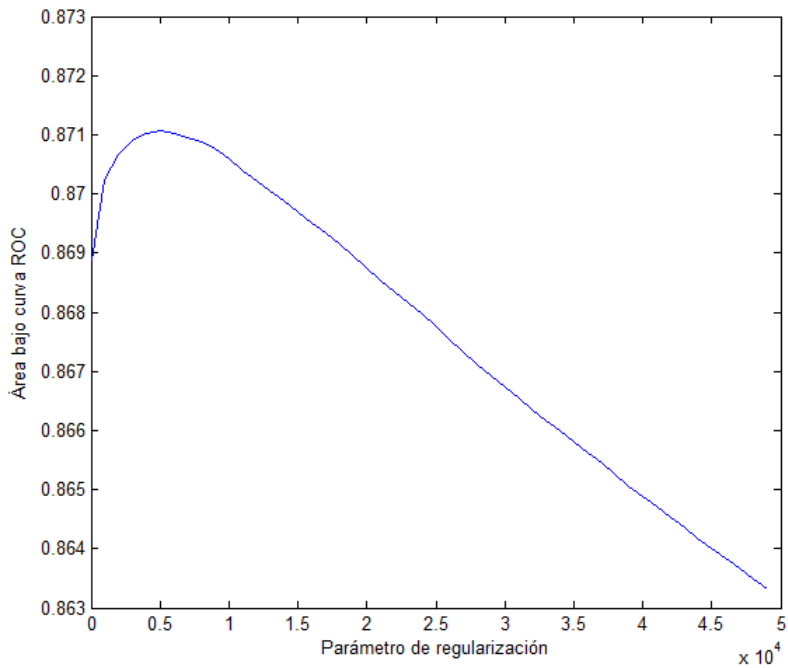


Figura 6.2: Área bajo la curva ROC para un barrido del parámetro de regularización del LDA. El valor óptimo es de 5001. Sujeto B.

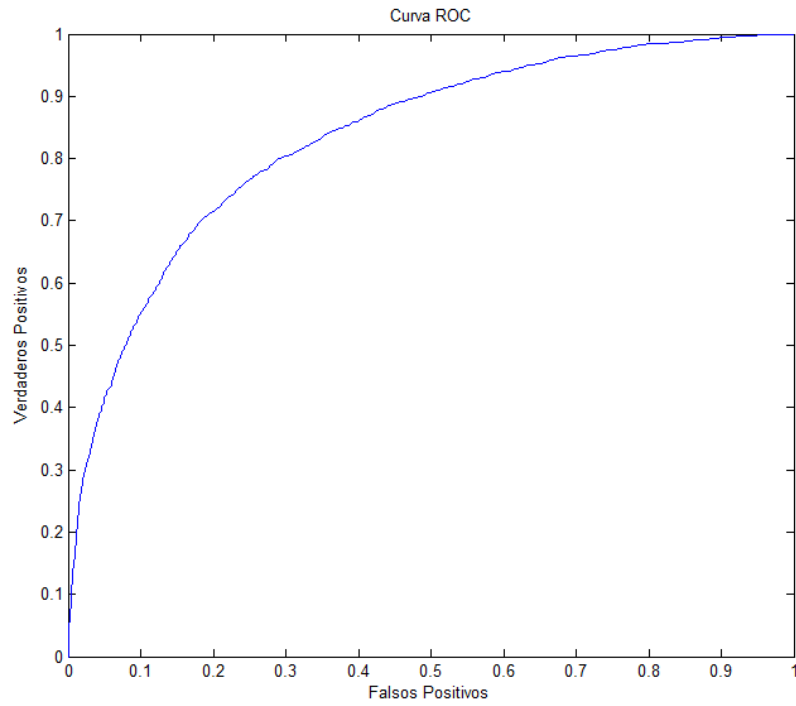


Figura 6.3: Curva ROC para el parámetro de regularización óptimo del Sujeto A. LDA.

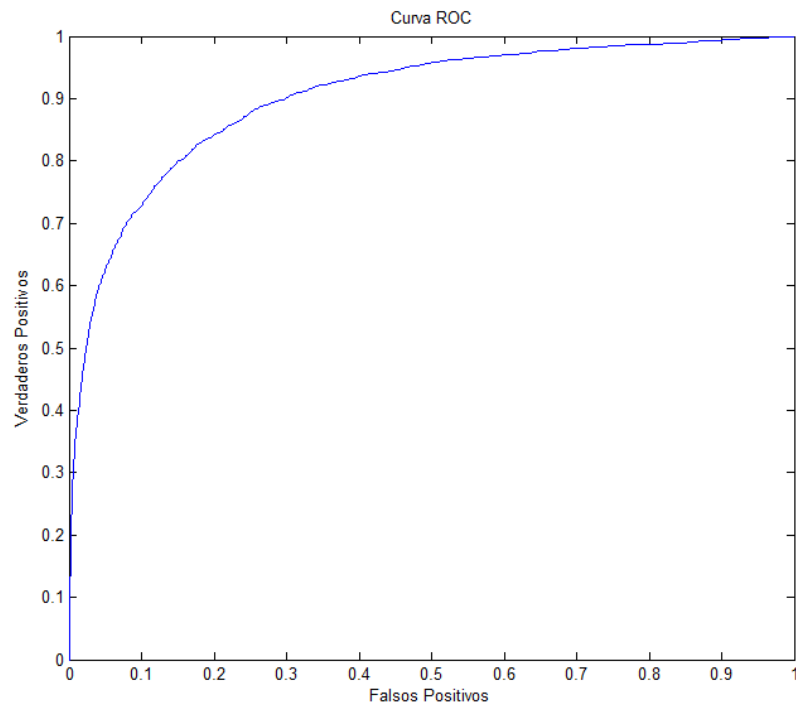


Figura 6.4: Curva ROC para el parámetro de regularización óptimo del Sujeto B. LDA.

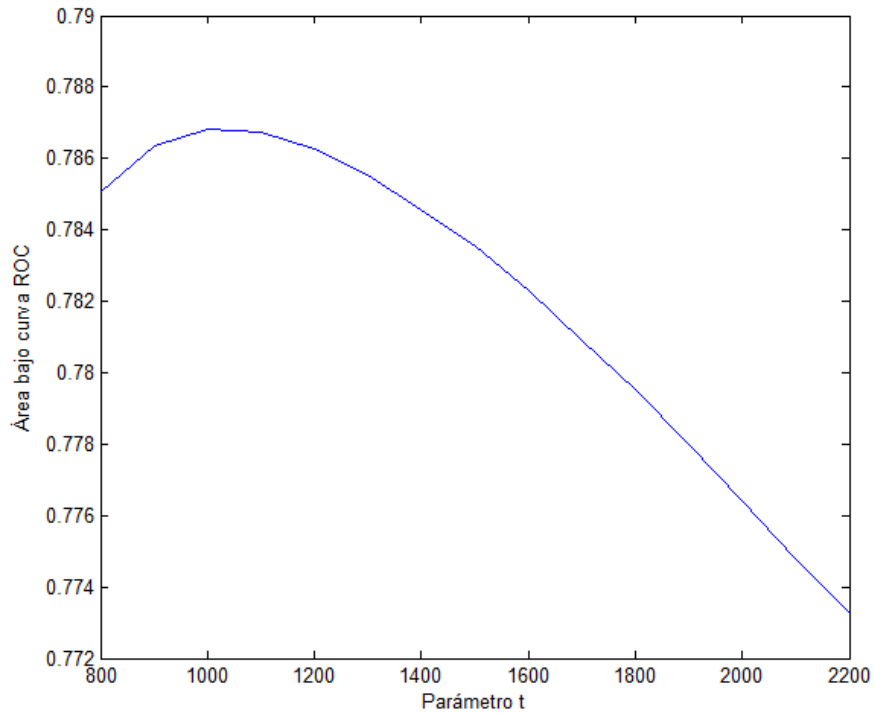


Figura 6.5: Área bajo la curva ROC para un barrido del parámetro t del SRKDA. El valor óptimo es de 1000. Para una mayor claridad solo se muestra el barrido para el valor del parámetro de regularización de 0.05. Sujeto A.

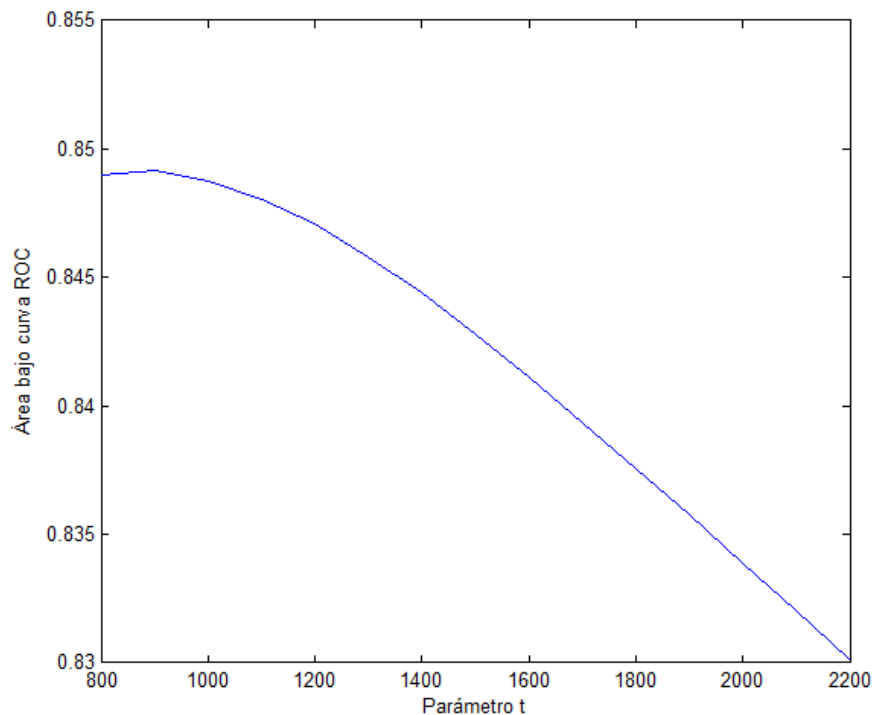


Figura 6.6: Área bajo la curva ROC para un barrido del parámetro t del SRKDA. El valor óptimo es de 900. Para una mayor claridad solo se muestra el barrido para el valor del parámetro de regularización de 0.05. Sujeto B.

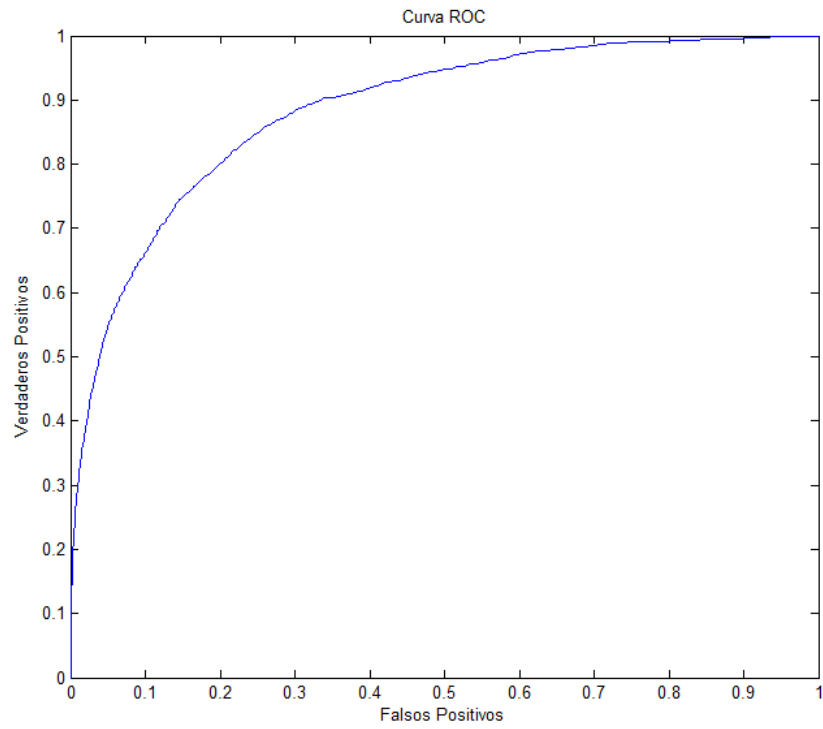


Figura 6.7: Curva ROC para el parámetro t óptimo del Sujeto A. SRKDA.

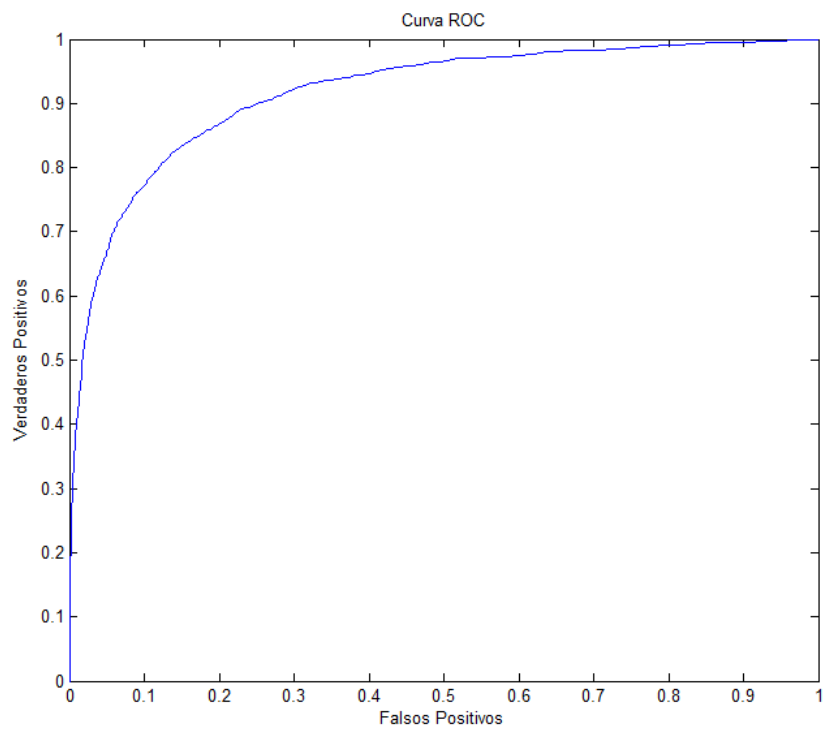


Figura 6.8: Curva ROC para el parámetro t óptimo del Sujeto B. SRKDA.

6.2.1.3. Conjunto de SVM

El área bajo la curva ROC que se obtuvo en el proceso de validación cruzada se puede observar en la figura 6.9 para el sujeto A y en la figura 6.10 para el sujeto B. El valor óptimo para el parámetro C fue de 30 para el sujeto A y el B, con un área bajo la curva ROC de 0.7834 para el A y un área bajo la curva ROC de 0.8711 para el B.

Una vez obtenido el parámetro C óptimo se procedió a determinar el número de SVM que se usarían en el conjunto. Dicho número tiene que ser impar para asegurar que nunca haya un mismo número de votos a favor de un resultado (1 o -1) en la clasificación. Para obtener el número ideal de máquinas se realizaron procedimientos de selección de caracteres sobre el conjunto de datos de entrenamiento, para 15 iluminaciones y barriendo desde 9 hasta 31 SVM. Los números que arrojaron mayor tasa de aciertos fueron 15, 17 y 19. Estos resultados se pueden ver en la tabla 6.1.

Es importante notar que SVM tiene un coste computacional que crece de forma cúbica con el número de señales de entrenamiento [61]. Teniendo en cuenta que cada SVM se entrena con una fracción de las señales de entrenamiento de $1/N$, donde N es el número de SVM del conjunto, es contraproducente usar pocas SVM. Finalmente se decidió usar 19 SVM puesto que entre aquellas con mayor tasa de aciertos era la que más máquinas contenía, por lo que requería un menor tiempo de entrenamiento.

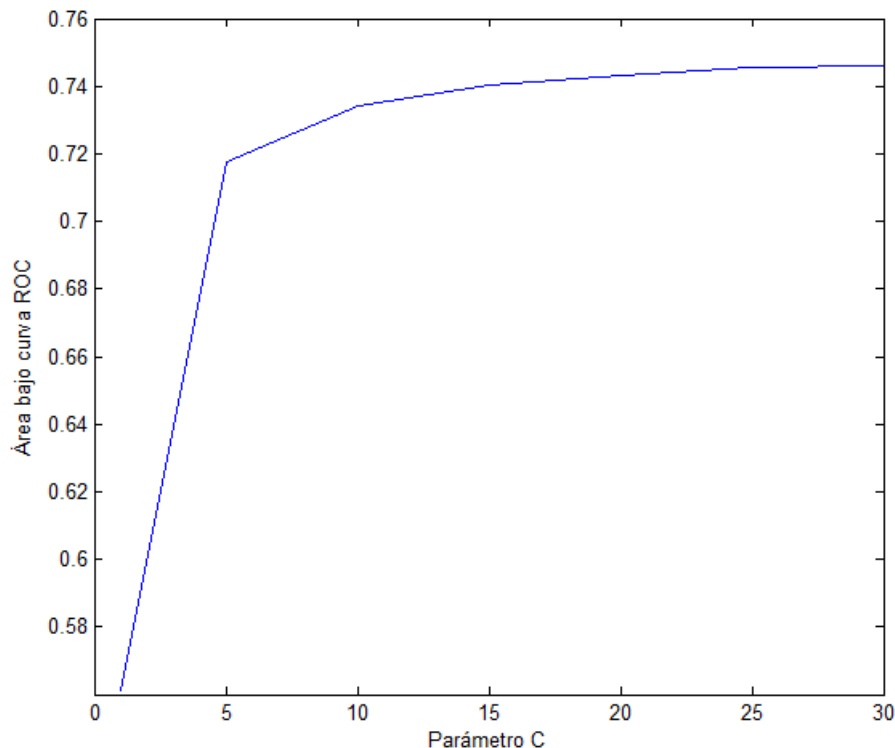


Figura 6.9: Área bajo la curva ROC para un barrido del parámetro C del conjunto de SVM. El valor óptimo de C es 30. Sujeto A.

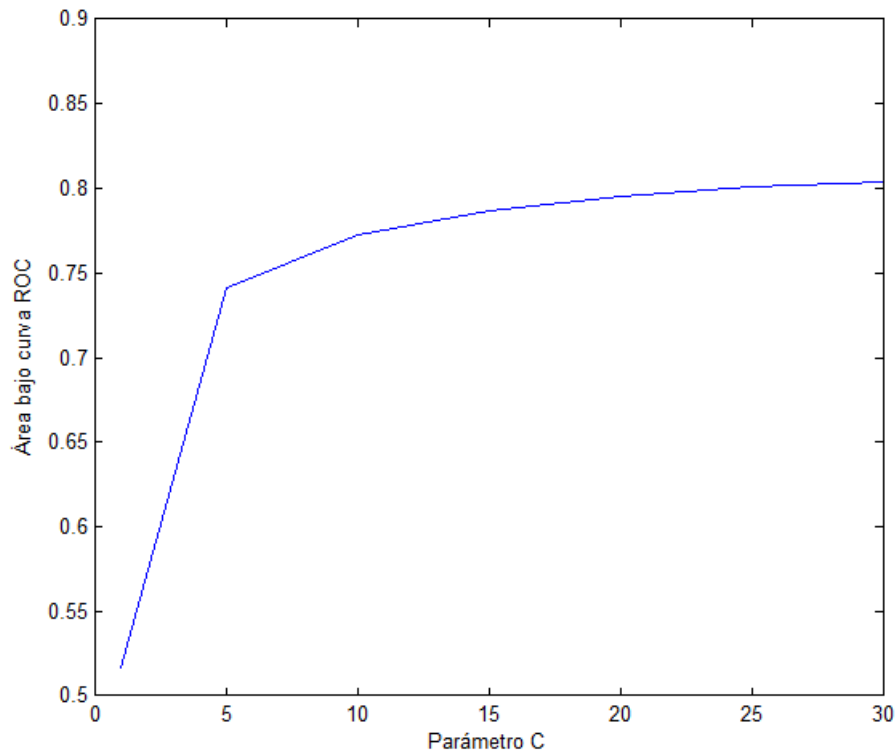


Figura 6.10: Área bajo la curva ROC para un barrido del parámetro C del conjunto de SVM. El valor óptimo de C es 30. Sujeto B.

	Número de SVM											
	9	11	13	15	17	19	21	23	25	27	29	31
	Porcentaje de aciertos (%)											
Sujeto A	98.82	98.82	98.82	100	100	100	98.82	97.65	100	98.82	98.82	97.65
Sujeto B	96.47	96.47	95,29	96,47	96,47	96,47	95,29	95,29	95,29	96,47	96,47	96,47
Media	97.64	97.64	97.05	98.23	98.23	98.23	97.64	96.47	97.64	97.64	97.64	97.06

Tabla 6.1: Resultados para 15 iluminaciones sobre el conjunto de datos de entrenamiento en función del número de SVM.

6.2.2. Resultados de los clasificadores

Una vez obtenidos los parámetros óptimos de los clasificadores se procedió a aplicar los mismos al conjunto de datos de la competición. Se realizó el proceso de clasificación para un número de iluminaciones desde 5 hasta el total de 15. Los resultados obtenidos se recogen en las tablas 6.2, 6.3 y 6.4.

6.2.2.1. LDA

La tabla 6.2 muestra los resultados obtenidos usando LDA con los parámetros de regularización óptimos para cada sujeto: 13001 para A y 5001 para B.

	Iluminaciones										
	5	6	7	8	9	10	11	12	13	14	15
	Porcentaje de aciertos (%)										
Sujeto A	65	73	74	82	85	89	95	97	98	97	98
Sujeto B	81	84	89	91	92	94	94	95	93	95	96
Media	73	78.5	81.5	86.5	88.5	91.5	94.5	96	95.5	96	97

Tabla 6.2: Resultados obtenidos mediante la clasificación usando LDA para distinto número de iluminaciones.

6.2.2.2. SRKDA

La tabla 6.3 muestra los resultados usando SRKDA y parámetros de regularización y t óptimos: 0.05 de regularización para ambos y parámetro t 1000 para A y 900 para B.

	Iluminaciones										
	5	6	7	8	9	10	11	12	13	14	15
	Porcentaje de aciertos (%)										
Sujeto A	77	79	82	83	81	89	93	96	99	99	100
Sujeto B	79	81	84	87	90	92	93	94	95	95	94
Media	78	80	83	85	85.5	90.5	93	95	97	97	97

Tabla 6.3: Resultados obtenidos mediante la clasificación usando SRKDA para distinto número de iluminaciones.

6.2.2.3. Conjunto de SVM

La tabla 6.4 muestra los resultados obtenidos usando 19 SVM y parámetro C óptimo para cada sujeto.

	Iluminaciones										
	5	6	7	8	9	10	11	12	13	14	15
	Porcentaje de aciertos (%)										
Sujeto A	63	64	77	76	82	86	89	91	90	94	96
Sujeto B	74	72	85	85	89	92	91	92	90	92	94
Media	68.5	68	81	80.5	85.5	89	90	91.5	90	93	95

Tabla 6.4: Resultados obtenidos mediante la clasificación usando un conjunto de 19 SVM para distinto número de iluminaciones.

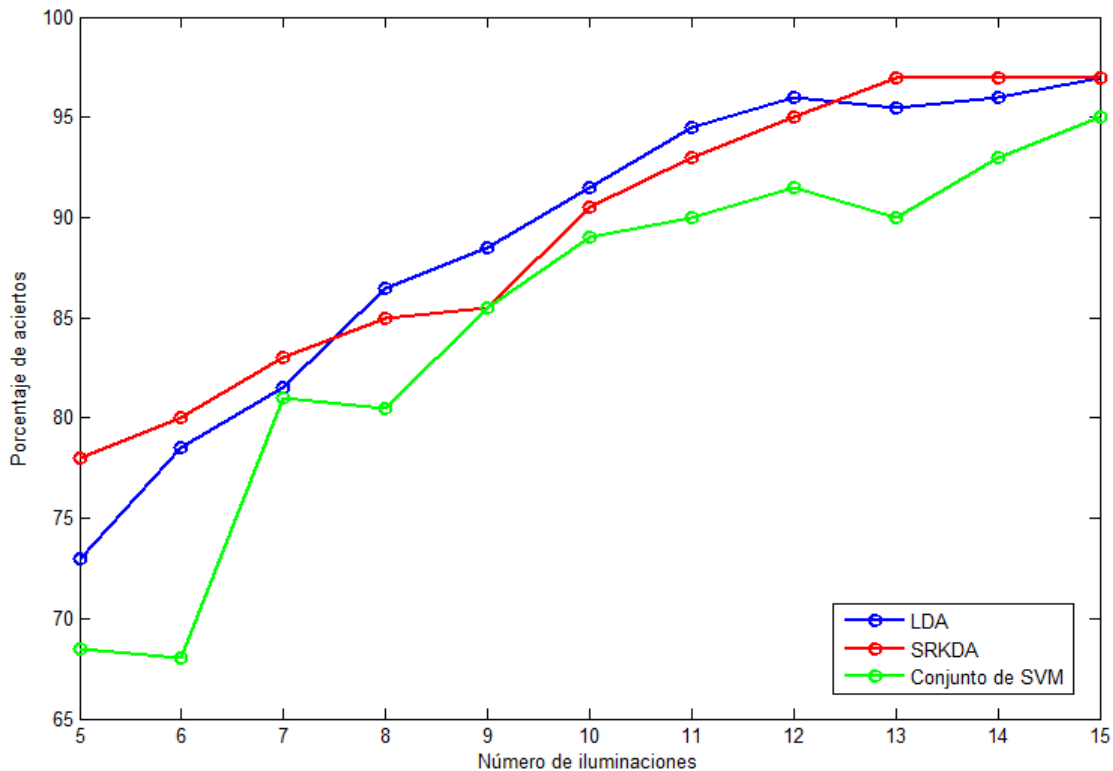


Figura 6.11: Comparativa porcentaje de aciertos de caracteres en función del método de clasificación y el número de iluminaciones utilizadas para la misma.

Finalmente, la figura 6.11 presenta una gráfica comparativa con los resultados medios obtenidos con los tres métodos de este estudio.

6.3. Segundo estudio

En este apartado se muestran los resultados del segundo estudio por etapas, de forma que se puede apreciar el proceso de optimización hasta llegar a los resultados finales que se discuten en el capítulo posterior.

6.3.1. Primera etapa: evaluación del sub-muestreo óptimo

6.3.1.1. LDA

En la tabla 6.5 se observan los resultados para cada sub-muestreo en el caso del sujeto A y en la 6.6 los del sujeto B. En el caso del sujeto A, el sub-muestreo que da mejores resultados para 15 iluminaciones es 10 muestras por canal e iluminación (0.1 - 15 Hz), con un porcentaje de aciertos del 100 %, por lo que se considera como sub-muestro óptimo. En el caso del sujeto B, los sub-muestrros de 8 y 10 muestras arrojan los mismos resultados para 15 iluminaciones (98 %), pero el de 8 muestras por canal e iluminación (0.1 - 12 Hz) da mejores resultados para 5 iluminaciones, así que se considera óptimo.

Por tanto, finalmente se seleccionan sub-muestras de **10 muestras** por canal e iluminación para el **sujeto A** y **8 muestras** por canal e iluminación para el **sujeto B**.

6.3.1.2. Conjunto de 2 SRKDA

En la tabla 6.7 se observan los resultados para cada sub-muestreo en el caso del sujeto A y en la 6.8 los del sujeto B.

En el caso del sujeto A, el sub-muestreo que da mejores resultados para 15 iluminaciones es 14 muestras por canal e iluminación (0.1 - 20 Hz), con un porcentaje de aciertos del 100 %, por lo que se considera como sub-muestro óptimo. En el caso del sujeto B, el sub-muestreo de 8 muestras por canal e iluminación (0.1 - 12 Hz) es el que obtiene mejores resultados, con un porcentaje de aciertos del 99 %

Por tanto, finalmente se seleccionan sub-muestras de **14 muestras** por canal e iluminación para el **sujeto A** y **8 muestras** por canal e iluminación para el **sujeto B**.

6.3.1.3. Conjunto de 19 SVM

En la tabla 6.9 se observan los resultados para cada sub-muestreo en el caso del sujeto A y en la 6.10 los del sujeto B.

En el caso del sujeto A, los sub-muestras de 12 y 16 muestras por canal e iluminación arrojan los mismos resultados para 15 iluminaciones (96 %) y para 5 (65 %). Para resolver este conflicto se selecciona el sub-muestreo de 12 muestras por canal e iluminación ya que al tener menos muestras suponen menos carga computacional para los clasificadores. Para el sujeto B el sub-muestreo de 14 muestras por canal e iluminación (0.1 - 20 Hz) arroja los mejores resultados para 15 iluminaciones, con una tasa de aciertos del 94 %.

Por tanto, finalmente se seleccionan sub-muestras de **12 muestras** por canal e iluminación para el **sujeto A** y **14 muestras** por canal e iluminación para el **sujeto B**.

6.3.2. Segunda etapa: selección de la Wavelet Madre óptima

6.3.2.1. LDA

En la tabla 6.11 se muestran los resultados para el sujeto A y en la tabla 6.12 los resultados del sujeto B.

6.3.2.2. Conjunto de 2 SRKDA

En la tabla 6.13 se muestran los resultados para el sujeto A y en la tabla 6.14 los resultados del sujeto B. Como se comprueba observando estas tablas, cada uno de los sujetos tiene una Wavelet Madre que mejora los resultados de la clasificación de alguna forma y se considera óptima para el sujeto y el número de muestras.

En el caso del sujeto A no seleccionamos ninguna Wavelet Madre óptima, ya que ninguna mejora el porcentaje de aciertos del 77 % para 5 iluminaciones, y el porcentaje del 100 % para 15 iluminaciones no es mejorable.

En el caso del sujeto B tampoco hay Wavelet Madre óptima, ya que ninguna supera el porcentaje de aciertos de 99 % para 15 iluminaciones ni el de 75 % para 5 iluminaciones.

Muestras por canal	Iluminaciones	
	5	15
	Porcentaje de aciertos (%)	
4 muestras (0.1 - 6 Hz)	30	75
6 muestras (0.1 - 9 Hz)	56	91
8 muestras (0.1 - 12 Hz)	65	95
10 muestras (0.1 - 15 Hz)	55	100
12 muestras (0.1 - 18 Hz)	62	98
14 muestras (0.1 - 20 Hz)	65	98
16 muestras (0.1 - 24 Hz)	61	97
18 muestras (0.1 - 27 Hz)	58	96
20 muestras (0.1 - 30 Hz)	59	95
22 muestras (0.1 - 33 Hz)	58	94

Tabla 6.5: Resultados obtenidos para el Sujeto A con el clasificador LDA para cada sub-muestreo. El mejor resultado para 15 iluminaciones aparece en negrita.

Muestras por canal	Iluminaciones	
	5	15
	Porcentaje de aciertos (%)	
4 muestras (0.1 - 6 Hz)	55	87
6 muestras (0.1 - 9 Hz)	74	97
8 muestras (0.1 - 12 Hz)	79	98
10 muestras (0.1 - 15 Hz)	78	98
12 muestras (0.1 - 18 Hz)	81	96
14 muestras (0.1 - 20 Hz)	81	96
16 muestras (0.1 - 24 Hz)	81	95
18 muestras (0.1 - 27 Hz)	79	95
20 muestras (0.1 - 30 Hz)	78	93
22 muestras (0.1 - 33 Hz)	77	93

Tabla 6.6: Resultados obtenidos para el Sujeto B con el clasificador LDA para cada sub-muestreo. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estos dos sub-muestreos el que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

Muestras por canal	Iluminaciones	
	5	15
	Porcentaje de aciertos (%)	
4 muestras (0.1 - 6 Hz)	34	74
6 muestras (0.1 - 9 Hz)	53	88
8 muestras (0.1 - 12 Hz)	64	94
10 muestras (0.1 - 15 Hz)	68	99
12 muestras (0.1 - 18 Hz)	74	100
14 muestras (0.1 - 20 Hz)	77	100
16 muestras (0.1 - 24 Hz)	73	100
18 muestras (0.1 - 27 Hz)	70	100
20 muestras (0.1 - 30 Hz)	76	100
22 muestras (0.1 - 33 Hz)	71	100

Tabla 6.7: Resultados obtenidos para el Sujeto A con el conjunto de 2 SRKDA como clasificador para cada sub-muestreo. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estos dos sub-muestreos el que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

Muestras por canal	Iluminaciones	
	5	15
	Porcentaje de aciertos (%)	
4 muestras (0.1 - 6 Hz)	49	79
6 muestras (0.1 - 9 Hz)	72	94
8 muestras (0.1 - 12 Hz)	75	99
10 muestras (0.1 - 15 Hz)	74	97
12 muestras (0.1 - 18 Hz)	76	97
14 muestras (0.1 - 20 Hz)	76	97
16 muestras (0.1 - 24 Hz)	79	97
18 muestras (0.1 - 27 Hz)	76	93
20 muestras (0.1 - 30 Hz)	75	92
22 muestras (0.1 - 33 Hz)	73	92

Tabla 6.8: Resultados obtenidos para el Sujeto B con el conjunto de 2 SRKDA como clasificador para cada sub-muestreo. El mejor resultado para 15 iluminaciones aparece en negrita.

Muestras por canal	Iluminaciones	
	5	15
	Porcentaje de aciertos (%)	
4 muestras (0.1 - 6 Hz)	29	69
6 muestras (0.1 - 9 Hz)	46	82
8 muestras (0.1 - 12 Hz)	59	93
10 muestras (0.1 - 15 Hz)	56	93
12 muestras (0.1 - 18 Hz)	65	96
14 muestras (0.1 - 20 Hz)	63	96
16 muestras (0.1 - 24 Hz)	65	96
18 muestras (0.1 - 27 Hz)	64	93
20 muestras (0.1 - 30 Hz)	62	95
22 muestras (0.1 - 33 Hz)	58	93

Tabla 6.9: Resultados obtenidos para el Sujeto A con el conjunto de 19 SVM como clasificador para cada sub-muestreo. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estos dos sub-muestreos el que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

Muestras por canal	Iluminaciones	
	5	15
	Porcentaje de aciertos (%)	
4 muestras (0.1 - 6 Hz)	38	64
6 muestras (0.1 - 9 Hz)	61	88
8 muestras (0.1 - 12 Hz)	66	91
10 muestras (0.1 - 15 Hz)	69	92
12 muestras (0.1 - 18 Hz)	73	92
14 muestras (0.1 - 20 Hz)	74	94
16 muestras (0.1 - 24 Hz)	77	93
18 muestras (0.1 - 27 Hz)	70	93
20 muestras (0.1 - 30 Hz)	69	93
22 muestras (0.1 - 33 Hz)	67	91

Tabla 6.10: Resultados obtenidos para el Sujeto B con el conjunto de 19 SVM como clasificador para cada sub-muestreo. El mejor resultado para 15 iluminaciones aparece en negrita.

Muestras por canal	Iluminaciones	
	5	15
	Porcentaje de aciertos (%)	
Normal	55	100
'haar'	55	100
'db1'	55	100
'db2'	56	100
'db3'	55	99
'db4'	56	99
'db5'	56	99
'db6'	56	99
'db7'	56	99
'db8'	56	99
'db9'	55	99
'db10'	55	99
'sym2'	56	100
'sym3'	55	99
'sym4'	57	99
'sym5'	56	99
'sym6'	56	99
'sym7'	56	99
'sym8'	54	99
'coif1'	57	100
'coif2'	56	99
'coif3'	56	99
'coif4'	56	99
'coif5'	56	99

Tabla 6.11: Resultados obtenidos para el Sujeto A con el clasificador LDA para cada Wavelet madre y 10 muestras por canal e iluminación. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estas Wavelets Madre la que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

Muestras por canal	Iluminaciones	
	5	15
Normal	79	98
'haar'	77	98
'db1'	77	98
'db2'	78	98
'db3'	77	98
'db4'	78	98
'db5'	79	98
'db6'	79	98
'db7'	79	98
'db8'	79	98
'db9'	79	98
'db10'	79	98
'sym2'	78	98
'sym3'	77	98
'sym4'	81	98
'sym5'	79	98
'sym6'	79	98
'sym7'	79	98
'sym8'	79	98
'coif1'	78	98
'coif2'	79	98
'coif3'	79	98
'coif4'	80	97
'coif5'	79	98

Tabla 6.12: Resultados obtenidos para el Sujeto B con el clasificador LDA para cada Wavelet madre y 8 muestras por canal e iluminación. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estas Wavelets Madre la que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

Muestras por canal	Iluminaciones	
	5	15
	Porcentaje de aciertos (%)	
Normal	77	100
'haar'	77	100
'db1'	77	100
'db2'	76	100
'db3'	75	100
'db4'	75	100
'db5'	76	100
'db6'	76	100
'db7'	77	100
'db8'	76	100
'db9'	74	100
'db10'	73	100
'sym2'	76	100
'sym3'	75	100
'sym4'	75	100
'sym5'	76	100
'sym6'	73	100
'sym7'	74	100
'sym8'	76	100
'coif1'	75	100
'coif2'	76	100
'coif3'	73	100
'coif4'	72	100
'coif5'	70	100

Tabla 6.13: Resultados obtenidos para el Sujeto A con el conjunto de 2 SRKDA como clasificador para cada Wavelet madre y 10 muestras por canal e iluminación. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estas Wavelets Madre la que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

Muestras por canal	Iluminaciones	
	5	15
Normal	75	99
'haar'	74	99
'db1'	74	99
'db2'	74	99
'db3'	72	98
'db4'	74	99
'db5'	73	99
'db6'	72	99
'db7'	72	95
'db8'	72	95
'db9'	71	95
'db10'	71	95
'sym2'	74	99
'sym3'	72	98
'sym4'	72	99
'sym5'	73	99
'sym6'	73	98
'sym7'	73	98
'sym8'	73	97
'coif1'	74	99
'coif2'	73	99
'coif3'	71	95
'coif4'	72	95
'coif5'	73	94

Tabla 6.14: Resultados obtenidos para el Sujeto B con el conjunto de 2 SRKDA como clasificador para cada Wavelet madre y 8 muestras por canal e iluminación. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estas Wavelets Madre la que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

Muestras por canal	Iluminaciones	
	5	15
Normal	65	96
'haar'	60	93
'db1'	62	91
'db2'	60	93
'db3'	57	94
'db4'	57	94
'db5'	63	94
'db6'	60	95
'db7'	62	93
'db8'	60	95
'db9'	62	92
'db10'	64	93
'sym2'	60	93
'sym3'	57	94
'sym4'	60	97
'sym5'	60	93
'sym6'	62	94
'sym7'	67	93
'sym8'	59	92
'coif1'	61	96
'coif2'	58	97
'coif3'	61	93
'coif4'	60	94
'coif5'	56	92

Tabla 6.15: Resultados obtenidos para el Sujeto A con el conjunto de 19 SVM como clasificador para cada Wavelet madre y 10 muestras por canal e iluminación. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estas Wavelets Madre la que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

Muestras por canal	Iluminaciones	
	5	15
Normal	74	94
'haar'	74	94
'db1'	74	94
'db2'	75	94
'db3'	74	92
'db4'	73	91
'db5'	75	93
'db6'	76	93
'db7'	75	93
'db8'	77	93
'db9'	69	93
'db10'	69	93
'sym2'	75	94
'sym3'	74	92
sym4'	69	92
'sym5'	70	93
'sym6'	72	93
'sym7'	73	94
'sym8'	77	93
'coif1'	73	93
'coif2'	71	94
'coif3'	75	93
'coif4'	73	93
'coif5'	76	92

Tabla 6.16: Resultados obtenidos para el Sujeto B con el conjunto de 19 SVM como clasificador para cada Wavelet madre y 8 muestras por canal e iluminación. Los mejores resultados para 15 iluminaciones aparecen en negrita. Dentro de estas Wavelets Madre la que mejores resultados arroja para 5 iluminaciones también se muestra en negrita.

6.3.2.3. Conjunto de 19 SVM

En la tabla 6.15 se muestran los resultados para el sujeto A y en la tabla 6.16 los resultados del sujeto B.

Como se comprueba observando estas tablas, cada uno de los sujetos tiene una Wavelet Madre que mejora los resultados de la clasificación de alguna forma y se considera óptima para el sujeto y el número de muestras.

En el caso del sujeto A la Wavelet Madre óptima es ‘sym4’, ya que mejora el porcentaje de aciertos del 96 % para 15 iluminaciones, pasando al 97 %, aunque la clasificación para 5 iluminaciones empeora, pasando del 65 % al 60 %.

En el caso del sujeto B la Wavelet Madre óptima queda empatada entre ‘db2’ y ‘sym2’, ya que mantienen el porcentaje de aciertos de 94 % para 15 iluminaciones y mejoran la clasificación para 5 iluminaciones, pasando del 74 % al 75 %. En este caso, por tanto, se calcularán los resultados de la siguiente etapa para ambas Wavelet Madre y para los resultados medios finales se seleccionará aquella que dé mejor puntuación de la clasificación para 14 iluminaciones, si se vuelve a dar un caso de empate, para 13, y así sucesivamente.

6.3.3. Tercera etapa: clasificación con los parámetros óptimos

En las tablas 6.17 y 6.18 se muestra un resumen de los resultados obtenidos para cada sujeto y cada clasificador.

La tabla 6.19 es una tabla resumen con los resultados medios de los clasificadores. Es necesario apuntar que la media de los resultados entre los dos sujetos no es una medida muy correcta, ya que se están sumando resultados obtenidos con el mismo clasificador pero distintos sub-muestreos y Wavelets Madre. Sin embargo, esta es una métrica que se pedía en la III Competición BCI, por lo que se aporta en este trabajo.

	Iluminaciones										
	5	6	7	8	9	10	11	12	13	14	15
	Porcentaje de aciertos (%)										
LDA	57	63	72	76	79	83	88	93	93	94	100
SRKDA	77	78	81	81	84	87	94	96	98	99	100
SVM	60	61	71	73	72	83	85	89	86	93	97

Tabla 6.17: Resultados obtenidos para el Sujeto A con los tres clasificadores para 10, 14 y 12 muestras por canal e iluminación para cada clasificador respectivamente (LDA, SRKDA, SVM) y extracción de características mediante Transformada Wavelet con Wavelet Madre ‘coif1’ para LDA, ninguna para el conjunto de 2 SRKDA y ‘sym4’ para el conjunto de SVM. Los mejores resultados para 15 y 5 iluminaciones aparecen en negrita.

	Iluminaciones										
	5	6	7	8	9	10	11	12	13	14	15
	Porcentaje de aciertos (%)										
LDA	81	87	87	91	95	94	97	97	97	98	98
SRKDA	74	78	88	92	92	95	96	99	99	98	99
SVM db2	75	78	85	85	89	92	93	93	93	94	94
SVM sym2	75	78	85	85	89	92	93	93	93	94	94

Tabla 6.18: Resultados obtenidos para el Sujeto B con los tres clasificadores para 8, 8 y 14 muestras por canal e iluminación para cada clasificador respectivamente (LDA, SRKDA, SVM) y extracción de características mediante Transformada Wavelet con Wavelet Madre ‘sym4’ para LDA, ninguna para el conjunto de 2 SRKDA y ‘sym2’ y ‘db2’ para el conjunto de SVM. Los mejores resultados para 15 y 5 iluminaciones aparecen en negrita.

	Iluminaciones										
	5	6	7	8	9	10	11	12	13	14	15
	Porcentaje de aciertos (%)										
LDA	69	75	79.5	83.5	87	88.5	92.5	95	95	96	99
SRKDA	75	78	84	86.5	88	91	95	97.5	98.5	98.5	99.5
SVM	67.5	69.5	78	79	80.5	87.5	89	91	89.5	93.5	95.5

Tabla 6.19: Resultados medios obtenidos para ambos sujetos con los tres clasificadores. Los mejores resultados para 15 y 5 iluminaciones aparecen en negrita.

En las figuras 6.12 y 6.13 se muestran los resultados para el sujeto A y B de forma gráfica para los tres clasificadores con sus sub-muestreos y Wavelet Madre óptimas. En la figura 6.14 se hace lo mismo para los valores medios de la clasificación. Es importante destacar que en el caso del sujeto B con SVM como clasificador no se resolvió el empate entre ‘db2’ y ‘sym2’, ya que obtuvieron exactamente los mismos resultados en la clasificación. Por tanto, se consideran ambos resultados como óptimos.

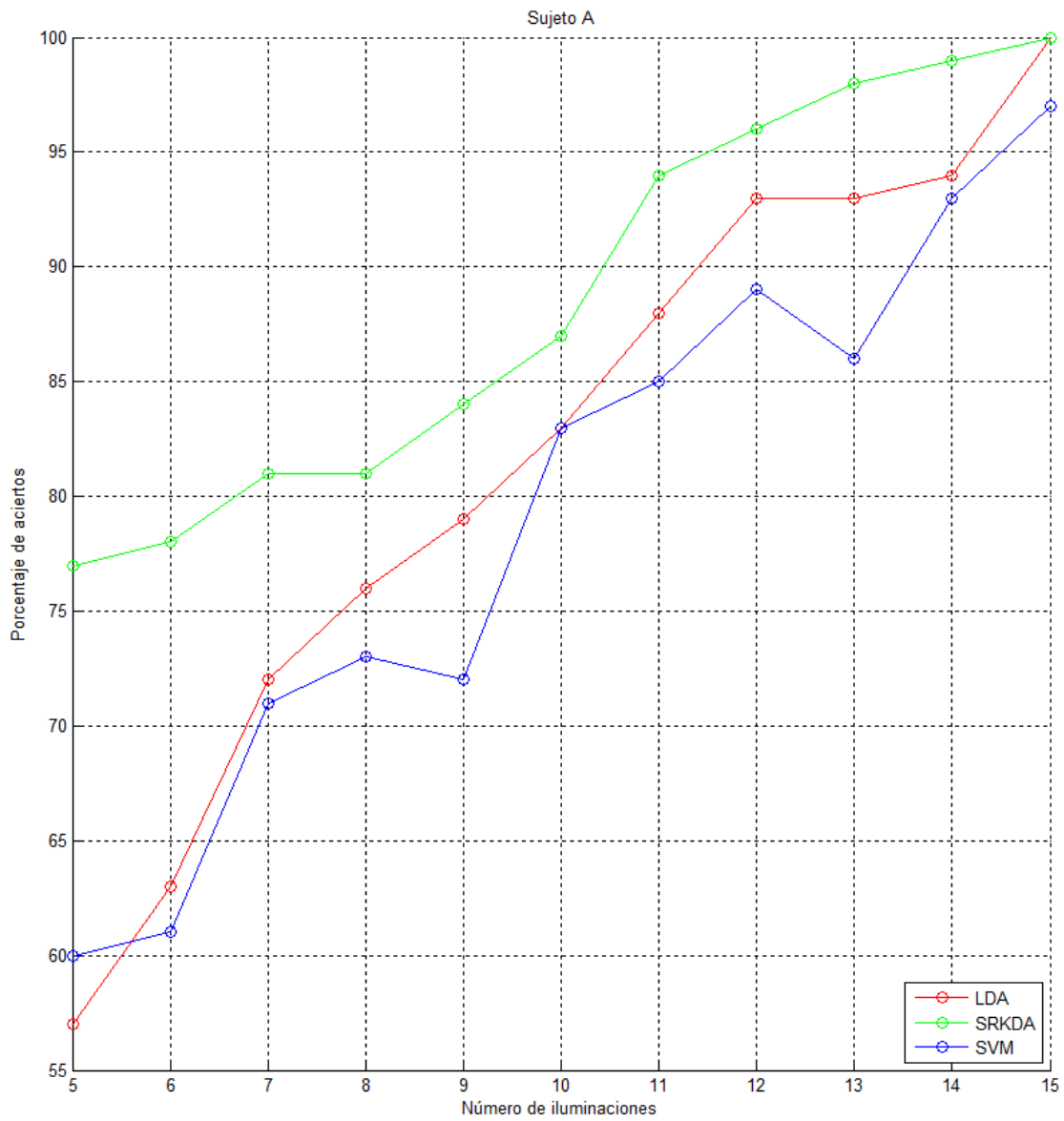


Figura 6.12: Representación gráfica de los valores obtenidos para el Sujeto A con los tres clasificadores y extracción de características mediante sub-muestreo y Transformada Wavelet.

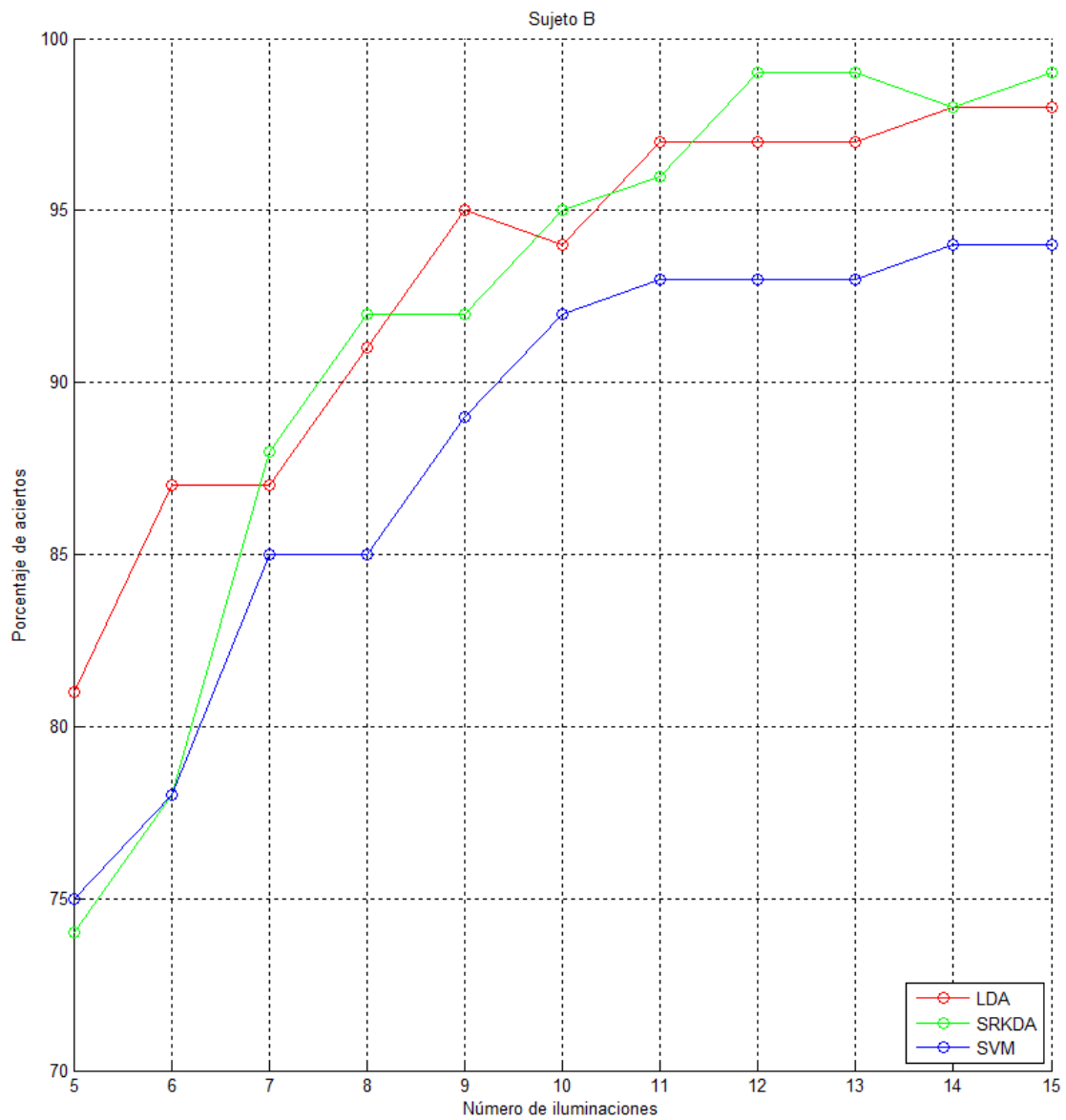


Figura 6.13: Representación gráfica de los valores obtenidos para el Sujeto B con los tres clasificadores y extracción de características mediante sub-muestreo y Transformada Wavelet.

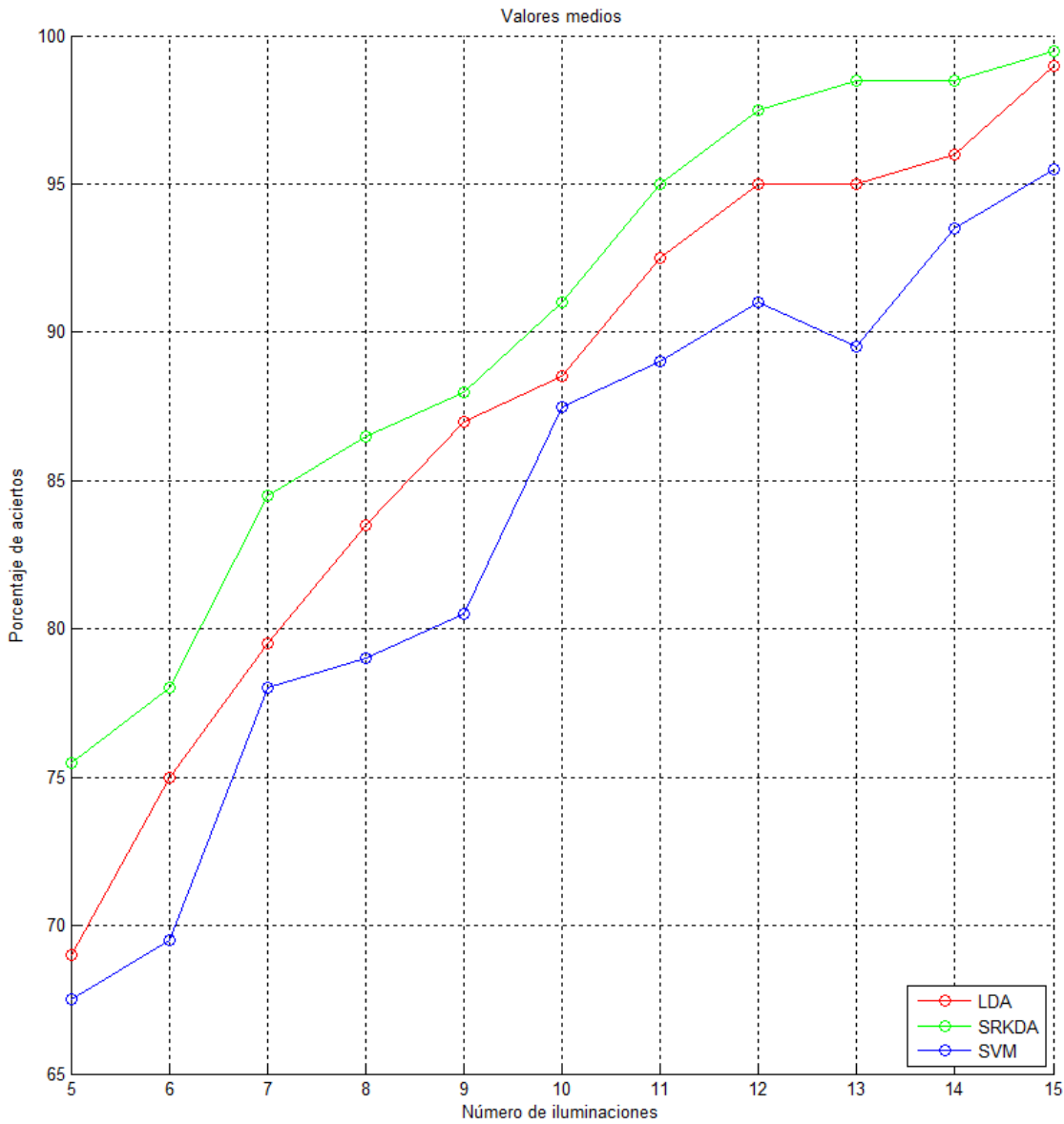


Figura 6.14: Representación gráfica de los valores obtenidos para los valores medios del Sujeto A y B con los tres clasificadores y extracción de características mediante submuestreo óptimo y Transformada Wavelet con sus respectivas Wavelets Madre.

6.3.4. Cuarta etapa: algoritmo de eliminación de canales

Este experimento se realiza a mayores del principal, con el objetivo de comparar los resultados cuando se aplica el algoritmo de eliminación de canales y cuando no. En las tablas 6.20 y 6.21 se muestran los resultados obtenidos para ambos sujetos tras la eliminación de canales. Se muestran en verde los resultados superiores a aquellos obtenidos sin eliminación de canales y en rojo los que sean inferiores.

Los canales eliminados para el sujeto A fueron, según el clasificador:

- **LDA:** 22, 24, 39, 52
- **SRKDA:** Ninguno
- **SVM:** 3, 7, 16, 18, 30, 60

Los canales eliminados para el sujeto B fueron, según el clasificador:

- **LDA:** Ninguno
- **SRKDA:** Ninguno
- **SVM:** 3, 8, 25, 28, 29, 35

	Iluminaciones										
	5	6	7	8	9	10	11	12	13	14	15
	Porcentaje de aciertos (%)										
LDA	57	66	73	77	79	84	89	95	93	93	99
SRKDA	77	78	81	81	84	87	94	96	98	99	100
SVM	57	58	68	72	75	81	85	88	86	93	91

Tabla 6.20: Resultados obtenidos para el Sujeto A con los tres clasificadores para 10 muestras por canal e iluminación, extracción de características mediante Transformada Wavelet y eliminación de canales. Los resultados que mejoran la clasificación respecto a cuando no hay eliminación de canales aparecen en verde y los que empeoran en rojo.

	Iluminaciones										
	5	6	7	8	9	10	11	12	13	14	15
	Porcentaje de aciertos (%)										
LDA	81	87	87	91	95	94	97	97	97	98	98
SRKDA	74	78	88	92	92	95	96	99	99	98	99
SVM	71	77	86	87	89	90	91	93	94	92	93

Tabla 6.21: Resultados obtenidos para el Sujeto B con los tres clasificadores para 8 muestras por canal e iluminación, extracción de características mediante Transformada Wavelet y eliminación de canales. Los resultados que mejoran la clasificación respecto a cuando no hay eliminación de canales aparecen en verde y los que empeoran en rojo.

En las figuras 6.15, 6.16 y 6.17 se muestran unas gráficas comparativas entre aquellos clasificadores en los cuáles se eliminaron canales al aplicar el algoritmo de eliminación. La figura 6.15 muestra la comparativa de LDA para el sujeto A y la 6.17 la comparativa del conjunto de SVM para el sujeto B.

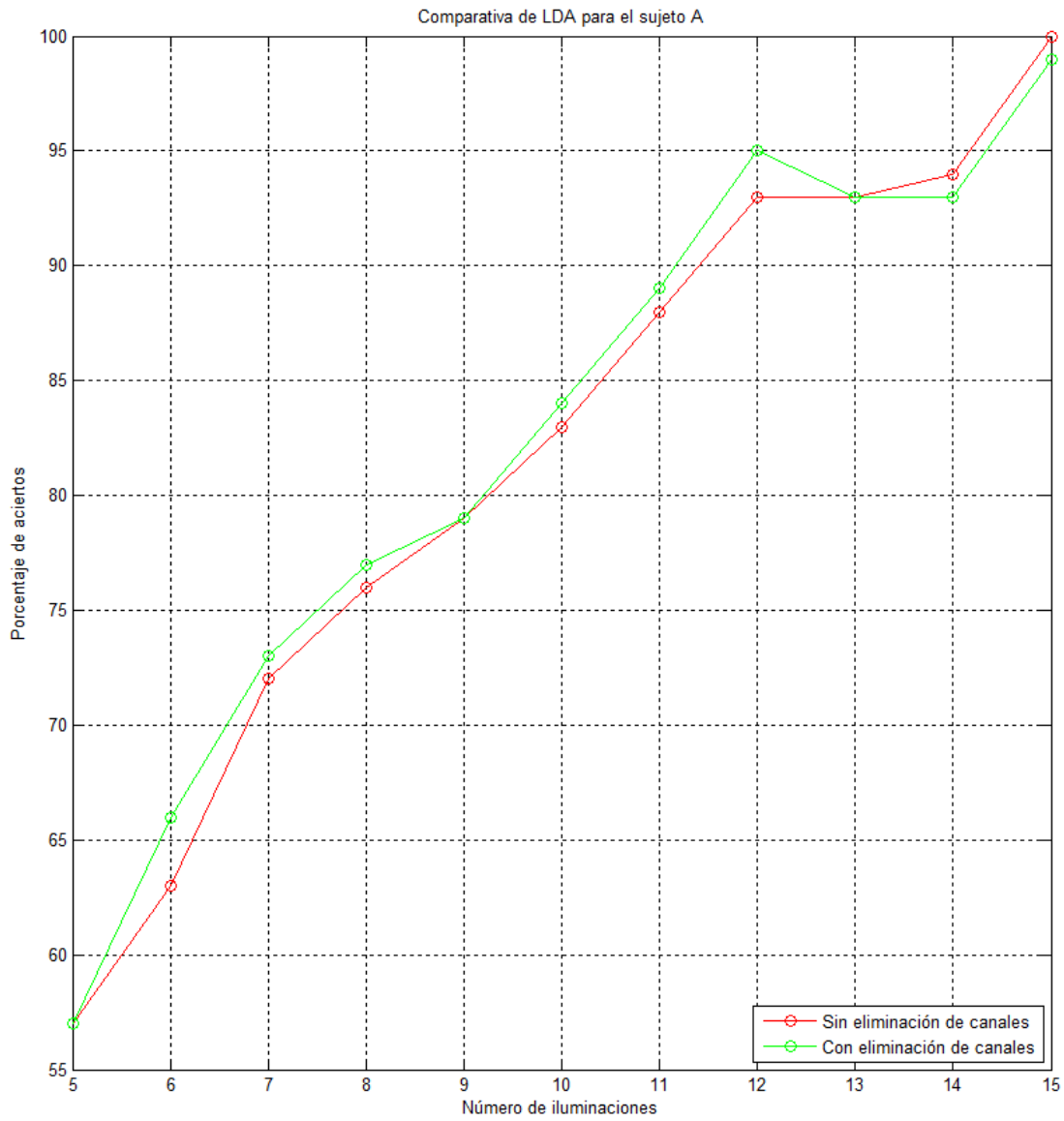


Figura 6.15: Representación gráfica de los valores obtenidos para el sujeto A con LDA, con y sin la aplicación del algoritmo de eliminación de canales.

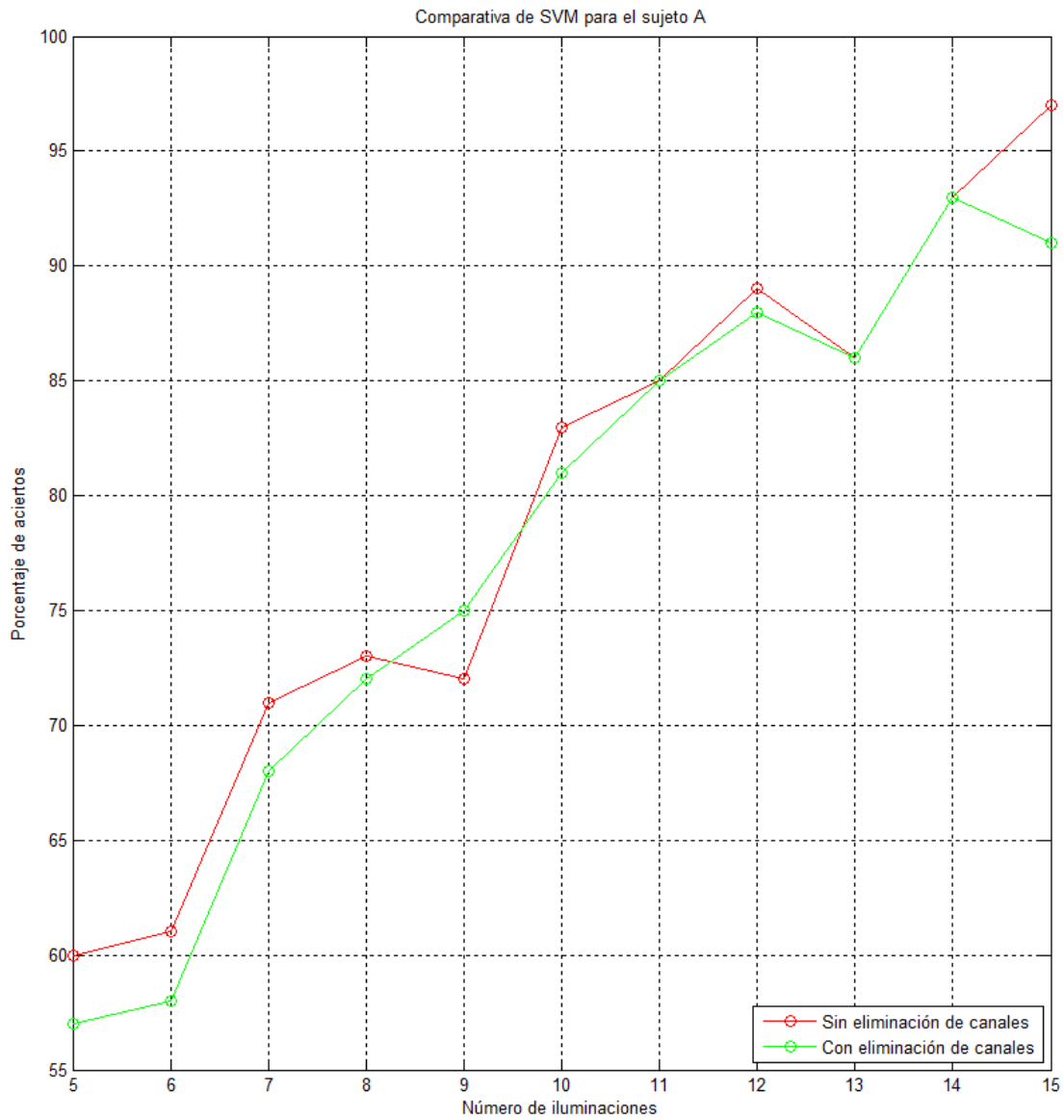


Figura 6.16: Representación gráfica de los valores obtenidos para el sujeto A con SVM, con y sin la aplicación del algoritmo de eliminación de canales.

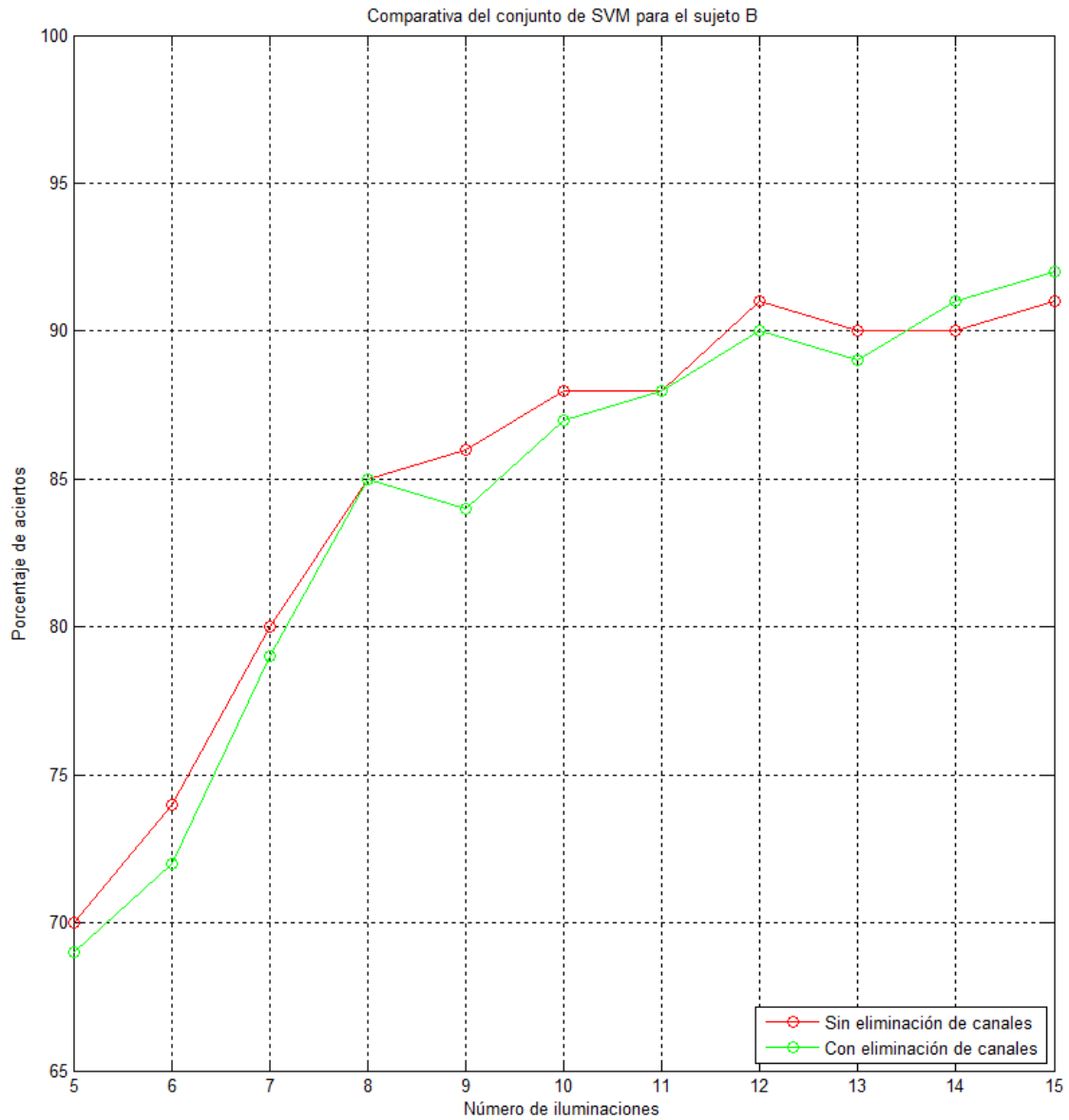


Figura 6.17: Representación gráfica de los valores obtenidos para el sujeto B con el conjunto de SVM, con y sin la aplicación del algoritmo de eliminación de canales.

Capítulo 7

Discusión de los resultados

7.1. Discusión de los resultados del sub-muestreo

Los resultados del sub-muestreo óptimo difieren para cada clasificador y sujeto, repitiéndose solo 8 muestras por canal e iluminación, para el sujeto B, con LDA y SRKDA de clasificadores respectivamente. Los límites inferior y superior de muestras óptimas son 8 y 16 (es decir, de 12 a 24 Hz como límites en frecuencia máxima) aunque nunca se llegan a seleccionar 16 muestras porque en ese caso con 12 muestras se obtienen los mismos resultados. Esto indica que cada sujeto y clasificador tienen distintas bandas de frecuencia óptimas, lo que contribuye a la idea de que en los sistemas BCI cada caso tiene que ser estudiado por separado y no existen soluciones únicas que se adapten a todos los sujetos.

7.2. Discusión de los resultados de la Wavelet Madre óptima

Los resultados de la Wavelet Madre óptima son diferentes para las distintas combinaciones de sujetos y clasificadores. En el caso de LDA es 'coif1' y 'sym4' para el sujeto A y B respectivamente. Para el conjunto de 2 SRKDA no se encuentra ninguna Wavelet madre óptima ya que ninguna supera los resultados de la clasificación sin análisis wavelet, por lo que se procede a la siguiente etapa sin aplicar Wavelet. Finalmente, con el conjunto de 19 SVM las Wavelet Madre óptima es 'sym4' para el sujeto A. En el caso del sujeto B tanto 'sym2' como 'db2' obtienen idénticos resultados y ambas se consideran óptimas. La única Wavelet madre que se repite es 'sym4'. De nuevo se llega a la misma conclusión: es necesario optimizar cada sujeto y clasificador por separado si se pretende alcanzar un resultado óptimo en todos los casos.

7.3. Discusión de los resultados de la clasificación

Los resultados de los métodos propuestos se evalúan en el conjunto de datos II de la III Competición BCI, disponible de forma pública. Se consideran tres métodos diferentes de clasificación de características: LDA, un conjunto de 2 SRKDA y un conjunto de 19 SVM. En primer lugar se obtienen los parámetros óptimos, luego se establece el sub-muestreo

adecuado para cada sujeto y después se determina la Wavelet Madre óptima. Una vez que se han realizado estos pasos previos se aplica Transformada Wavelet a los datos de entrenamiento y test, se entrena al modelo mediante el conjunto de datos de entrenamiento, se aplica dicho modelo al conjunto de datos de test y finalmente se decide el carácter mediante la información proporcionada por dicha clasificación. La tabla 7.1 compara los tres métodos para un total de 5 y 15 iluminaciones, ya que en la III Competición BCI se pedían los resultados para dichos números.

Método	Sujetos					
	A		B		MEDIA	
	Número de iluminaciones					
	5	15	5	15	5	15
LDA	57	100	81	98	69	99
SRKDA	77	100	74	99	75	99.5
SVM	60	97	75	94	67.5	95.5

Tabla 7.1: Comparativa de resultados de 5 y 15 iluminaciones para los distintos métodos de clasificación de este estudio. Los valores máximos aparecen en negrita.

La tabla 7.2 muestra los resultados obtenidos en el estudio previo con 14 muestras por canal e iluminación y sin Transformada Wavelet.

Método	Sujetos					
	A		B		MEDIA	
	Número de iluminaciones					
	5	15	5	15	5	15
LDA	65	98	81	96	73	97
SRKDA	77	100	79	94	78	97
SVM	63	96	74	94	68.5	95

Tabla 7.2: Comparativa de resultados de 5 y 15 iluminaciones para los distintos métodos de clasificación del primer estudio. Los valores máximos aparecen en negrita.

Observando la primera tabla y comparando los resultados se aprecia que el algoritmo que da mejores resultados es el mismo para ambos sujetos. En el caso del sujeto A, LDA obtiene una clasificación perfecta (100 %), aunque la clasificación del mismo para 5 iluminaciones es pobre, llegando solo al 57 %. Sin embargo, con SRKDA se obtienen mejores resultados, ya que para 15 iluminaciones se llega al 100 % y para 5 al 77 %, que es un resultado muy superior. El criterio de la III Competición BCI es que el ganador

es el que obtenga mejores resultados para 15 iluminaciones y en caso de empate, para 5 iluminaciones, así que se selecciona SRKDA como clasificador óptimo. Cabe preguntarse si este criterio es el ideal, ya que en la práctica 15 iluminaciones por carácter se pueden considerar demasiadas para un uso viable. Quizás sería conveniente usar como criterio el que mejores resultados obtenga para un número bajo de iluminaciones, en cuyo caso no se llegaría a la misma selección de sub-muestreo en algunos casos.

En el caso del sujeto B, los mejores resultados con 15 iluminaciones se obtienen para SRKDA, con un porcentaje de aciertos del 99 %, lo que lleva a que se elija como óptimo. Sin embargo, al contrario que en el caso del sujeto A, LDA proporciona mejores resultados para 5 iluminaciones, con un 81 % de aciertos. El conjunto de 19 SVM es el que arroja peores resultados en ambos casos, con un 97 % y un 94 % para el sujeto A y B respectivamente. Estos resultados son inferiores pero en absoluto negativos, ya que se trata de tasas de aciertos muy elevadas. Los valores medios obtenidos para LDA y SRKDA (recordando que difieren en sub-muestreo y Wavelet Madre) son muy buenos (99.5 % para SRKDA y 15 iluminaciones y 99 % para LDA y 15 iluminaciones). Estos resultados demuestran que es de gran importancia la optimización individualizada de cada sujeto y clasificador.

Comparando con los resultados del estudio previo, en el caso de LDA se aprecia una clara mejoría para 15 iluminaciones en ambos sujetos, aunque el A pierde mucha precisión en la clasificación para 5 iluminaciones.

En el caso de SRKDA los resultados son iguales para el sujeto A tanto para 15 como 5 iluminaciones (100 % y 77 % respectivamente), pero en el caso del sujeto B mejoran para 15 iluminaciones, pasando del 94 % al 99 %, aunque esto viene acompañado de peores resultados para 5 iluminaciones (74 % frente a 79 %). Es importante recordar que en este estudio es un conjunto de 2 SRKDA y en el anterior era solo uno. Dividir el conjunto de entrenamiento en dos partes mejora notablemente el tiempo de cálculo del modelo, haciéndolo computacionalmente menos exigente. El conjunto de 19 SVM mejora para 15 iluminaciones en el sujeto A, pasando del 96 % de aciertos al 97 % para 15 iluminaciones, aunque bajando del 63 % al 60 % para 5 iluminaciones. En el sujeto B los resultados se mantienen para 15 iluminaciones (94 %) y suben del 74 % al 75 % para 5 iluminaciones.

Fijándonos estrictamente en los valores medios, para LDA y SRKDA hay una mejora del 97 % al 99 % y 99.5 % de aciertos para 15 iluminaciones respectivamente, aunque para 5 disminuye la precisión del 73 % y 78 % al 69 % y 75 % respectivamente. Sin embargo, estos resultados se pueden considerar mejores bajo este criterio ya que predominan siempre los de 15 iluminaciones. Para SVM también hay una mejora en 15 iluminaciones (pasando del 95 % al 97 %) y un descenso de la precisión para 5 iluminaciones (del 68.5 % al 67.5 %)

A diferencia del estudio anterior ya no hay tanta diferencia para 15 iluminaciones entre el sujeto A y el B. Previamente el sujeto B obtenía peores resultados para 15 iluminaciones tanto para LDA como para SRKDA, mientras que ahora son comparables (aunque algo inferiores en ambos casos). Es interesante destacar también que el método óptimo ha cambiado para el sujeto B, pasando de LDA a SRKDA, siempre que se sigan los criterios de la III Competición BCI. Sería ciertamente discutible afirmar que SRKDA es superior a LDA cuando la precisión para 5 iluminaciones es del 74 % en SRKDA y del 81 % en LDA.

En este estudio no es posible elegir un método de clasificación “superior” globalmente ya que se ha demostrado que los métodos de clasificación, el sub-muestreo y la extracción de características están muy relacionados entre sí. Cada sujeto tendrá una combinación óptima de los mismos, sin poderse obtener un método universal. Puede incluso que esta combinación no sea fija para un sujeto y varíe en el tiempo por distintos factores que afectan a señales muy dinámicas como son los P300.

7.3.1. Discusión de los resultados con eliminación de canales

Los resultados arrojados al aplicar dicho algoritmo de forma previa a la clasificación son interesantes y se analizan aquí de forma separada. En primer lugar llama la atención la alta variabilidad de canales eliminados en función de que sujetos y clasificadores. En el caso del sujeto A solamente se eliminan canales con el LDA y con SVM, concretamente los correspondientes a los electrodos 22, 24, 52 y 39 en el caso del LDA y los canales 3, 7, 16, 18, 30 y 60 para el conjunto de SVM. Por su parte en el caso del sujeto B se descartan los canales 3, 8, 25, 28, 29 y 35 con el conjunto de SVM y ninguno en el resto de clasificadores. Es interesante que el único canal que se repite es el 3, para el sujeto A y el B con SVM, lo que puede llevar a pensar que las zonas donde predomina la actividad cerebral asociada a los potenciales evocados P300 no están centradas exactamente en las mismas zonas del cuero cabelludo para todas las personas.

Las figuras 7.1, 7.2 y 7.3 muestran los canales descartados en cada caso con el fin de apreciar mejor en qué zonas se encuentran localizados. Se observa que para el sujeto A con LDA se corresponden con las zonas más externas del cuero cabelludo respecto al electrodo Cz (central). Sin embargo, con SVM los canales descartados están más dispersos e incluyen algunos centrales. El hecho de que los canales descartados para un mismo sujeto varíen con el clasificador también es notable. Es difícil extraer conclusiones de esto último.

En segundo lugar también son destacables los resultados de la clasificación: en el caso del sujeto A con LDA los resultados son iguales o superiores a aquellos obtenidos sin descartes, excepto para 14 o 15 iluminaciones, con lo cual se concluye que el algoritmo puede tener utilidad para un número bajo de iluminaciones. En el caso del conjunto de SVM tanto para el sujeto A como el B el resultado es muy distinto, siendo inferior o igual la clasificación para todo número de iluminaciones excepto para 9 en el A 8 y 13 en el B. Se observa que el algoritmo no ofrece resultados satisfactorios para el clasificador SVM. Estos resultados son completamente dispares, y aún más llamativo es comparar los resultados con los del método ganador de la competición [12]. Si observamos los canales que obtenían una mayor puntuación en su algoritmo de eliminación de canales (figura 7.4) podemos observar que en el sujeto A con LDA como clasificador, el canal 22, uno de los eliminados con nuestro algoritmo, tiene una puntuación media. Es decir, que nuestro algoritmo ha determinado como descartable un canal que los ganadores de la competición determinaron como óptimo. Con SVM se descarta el 60, que tiene también una alta puntuación para los ganadores.

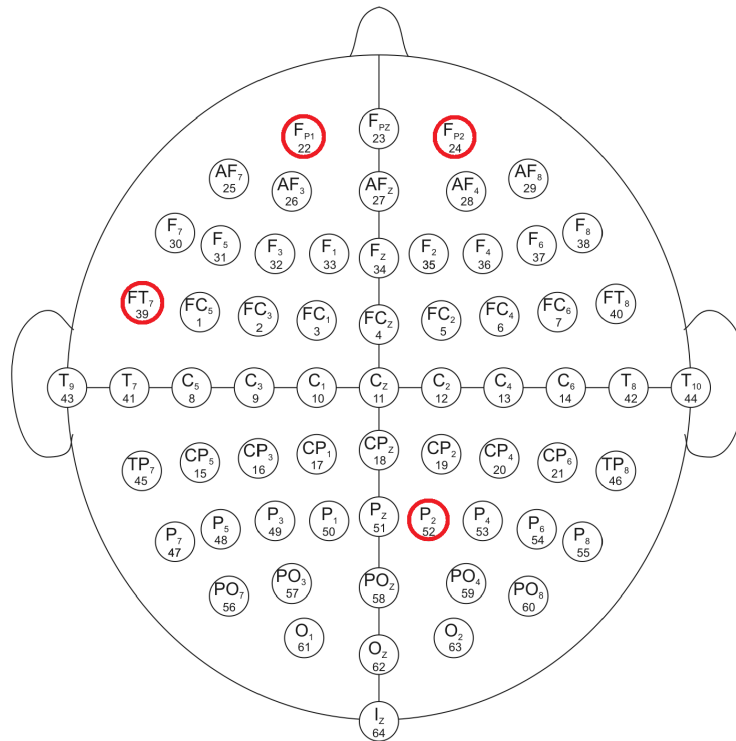


Figura 7.1: Canales descartados por el algoritmo de eliminación de canales para el sujeto A y el clasificador LDA.

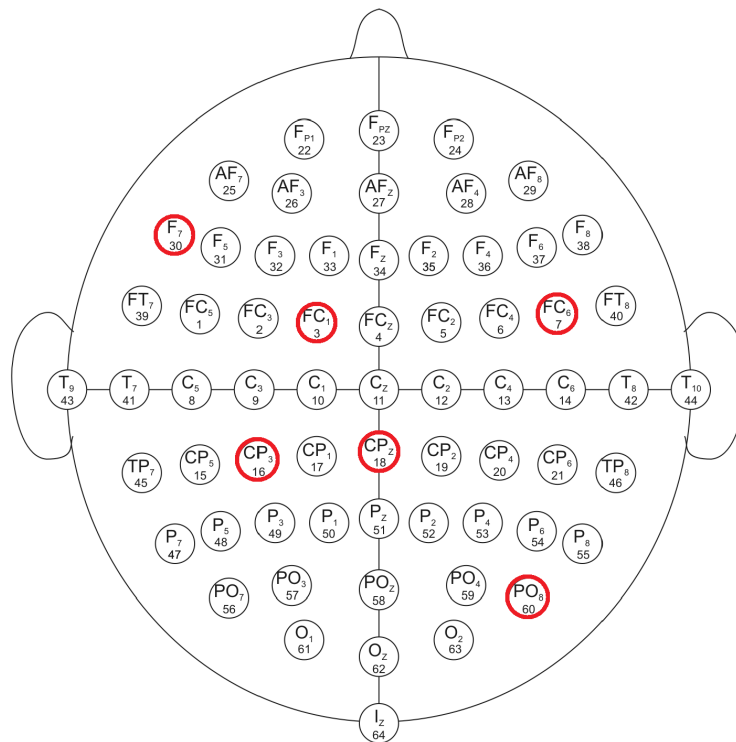


Figura 7.2: Canales descartados por el algoritmo de eliminación de canales para el sujeto A y el clasificador SVM.

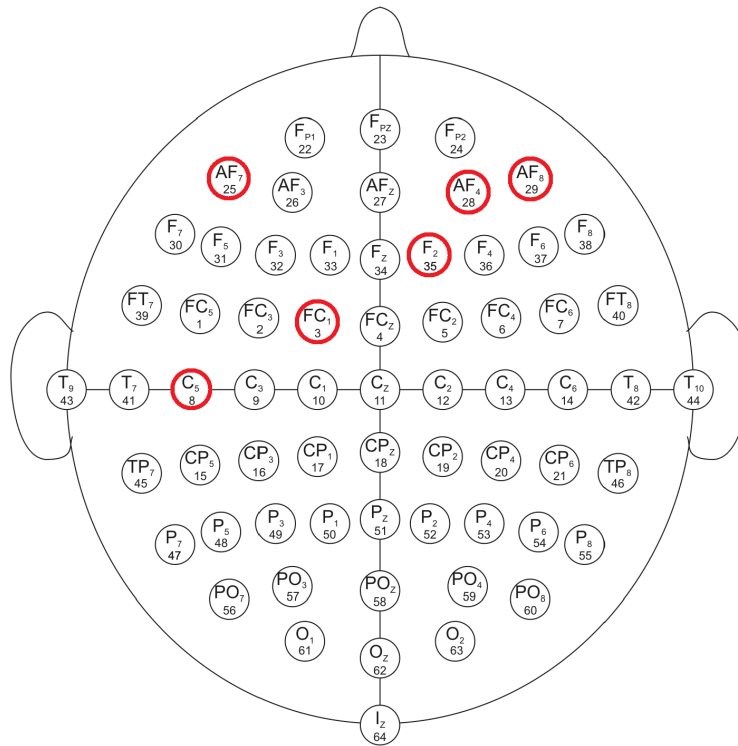


Figura 7.3: Canales descartados por el algoritmo de eliminación de canales para el sujeto B y el clasificador conjunto de SVM.

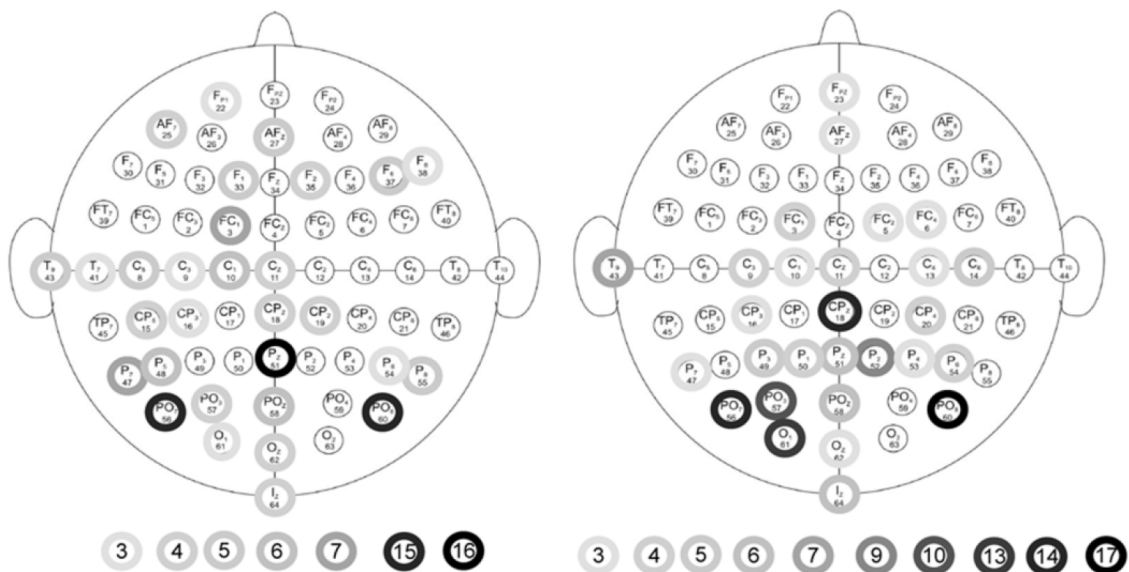


Figura 7.4: Histograma topológico de la clasificación de los canales en el método ganador de la competición, para el Sujeto A (izquierda) y el Sujeto B (derecha). Cada canal está rodeado por un círculo. La escala de grises indica el número de veces, dado en los círculos que aparecen en la parte inferior, que un canal ha sido incluido entre los 12 mejores. En total se realizaron 17 clasificaciones para cada sujeto (17 SVM entrenadas, es decir, el conjunto de datos de entrenamiento se divide en 17 partes).

En el caso del sujeto B con SVM como clasificador, no ocurre esto, y los canales descartados no están entre los óptimos según los ganadores. Estos resultados son interesantes, aunque es cierto que han sido obtenidos con algoritmos muy diferentes en cuanto a la implementación. También es cierto que en el caso del Sujeto A, el clasificador que eliminó el canal 22 fue LDA, no SVM. En el caso del sujeto B no se realizó el proceso con cada SVM como clasificador de forma individual y luego se obtuvo la media, como se hizo en el estudio ganador, sino que se hace la clasificación con las 19 SVM directamente para disminuir el tiempo de procesamiento.

Globalmente estos resultados podían haber sido más satisfactorios pero aún resta mucho margen de mejora para este algoritmo, el cual se basa en parte en la idea de los ganadores de la competición pero en general es de desarrollo propio. La eliminación de canales tiene una gran importancia no solo por conseguir el descarte de parte de los datos que no aportan información relevante, sino porque esto también supone una reducción de la carga computacional de la clasificación, lo que supondrá un tiempo reducido de la misma y un mayor rendimiento. Esta es una vía de investigación prometedora que aquí solo ha sido tratada de forma superficial.

7.4. Comparación con otros métodos

En la tabla 7.3 se comparan los resultados de los métodos propuestos con los obtenidos por el método ganador.

Los resultados globales para 15 iluminaciones son muy superiores al método ganador tanto para LDA como SRKDA. Por otra parte para 5 iluminaciones, SRKDA supera al ganador de la competición, aunque ni LDA ni SVM lo alcanzan. Siguiendo el criterio de la competición el ganador global sería el conjunto de 2 SRKDA, aunque ya se ha comentado que en este estudio no se pretende elegir un método global de clasificación.

En la tabla 7.4 se comparan los resultados con otros métodos del estado del arte que usan la misma base de datos ordenados de mayor a menor número de aciertos.

Método	Sujetos					
	A		B		MEDIA	
	Número de iluminaciones					
	5	15	5	15	5	15
LDA	57	100	81	98	69	99
SRKDA	77	100	74	99	75	99.5
SVM	60	97	75	94	67.5	95.5
Rakotomamonjy y Guigue (2008) [12]	72	97	75	96	73.5	96.5

Tabla 7.3: Comparativa de los resultados con el método ganador. En negrita el método superior en cada caso.

Los resultados finales son muy satisfactorios, puesto que el conjunto de 2 SRKDA y el LDA, previo ajuste de sub-muestreo y extracción de características mediante Transformada Wavelet se colocan en primer y segundo lugar de la clasificación global. El conjunto de 19 SVM se coloca en el puesto 12, no siendo este resultado tan positivo aunque es satisfactorio. Aunque como se comentará después, puede que la base de datos de la III Competición BCI ya haya llegado al límite de su utilidad y quizás se deberían empezar a introducir nuevos registros de otros sujetos, preferiblemente en mayor número.

Nº	Método	Aciertos (%)
1	Conjunto de 2 SRKDA	99.50
2	LDA	99.00
3	Amini et al (2012) [35]	97.40
4	SRKDA (Estudio previo)	97.00
5	LDA (Estudio previo)	97.00
6	El Dabbagh y Fakhr (2011) [44]	97.00
7	Shahriari y Erfanian (2011) [39]	96.90
8	Mirghasemi et al (2006) [32]	96.77
9	Rakotomamonjy y Guigue (2008) [12]	96.50
10	Sakamoto y Aono (2009) [31]	96.50
11	Tomioka y Haufe (2008) [45]	96.50
12	Conjunto de 19 SVM	95.50
13	Cecotti y Gräser (2011) [38]	95.50
14	Conjunto de 19 SVM (Estudio previo)	95.00
15	Salvaris et al (2009) [30]	95.00
16	Çelik y Anca (2010) [43]	95.50
17	Onishi et al (2012) [33]	92.00

Tabla 7.4: Comparación de resultados con otros métodos del estado del arte ordenados por porcentaje de aciertos para 15 iluminaciones.

7.5. Limitaciones de este estudio

A diferencia del estudio realizado previamente, en este sí se han aplicado métodos de extracción de características, lo que hace que los resultados sean superiores que cuando se eliminó esta etapa, como era de esperar. Como se observa en la tabla 7.2, en dos de los tres casos la extracción de características ha permitido mejorar los resultados de la clasificación. Sin embargo, algunas limitaciones anteriores siguen persistiendo en este caso.

La principal limitación heredada, como ya se adelantaba, es el propio formato de la III Competición BCI: dos sujetos es una muestra de población demasiado reducida como para que los resultados de un estudio se puedan generalizar. Por esta misma razón hacer la media de los resultados entre dos sujetos no es de demasiado interés, ya que cada sujeto necesitará un método de extracción y de clasificación adecuado, y una combinación que da resultados óptimos para una persona no tiene por qué ser adecuada para otra. Por todo esto sería deseable contar con un conjunto de datos extraídos de un número mayor de sujetos.

Las pruebas realizadas se han hecho sobre bases de datos registradas, no sobre usuarios en tiempo real, por lo que no se ha podido comprobar si son viables para una aplicación real debido a problemas potenciales como un excesivo tiempo de procesamiento previo a la clasificación.

Los códigos desarrollados en Matlab están preparados para clasificar las señales una vez han sido registradas y adaptadas al formato de la III Competición BCI. Si se quisiera poder usar este código para una aplicación BCI en tiempo real como BCI2000 sería necesaria una adaptación del mismo, incluyendo su reescritura en otro lenguaje de programación (C++) y su encaje en el esquema global del mismo.

En este trabajo se han tratado solamente tres algoritmos de clasificación de características, el estudio podría haberse ampliado con otros métodos, tales como la regresión logística.

Se ha desarrollado un algoritmo de selección de canales, cuya carga computacional es elevada y por tanto tiene un tiempo de ejecución alto. Al solo tener que ser ejecutado una vez, durante el entrenamiento, esto no supone un mayor problema. Este algoritmo ha obtenido resultados prometedores para el sujeto A y LDA como clasificador, pero aún no tiene un rendimiento satisfactorio. A pesar de ello tiene mucho margen de mejora, ya que aún se encuentra en una primera versión y todavía es bastante simple.

El conjunto de datos II de la III Competición BCI se obtuvo mediante métodos de adquisición descritos previamente. Las señales podrían haberse extraído con diferentes procedimientos con vistas a ver cuál de ellos proporciona mejores resultados para un mismo algoritmo. Es por tanto útil plantearse probar el algoritmo de clasificación desarrollado con bases de datos de extracción propia.

Capítulo 8

Conclusiones y líneas futuras

8.1. Conclusiones

A lo largo de este trabajo se han estudiado los sistemas BCI, con especial hincapié en aquellos basados en potenciales evocados P300. Estos sistemas permiten que usuarios en situación de discapacidad motora puedan interactuar con dispositivos sin necesidad de los músculos periféricos. Las señales han sido registradas mediante el procedimiento del EEG de tipo no invasivo. Dichas señales han sido potenciales evocados P300, las cuales tienen la ventaja de que no necesitan entrenamiento previo por parte del usuario puesto que son respuestas espontáneas.

El estudio realizado se ha centrado en la aplicación de un sub-muestreo, un método de extracción de características (Transformada wavelet), un algoritmo de eliminación de canales y tres métodos de clasificación de características diferentes a las señales del conjunto de datos II de la III Competición BCI de 2004, proporcionados por el centro Wadsworth (*NYS Department of Health*) [8], que consisten en una serie de potenciales evocados P300 registrados a dos usuarios mediante EEG. Dichas señales se dividen en dos subconjuntos: uno de entrenamiento y uno de test. En primer lugar se ha obtenido un sub-muestreo óptimo para cada sujeto y se ha seleccionado una Wavelet Madre óptima para cada uno. Tras esto se ha procedido a la clasificación de características (con LDA, dos SRKDA y un conjunto de SVM), previa aplicación de transformación wavelet con las Wavelet Madre correspondientes. Los resultados de la clasificación se han utilizado para elegir el carácter buscado en cada caso. Dichos resultados se han comparado con los obtenidos por el método ganador de la competición [12], consistente en un conjunto de 17 SVM con selección de canales, así como con otros métodos actuales [30–33, 35, 38, 39, 43–45].

Las principales conclusiones a las que se han llegado se pueden resumir en los puntos siguientes:

- La precisión media en la clasificación obtenida por LDA y SRKDA tras el sub-muestreo y la aplicación de la extracción de características mediante transformada wavelet (99 % para LDA y 99.5 % para el conjunto de 2 SRKDA) ha superado a la del método ganador de la competición. Además, SRKDA supera al ganador de la competición para 5 iluminaciones (75 % frente a 73.5 %), por lo que se puede decir que los resultados de SRKDA son una mejora absoluta respecto a los ganadores de la competición.

- En el caso del sujeto A y LDA como clasificador se ha seleccionado como ideales un sub-muestreo de 10 muestras por canal y en el sujeto B de 8 muestras por canal. La wavelet madre óptima para el sujeto A ha sido 'coif1' y para el B 'sym4'.
- Cuando se usa como clasificador el conjunto de 2 SRKDA, el sub-muestreo óptimo para el sujeto A es 14 muestras por canal y para el B 8 muestras por canal. En este caso no se obtiene una Wavelet Madre óptima para ninguno de los sujetos.
- En el caso del sujeto A y el conjunto de 19 SVM como clasificador se ha seleccionado como ideales un sub-muestreo de 12 muestras por canal y en el sujeto B de 14 muestras por canal. La wavelet madre óptima para el sujeto A ha sido 'sym4' y para el B 'db2' y 'sym2'.
- El sub-muestreo, Wavelet Madre y clasificadores utilizados están muy relacionados y deben ser elegidos en conjunto. Una Wavelet Madre o un sub-muestreo pueden ser más apropiados para un clasificador que para otro. Por tanto el proceso de selección de parámetros realizado se debe hacer para cada clasificador por separado.
- LDA cuenta con la ventaja de ser un método computacionalmente poco exigente y rápido en comparación con SRKDA.
- SRKDA requiere un tiempo considerable para la construcción del modelo. Esto no es problemático debido a que esta operación se puede realizar entre la sesión de entrenamiento y la de test. El tiempo de clasificación a partir de un modelo ya construido es aceptable aunque mayor que el de LDA.
- Se ha propuesto usar un conjunto de dos SRKDA para la clasificación, entrenados con la mitad de los datos de entrenamiento. Esto ha proporcionado un gran aumento de la velocidad de cálculo del modelo en la etapa de entrenamiento sin que suponga pérdidas de rendimiento en la clasificación, por lo que la propuesta ha tenido éxito.
- Comparando con otros métodos del estado del arte, SRKDA y LDA con extracción previa de características mediante transformación wavelet han quedado en primer y segundo puesto en cuanto a precisión para 15 iluminaciones, por lo que se pueden considerar sus resultados como satisfactorios y relevantes.
- El conjunto de 2 SRKDA ha sido el clasificador que mejores resultados ha aportado excepto en el sujeto B para 5 iluminaciones. Se considera que es el mejor clasificador globalmente.
- LDA ha resultado ser el método más rápido y de menor carga computacional, por lo que su aplicación a sistemas en tiempo real es la más viable, aunque usar dos SRKDA en conjunto, entrenados con la mitad de los datos de entrenamiento, ha reducido considerablemente su tiempo de cómputo sin sacrificar rendimiento.
- El algoritmo de eliminación y selección de canales propuesto ha obtenido resultados aceptables para el clasificador LDA en el caso del sujeto A, mejorando los resultados excepto para 14 y 15 iluminaciones. Con el sujeto A y SVM mejoran para 9 iluminaciones, se igualan para 11, 13 y 14, y empeoran para el resto. En

el caso del sujeto B y SVM como clasificador los resultados mejoran para 8 y 13 iluminaciones, se igualan para 8 y 12 y empeoran en el resto. En los casos restantes no se elimina ningún canal.

- Existe mucho margen de mejora para el algoritmo de eliminación y selección de canales, ya que se trata de una primera versión sencilla que usa criterios simples para decidir qué canales son mejores y más apropiados. Sin embargo, los resultados no son excesivamente satisfactorios excepto para LDA y el sujeto A.
- La base de datos II de la III Competición BCI ha llegado al límite de su vida útil. Se trata de una base de datos que solo cuenta con dos sujetos, lo que la limita mucho a la hora de servir para extraer conclusiones generalizadas. Además se ha llegado a una tasa media de aciertos del 99 %, con lo que el margen de mejora es mínimo.

8.2. Líneas futuras

Llegados al fin de este estudio es procedente señalar algunas limitaciones que han ido apareciendo y en base a las mismas proponer nuevas líneas de investigación.

- El formato de la base de datos II de la III Competición BCI es una de las principales limitaciones del estudio, ya que se compone solo de dos sujetos. Esto provoca que los resultados de cualquier estudio que use estos registros no puedan generalizarse, ya que dos sujetos son muy pocos para tal fin. En este estudio se realiza la media de resultados entre los dos sujetos, pero no es un resultado relevante excepto para ser comparado con otros trabajos que usen esta misma base de datos. Sería interesante, por tanto, aplicar todos los métodos aquí desarrollado a otras bases de datos menos limitadas para observar los resultados que ofrecen y extraer conclusiones generalizadas.
- Sería de gran interés aplicar todos estos algoritmos y métodos desarrollados con personas en tiempo real y analizar la precisión obtenida, el tiempo necesario para la clasificación y la viabilidad en general de los mismos para aplicaciones en tiempo real. En concreto la aplicación BCI2000 es una buena herramienta para este fin.
- Una línea de desarrollo futura podría ser la adaptación del código y algoritmos desarrollados al lenguaje de programación C++ para poder ser usado por la aplicación BCI2000. Una migración completa y adaptada del código podría ser muy interesante para alcanzar el objetivo del punto anterior.
- Se podrían ampliar etapas del estudio usando más algoritmos de extracción de características o de clasificación, como PCA o regresión logística.
- El algoritmo de selección y eliminación de canales es prometedor pero aún tiene mucho margen de desarrollo. Por tanto, es una línea de desarrollo futuro interesante y de gran utilidad para reducir las dimensiones de los registros. Se podrían estudiar diversos algoritmos de otros estudios e intentar llegar a uno propio que los supere en eficiencia y carga computacional.

Los resultados de este estudio se pueden considerar exitosos. Se ha conseguido superar al método ganador con dos clasificadores (LDA y conjunto de dos SRKDA) y además se ha llegado al primer y segundo puesto del estado del arte con los mismos. Se han reducido además las dimensiones de los registros, ya que el sub-muestreo utilizado en ambos casos es inferior al de los ganadores de la competición. Por otra parte la extracción de características mediante transformada wavelet no ha mejorado los resultados tanto como se esperaba, aunque finalmente sí que supera ligeramente a los obtenidos sin ella. Se ha llegado al final de la vida útil de la base de datos II de la III Competición BCI, por lo que en el futuro habrá que plantearse usar nuevos registros de personas. A pesar de esto, la perspectiva de usar estos métodos desarrollados en nuevas personas para aplicaciones incluso en tiempo real, así como el futuro desarrollo del algoritmo de selección y eliminación de canales son prometedores.

Referencias

- [1] “Biomedical Signal Processing”, About Biomedical Engineering, IEEE Engineering in Medicine & Biology Society, <http://www.embs.org/about-biomedical-engineering/our-areas-of-research/biomedical-signal-processing>
- [2] E. N. Bruce, “Biomedical signal processing and signal modelling”, John Wiley and Sons, 2001.
- [3] E. Niedermeyer, F.L. da Silva. “Electroencephalography: Basic Principles, Clinical Applications, and Related Fields”, Lippincot Williams & Wilkins, 2004.
- [4] G. H. Klem, H. O. Lüders, H. H. Jasper, C. Elger, “The ten twenty electrode system of the international federation”, *Electroencephalography and Clinical Neurophysiology*, vol. 10, pp. 214–215, 1958.
- [5] Medline Plus (Consultado Septiembre 2015). *EEG* [Online]. Disponible: <http://www.nlm.nih.gov/medlineplus/ency/article/003931.htm>
- [6] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, T. M. Vaughan, “Brain–computer interfaces for communication and control”, *Clinical Neurophysiology*, vol. 113, pp. 767–791, 2002.
- [7] A. Kübler, K. Müller, “An Introduction to Brain-Computer Interfacing”, *Toward Brain-Computer Interfacing*, MIT Press, Cambridge, Massachusetts, pp. 1-25, 2007.
- [8] D. Krusienski, G. Schalk, “BCI Competition III Challenge 2004”, 2004: http://www.bbci.de/competition/iii/desc_II.pdf
- [9] E.J. Speckmann, C.E. Elger y A. Gorji, “Neurophysiologic Basis of EEG and DC Potentials”, *Niedermeyer’s Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*, 6th ed. D.E. Schomer y F.H.L. da Silva, Eds. Philadelphia, Pennsylvania: Lippincott Williams & Wilkins, 2011.
- [10] K. E. Misulis, T. Fakhoury, ”Spehlmann’s Evoked Potential Primer”, Butterworth-Heinemann., 2001.
- [11] D. J. Creel, “Visually Evoked Potentials”, Webvision, 2012.
- [12] A. Rakotomamonjy, V. Guigue, “BCI Competition III: Dataset II – Ensemble of SVMs for BCI P300 Speller”, *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 3, pp. 1147–1154, 2008.

- [13] X. Gao, D. Xu, M. Cheng, S. Gao, "A BCI-based environmental controller for the motion-disabled", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, pp. 137–140, 2003.
- [14] K.D. Nielsen, A.F. Cabrera, O.F. do Nascimento, "EEG based BCI-towards a better control. Brain-computer interface research at Aalborg University", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, pp. 202–204, 2006.
- [15] K.C. Squires, K.C.E. Donchin, R.I. Herning, G. McCarthy, "On the influence of task relevance and stimulus probability on event-related-potential components", *Electroencephalography and Clinical Neurophysiology*, vol. 42, pp.1–14, 1977.
- [16] L.A. Farwell, E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials", *Electroencephalography and Clinical Neurophysiology*, vol. 70, pp. 510–523, 1988.
- [17] R. Corralejo, "Extracción de Características de Electroencefalogramas. Aplicación a Brain Computer Interface", Proyecto Fin de Carrera, Departamento de Teoría de la Señal y Comunicaciones, Valladolid, Universidad de Valladolid, 2008.
- [18] K. R. Müller, C. W. Anderson, G. E. Birch, "Linear and Nonlinear Methods for Brain Computer Interfaces," *IEEE Transaction on Rehabilitation Engineering*, vol. 11, no. 2, pp. 165–169, 2003.
- [19] L. Mayaud, M. Congedo, A. Van Laghenhove, D. Orlikowski, M. Figère, E. Azabou, F. Cheliout-Heraut, "A comparison of recording modalities of P300 event-related potentials (ERP) for brain-computer interface (BCI) paradigm", *Neurophysiologie Clinique/Clinical Neurophysiology*, vol. 43, no. 4, pp 217–227, 2013.
- [20] F. Akram, M.K. Metwally, Hee-Sok Han, Hyun-Jae Jeon, Tae-Seong Kim, "A Novel P300-Based BCI System for Words Typing", *International Winter Workshop on Brain-Computer Interface (BCI)*, pp. 24–25, 2013.
- [21] S.T. Ahi, H. Kambara, Y. Koike, "A Dictionary-Driven P300 Speller with a Modified Interface", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 1, pp. 6–14, 2011.
- [22] D. J. McFarland, W.A. Sarnacki, G. Townsend, T. Vaughan, J. R. Wolpaw, "The P300-based brain-computer interface (BCI): Effects of stimulus rate", *Clinical Neurophysiology*, vol. 122, no. 4, pp. 731–737, 2011.
- [23] J. Guo, S. Gao, B. Hong "An Auditory Brain-Computer Interface Using Active Mental Response", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 3, pp. 230–235, 2010.
- [24] D. Aminaka, K. Mori, T. Matsui, S. Makino, T.M. Rutkowski, "Bone-conduction-based brain Computer Interface Paradigm-EEG Signal Processing, Feature Extraction and Classification", *International Conference on Signal-Image Technology & Internet-Based Systems*, pp. 818–824, 2013.

- [25] A.S. Elsayy, S. Eldawlatly, M.Taher, M. Aly, “A Principal Component Analysis Ensemble Classifier for P300 Speller Applications”, *8th International Symposium on Image and Signal Processing and Analysis*, pp. 444–449, 2013.
- [26] Y. Wang, J. Shen, J. Liang, Y. Ji, “Research of P300 Feature Extraction Algorithm based on ICA and Wavelet Transform”, *Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp. 41–45, 2014.
- [27] L. He, Z. Gu, Y. Li, Z. Yu, “Feature extraction with multiscale autoregression of multichannel time series for P300 speller BCI”, *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 610–613, 2010.
- [28] X. Li, F. Wang, X. Chen, R.K. Ward, “A P300-based BCI classification algorithm using median filtering and Bayesian feature extraction”, *Multimedia Signal Processing (MMSP)*, IEEE 14th International Workshop, pp. 305–308, 2012.
- [29] Y. Shahriari, A. Erfanian, “Improving the performance of P300-based brain–computer interface through subspace-based filtering”, *Neurocomputing*, vol. 121, pp. 434–441, 2013.
- [30] M. Salvaris, F. Sepúlveda, “Wavelets and Ensemble of FLDs for P300 Classification”, *Proceedings of the 4th International IEEE EMBS Conference on Neural Engineering*, pp. 339–342, 2009.
- [31] Y. Sakamoto, M. Aono, “Supervised Adaptive Downsampling for P300-based Brain Computer Interface”, *31st Annual International Conference of the IEEE EMBS*, pp. 567–570, 2008.
- [32] H. Mirghasemi, M.B. Shamsollahi, R. Fazel-Rezai, “Assessment of Preprocessing on Classifiers Used in the P300 Speller Paradigm”, *Proceedings of the 28th IEEE EMBS Annual International Conference*, pp. 1319–1322, 2006.
- [33] A. Onishi, A. H. Phan, K. Matsuoka, A. Cichocki, “Tensor Classification for P300-based Brain Computer Interface”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 581–584, 2012.
- [34] B. Dal Seno, M. Matteucci, “A Genetic Algorithm for Automatic Feature Extraction in P300 Detection”, *IEEE International Joint Conference on Neural Networks*, pp. 3145 – 3152, 2008.
- [35] Z. Amini, V. Abootalebu, M. T. Sadeghi, “Comparison of Performance of Different Feature Extraction Methods in Detection of P300”, *Biocybernetics and Biomedical Engineering*, vol. 33, no. 1, pp. 3–20, 2013.
- [36] A. Turnip, A. Widyotriatmo and Suprijanto, “Classification of EEG-P300 signals using Fisher’s linear discriminant analysis”, *3rd International Conference on Instrumentation Control and Automation (ICA)*, pp. 98–103, 2013.

- [37] Y. Liu, Z. Zhou, D. Hu, G. Dong, “T-weighted Approach for Neural Information Processing in P300 based Brain-Computer Interface”, *International Conference on Neural Networks and Brain (ICNN&B)*, pp. 1535–1539, 2005.
- [38] H. Cecotti, A. Graser, “Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 433–445, 2011.
- [39] Y. Shahriari, A. Erfanian, “A Mutual Information Based Channel Selection Scheme for P300-based Brain Computer Interface”, *5th International IEEE EMBS Conference on Neural Engineering*, pp. 434–437, 2011.
- [40] W. Speier, C. Arnold, J. Lu, A. Deshpande, N. Pouratian, “Integrating Language Information With a Hidden Markov Model to Improve Communication Rate in the P300 Speller”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, por publicar, 2014.
- [41] A. Combaz, N. Chumerin, N.V. Manyakov, A. Robben, J.A.K. Suykens, M.M. Van Hulle, “Towards the detection of error-related potentials and its integration in the context of a P300 speller brain–computer interface”, *Neurocomputing*, vol. 80, pp. 73–82, 2012.
- [42] A. González, I. Nambu, H. Hokari, M. Iwahashi, Y. Wada, “Towards the Classification of Single-trial Event-Related Potentials using Adapted Wavelets and Particle Swarm Optimization”, *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3089–3094, 2013.
- [43] Umut Çelik, Sami Anca, “Classification of the Visual Evoked EEG Using Multi-resolution Approximation Based on Excitatory Post-Synaptic Potential Waveform”, *IEEE 26th Convention of Electrical and Electronics Engineers in Israel (IEEEI)*, pp. 929–933, 2010.
- [44] H. El Dabbagh, W. Fakhr, “Multiple classification algorithms for the BCI P300 speller diagram using ensemble of SVMs”, *IEEE GCC Conference and Exhibition (GCC)*, pp. 393–396, 2011.
- [45] R. Tomioka, S. Haufe, “Combined classification and channel/basis selection with L1-L2 regularization with application to P300 speller system”, *4th International Brain-Computer Interface Workshop and Training Course*, 2008.
- [46] BCI Competition III, Final Results, <http://www.bbci.de/competition/iii/results/>
- [47] B. Rebsamen, C. Guan, H. Zhang, C. Wang, C. Teo, M. H. Ang, Jr., E. Burdet, “A Brain Controlled Wheelchair to Navigate in Familiar Environments”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 6, pp. 590–598, 2010.

- [48] H. Riechmann, N. Hachmeister, H. Ritter, A. Finke, "Asynchronous, parallel on-line classification of P300 and ERD for an efficient hybrid BCI", *5th International IEEE EMBS Conference on Neural Engineering*, pp. 412–415, 2011.
- [49] Y. Li, J. Long, T. Yu, Z. Yu, C. Wang, H. Zhang, C. Guan, "An EEG-Based BCI System for 2-D Cursor Control by Combining Mu/Beta Rhythm and P300 Potential", *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 10, pp. 2495–2505, 2010.
- [50] S. Yun, M. Lee, S. Cho, "P300 BCI based Planning Behavior Selection Network for Humanoid Robot Control", *Ninth International Conference on Natural Computation*, pp. 354-358, 2013.
- [51] J. P. Rosenfeld, X. Hu, E. Labkovsky, J. Meixner, M. R. Winograd, "Review of recent studies and issues regarding the P300-based complex trial protocol for detection of concealed information", *International Journal of Psychophysiology*, vol. 90, no. 2, pp. 118–134, 2013.
- [52] P. Rosenfeld, "P300 in detecting concealed information", en *Memory Detection: Theory and application of the Concealed Information Test*, B. Verschuere, G. Ben-Shakar, E. Meijer, Eds., Cambridge University Press, 2011, pp. 63–89.
- [53] V.J. Samar, A. Bopardikar, R. Rao, K. Swartz, "Wavelet Analysis of Neuroelectric Waveforms: A Conceptual Tutorial", *Brain and Language*, Vol. 66, pp. 7–60, 1999.
- [54] O. Rioul, M. Vetterli, "Wavelets and Signal Processing", *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, 1991.
- [55] C.M. Bishop, "Pattern Recognition and Machine Learning", Springer, New York, 2006.
- [56] L.F Nicolás-Alonso, R. Corralejo, J. Gómez-Pilar, D. Álvarez, R. Hornero, "Adaptive semi-supervised classification to reduce intersession non-stationarity in multi-class motor imagery-based brain computer interfaces", *Journal of Neural Engineering*, 2014. In press.
- [57] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K. Mullers, "Fisher discriminant analysis with kernels", *Proceedings of IEEE Signal Processing Society Workshop*, (Madison, WI, Aug. 23–25) pp 41–48, 1999.
- [58] D. Cai, X. He, J. Han, "Speed up kernel discriminant analysis", *The VLDB Journal* no. 20 pp 21–33, 2011.
- [59] S. Haykin, "Neural Networks. A Comprehensive Foundation", Prentice Hall International, 1999.
- [60] M. García, "Procesado de retinografías basado en redes neuronales para la detección automática de lesiones asociadas a la retinopatía diabética", M. S. Thesis, Departamento de Teoría de la Señal y las Comunicaciones, Valladolid, Universidad de Valladolid, 2008.

- [61] L. Bottou, C.-J. Lin, “Support vector machine solvers”, in *Large-Scale Kernel Machines*, L. Bottou, O. Chapelle, D. DeCoste and J. Weston (Eds.), The MIT Press, pp. 1–27, 2007.

Anexo A

Listado del código desarrollado en Matlab

Fichero canalOptimo2.m

```
function [canales, canalesEliminados] = canalOptimo2(features, class, metodoClas, ...
    nMuestras, params)
%Esta funci\{o}n selecciona los canales \{o}ptimos mediante validaci\{o}n
%cruzada en 10 particiones sobre las se\{n}ales de entrenamiento. Se basa
%en eliminaci\{o}n recursiva de canales cuya eliminaci\{o}n mejore el ratio
%truePositives/(truePositives+falsePositives+falseNegatives). No se
%tienen en cuenta trueNegatives al estar las clases descompensadas a
%favor de los negativos.

[caracteres,iluminaciones,muestras]=size(features);
canales=1:64;
token=1;

%Vamos haciendo validaci\{o}n cruzada eliminando canales de uno en uno

%% Primero calculamos el ratio con todos los canales
ratioTotal=ratioCruzado(canales, features, class, nMuestras, metodoClas, ...
caracteres, iluminaciones, params);
fprintf('Calculado el ratio con todos los canales\n');
%% Ahora vamos eliminando canales y viendo si mejora el ratio

flag=1;
canalesTotal=length(canales);

while(flag==1)
    canalEliminar=0;
    for c=1:canalesTotal
        canalesAux=canales(find(canales~=canales(c))); %Seleccionamos todos los
        % canales excepto el que corresponda
        ratio=ratioCruzado(canalesAux, features, class, nMuestras, ...
            metodoClas, caracteres, iluminaciones, params);

        if(ratio>ratioTotal)
            canalEliminar=c;
            canalesEliminados(token)=canalEliminar;
            token=token+1;
            ratioTotal=ratio;
        end
    end
    if(canalEliminar==0)
        flag=0; %Si no se ha eliminado ning\{u}n canal acabamos
    end
    if(canalEliminar~=0)
        canales=canales(find(canales~=canalEliminar)); %Si se ha eliminado
        % algun canal lo quitamos
    end
end
```

```

        canalesTotal=length(canales);
        fprintf('Eliminado un canal\n');
        fprintf('%d\n',canalEliminar);
    end

end

if (canalesTotal==64)
    canalesEliminados=NaN;
    fprintf('\nNo se ha eliminado ning\{u\}n canal\n');
else
    fprintf('\nCanales eliminados: ');
    for a=1:length(canalesEliminados)
        fprintf('%d ',canales(a));
    end
end
end
end

```

Fichero cargarDatos.m

```

function [ Data ] = cargarDatos( nombreFichero )
%Extractor de Datos en estructura

    Data=load(nombreFichero);

end

```

Fichero clasificacion.m

```

function [p,clase] = clasificacion( trainFeatures, testFeatures, trainClass, metodo)
%Funci\{o\}n que llama a los distintos clasificadores

%Clasificador SRKDA
if (strcmp(metodo,'SRKDA'))
    %Opciones del clasificador
    options = [];
    options.KernelType = 'Gaussian';
    options.ReguType = 'Ridge';
    options.ReguAlpha = 0.05;
    options.t = 900;

    [clase,p]=clasificacionSRKDA(trainFeatures,testFeatures,trainClass,options);
end

if (strcmp(metodo,'LDA'))
    %Opciones del clasificador
    [clase,err,p]=classify(testFeatures,trainFeatures,trainClass,'linear');
    p=p(:,2);
end

if (strcmp(metodo,'SWLDA'))
    %Opciones del clasificador
    options=[];
    options.Regu = 1;
    %options.ReguAlpha = params;
    [clase,p]=clasificacionSWLDA(trainFeatures, trainClass, testFeatures,options);
end

if (strcmp(metodo,'LDA2'))
    %Opciones del clasificador
    options=[];
    options.Regu = 1;
    options.ReguAlpha = 13001;
    [clase,p]=clasificacionLDA(trainFeatures, trainClass, testFeatures,options);
    p=p(:,1);
end

if (strcmp(metodo,'LDA2conjunto'))

```



```

%Opciones del clasificador
options=[];
options.Regu = 1;
options.ReguAlpha = 13001;
%N\{u}mero de divisiones k
k=2;
longitud=ceil(length(trainClass)/k);
[longitudTest,~]=size(testFeatures);
claseaux=NaN(longitudTest,k);
p=NaN(longitudTest,1);
clase=p;
N = size(trainFeatures,1);
for ii=1:k
    featuresTrain=(trainFeatures(longitud*(ii-1)+1:min(longitud*ii,N),:));
    claseTrain=(trainClass(longitud*(ii-1)+1:min(longitud*ii,N)));
    [claseaux(:,ii),p]=clasificacionLDA(featuresTrain, claseTrain, testFeatures,...
    options);
    paux(:,ii)=p(:,1);
end

p=sum(paux,2)/k;
%{
for ii=1:longitudTest
    cont=0;
    for a=1:k
        if(claseaux(ii,a)==1)
            cont=cont+1;
        end
    end
    p(ii)=cont/k;
    if(mean(claseaux(ii,:))>=0)
        clase(ii)=1;
    else
        clase(ii)=-1;
    end
end
%}
end

if (strcmp(metodo,'SRKDAconjunto'))

%Opciones del clasificador
options = [];
options.KernelType = 'Gaussian';
options.ReguType = 'Ridge';
options.ReguAlpha = 0.05;
options.t = 900;

%N\{u}mero de divisiones k
k=2;
longitud=ceil(length(trainClass)/k);
[longitudTest,~]=size(testFeatures);
claseaux=NaN(longitudTest,k);
p=NaN(longitudTest,1);
clase=p;
N = size(trainFeatures,1);

for ii=1:k
    featuresTrain=(trainFeatures(longitud*(ii-1)+1:min(longitud*ii,N),:));
    claseTrain=(trainClass(longitud*(ii-1)+1:min(longitud*ii,N)));
    [claseaux(:,ii),p]=clasificacionSRKDA(featuresTrain, testFeatures, claseTrain,...
    options);
    paux(:,ii)=p(:,1);
end

p=sum(paux,2)/k;

end

```

```

if (strcmp (metodo,'SVM'))
    %N\{u}mero de divisiones k
    addpath('SVM\libsvm-3.18\matlab')
    k=19;
    longitud=ceil(length(trainClass)/k);
    [longitudTest,nada]=size(testFeatures);
    claseaux=NaN(longitudTest,k);
    p=NaN(longitudTest,1);
    clase=p;
    N = size(trainFeatures,1);
    parfor ii=1:k
        featuresTrain=(trainFeatures(longitud*(ii-1)+1:min(longitud*ii,N),:));
        claseTrain=(trainClass(longitud*(ii-1)+1:min(longitud*ii,N)));
        claseaux(:,ii)=clasificacionSVM(featuresTrain, claseTrain, testFeatures);
    end
    for ii=1:longitudTest
        cont=0;
        for a=1:k
            if(claseaux(ii,a)==1)
                cont=cont+1;
            end
        end
        p(ii)=cont/k;
        if(mean(claseaux(ii,:))>=0)
            clase(ii)=1;
        else
            clase(ii)=-1;
        end
    end
end

end

```

Fichero clasificacion_master.m

```

function [p,clase] = clasificacion( trainFeatures, testFeatures, trainClass, metodo)
%Funci\{o}n que llama a los distintos clasificadores

%Clasificador SRKDA
if (strcmp(metodo,'SRKDA'))
    %Opciones del clasificador
    options = [];
    options.KernelType = 'Gaussian';
    options.ReguType = 'Ridge';
    options.ReguAlpha = 0.05;
    options.t = 900;

    [clase,p]=clasificacionSRKDA(trainFeatures,testFeatures,trainClass,options);
end

if (strcmp(metodo,'LDA'))
    %Opciones del clasificador
    [clase,err,p]=classify(testFeatures,trainFeatures,trainClass,'linear');
    p=p(:,2);
end

if (strcmp(metodo,'LDA2'))
    %Opciones del clasificador
    options=[];
    options.Regu = 1;
    options.ReguAlpha = 13001;
    [clase,p]=clasificacionLDA(trainFeatures, trainClass, testFeatures,options);
    p=p(:,1);
end

if (strcmp (metodo,'SVM'))
    %N\{u}mero de divisiones k
    addpath('SVM\libsvm-3.18\matlab')
    k=19;

```

```

longitud=ceil(length(trainClass)/k);
[longitudTest,nada]=size(testFeatures);
claseaux=NaN(longitudTest,k);
p=NaN(longitudTest,1);
clase=p;
N = size(trainFeatures,1);
parfor ii=1:k
    featuresTrain=(trainFeatures(longitud*(ii-1)+1:min(longitud*ii,N),:));
    claseTrain=(trainClass(longitud*(ii-1)+1:min(longitud*ii,N)));
    claseaux(:,ii)=clasificacionSVM(featuresTrain, claseTrain, testFeatures);
end
for ii=1:longitudTest
    cont=0;
    for a=1:k
        if(claseaux(ii,a)==1)
            cont=cont+1;
        end
    end
    p(ii)=cont/k;
    if(mean(claseaux(ii,:))>=0)
        clase(ii)=1;
    else
        clase(ii)=-1;
    end
end
end

end

if (strcmp(metodo,'LDA2_pca'))
    %Opciones del clasificador
    options=[];
    options.Regu = 1;
    options.ReguAlpha = 13001;

    k=14;
    [longitudTest,nada]=size(testFeatures);
    [longitudTrain,nada]=size(trainFeatures);
    claseaux=NaN(longitudTest,k);
    featuresTrain=NaN(longitudTrain,64);
    featuresTest=NaN(longitudTest,64);
    for ii=1:14 %para cada clasificador
        for iii=1:64 %para cada canal
            featuresTrain(:,iii)=trainFeatures(:,ii+(iii-1)*k);
            featuresTest(:,iii)=testFeatures(:,ii+(iii-1)*k);
        end
        [claseaux(:,ii),p]=clasificacionLDA(featuresTrain, trainClass, featuresTest,...
        options);
    end
    for ii=1:longitudTest
        cont=0;
        for a=1:k
            if(claseaux(ii,a)==1)
                cont=cont+1;
            end
        end
        p(ii)=cont/k;
        if(mean(claseaux(ii,:))>=0)
            clase(ii)=1;
        else
            clase(ii)=-1;
        end
    end
end
end
end

```

Fichero clasificacionCV.m

```

clc
close all

```

```

% Par\{a}metros
dir='..\..\02mat';
metodoExt='ganador';
metodoClas='SVM';
sujetos={'Sujeto A', 'Sujeto B'};
tipoCurva='ROC';

Nsubjectos=size(sujetos,2);

for it=1:Nsubjectos
    fprintf('%s\n',sujetos{it})
    suj=sprintf('%s SVM',sujetos{it});

    train=sprintf('%s\\%s\\train\\%s.mat',dir,metodoExt,sujetos{it});
    test=sprintf('%s\\%s\\test\\%s.mat',dir,metodoExt,sujetos{it});
    trainData=cargarDatos(train);
    testData=cargarDatos(test);

    %% Obtenci\{o}n par\{a}metros ideales
    trainData.features=reshape(permute(trainData.features,[2 1 3]),15300,896);
    trainData.class=reshape(permute(trainData.class,[2 1]),15300,1);

    %features = trainData.features(1:2000,:);
    %class = trainData.class(1:2000,:);
    features = trainData.features;
    class = trainData.class;

    %a = 0.05:0.01:0.1; %SRKDA
    %t = 800:100:2200; %SRKDA
    %regu=1:1000:50000; %LDA2
    %matriz = crossvalmatrix( a,t ); %SRKDA
    %N = length(matriz); %SRKDA
    %N=length(regu); %LDA
    %nucleo=0;
    %cparam=[0.01 0.05 0.1 0.5 1]; %kernel lineal
    cparam=[1 5 10 15 20 25 30];
    %nucleo=0;
    %gammainv=10000:1000000:100000000;
    gammainv=1000000;

    matriz=crossvalmatrix( gammainv,cparam ); %SVM
    N=length(matriz); %SVM
    cvMCRplot=NaN(1,N);
    for i = 1:N
        %% Param
        %a=matriz(i,1); %SRKDA
        %t=matriz(i,2); %SRKDA
        gamma=1/matriz(i,1); %SVM
        cparam=matriz(i,2); %SVM
        %% CV
        s = RandStream('mt19937ar','Seed',2014);
        RandStream.setGlobalStream(s);
        cp = cvpartition(class,'k',10);
        string=sprintf('-g %f -c %f -w1 11 -w-1 1 -q', gamma, cparam);
        %perf = @(xtrain, ytrain,xtest, ytest) curvas(ytest, clasificacionParams...
        (xtrain,xtest,ytrain,metodoClas,a(1,i)),tipoCurva);
        %perf = @(xtrain, ytrain,xtest, ytest) curvas(ytest, clasificacionParams...
        (xtrain,xtest,ytrain,metodoClas,[a t]),tipoCurva);
        %perf = @(xtrain, ytrain,xtest, ytest) curvas(ytest, clasificacionParams...
        (xtrain,xtest,ytrain,metodoClas,regu(i)),tipoCurva);
        perf = @(xtrain, ytrain,xtest, ytest) curvas(ytest, clasificacionParams...
        (xtrain,xtest,ytrain,metodoClas,string),tipoCurva);
        cvMCR = crossval(perf,features,class,'partition',cp);
        cvMCR=mean(cvMCR);
        fprintf('Par1: %.02f Par2: %.02f\n', a, t); %SRKDA
        fprintf('Par\{a}metro regularizaci\{o}n: %.02f\n', regu(i)); %LDA2
        fprintf('Gamma: %d c: %.02f\n', gamma, cparam); % SVM
        fprintf('Performance: %.02f\n\n', cvMCR);
    end
end

```

```

        cvMCRplot(i)=cvMCR;
    end
    figure
    %plot(regu,cvMCRplot);
    plot(cvMCRplot);
    [maximo,posicion]=max(cvMCRplot)
    %aopt=matriz(posicion,1) %SRKDA
    %topt=matriz(posicion,2) %SRKDA
    %reguOpt=regu(posicion); %LDA2
    gopt=1/matriz(posicion,1);
    %gopt=1/(matriz(posicion,2));
    copt=matriz(posicion,2);

    %% Clasificaci\{o}n
    testData.features=reshape(permute(testData.features,[2 1 3]),18000,896);
    testData.class=reshape(permute(testData.class,[2 1]),18000,1);

    % [p,clase] = clasificacionParams( trainData.features, testData.features,...
    trainData.class,metodoClas,[aopt topt]);
    % [p,clase] = clasificacionParams( trainData.features, testData.features,...
    trainData.class,metodoClas,reguOpt);
    string=sprintf('-g %f -c %f -w1 11 -w-1 1 -q', gopt, copt);
    [p,clase] = clasificacionParams( trainData.features, testData.features,...
    trainData.class,metodoClas,string);
    %% Letras
    testData.code=reshape(permute(testData.code,[2 1]),18000,1);
    [letras, caract]=determinarLetra(p,testData.code,testData.matrix);

    %% Evaluaci\{o}n
    aciertos=posicionLetras(letras,testData.target,caract);
    %save(suj,'cvMCRplot','reguOpt','aciertos','regu') %LDA2
    %save(suj,'cvMCRplot','aopt','topt','aciertos','matriz') %SRKDA
    save(suj,'cvMCRplot','gopt','copt','aciertos','matriz') %SVM
    %Curva ROC

    % [X,Y,T,AUC] = perfcurve(testData.class,p(:,2),1);
    %figure(it)
    %plot(X,Y)
    %title('Curva ROC')
    %drawnow

    % [AUC,F] = curvas(testData.class,p,tiposCurva);
end

```

Fichero clasificacionLDA.m

```

function [clase,p]=clasificacionLDA(trainFeatures, trainClass, testFeatures,options)
    addpath('LDA')

    s = RandStream('mt19937ar','Seed',0);
    RandStream.setGlobalStream(s);
    N=size(trainClass,1);
    w=NaN(N,1);
    num1=length(find(trainClass==1));
    num2=length(find(trainClass==-1));
    p1=N/num1;
    p2=N/num2;
    w(trainClass==-1,1)=(p2)/(p1+p2);
    w(trainClass==1,1)=(p1)/(p1+p2);
    y = randsample(1:N,100000,true,w);

    % trainFeatures=trainFeatures(y,:);
    % trainClass=trainClass(y,:);
    model = LDAtrain(trainFeatures, trainClass, options);
    [accuracy, clase, p] = LDApredict(trainFeatures, trainClass, model);
    if mean(p(trainClass == -1,1)) > mean(p(trainClass ~= -1,1))
        model.eigvector = -model.eigvector;
    end
    N = size(testFeatures,1);

```

```
[accuracy, clase, p] = LDAPredict(testFeatures, ones(N,1), model);
end
```

Fichero clasificacionParams.m

```
function [p,clase] = clasificacionParams( trainFeatures, testFeatures, trainClass,...
metodo, params)
%Funci\ '{o}n que llama a los distintos clasificadores

%Clasificador SRKDA
if (strcmp(metodo,'SRKDA'))
    %Opciones del clasificador
    options = [];
    options.KernelType = 'Gaussian';
    options.ReguType = 'Ridge';
    options.ReguAlpha = params(1);
    options.t = params(2);

    [clase,p]=clasificacionSRKDA(trainFeatures,testFeatures,trainClass,options);
end

if (strcmp(metodo,'LDA'))
    %Opciones del clasificador

    [clase,err,p]=classify(testFeatures,trainFeatures,trainClass,'linear');
end

if (strcmp(metodo,'LDA2'))
    %Opciones del clasificador
    options=[];
    options.Regu = 1;
    options.ReguAlpha = params;
    [clase,p]=clasificacionLDA(trainFeatures, trainClass, testFeatures,options);
end

if (strcmp (metodo,'SVM'))
    %N\ '{u}mero de divisiones k
    addpath('SVM\libsvm-3.18\matlab')
    k=17;
    longitud=ceil(length(trainClass)/k);
    [longitudTest,nada]=size(testFeatures);
    claseaux=NaN(longitudTest,k);
    p=NaN(longitudTest,1);
    clase=p;
    string=params;
    N = size(trainFeatures,1);
    parfor ii=1:k
        featuresTrain=(trainFeatures(longitud*(ii-1)+1:min(longitud*ii,N),:));
        claseTrain=(trainClass(longitud*(ii-1)+1:min(longitud*ii,N)));
        claseaux(:,ii)=clasificacionSVMparams(featuresTrain, claseTrain, testFeatures,...
        string);
    end

    for ii=1:longitudTest
        cont=0;
        for a=1:k
            if(claseaux(ii,a)==1)
                cont=cont+1;
            end
        end
        p(ii)=cont/k;
        if(mean(claseaux(ii,:))>=0)
            clase(ii)=1;
        else
            clase(ii)=-1;
        end
    end
end
end
```

```

if (strcmp(metodo,'SRKDAconjunto'))

    %Opciones del clasificador
    options = [];
    options.KernelType = 'Gaussian';
    options.ReguType = 'Ridge';
    options.ReguAlpha = 0.05;
    options.t = params;

    %N\{u}mero de divisiones k
    k=2;
    longitud=ceil(length(trainClass)/k);
    [longitudTest,~]=size(testFeatures);
    claseaux=NaN(longitudTest,k);
    p=NaN(longitudTest,1);
    clase=p;
    N = size(trainFeatures,1);

    for ii=1:k
        featuresTrain=(trainFeatures(longitud*(ii-1)+1:min(longitud*ii,N),:));
        claseTrain=(trainClass(longitud*(ii-1)+1:min(longitud*ii,N)));
        [claseaux(:,ii),p]=clasificacionSRKDA(featuresTrain, testFeatures, claseTrain,...
        options);
        paux(:,ii)=p(:,1);
    end

    p=sum(paux,2)/k;

end

```

Fichero clasificacionSRKDA.m

```

function [clase,p]=clasificacionSRKDA(trainFeatures,testFeatures,trainClass,options)

    %SRKDA
    addpath('SRKDA')
    %Entrenamiento
    model = SRKDAttrain(trainFeatures, trainClass, options);
    %Clasificaci\{o}n
    N = size(testFeatures,1);
    [~, clase, p] = SRKDApredict(testFeatures, ones(N,1), model);
end

```

Fichero clasificacionSVM.m

```

function [clase,p] = clasificacionSVM( trainFeatures, trainClass, testFeatures )
    model = svmtrain(trainClass, trainFeatures, '-g 1e-6 -c 30 -w1 11 -w-1 1 -q');
    N = size(testFeatures,1);
    clase = svmpredict(ones(N,1), testFeatures, model, '-q');

end

```

Fichero clasificacionSVMparams.m

```

function [clase,p] = clasificacionSVMparams( trainFeatures, trainClass, testFeatures,...
string )
    model = svmtrain(trainClass, trainFeatures, string);
    N = size(testFeatures,1);
    clase = svmpredict(ones(N,1), testFeatures, model, '-q');

end

```

ClasificacionTestWaveletFinal.m

```

clc
%close all

% Par\{a}metros
dir='..\..\02mat';
metodoExt='ganador';
%metodoClas='SRKDAconjunto';
metodoClas={'LDA2','SRKDAconjunto','SVM'};
%metodoClas={'SVM'};
sujetos={'Sujeto B'};
reguOptLDA=[13001, 5001]; %LDA
reguOptSRKDA=[1000, 900]; %SRKDA
wmadres={'sym4'};
Nobjetos=size(sujetos,2);
Nwavelets=size(wmadres,2);
Nclasificadores=size(metodoClas,2);
diezmado=3; %Flag para decidir qu\{e} datos se usan

for itt=1:Nwavelets
    for it=1:Nobjetos
        if(diezmado==0)
            train=sprintf('%s\\%s\\train\\%s.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s.mat',dir,metodoExt,sujetos{it});
            mCanal=160;
        end
        if(diezmado==1)
            train=sprintf('%s\\%s\\train\\%s 4.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 4.mat',dir,metodoExt,sujetos{it});
            mCanal=4;
        end
        if(diezmado==2)
            train=sprintf('%s\\%s\\train\\%s 6.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 6.mat',dir,metodoExt,sujetos{it});
            mCanal=6;
        end
        if(diezmado==3)
            train=sprintf('%s\\%s\\train\\%s 8.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 8.mat',dir,metodoExt,sujetos{it});
            mCanal=8;
        end
        if(diezmado==4)
            train=sprintf('%s\\%s\\train\\%s 10.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 10.mat',dir,metodoExt,sujetos{it});
            mCanal=10;
        end
        if(diezmado==5)
            train=sprintf('%s\\%s\\train\\%s 13.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 13.mat',dir,metodoExt,sujetos{it});
            mCanal=13;
        end
        if(diezmado==6)
            train=sprintf('%s\\%s\\train\\%s 14.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 14.mat',dir,metodoExt,sujetos{it});
            mCanal=14;
        end
        if(diezmado==7)
            train=sprintf('%s\\%s\\train\\%s 16.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 16.mat',dir,metodoExt,sujetos{it});
            mCanal=16;
        end
        if(diezmado==8)
            train=sprintf('%s\\%s\\train\\%s 18.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 18.mat',dir,metodoExt,sujetos{it});
            mCanal=18;
        end
        if(diezmado==9)
            train=sprintf('%s\\%s\\train\\%s 20.mat',dir,metodoExt,sujetos{it});
            test=sprintf('%s\\%s\\test\\%s 20.mat',dir,metodoExt,sujetos{it});
            mCanal=20;
        end
    end
end

```



```

end
if(diezmado==10)
    train=sprintf('%s\\%s\\train\\%s 22.mat',dir,metodoExt,sujetos{it});
    test=sprintf('%s\\%s\\test\\%s 22.mat',dir,metodoExt,sujetos{it});
    mCanal=22;
end
trainData=cargarDatos(train);
testData=cargarDatos(test);
%% Extracci\{o\}n de caracter\{i\}sticas

channels=1:64;

[trainData.features, mCanal2]=waveletExtract(trainData.features, mCanal,...
channels, wmadres{itt});
[testData.features, mCanal]=waveletExtract(testData.features, mCanal,...
channels, wmadres{itt});
mCanal=mCanal2;

for ittt=1:Nclasificadores
    if strcmp(metodoClas{ittt},'SVM')
        [channels, channelsOut] = canalOptimo2(trainData.features,...
        trainData.class, metodoClas{ittt}, mCanal, NaN(1));
    end
    if strcmp(metodoClas{ittt},'LDA2')
        [channels, channelsOut] = canalOptimo2(trainData.features,...
        trainData.class, metodoClas{ittt}, mCanal, reguOptLDA(it));
    end
    if strcmp(metodoClas{ittt},'SRKDAconjunto')
        [channels, channelsOut] = canalOptimo2(trainData.features,...
        trainData.class, metodoClas{ittt}, mCanal, reguOptSRKDA(it));
    end
    numChannels=length(channels); %N\{u\}mero de canales

    if (numChannels~=0)
        trainData.features2=selectorCanal(channels, trainData.features, 'train',...
        mCanal);
        testData.features2=selectorCanal(channels, testData.features, 'test',...
        mCanal);
    end

    matrizResults=NaN(1,11);
    suj=sprintf('%s %s %s %d',sujetos{it},wmadres{itt},metodoClas{ittt},diezmado);
    fprintf('%s \n',sujetos{it})

for noIllumCont=5:1:15
    %% Clasificaci\{o\}n
    noIllum=noIllumCont;
    noIllum=((180)/15)*noIllum;
    noIllumtot=noIllum*100;
    testData.featuresAux=testData.features2(:,1:noIllum,:);
    testData.classAux=testData.class(:,1:noIllum);
    testData.codeAux=testData.code(:,1:noIllum);
    trainData.featuresAux=reshape(permute(trainData.features2,[2 1 3]),15300,...
    Canal*numChannels);
    testData.featuresAux=reshape(permute(testData.featuresAux,[2 1 3]),...
    noIllumtot,mCanal*numChannels);
    trainData.classAux=reshape(permute(trainData.class,[2 1]),15300,1);
    testData.classAux=reshape(permute(testData.classAux,[2 1]),noIllumtot,1);

    if strcmp(metodoClas{ittt},'SVM')
        [p,clase] = clasificacion( trainData.featuresAux,...
        testData.featuresAux, trainData.classAux, metodoClas{ittt});
    end
    if strcmp(metodoClas{ittt},'LDA2')
        [p,clase] = clasificacionParams( trainData.featuresAux,...
        testData.featuresAux, trainData.classAux, metodoClas{ittt},...
        reguOptLDA(it));
    end
end

```

```

    if strcmp(metodoClas{ittt},'SRKDAconjunto')
        [p,clase] = clasificacionParams( trainData.featuresAux,...
            testData.featuresAux, trainData.classAux, metodoClas{ittt},...
            reguOptSRKDA(it));
    end

    %% Letras
    testData.codeAux=reshape(permute(testData.codeAux,[2 1]),noIllumTot,1);
    [letras, caract]=determinarLetra(p,testData.codeAux,...
        estData.matrix,noIllum*15/180);

    %% Evaluaci\{o}n
    aciertos=posicionLetras(letras,testData.target,caract);
    matrizResults(noIllumCont-4)=aciertos;

    end
    complejidad=1;
    save(suj,'matrizResults','complejidad','channelsOut');
end
end
end
end

```

Fichero crossvalmatrix.m

```

function matriz = crossvalmatrix( par1,par2,varargin )
    if nargin==2
        [n1,n2] = ndgrid(par2,par1);
        matriz=[n2(:) n1(:)];
    end
    if nargin==3
        [n1,n2,n3] = ndgrid(varargin{1},par2,par1);
        matriz=[n3(:) n2(:) n1(:)];
    end

end

end

```

Fichero curvas.m

```

function [out_perf,X,Y] = curvas(class,p,type)

    if (strcmp(type,'ROC'))
        [X,Y,T,out_perf] = perfcurve(class,p,1);
        figure
        plot(X,Y)
        title('Curva ROC')
        drawnow
    end

    if(strcmp(type,'F'))
        [X,Y1,T1,AUC(1)] = perfcurve(class,p,1,'YCrit','TPR');
        figure
        plot(T1,Y1)
        title('Sensibilidad')
        [X,Y2,T2,AUC(2)] = perfcurve(class,p,1,'YCrit','PPV');
        figure
        plot(T2,Y2)
        title('Precisi\{o}n')
        F=(Y2.*Y1)/(Y2+Y1);
        figure
        plot(T2,F)
        title('F-Score')
        drawnow
        X = T2;
        Y = F;
        out_perf=max(F);
    end

end

end

```

Fichero determinarLetra.m

```
function [letras,caract]=determinarLetra(p,code,matrix,noIllum)
%Determinar letras
  noIllum=15; %Seleccionar n\{u}mero de iluminaciones
  noIllum=((180)/15)*noIllum;
  p = reshape(p,noIllum,[]);
  [nada,caract]=size(p);
  for n=1:caract

      v=NaN(1,12);
      for i= 1:12 % Cada fila-columna (12*15 rep= 180 ilum)
          v(1,i)=mean(p(code(1+(n-1)*noIllum:noIllum+(n-1)*noIllum,1)==i,n));
          %En cada elemento de v guardamos la media de las probabilidades de cada fila/columna
      end

      v1=repmat(v(1,1:6),6,1);
      v2=repmat(v(1,7:12)',1,6);

      V=(v1+v2)/2;

      valor=max(V(:));
      [x,y]=find(V==valor,1);

      letras(n)=matrix(x,y);

  end

  clear nada
end
```

Fichero ratioCruzado.m

```
function [ ratioTotal ] = ratioCruzado( canales, features, class, nMuestras,...
metodoClas, caracteres, iluminaciones, params)
  % Esta funcion encuentra el ratio
  % truePositives/(truePositives+falsePositives+falseNegatives) mediante
  % validacion cruzada en 10 particiones sobre las se~{n}ales de
  % entrenamiento

  canalesTotal=length(canales);
  featuresMat=NaN(10,caracteres*iluminaciones/10,nMuestras*canalesTotal);
  %(parte, iluminaciones por caracter, muestras por caracter*numeroCanales)
  classMat=NaN(10,caracteres*iluminaciones/10);
  %Particion de la validacion cruzada

  featuresAux=selectorCanal(canales, features, 'train', nMuestras);
  featuresMatAux=reshape(permute(featuresAux,[2 1 3]),15300,nMuestras*canalesTotal);
  classAux=reshape(permute(class,[2 1]),15300,1);

  for a=1:10
      featuresMat(a,:,:) = featuresMatAux(((a-1)*(caracteres*iluminaciones/10)+1)...
      :(a*caracteres*iluminaciones/10),:);
      classMat(a,:)=classAux(((a-1)*(caracteres*iluminaciones/10)+1)...
      :(a*caracteres*iluminaciones/10));
  end

  for a=1:10
      featuresTrain=[];
      classAux=[];
      featuresTest=NaN(1530,nMuestras*canalesTotal);
      for aa=1:10 % Creamos el vector de entrenamiento
          if (aa~=a)
              featuresTrain=[featuresTrain featuresMat(a,:,:)];
              classAux=[classAux classMat(a,:)];
          end
      end
      featuresTrain=reshape(permute(featuresTrain,[2 1 3]),15300*9/10,...
      nMuestras*canalesTotal);
```

```

classAux=reshape(permute(classAux,[2 1]),15300*9/10,1);
featuresTest=reshape(permute(featuresMat(a, :, :), [2 1 3]),1530,...
nMuestras*canalesTotal);

if strcmp(metodoClas,'SVM')
    [p,clase] = clasificacion( featuresTrain, featuresTest, classAux,...
    metodoClas);
else
    [p,clase] = clasificacionParams( featuresTrain, featuresTest, classAux,...
    metodoClas, params);
end

%% Encontramos el ratio
indicesUnosReal=find(classMat(a, :)==1);
indicesMenosUnosReal=find(classMat(a, :)==-1);
%Encontramos verdaderos positivos
truePositives=length(find(clase(indicesUnosReal)==1));

%Encontramos falsos positivos
falsePositives=length(find(clase(indicesMenosUnosReal)==1));

%Encontramos falsos negativos
falseNegatives=length(find(clase(indicesUnosReal)==-1));

%Calculamos ratio
dif=truePositives/(truePositives+falsePositives+falseNegatives);

end

ratioTotal=sum(dif)/10; %Ponderamos el ratio total

end

```

Fichero selectorCanal.m

```

function [featuresOut] = selectorCanal(channels, features, tipo, mCanal)
    %Cada canal tiene 160 muestras por iluminaci\{o\}n sin diezmar
    %mCanal = uestras por canal
    numChannels=length(channels); %N\{u\}mero de canales seleccionados

    if (strcmp(tipo,'train'))
        featuresaux=NaN(85,180,mCanal);
    end
    if (strcmp(tipo,'test'))
        featuresaux=NaN(100,180,mCanal);
    end

    for ii=1:numChannels
        featuresaux(:, :, (ii-1)*mCanal+1:(ii)*mCanal)=features(:, :, (channels(ii)-1)*mCanal+1...
        :channels(ii)*mCanal);
    end

    featuresOut=featuresaux;

end

```

Fichero waveletExtract.m

```

function [featuresOut, mCanalOut] = waveletExtract( featuresIn, mCanal, channels, wmadre )

[caract, ilum, ~]=size(featuresIn);
featuresAux=NaN(1,mCanal);
iter=3; %N\{u\}mero de iteraciones DWT

for a=1:caract
    for aa=1:ilum
        for ii=1:length(channels)
            featuresAux(:, :)=featuresIn(a, aa, (channels(ii)-1)*mCanal+1:channels(ii)...

```

```
*mCanal);

[a1,d1]=dwt(featuresAux,wmadre);
%[a1,d2]=dwt(a1,wmadre);
%[a1,d3]=dwt(a1,wmadre);
mCanalOut=length([a1 d1]);
muestrasCanal(ii,1:mCanalOut)=[a1 d1];
%{
for aaa=2:iter
    [a1,d1]=dwt(a1,'db5');
    mCanalOut=length([a1 d1]);
    %featuresOut(a,aa,(ii-1)*mCanal+1:(ii)*mCanalOut)=[a1 d1];
    muestrasCanal(ii,1:mCanalOut)=[a1 d1];
end
%}
end
featuresOutAux=NaN(1,length(channels)*mCanalOut);
for ii=1:length(channels)
    %featuresOut(a,aa,(ii-1)*mCanal+1:(ii)*mCanalOut)=muestrasCanal(ii,:);
    featuresOutAux((ii-1)*mCanalOut+1:(ii)*mCanalOut)=muestrasCanal(ii,:);
end

    featuresOut(a,aa,:)=featuresOutAux;
end
    %fprintf('Acabada dwt de un caracter %d \n', a);
end
    %featuresOut(a,aa,(ii-1)*mCanal+1:(ii)*mCanalOut)=muestrasCanal(ii,:);
end
```