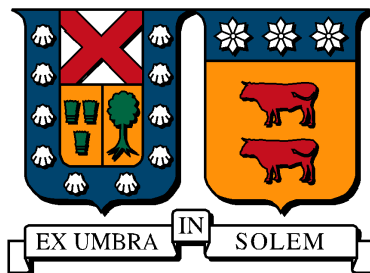


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA



Desarrollo de técnicas de colocación y reducción del número de electrodos para EEG basado en GSP

Tesis de grado presentada por
Sebastián Andre Espinoza Toro

Como requisito parcial para optar al grado de
Magíster en Ciencias de la Ingeniería Electrónica

Director de Tesis
Alejandro Weinstein, Ph.D.

Co-Director de Tesis
Matías Zañartu, Ph.D.

Valparaíso, 12 de marzo de 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título Tesis de Postgrado

Título del trabajo: Desarrollo de técnicas de colocación y reducción del número de electrodos para EEG basado en GSP

Nombre del candidato(a): Sebastián Andre Espinoza Toro

Carrera / Grado: Magister en ciencias de la ingeniería electrónica, mención computadores

Campus: Casa Central Departamento: Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Alejandro Weinstein, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

6 meses 12 meses 2 años 3 años 5 años 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 13-3-2026

Firma: A. Weinstein D.

Estudiante o Candidato(a):

Fecha: 12/03/2026

Firma: Sebastián E.

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Departamento de Ingeniería Electrónica

Desarrollo de técnicas de colocación y reducción del número de electrodos para EEG basado en GSP

Tesis de grado presentada por
Sebastián Andre Espinoza Toro

Como requisito parcial para optar al grado de
Magíster en Ciencias de la Ingeniería Electrónica

Director de Tesis
Alejandro Weinstein, Ph.D.

Co-Director de Tesis
Matías Zañartu, Ph.D.

Valparaíso, 12 de marzo de 2026

Tesis:

Desarrollo de técnicas de colocación y reducción del número de electrodos para EEG basado en GSP

Autor:

Sebastián Andre Espinoza Toro

TRABAJO DE TESIS, presentado en cumplimiento parcial de los requisitos para el grado de Magíster en Ciencias de la Ingeniería Electrónica de la Universidad Técnica Federico Santa María, Valparaíso, Chile.

Director de Tesis,

Alejandro Weinstein, Ph.D.

Co-Director de Tesis,

Matías Zañartu, Ph.D.

Examinador Externo,

Alejandro Veloz, Ph.D.

(Universidad de Valparaíso)

Agradecimientos

Agradecimientos a mis padres, familia, amigos, y Sofía. Han sido largos años, de los que probablemente en algún momento pensé en cambiar cosas del pasado o haber hecho algo distinto. La verdad, es que a día de hoy, no cambiaría nada de mi vida, porque mi presente es perfecto gracias a todos ustedes.

*Agradezco enormemente a mi profesor Alejandro Weinstein por su apoyo, comprensión y por no ser solo un buen profesor, sino que también muy buena persona. Finalmente, aprovecho de dar los agradecimiento a **FONDECYT REGULAR 1231132, ANILLO ACT210053 y BASAL FB0008** por el financiamiento y por permitirme trabajar en este proyecto.*

Contenido

Resumen	III
Lista de Figuras	IV
1 INTRODUCCIÓN	1
1.0.1 Motivación	1
1.1 Objetivos	1
1.2 Principales contribuciones	2
1.3 Organización de la Tesis	2
2 FUNDAMENTOS TEÓRICOS	3
2.1 Electroencefalografía	3
2.1.1 Áreas del cerebro	3
2.1.2 Características de las señales EEG	4
2.1.3 Colocación de electrodos	5
2.1.4 Artefactos	8
2.2 Preprocesamiento de señales EEG	8
2.2.1 Independent Component Analysis	9
2.3 Brain-Computer Interface (BCI)	9
2.4 Graph Signal Processing (GSP)	12
2.4.1 Construcción de grafos basados en EEGs	13
2.5 Técnicas de selección y reducción de nodos	14
2.5.1 Algoritmo 1	14
2.5.2 Algoritmo 2	16
2.5.3 Algoritmo 3	17
2.6 Extracción de características	18
2.6.1 Common Spatial Patterns (CSP)	18
2.7 Clasificación	20
2.7.1 Linear Discriminant Analysis (LDA)	20
2.7.2 Validación de modelos	23
3 METODOLOGÍA E IMPLEMENTACIÓN	25
3.1 Datasets utilizados	25
3.1.1 Dataset A: EEG Motor Movement/Imagery Dataset[1]	26
3.1.2 Dataset B: Data set IVa <motor imagery, small training sets[2]	26
3.1.3 Dataset C: Data set IVb <motor imagery, uncued classifier application>[2]	27

3.1.4	Dataset D:dataset for simultaneous EEG and NIRS brain-computer interface (BCI) [3]	27
3.2	Preprocesamiento de datos EEG	28
3.2.1	Segmentación en épocas	28
3.2.2	Filtrado y eliminación de artefactos mediante ICA	28
3.3	Construcción de grafos	31
3.4	Aplicación de algoritmos de selección y reducción de electrodos	33
3.5	Extracción de características y clasificación	34
3.5.1	Diseño Experimental y Validación	34
4	RESULTADOS Y ANÁLISIS	36
4.1	Resultados por dataset	36
4.2	Desempeño Base (Linebase performance) y Panorama General	36
4.3	Análisis de rendimiento: Dataset A	37
4.3.1	tarea 3	37
4.3.2	tarea 4	37
4.4	Análisis de rendimiento: Dataset B	37
4.5	Análisis de rendimiento: Dataset C	39
4.6	Análisis de rendimiento: Dataset D	41
4.6.1	task 1	41
4.6.2	task 2	43
4.7	Resultados generales	44
4.8	Visualización de subconjuntos de electrodos	44
5	CONCLUSIONES Y TRABAJO FUTURO	47
5.1	Conclusiones Generales	47
5.2	Trabajo Futuro	48
	BIBLIOGRAFÍA	48

Resumen

Esta tesis de postgrado se basa en la utilización de Graph Signal Processing (GSP) para modelar señales de electroencefalografía (EEG) y así usar técnicas que reducen la cantidad de electrodos y mejorar el desempeño en comparación a métodos ya utilizados en tareas de Brain-Computer Interface (BCI).

GSP es un campo relativamente moderno que combina la teoría de grafos con el procesamiento de señales, permitiendo el análisis y modelado de señales de EEG. En especial, GSP permite la caracterización de patrones espaciales y temporales. Considerando lo nuevo que es el uso de GSP, especialmente en señales de EEG, se propone la reducción de la cantidad de electrodos utilizados mediante técnicas de GSP que permiten la selección y reducción de sensores más representativos.

Mediante el uso de técnicas de Machine Learning, utilizando Linear Discriminant Analysis (LDA) al implementarlo en tareas de BCI con todos los electrodos y con subconjuntos de electrodos reducidos, y se compara su rendimiento. Para el desarrollo de las implementaciones de EEG en tareas de BCI se utilizará principalmente *EEG Motor Movement/Imagery Dataset* [1], que contiene registros de 109 sujetos con 64 canales de EEG que realizaron una serie de tareas, como abrir y cerrar el puño izquierdo o derecho, abrir y cerrar ambos puños o ambos pies, o imaginar la realización de estas.

El trabajo a realizar se enmarca en el proyecto **FONDECYT 1231132**. En particular, está relacionado con el segundo objetivo específico de este proyecto.

Palabras clave: Electroencefalograma, Brain-Computer Interface, Graph Signal Processing, Common Spatial Patterns, Reducción de electrodos.

Lista de Figuras

2.1	Cerebro y sus cuatro lóbulos.	3
2.2	Ritmos EEG	4
2.3	Referencias para colocación de electrodos utilizando sistema 10-20.	5
2.4	Ubicación de electrodos utilizando el sistema 10-20.	6
2.5	Ubicación de electrodos utilizando el sistema 10-10.	6
2.6	Layout de EEG con cantidad variable de electrodos.	7
2.7	Layout de EEG con cantidad fija de electrodos.	7
2.8	Ejemplo de artefacto producido por parpadeo.	8
2.9	Diseño básico y operación de un sistema [4].	9
2.10	Artículos sobre BCI hasta 2010[4].	10
2.11	Conjunto de electrodos de electrocorticografía. [4]	11
2.12	Matriz de microelectrodos. [4]	11
2.13	Ejemplo grafo no dirigido (superior) y grafo dirigido (inferior). [5]	12
2.15	Covarianzas de dos clases antes de aplicar CSP [6]	19
2.16	Covarianzas de dos clases luego de aplicar CSP [6]	20
2.17	Separación entre las proyecciones de dos clases sin el uso del criterio de Fisher y su mejora utilizando el criterio de Fisher.[7]	22
2.18	Distribuciones Gaussianas de dos clases proyectadas	23
2.19	Ejemplo modelo con underfitting (izquierda), buena generalización (centro) y overfitting (derecha).	24
3.1	Diagrama de flujo general de la metodología implementada para la Interfaz Cerebro-Computador.	26
3.2	Primero componentes ICA en dataset C.	29
3.3	Componentes finales ICA en dataset C.	30
3.5	Topologías de los grafos generados para el dataset A al variar empíricamente sus parámetros de conectividad.	31
3.6	Topologías de los grafos generados para los datasets B y C.	32
3.7	Topologías de los grafos generados para el dataset D.	33
4.1	Accuracy tarea 3 dataset A.	38
4.2	Accuracy tarea 4 dataset A.	38
4.3	Accuracy en dataset B.	39
4.4	Accuracy dataset C.	41
4.5	Accuracy dataset D para task 1.	42

4.6	Accuracy dataset D para task 2.	43
4.7	Selección espacial de electrodos para grafo 1 algoritmo 1 en dataset B y C. .	45
4.8	Selección espacial de electrodos para grafo 1 algoritmo 2 en dataset B y C. .	46
4.9	Selección espacial de electrodos para grafo 1 algoritmo 3 en dataset B y C. .	46

CAPÍTULO



INTRODUCCIÓN

1.0.1. Motivación

Las interfaces Cerebro-Computador (BCI) han demostrado un potencial transformador, permitiendo traducir la intención motora de un usuario y traducirla en un comando, que pueda potencialmente facilitar su vida. Aunque para algunos es algo que suene a ciencia ficción, es algo que ya existe. Sin embargo, la transición a estas tecnologías está alejada de ser humano corriente, pues sigue teniendo restricciones que limitan su uso y práctica.

Un problema que surge en la captura de señales de EEG, es la alta densidad de electrodos que se utilizan en ciertos contextos, alcanzo más de 100 en su cantidad. Teniendo una perspectiva de hardware, aquella cantidad de sensores resulta impráctico por temas de espacio y complejidad. Puede resultar ser incómodo y de difícil acceso para el usuario.

Ahora, desde el punto de vista del procesamiento de señales, no toda información capturada por los sensores es siempre útil, pues aportan datos redundantes, tienen ruido y puede incluso provocar sobreajustes en el contexto de machine learning. Es por este motivo que surge la necesidad de desarrollar sistemas eficientes y más simples, sin perder rendimiento. Así, se puede hacer uso del procesamiento de señales en grafos, y explorar las herramientas que brinda en contextos donde la posición de los sensores sí importa.

1.1. Objetivos

El objetivo principal de esta tesis es el desarrollo e implementación de algoritmos de Graph Signal Processing para la reducción de electrodos en electroencefalogramas. GSP es un área relativamente nueva dentro del procesamiento de señales y cuyas aplicaciones no han

sido profundamente estudiadas en el área de la salud, específicamente para el problema de selección de electrodos en EEGs, cuya estructura permite un análisis como si fuera un grafo. De forma paralela, se compararán los resultados con subconjuntos de electrodos de un uso estándar y que servirán para comprobar si el uso de estas nuevas técnicas aplicadas en EEG obtienen rendimientos superiores en la clasificación de tareas de Brain-Computer Interface.

1.2. Principales contribuciones

Las principales contribuciones de esta Tesis se pueden resumir de la siguiente manera:

1. Se implementan algoritmos de Graph Signal Processing en electroencefalogramas con la novedad de reducir el número de electrodos.
2. Se compara el rendimiento de técnicas de GSP en tareas de Brain-Computer interface para la validación del uso de GSP en este contexto.
3. Se contribuye en la literatura con resultados comparativos en distintos datasets públicos que pueden servir como referencia futura en el emergente campo de GSP.
4. Se implementan algoritmos en la librería pygsp2 de python.
5. Se envía un artículo con el trabajo realizado a la revista Biomedical Signal Processing and Control.

1.3. Organización de la Tesis

La presente Tesis se organiza mediante capítulos ordenados de la siguiente manera:

- En el Capítulo 2 se detallan definiciones y notaciones importantes para el desarrollo de la Tesis.
- El Capítulo 3, de metodología e implementación, se describe cómo se lleva a cabo el estudio. Se explica el diseño de la investigación, los métodos y las técnicas utilizadas, así como también los procedimientos de procesamiento de los datos, parámetros y herramientas utilizadas.
- En el Capítulo 4 se presentan los datos y hallazgos principales del estudio. Se incluyen gráficos, tablas y análisis que reflejan el desempeño de las tareas realizadas.
- Finalmente en el Capítulo 5 se mencionan las conclusiones y trabajos futuros.

CAPÍTULO

2

FUNDAMENTOS TEÓRICOS**2.1. Electroencefalografía**

La electroencefalografía (EEG) es una técnica no invasiva que registra la actividad eléctrica del cerebro utilizando electrodos en el cuero cabelludo. El EEG es una herramienta fundamental para el diagnóstico clínico, neurociencia y el desarrollo de sistemas de Brain-Computer Interface.

2.1.1. Áreas del cerebro

La corteza cerebral, observada en la Figura 2.1 se divide en cuatro partes llamadas lóbulos, donde cada una es responsable de distintas funciones del sistema nervioso.

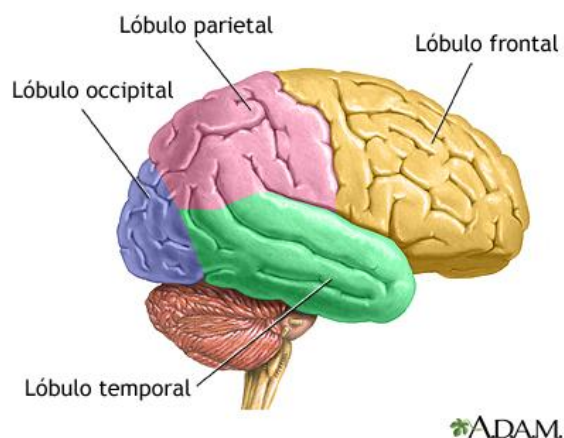


Figura 2.1: Cerebro y sus cuatro lóbulos.

- Frontal: Se encarga del razonamiento, control del habla y del movimiento, emociones y resolución de problemas.
- Central: Sensoriomotor.
- Parietal Atención, percepción y procesamiento de los estímulos relacionados con los sentidos.
- Occipital: Se encarga de la visión, nos hace ser consciente de lo que vemos.
- Temporal: Se encarga de la memoria, el significado y la interpretación, y procesamiento de los estímulos auditivos.

2.1.2. Características de las señales EEG

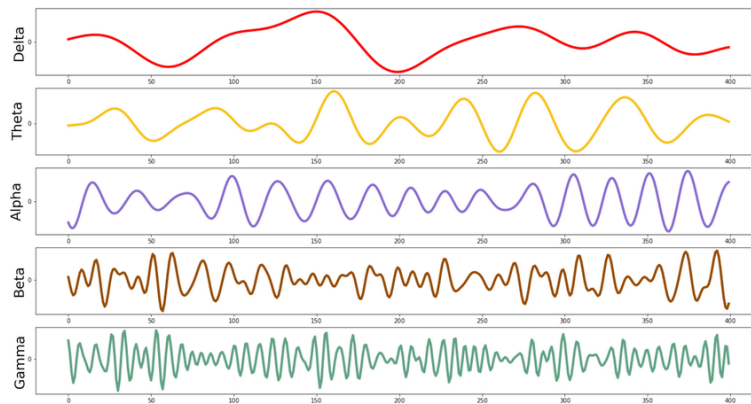


Figura 2.2: Ritmos EEG

Las señales registradas por el EEG están compuestas por distintos ritmos, como se aprecia en la Figura 2.2. Cada ritmo está asociado a un estado y función distinto.

- Ondas delta δ (0.5-4 Hz): Es el ritmo más lento y aparece en el sueño profundo.
- Ondas theta θ (4-8 Hz): Es el ritmo observado en el sueño ligero, somnolencia y estados de relajación profunda.
- Ondas alpha α (8-13 Hz): Corresponden al ritmo presente en la actividad eléctrica de un cerebro adulto despierto.
- Ondas beta β (13-30 Hz): Ocurre en estados de pensamiento activo, alerta, atención, y actividades motoras.
- Ondas gamma γ (sobre 30 Hz): Es el ritmo con las ondas de mayor frecuencia, y está relacionado a procesos cognitivos, aprendizaje y actividades mentales complejas.

2.1.3. Colocación de electrodos

La cantidad de electrodos utilizados puede variar según su aplicación. Además, la colocación correcta de estos es de vital importancia, pues su ubicación determina qué áreas corticales se están registrando, lo cual es clave para la obtención de datos precisos y su posterior análisis o reproducción. Es por esto que se utilizan los sistemas 10-20 y 10-10.

Sistema 10-20

El sistema internacional de colocación de EEG [8] toma los puntos craneales de referencia universales nasion, inion y ambos puntos pre-auricular, distribuyendo proporcionalmente los electrodos del EEG sobre la superficie de la cabeza. Los números 10-20 hacen referencia al porcentaje (10 y 20 %) de distancia en la que deben estar colocados los electrodos en la parte frontal (Nasion), occipital (Inion), de lado derecho e izquierdo del cráneo (anterior al oído). Esto se puede observar en la Figura 2.3.

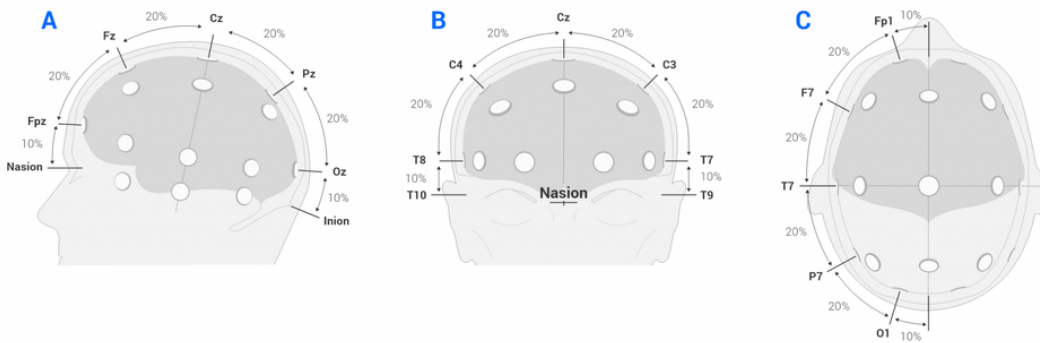


Figure 3 (A-C): Placement of the standard electrodes of the 10–20-system. Modified from: Clinical Neurophysiology, 2017.

Figura 2.3: Referencias para colocación de electrodos utilizando sistema 10-20.

Se tiene entonces:

- Los electrodos son etiquetados según el área de la corteza cerebral en la que se encuentren, las cuales son frontopolar (Fp), frontal (F), central (C), temporal (T), parietal (P) y occipital (O).
- Los números impares (1, 3, 5, 7) corresponden a los electrodos colocados en el hemisferio izquierdo, mientras que los números pares (2, 4, 6, 8) corresponden a los colocados en el hemisferio derecho.
- Los electrodos que están sobre la línea media son etiquetados con zero (z).

En la Figura 2.4 se puede apreciar la distribución de 21 electrodos utilizando el sistema 10-20.

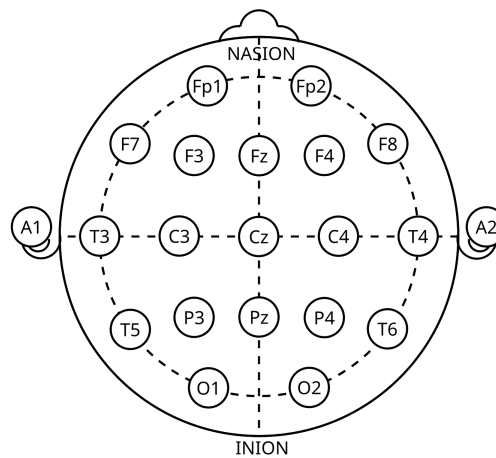


Figura 2.4: Ubicación de electrodos utilizando el sistema 10-20.

Sistemas con mayor resolución

Existe el sistema 10-10, que es una extensión del sistema 10-20 y utiliza distancias de 10% entre electrodos adyacentes, lo cual permite una mayor densidad de electrodos, y por consiguiente, más datos. Este sistema posee una nomenclatura distinta debido a su complejidad, añadiendo códigos en las posiciones de electrodos intermedios como AF (entre Fp y F), FC (entre F y C), FT (entre F y T), etc. En la Figura 2.5 se puede observar esta nueva distribución.

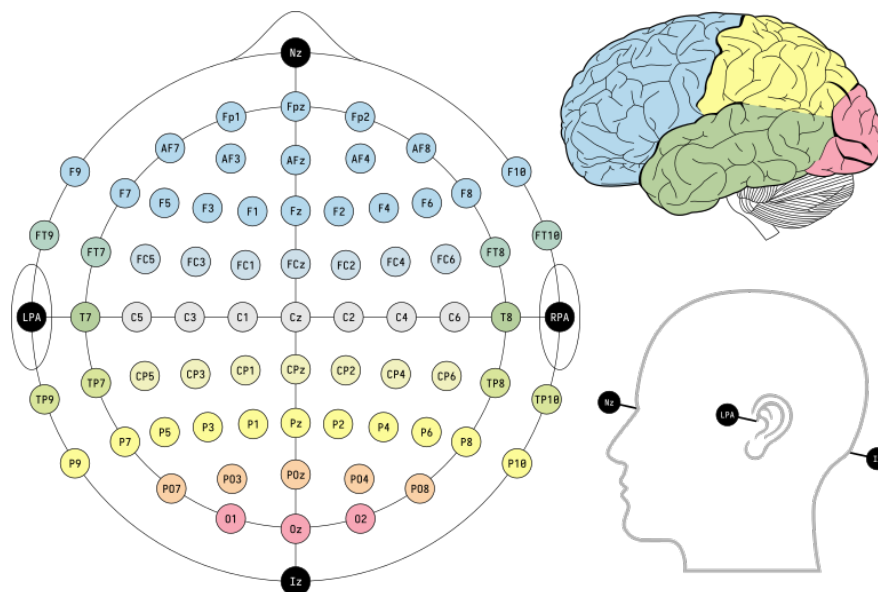


Figura 2.5: Ubicación de electrodos utilizando el sistema 10-10.

Existe incluso el sistema 10-5, siendo también una extensión del sistema 10-20, que permite hasta 320 electrodos.

Layouts de EEG

Existen dos tipos de *layouts*. En el primer tipo se tiene una cantidad variable de electrodos, permitiendo ubicarlos en distintas posiciones (Figura 2.6). En el segundo tipo la cantidad y posición de los electrodos es fija (Figura 2.7).

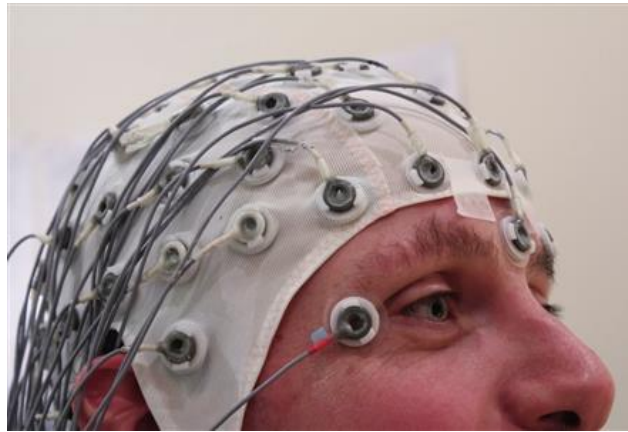


Figura 2.6: Layout de EEG con cantidad variable de electrodos.



Figura 2.7: Layout de EEG con cantidad fija de electrodos.

Los sistemas variables permiten ajustar las posiciones para diferentes experimentos, lo que es ideal en un entorno de investigación. Por otro lado, los sistemas fijos son diseñados para medir procesos específicos, y que dependen netamente de la aplicación que se les dará.

2.1.4. Artefactos

Aunque el EEG está diseñado para registrar la actividad cerebral, también registra actividad eléctrica proveniente de otros sitios además del cerebro[9]. Toda actividad registrada que no surge del cerebro mismo es considerada un artefacto. Los artefactos más comunes son causados por parpadeos, movimientos laterales de los ojos, masticación, movimiento de la lengua, tener los electrodos sueltos, interferencia eléctrica de 60 Hz, etc [10].

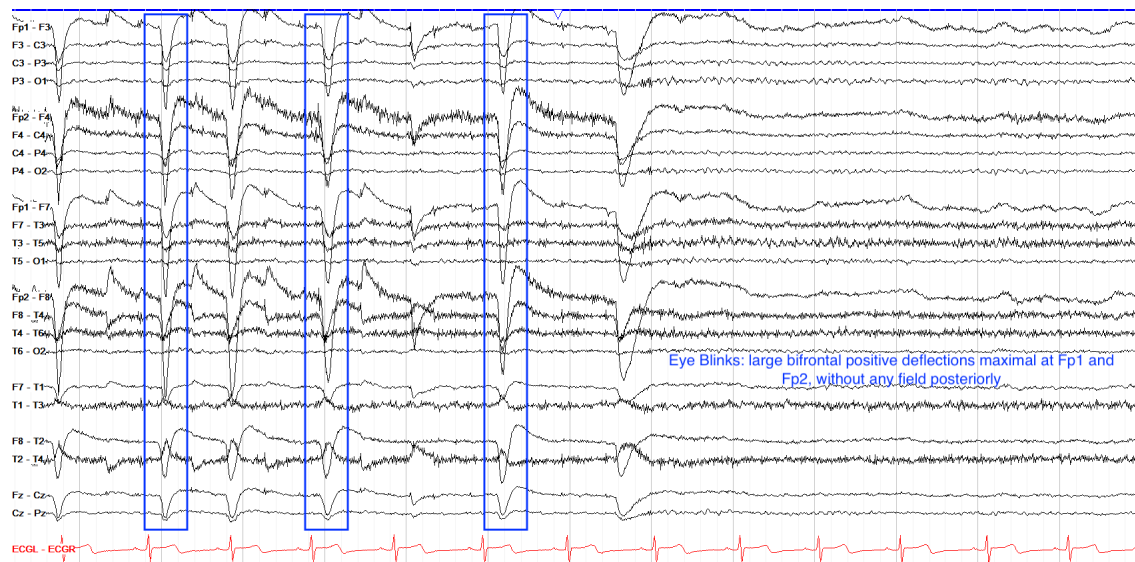


Figura 2.8: Ejemplo de artefacto producido por parpadeo.

En la Figura 2.8 se muestra un artefacto producido por parpadeo, y como es evidente, la magnitud de este ruido es considerablemente mayor a la señal que se quiere analizar, por lo que la presencia de artefactos genera un gran problema cuando es necesario caracterizar los registros obtenidos. Es por esto que, en lo posible, se debe intentar eliminar los artefactos mediante técnicas como Independent Component Analysis (ICA) y electrooculograma (EoG).

2.2. Preprocesamiento de señales EEG

Para el manejo de datos, primero se obtienen las épocas en el experimento realizado. Una época es un segmento temporal de la señal de EEG registrada que se saca a partir de un evento o estímulo, como por ejemplo apretar un puño dado un cue visual. Una época es un arreglo de $n \times m$, donde n representa la cantidad de canales (electrodos) y m representa la cantidad de muestras.

Para evitar ruido innecesario y que se permita una separación más clara de artefactos, se filtran las señales en el rango de 0.5 a 45 Hz, para luego aplicar ICA. Posteriormente se aplica un filtro pasa banda en el rango de 8 a 30 Hz, que corresponde al rango de frecuencias

relevante para BCI.

2.2.1. Independent Component Analysis

Para poder trabajar con una señal lo más limpia posible, se utiliza Independent Component Analysis [11] (ICA). ICA es una técnica de separación de fuentes usada para separar componentes estadísticamente independientes en datos multidimensionales, como lo son en este caso registros de EEGs. Como los componentes son estadísticamente independientes, entonces tampoco están correlacionados.

2.3. Brain-Computer Interface (BCI)

Brain-Computer Interface (BCI) es un sistema que mide la actividad del Sistema Nervioso Central (SNC) y lo convierte en una salida artificial que reemplaza, restaura, mejora, complementa o mejora la salida natural de SNC, y por lo tanto, cambia las interacciones continuas entre el SNC y su entorno externo o interno. Es decir, un BCI registra las señales cerebrales, extrae medidas (o características) específicas de ellas y las convierte (o traduce) en salidas artificiales que actúan en el mundo exterior o sobre el propio cuerpo. Un ejemplo de aplicación de un sistema BCI es ayudar a recuperar/reemplazar un función perdida producto de una enfermedad o lesión, como lo puede ser una persona que no puede hablar y use un sistema BCI para hablar por medio de un sintetizador de voz. En la Figura 2.9 se puede apreciar el diseño básico y operación de un sistema BCI [4]

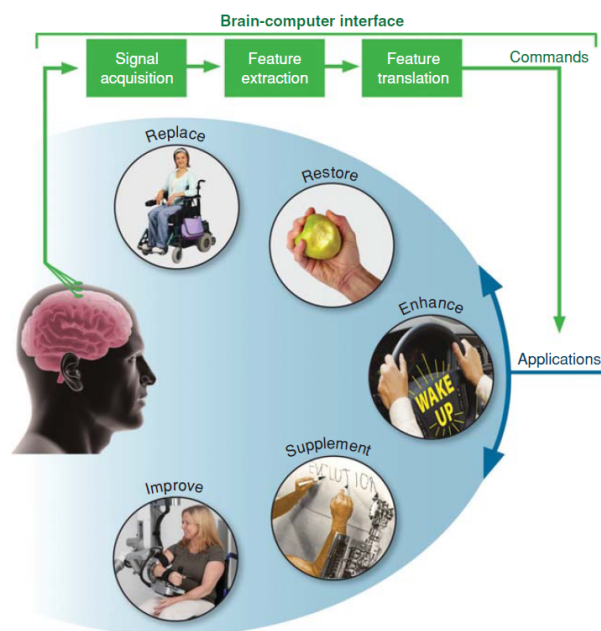


Figura 2.9: Diseño básico y operación de un sistema [4].

Los sistemas BCI pueden usar distintos tipos de señales cerebrales, por lo que es una posibilidad encontrarse con artefactos que previamente deben ser eliminados.

La investigación de sistemas BCI a aumentado a lo largo de los años, y como se ve en la Figura 2.10, la tendencia sigue y sigue. Un ejemplo actual y conocido por el público general es la empresa *Neuralink* que desarrolla BCI con el objetivo de ayudar a personas con parálisis [12].

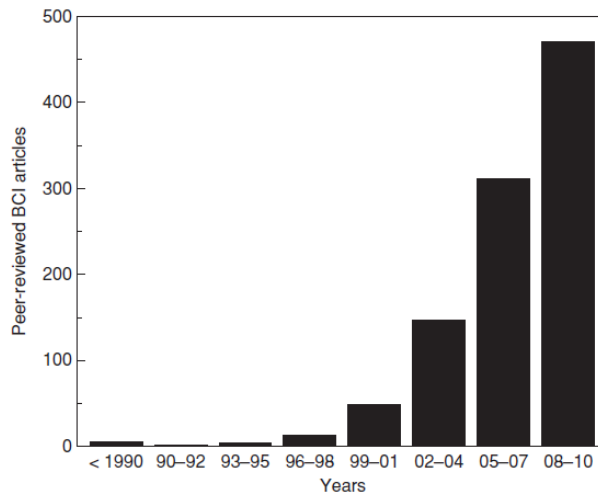


Figura 2.10: Artículos sobre BCI hasta 2010[4].

Un desafío dentro en el diseño de sistemas BCI es que la interfaz debe ser segura y confiable, no debe ser más invasiva de lo necesario y las señales neuronales deben tener la información suficiente para el uso correcto de la interfaz. Considerando esto, hay tres tipos de interfaces.

- Conjunto de electrodos de electroencefalografía (EEG) no invasivos. (Figura 2.6).
- Conjunto de electrodos de electrocorticografía (ECoG) que se colocan quirúrgicamente en el cerebro. (Figura 2.11).

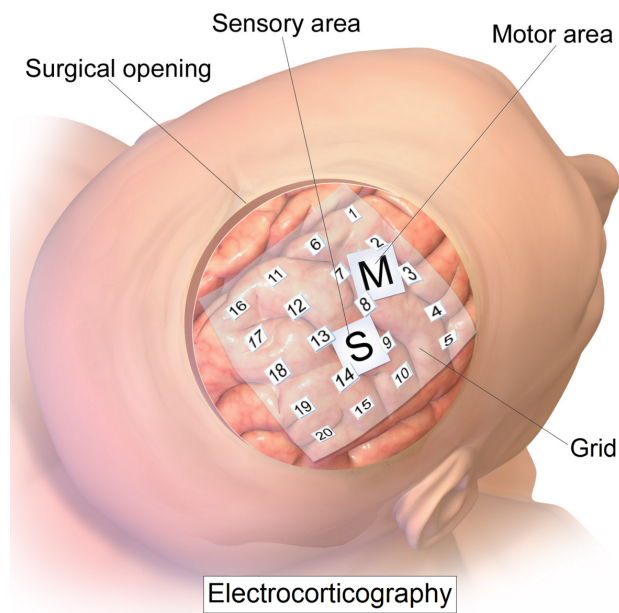


Figura 2.11: Conjunto de electrodos de electrocorticografía. [4]

- Matrices de microelectrodos miniaturizados que se insertan quirúrgicamente en la corteza cerebral.

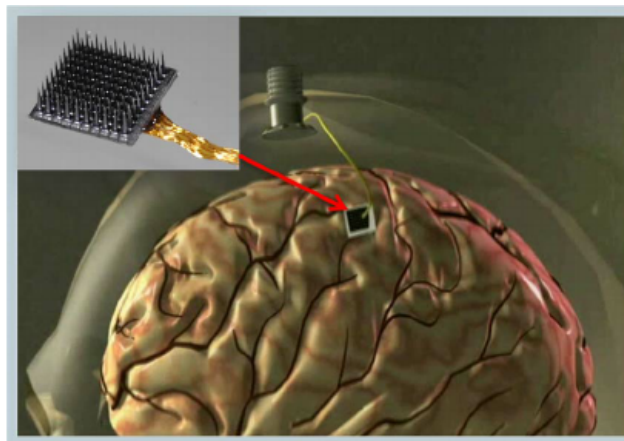


Figura 2.12: Matriz de microelectrodos. [4]

Debido a su no invasividad, accesibilidad, portabilidad y bajo costo, en comparación al resto de técnicas, el uso de EEGs para sistemas BCI posee una gran ventaja, siendo práctico tanto en usos clínicos como no clínicos.

2.4. Graph Signal Processing (GSP)

Los grafos son una estructura de datos que permite representar una gran cantidad de fenómenos al poder modelar las relaciones entre elementos, como redes sociales (usuarios y sus conexiones), redes de transporte (lineas de metro), mapas, y rutas. De esta forma, un grafo se puede entender como una serie de vértices (o nodos), conectados por medio de aristas.

Formalmente, un grafo se define como $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, donde \mathcal{V} es el conjunto de vértices (o nodos) con cardinalidad $|\mathcal{V}| = N$ y \mathcal{E} es el conjunto de aristas. Una arista representa una conexión entre dos vértices [5]. En el contexto de procesamiento de señales, cada nodo V_i en un grafo tiene un valor escalar asociado. Si se considera cada muestra como un instante en el tiempo, cada nodo puede ir variando su valor como si se tratara de una señal de 1 dimensión en el procesamiento de señales (DSP).

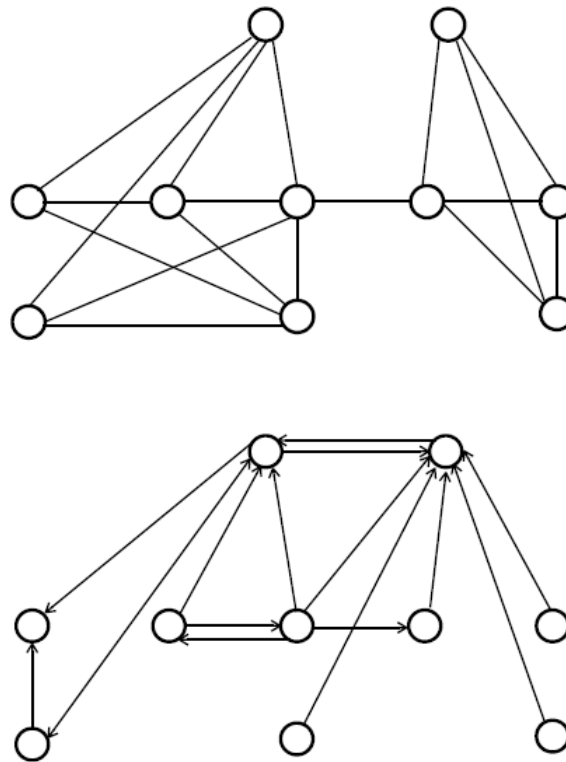


Figura 2.13: Ejemplo grafo no dirigido (superior) y grafo dirigido (inferior). [5]

Hay distintos tipos de grafos, encontrándose los grafos dirigidos y no dirigidos (ver Figura 2.13). Para el contexto de EEG, se considerarán grafos no dirigidos, en lo que se puede ir de un nodo A a un nodo B y vice versa, a diferencia de los grafos dirigidos en que puede existir una conexión del nodo A a B, pero no B hacia A. También, hay grafos en donde a

cada arista se le asigna un valor que representa un tipo de coste, peso o distancia, y que permite modelar de forma cuantitativa la relación entre dos vértices. Esto es representado por la matriz de adyacencia $\mathcal{W}_{N \times N}$, que al estar trabajando con un grafo no dirigido, es simétrica. Este tipo de grafo es denominado *no dirigido y ponderado*.

Una vez construido el grafo se puede comenzar a hacer uso de GSP, donde se define la matriz Laplaciana $L = \mathcal{D} - \mathcal{W}$, siendo \mathcal{D} una matriz diagonal que contiene la información sobre el grado de cada nodo en el grafo, es decir, el número de aristas conectadas a ese nodo. Debido a que L es una matriz simétrica real, tiene un conjunto de vectores propios ortonormales denotados por $\{u\}_{=0,1,\dots,N-1}$. Estos vectores propios tienen asociados valores propios reales y no negativos denotados por $\{\lambda\}_{=0,1,\dots,N-1}$ tal que $\mathcal{L}u = \lambda u$ para $= 0, 1, \dots, N - 1$. Así, los valores propios del Laplaciano son ordenados como $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{N-1} := \lambda_{max}$ y se define el conjunto $\sigma(L) = \{\lambda_0, \lambda_1, \dots, \lambda_{N-1}\}$ como el espectro de la matriz Laplaciana. Estas propiedades permiten una transformada de Fourier en el dominio de un grafo y los vectores propios proporcionan una noción de frecuencia. Así, mayores valores propios son asociados a mayores frecuencias.

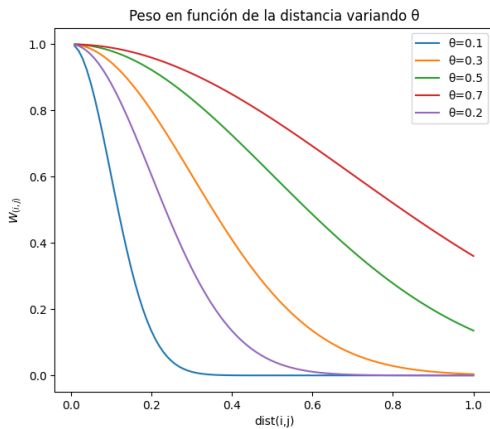
2.4.1. Construcción de grafos basados en EEGs

Para la construcción de grafos a partir de los datos de EEG, se modela cada electrodo como un nodo y las conexiones (o edges) como pesos ponderados. En este punto solo es relevante la información espacial del EEG, es decir, la ubicación de los electrodos. En el grafo, definido por $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ se construye con el método *threshold Gaussian Kernel*[13]. De esta manera, el peso entre 2 vértices i, j estará dado por 2.1,

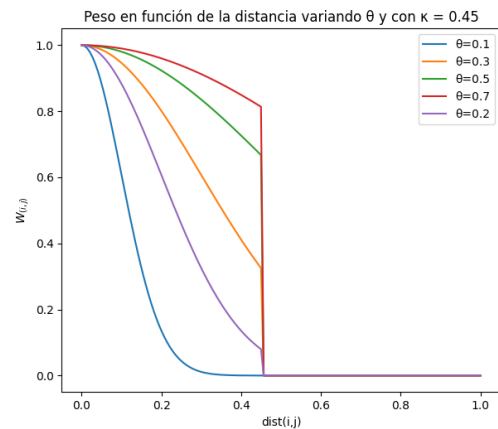
$$W_{i,j} = \begin{cases} \exp\left(-\frac{[dist(i,j)]^2}{2\theta^2}\right) & \text{si } dist(i,j) \leq \kappa \\ 0 & \text{en otro caso} \end{cases} \quad (2.1)$$

donde $dist(i, j)$ es la distancia física o euclidiana entre los vértices i y j , κ es el umbral a utilizar y θ es un parámetro que controla el ancho del kernel, es decir, la desviación estándar.

En la Figura 2.14a se pueden apreciar cómo varía el peso W en función de la distancia para distintos valores de θ (sin considerar el valor de κ). En la Figura 2.14b se puede apreciar el efecto de parámetro κ . Este método de construcción implica que mientras más cercano entre sí están dos nodos i y j , mayor será su conexión y relación, y que se ve delimitado por un umbral. Cuando el peso entre dos nodos es cero, quiere decir que estos no se encuentran conectados.



(a) Peso en función de distancia sin umbral



(b) Peso en función de distancia con umbral

2.5. Técnicas de selección y reducción de nodos

En el procesamiento de señales clásico existen distintas técnicas como el filtrado, muestreo, transformadas, modulaciones, etc. Estas técnicas son bastante intuitivas, debido a que se está acostumbrado a la representación temporal de las señales. Por ejemplo el filtrado se puede utilizar para la atenuación de ruido, tratando de borrar información que entorpece el análisis de una señal. Otro ejemplo es el muestreo, donde el uso de menos muestras puede agilizar el procesamiento, usar menos almacenamiento y hasta reducir el uso de energía en un dispositivo [14]

Todos estos conceptos comienzan a ser más abstractos o complejos de entender cuando se pasa a trabajar en el dominio de los grafos. Se pierde la estructura regular que provee el tiempo. Entonces, ¿qué significa en el procesamiento de señales en grafos usar técnicas de muestreo? En DSP clásico, se puede entender simplemente como usar menos datos, y en un caso concreto, tomar la mitad de las muestras eligiendo una por medio. ¿Esto significa que en GSP se debe utilizar un nodo sí y otro no?, ¿cómo es posible si un grafo es una estructura irregular?.

Como el objetivo de este trabajo es reducir el total de electrodos utilizados en un EEG para tareas BCI, se considera cada electrodo como un nodo, y se modela todo el conjunto como un grafo. Para esto, a continuación se presentan tres algoritmos de GSP que tienen como objetivo encontrar un subconjunto de nodos dentro del grafo original.

2.5.1. Algoritmo 1

El algoritmo 1 [15] propuesto busca encontrar el conjunto de muestreo más pequeño posible que garantice la recuperación única de una señal de grafo con un ancho de banda específico. En lugar de evaluar todas las combinaciones posibles, utiliza heurística. Comienza con un

conjunto vacío y añade un nodo a la vez, donde el objetivo es maximizar el incremento en la estimación de la frecuencia de corte.

En el contexto de señales biológicas de alta densidad, este enfoque asegura que los canales seleccionados retienen la estructura espectral de la señal original. Al detener la iteración exactamente cuando la frecuencia de corte del conjunto alcanza el ancho de banda objetivo ω_c , se garantiza un muestreo eficiente sin pérdida crítica de información espacial. La desventaja de este algoritmo, es que no permite escoger la cantidad de electrodos, por lo que sus parámetros deben variar para ajustarse a la solución deseada.

Se busca el conjunto de nodos más pequeño posible que garantice una recuperación única para una señal de un ancho de banda determinado.

Algorithm 1 Greedy heuristic for estimating S_{opt}

- 1: **Input:** $G = V, E, \mathcal{L}$, bandwidth ω_c , some $k \in \mathbf{Z}^+$
 - 2: **Initialize:** $\mathcal{S} = \emptyset, \omega = 0$
 - 3: **while** $\omega \leq \omega_c$ **do**
 - 4: For \mathcal{S} , compute cut-off estimate $\Omega_k(\mathcal{S})$ and corresponding smoothest signal $\phi_k^* \in L_2(\mathcal{S}^c)$ using proposition 1.
 - 5: $\omega \leftarrow \Omega_K(\mathcal{S}), v \leftarrow \operatorname{argmax}_i (\phi_k^*(i))^2$
 - 6: $\mathcal{S} \leftarrow \mathcal{S} \cup v$
 - 7: **end while**
 - 8: $\mathcal{S}_{est} \leftarrow \mathcal{S}$
-

El algoritmo 1 recibe como parámetros de entrada la estructura del grafo \mathcal{G} , su matriz Laplaciana normalizada \mathcal{L} , la frecuencia de corte objetivo ω_c y un parámetro entero k , que define el orden de la aproximación de los valores propios. En la salida se entrega un subconjunto de nodos S_{est} .

En la matemática del algoritmo se considera un grafo $\mathcal{G} = \{\mathcal{V}, E\}$ con un conjunto de nodos \mathcal{V} , aristas E , matriz de adyacencia (o pesos) W y matriz diagonal D . Se utiliza la matriz Laplaciana normalizada $\mathcal{L} = D^{-1/2} L D^{-1/2}$, siendo L la matriz Laplaciana. \mathcal{L} es una matriz semidefinida positiva simétrica y tiene un conjunto de valores propios reales $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N \leq 2$ y su correspondiente conjunto ortogonal de vectores propios $U = \{u_1, u_2, \dots, u_N\}$. Como se mencionó en 2.4, los valores propios de \mathcal{L} pueden ser interpretados como frecuencias.

Dada una señal de grafo f y g de un ancho de banda ω_c , se busca el conjunto S_{opt} más pequeño de tal manera que la señal es únicamente representada por $f(S_{opt})$. Sea K_c el número de valores propios de \mathcal{L} bajo ω_c , por consideraciones de dimensionalidad $|S_{opt}| \geq K_c$. El espacio que agrupa todas las señales limitadas por esta banda ω_c se conoce como espacio Paley-Wiener, denotado como $PW_{\omega_c}(G)$. A su vez, se define el concepto de conjunto de unicidad (uniqueness set) como un subconjunto de nodos $S \subset \mathcal{V}$ para un espacio $PW_w(G)$, si para dos señales cualesquiera de $PW_w(G)$, el hecho de que coincidan en S implica que

coinciden en \mathcal{V} , es decir, $\forall f, g \in PW_w(G), f(S) = g(S) \Rightarrow f = g$.

La estimación de la frecuencia de corte de orden k para un subconjunto de nodos S (con S^c como su complemento) se define como $\Omega_k(S)$. Esta estimación se calcula obteniendo el valor propio más pequeño de la matriz Laplaciana normalizada reducida $(\mathcal{L}^k)_{S^c}$, siendo siempre menor que $\omega_c(\mathcal{S})$. Si $\psi_{1,k}$ es su correspondiente vector propio, este representa la señal más suave posible, denotada como ψ_k^* , en el espacio de muestreo $(\mathcal{L})_{S^c}$ y el algoritmo selecciona el nodo v donde esta señal concentra su máxima energía:

$$v = \arg \max_i (\phi_k^*(i))^2 \quad (2.2)$$

Como $S_{est} \in PW_{w_c}(G)$, teniendo las señales de los nodos S_{est} es posible reconstruir matemáticamente los datos de todos los nodos en \mathcal{V} . Al ser un algoritmo que se basa en la concentración de energía de la señal más suave del grafo, este método asegura que en el contexto de EEG, se puedan retener los electrodos que aportan la mayor cantidad de información espacial, optimizando la reducción de la dimensionalidad

2.5.2. Algoritmo 2

Algorithm 2 Sampling algorithm

- 1: **Input:** Sampling set size n , first n eigenvectors of L , i.e. Q_n
 - 2: **Output:** Minimum uniqueness set \mathcal{S}_{opt}
 - 3: **Initialize:** $\mathcal{S} = \emptyset$
 - 4: $\mathcal{S} \leftarrow \mathcal{V}_i$, where i is the index of any nonzero element of first eigenvector q_1
 - 5: **for** $m = 2$ **to** n **do**
 - 6: create matrix $Q_m(\mathcal{S})$
 - 7: compute $x = \text{null}(Q_m(\mathcal{S}))$
 - 8: compute $b = Q_m(\mathcal{S}^c)x$
 - 9: $i \leftarrow \arg \max_i |b(i)|$
 - 10: $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}^c(i)$
 - 11: **end for**
 - 12: $\mathcal{S}_{opt} \leftarrow \mathcal{S}$
-

El algoritmo 2 [16] propone encontrar el conjunto de nodos más pequeño que garantice una reconstrucción de cualquier señal de banda limitada en un grafo.

Se buscan iterativamente los nodos que llevan a un mínimo error de reconstrucción, minimizando el valor propio máximo de la matriz de covarianza de error. El algoritmo requiere como entrada el tamaño n deseado del conjunto de muestreo y los primeros n vectores propios de L , Q_n . En la salida se obtiene el conjunto de nodos \mathcal{S}_{opt} .

Para una frecuencia límite de banda λ_n se busca el conjunto mínimo de unicidad \mathcal{S}_{opt} de tamaño n para el espacio $PW_{\lambda_n}(G)$ que pueda garantizar la reconstrucción de cualquier señal $f \in PW_{\lambda_n}(G)$.

El algoritmo busca n filas linealmente independientes de la matriz compuesta de los primeros n vectores propios de L . Se puede obtener de forma iterativa un conjunto mínimo de unicidad S de cualquier tamaño añadiendo una muestra a un conjunto mínimo de unicidad calculado anteriormente.

En cada iteración del algoritmo se evalúa una submatriz conformada por los nodos ya seleccionados, $Q_m(S)$ y se obtiene un vector x que representa el espacio nulo de aquella submatriz, que es único y ortogonal a todas las filas de la submatriz $Q_m(S)$. Luego se realiza una proyección de los nodos que no han sido seleccionados (S^c) sobre el vector x y se selecciona el nodo que maximice el valor absoluto de aquella proyección. Para el análisis de señales EEG en BCI, este algoritmo presenta la ventaja de no requiere de hiperparámetros y mantiene un equilibrio entre la complejidad computacional y la reconstrucción generada. Pese a que es un algoritmo que tiene iteraciones, solo calcula la matriz Laplaciana al comienzo, evitando cálculos repetidos e innecesarios. lo que puede ser costoso computacionalmente.

Este algoritmo evita operaciones complejas, a diferencia del algoritmo 1, y no requiere que el usuario seleccione distintos parámetros.

El algoritmo se basa en asegurar que el conjunto de muestreo sea siempre un conjunto de unicidad mínimo". Utilizando la matriz Q_n , compuesta por los primeros n vectores propios del Laplaciano del grafo, el algoritmo procede iterativamente. En cada iteración m , calcula el espacio nulo (nullspace) de la submatriz de los nodos ya seleccionados, obteniendo un vector x . Luego, calcula el producto $b = Q_{m+1}(S_m^c)x$ para los nodos candidatos y selecciona aquel que presenta el valor absoluto máximo en b , lo que garantiza independencia lineal. Para aplicaciones de BCI, esta eficiencia computacional es crítica, ya que permite identificar rápidamente los canales más informativos sin incurrir en el costo de descomposiciones de valores singulares (SVD) repetitivas en grafos de alta densidad.

2.5.3. Algoritmo 3

El algoritmo 3 [17] se basa en el operador de localización que considera tanto las localizaciones del dominio de vértice como las de dominio espectral.

Este algoritmo ataca el problema de selección de nodos utilizando el operador de localización. A diferencia de los enfoques anteriores, no tiene la necesidad de calcular la descomposición en valores propios de la matriz Laplaciana L , lo que reduce el tiempo de ejecución. Se necesita la matriz Laplaciana normalizada \mathcal{L} , el número de nodos F para seleccionar, el ancho de banda de la señal de grafo, y un kernel de filtrado $g(\lambda)$. En la salida se obtiene el conjunto de nodos S^* . Se tiene el operador de localización $T_{g,i}$ que concentra la energía de la señal alrededor de un nodo i . En forma matricial se calcula como

$$T = [T_{g,0}T_{g,1}\dots T_{g,N-1}] = \sqrt{N}Ug(\Lambda)U^H \quad (2.3)$$

donde $\Lambda = \text{diag}(\lambda_0, \dots, \lambda_{N-1})$ El objetivo es seleccionar los nodos que maximicen el *área de cobertura* del operador de localización, considerando incluso el caso en nodos se superpongan (por ejemplo $T_{g,i}$ y $T_{g,j}$ con $i \neq j$). Para esto, en cada iteración se busca un nodo y^* que maximice la función:

$$y^* = \arg \max_{y \in \mathcal{S}_m^c} [|T|w](y) \quad (2.4)$$

donde el vector de pesos w se define mediante una función rampa R que anula la ganancia en áreas ya cubiertas

$$w = R \left(\eta 1_{N \times 1} - \sum_{j \in \mathcal{S}_m} |T_{g,j}| \right) \quad (2.5)$$

Así, durante cada ciclo de las F iteraciones, se actualiza el vector de pesos w y evalúa $[|T|w](y)$ para cada uno de los nodos candidatos, agregando al conjunto S el vértice que maximice y^* .

En el diseño de BCI, el uso de EEG con mallas de electrodos de alta densidad puede aumentar la complejidad computacional, por lo que este algoritmo permite identificar los canales más informativos sin ninguna complicación.

2.6. Extracción de características

Una vez reducida la dimensionalidad espacial de los datos de EEG mediante los algoritmos de selección y reducción de electrodos, las señales siguen teniendo una complejidad al momento de analizarlas. Muchas de ellas pueden ser redundantes o poseer características irrelevantes, afectando de forma negativa a la clasificación [18]. Especialmente en el contexto de tareas cognitivas o motoras, es necesario traducir los datos actuales a un subespacio donde la diferencia entre las tareas sea más evidente, y es donde destaca la extracción de características. Básicamente se busca transformar los datos que se tienen para que el clasificador obtenga mejores resultados, aislando los distintos patrones. Para esto se utilizará el algoritmo *Common Spatial Patterns*.

2.6.1. Common Spatial Patterns (CSP)

El algoritmo de Common Spatial Patterns [6] es un método que usa filtros espaciales para maximizar la discriminabilidad de dos clases, y ha sido ampliamente usado en sistemas de BCI para imágenes motoras. Su objetivo es encontrar filtros espaciales, que corresponden a combinaciones lineales de los electrodos, que puedan maximizar la varianza de la señal para una clase, y al mismo tiempo minimizar la varianza para la otra. En el contexto de EEG, la varianza de la señal se relaciona con la energía o potencia de la actividad cerebral

de una banda de frecuencia específica.

El algoritmo supone que la banda de frecuencia y la ventana de tiempo son conocidas.

Sean Σ^{C_1} y $\Sigma^{C_2} \in \mathbb{R}^{N \times N}$ las estimaciones de las matrices de covarianza para la señal de EEG filtrada para unas clases C_1 y C_2 , respectivamente. Entonces, el método de CSP se basa en la diagonalización simultánea de sus dos matrices de covarianza

$$W^T \Sigma^{C_1} W = \Lambda^{C_1} \quad (2.6)$$

$$W^T \Sigma^{C_2} W = \Lambda^{C_2} \quad (2.7)$$

donde W se determina de tal manera que $\Lambda^{C_1} + \Lambda^{C_2} = I$, donde I representa la matriz identidad. 2.7 se resuelve como un problema de valores propios generalizados definido por:

$$\Sigma^{C_1} w = \lambda \Sigma^{C_2} w \quad (2.8)$$

Los valores propios, λ , asociados a la clase 1 son cercanos a 0, y los valores propios cercanos a 1 son asociados a la clase 2. Así, los primeros vectores propios w entregarán las mayores varianzas para una clase, mientras que los últimos vectores propios entregarán las mayores varianzas para la clase opuesta. Se escoge una cierta cantidad de vectores propios, que representa la cantidad de *filtros espaciales* que se aplican para discriminar ambas clases. La señal filtrada (o mejor dicho proyectada) se calcula como $X_{CSP} = W^T X$, siendo X , la señal preprocesada.

En las Figuras 2.15 y 2.16 se puede apreciar la diferencia en la covarianza de dos clases antes y después de aplicar CSP.

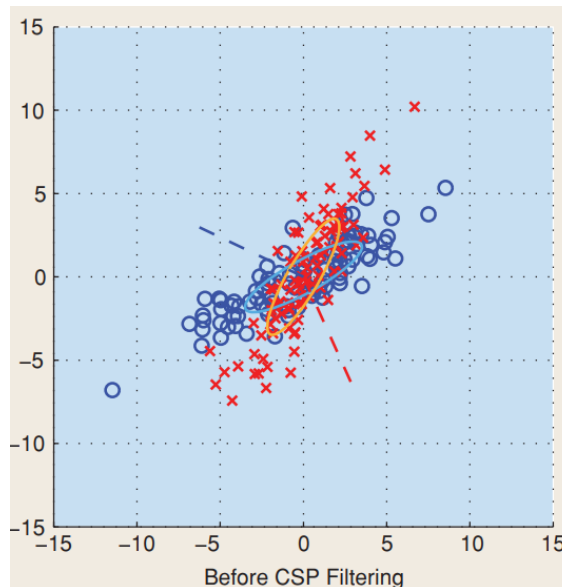


Figura 2.15: Covarianzas de dos clases antes de aplicar CSP [6]

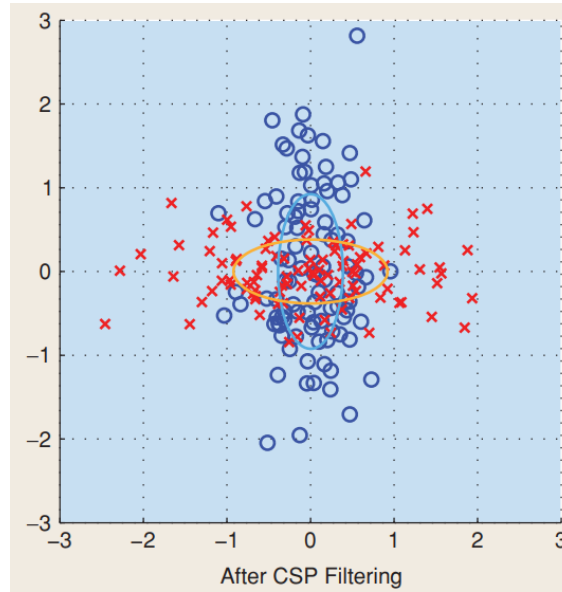


Figura 2.16: Covarianzas de dos clases luego de aplicar CSP [6]

2.7. Clasificación

2.7.1. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA), o Normal Discriminant Analysis, es un método de clasificación de aprendizaje supervisado y reducción de dimensionalidad, utilizado en el diseño de sistemas de BCI debido a su baja complejidad computacional y alta robustez. Busca encontrar una combinación lineal de características que mejor separe las clases. Esto lo hace maximizando la diferencia (o distancia) entre las clases y minimizando la dispersión dentro de cada clase [7].

Suponiendo que se tiene un vector de entrada D -dimensional y se proyecta en una dimensión usando

$$y = w^T x \quad (2.9)$$

Se supone además un umbral en y , clasificando $y \geq -w_o$ como clase C_1 y como C_2 en caso contrario. Se considera un problema con dos clases donde hay N_1 de la clase C_1 y N_2 puntos de la clase C_2 , tal que su media está dada por los vectores

$$\mu_1 = \frac{1}{N_1} \sum_{n \in C_1} x_n \quad (2.10)$$

$$\mu_2 = \frac{1}{N_2} \sum_{n \in C_2} x_n \quad (2.11)$$

Así, la separación entre clases, cuando se proyecta en w corresponde a la separación de la

media de las clases proyectadas. Esto permite escoger w para maximizar

$$\hat{\mu}_2 - \hat{\mu}_1 = w^T(\mu_2 - \mu_1) \quad (2.12)$$

donde

$$\hat{\mu}_k = w^T \mu_k \quad (2.13)$$

es la media de la proyección de los datos de la clase C_k .

Se propone el criterio de Fisher para maximizar una función que de una gran separación entre las clases proyectadas al mismo tiempo que cada clase tiene una varianza pequeña, minimizando el solapamiento entre las clases. La varianza de cada clase C_k está dada por

$$s_k^2 = \sum_{n \in C_k} (y_n - \hat{\mu}_k)^2 \quad (2.14)$$

donde $y_n = w^T x_n$. Se puede definir la varianza total dentro de cada clase para todos los datos como $s_1^2 + s_2^2$. Así, el criterio de Fisher, definido como la relación entre la varianza de las clases con la varianza dentro de las clases es

$$J(w) = \frac{(\hat{\mu}_2 - \hat{\mu}_1)^2}{s_1^2 + s_2^2} \quad (2.15)$$

Dejando de forma explícita la dependencia del vector w , el criterio de Fisher se puede escribir como

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \quad (2.16)$$

donde S_B es la matriz de covarianza entre clases, dada por

$$S_B = (\mu_2 - \mu_1)(\mu_2 - \mu_1)^T \quad (2.17)$$

y S_W es la matriz de covarianza total dentro de cada clase, dada por

$$S_W = \sum_{n \in C_1} (x_n - \mu_1)(x_n - \mu_1)^T + \sum_{n \in C_2} (x_n - \mu_2)(x_n - \mu_2)^T \quad (2.18)$$

Diferenciando 2.16 respecto a w , se encuentra que $J(w)$ se maximiza cuando

$$(w^T S_B w) S_W w = (w^T S_W w) S_B w \quad (2.19)$$

De 2.17 se nota que $S_B w$ está siempre en la dirección de $(\mu_2 - \mu_1)$. Como no interesa la magnitud de w , se ignoran los factores escalares $(w^T S_B w)$ y $(w^T S_W w)$. Ahora, multiplicando

ambos lados de 2.19 por S_W^{-1} se obtiene

$$w \propto S_W^{-1}(\mu_2 - \mu_1) \quad (2.20)$$

El resultado 2.20 es conocido como *discriminante lineal de Fisher*. En la Figura 2.17 se pueden apreciar dos clases en que sus proyecciones se solapan (izquierda) y en que una proyección basada en el discriminante lineal de Fisher mejora la separación de las clases (derecha).

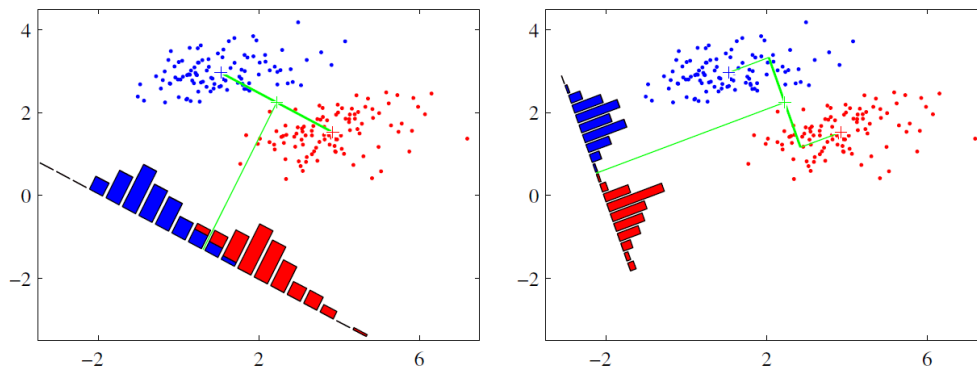


Figura 2.17: Separación entre las proyecciones de dos clases sin el uso del criterio de Fisher y su mejora utilizando el criterio de Fisher.[7]

El método de Fisher es la base de LDA, pues proporciona la dirección de proyección óptima, pero no proporciona un valor exacto para el umbral entre las dos clases. Para esto, LDA supone una distribución normal de las clases y la misma covarianza para ambas clases. Para definir el umbral de clasificación, se asume que los datos proyectados de cada clase siguen una distribución normal, lo que permite calcular qué tan probable es que un nuevo dato pertenezca a un grupo o al otro y el umbral de decisión corresponde al punto donde ambas distribuciones se cruzan. En la Figura 2.18 se pueden apreciar las distribuciones gaussianas para dos clases posterior a su proyección.

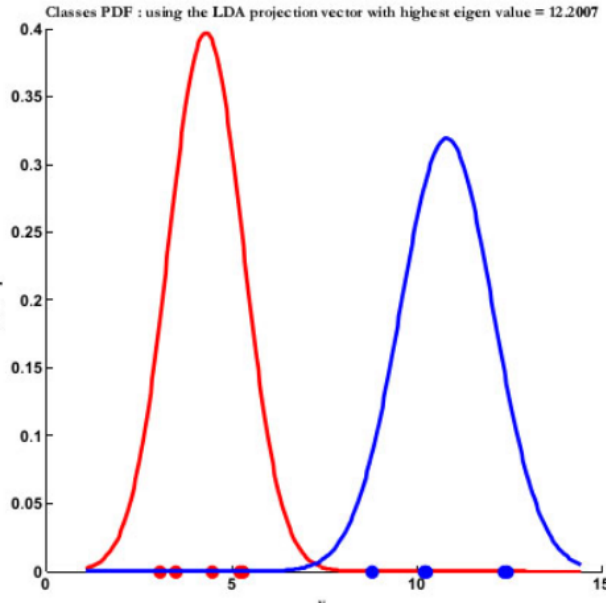


Figura 2.18: Distribuciones Gaussianas de dos clases proyectadas

2.7.2. Validación de modelos

El rendimiento de una clasificación es evaluado comparando los valores obtenidos con los valores esperados. Se mide su **accuracy** (exactitud), que corresponde al porcentaje de resultados totales correctos, dado por la ecuación 2.21

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (2.21)$$

donde

- True Positive (TP): El modelo predijo 1 y la etiqueta real era 1.
- True Positive (TN): El modelo predijo 0 y la etiqueta real era 0.
- False Positive (FP): El modelo predijo 1 y la etiqueta real era 0.
- False Negative (FN): El modelo predijo 0 y la etiqueta real era 1.

La elección de las métricas depende mucho del contexto. Accuracy es una de las más utilizadas para evaluar un modelo, pues provee una interpretación directa de la probabilidad de que el clasificador tome la decisión correcta. Esto último depende de que el costo de los errores (FP y FN) es simétrico y que las clases a predecir son balanceadas (cantidad similar de datos).

En escenarios donde, si bien las clases están equilibradas, la cantidad de datos con los que se trabaja no es mucha y no es posible incrementar su tamaño (*data augmentation*),

por lo que pueden aparecer problemas en la clasificación. El objetivo de cualquier modelo de aprendizaje automático no es memorizar los datos, sino descubrir patrones. A esto se le llama *generalizar*. Que un modelo no generalice bien puede traer dos problemas: El sobreajuste (*overfitting*), que consiste en que el modelo memoriza los datos y no aprende sus patrones; y el subajuste (*underfitting*), que se produce cuando un modelo es demasiado simple que no puede capturar los patrones en los datos. En la Figura 2.19 se puede apreciar como cambias las predicciones de un modelo cuando tiene *underfitting*, cuando generaliza bien, y cuando tiene *overfitting*[19].

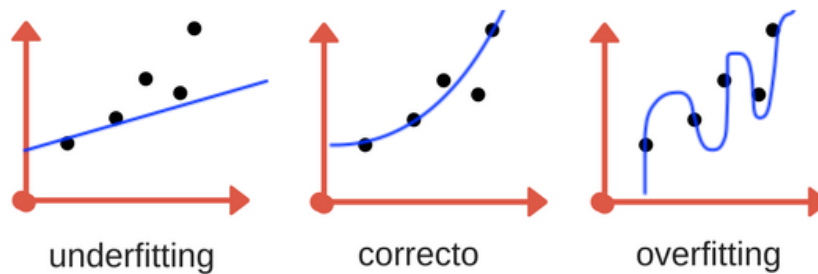


Figura 2.19: Ejemplo modelo con *underfitting* (izquierda), buena generalización (centro) y *overfitting* (derecha).

Para compensar la poca cantidad de datos, se utiliza una técnica llamada *Monte Carlo cross validation* (validación cruzada de Monte Carlo)[20]. Este método consiste en particionar de forma aleatoria el conjunto de datos en un subconjunto de entrenamiento y uno de prueba en cada una de sus k iteraciones (o experimentos), donde el tamaño de los conjuntos es un tamaño fijo y determinado. Cada iteración obtiene una *accuracy*, y su promedio corresponde al *accuracy* del modelo.

CAPÍTULO



METODOLOGÍA E IMPLEMENTACIÓN

En este capítulo se describen las metodologías empleadas para el procesamiento y análisis de las señales de EEG. Se habla sobre los datos utilizados, su preprocesamiento, la construcción de grafos a partir de los datos, los algoritmos de selección de electrodos y el flujo de trabajo con el que será evaluado el rendimiento de los algoritmos. Este trabajo fue desarrollado en python, en su versión 3.12.6. Las principales librerías utilizadas son numpy, MNE [21], scikit-learn, [22], ee grasp y matplotlib y el código se puede encontrar en https://github.com/sebastianespinozat/gsp_eeg_reduccion.

Antes de detallar las configuraciones y parámetros de cada etapa, en la Figura 3.1 se presenta el diagrama de flujo general del trabajo realizado. Se parte por la adquisición de los datos, que por un lado son preprocesados, donde se realizan filtrados y eliminan artefactos, y en paralelo se crean los grafos que permite realizar la selección de subconjuntos de nodos. Posteriormente los datos, que ahora son épocas son entregados a un clasificador que evalúa su rendimiento en distintas tareas de BCI, repitiendo el proceso con todos los subconjuntos de nodos obtenidos. Finalmente se realiza un análisis con los datos obtenidos. Este diagrama sirve como una guía para entender la estructura, orden y subsecciones de este capítulo.

3.1. Datasets utilizados

Para el desarrollo de este trabajo se utilizan 4 datasets (conjuntos de datos). Para un mayor entendimiento, se referirá a estos como dataset A, B, C y D. Todos los dataset utilizados corresponden a registros de EEG de sujetos sanos en tareas de BCI, donde varía el número de sujetos, el número de electrodos utilizados y las tareas realizadas por los sujetos.

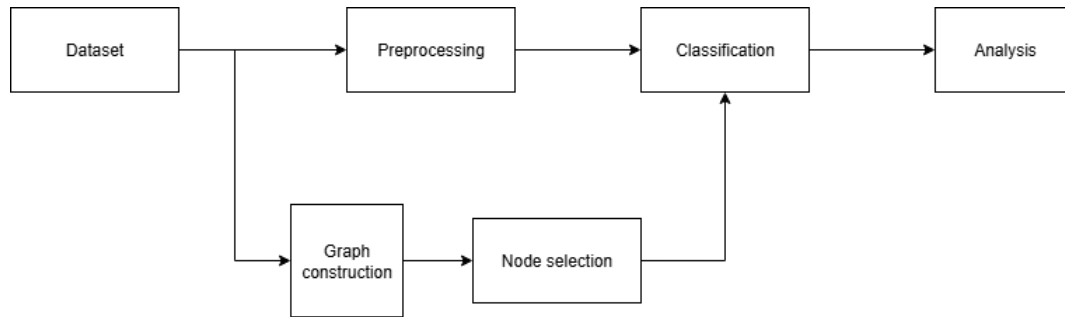


Figura 3.1: Diagrama de flujo general de la metodología implementada para la Interfaz Cerebro-Computador.

3.1.1. Dataset A: EEG Motor Movement/Imagery Dataset[1]

Este dataset corresponde a registros de 109 sujetos de EEG con 64 electrodos según el sistema 10-10. Por cada sujeto se realizaron 14 experimentos, donde se realizaban distintas tareas luego de que en una pantalla apareciera una señal: La tarea 1 corresponde a abrir y cerrar el puño izquierdo o derecho. La tarea 2 corresponde a imaginar abrir y cerrar el puño izquierdo. La tarea 3 corresponde a abrir y cerrar ambos puños o ambos pies. La tarea 4 corresponde a imaginar abrir y cerrar ambos puños o pies. En los experimentos 1 y 2 no se realiza ninguna actividad y solo se mantienen los ojos abiertos (experimento 1) y los ojos cerrados (experimento 2). El resto de tareas realizadas por experimento se muestra a continuación:

- Tarea 1: Experimentos 3, 7 y 11.
- Tarea 2: Experimentos 4, 8 y 12.
- Tarea 3: Experimentos 5, 9 y 13.
- Tarea 4: Experimentos 6, 10 y 14.

Cada experimento tiene anotaciones las anotaciones **T0**, **T1** o **T2**. **T0** corresponde a reposo. **T1** corresponde al inicio del movimiento (real o imaginado) para el puño izquierdo (experimentos 3, 4, 7, 8, 11 y 12) y para ambos puños (experimentos 5, 6, 9, 10, 13 y 14). **T2** corresponde al inicio del movimiento (real o imaginado) para el puño derecho (experimentos 3, 4, 7, 8, 11 y 12) y para ambos pies (experimentos 5, 6, 9, 10, 13 y 14).

3.1.2. Dataset B: Data set IVa <motor imagery, small training sets[2]

Este dataset corresponde al registro de 5 sujetos sanos de EEG con 118 electrodos según una versión extendida del sistema 10-20. Se proporcionan 280 señales para cada uno de los 5 sujetos (*aa*, *al*, *av*, *aw*, *ay*). Se dan señales de forma visual por 3.5 segundos que

indican 3 acciones que los sujetos deben imaginar realizar: Mano izquierda (L), mano derecha (R), pie derecho (F). Entre señales, había un periodo en el que el sujeto permanecía relajado. La estimulación visual que recibían eran de dos tipos: (1) donde los objetivos se indicaban mediante letras que aparecían detrás de una cruz de fijación (lo que podía inducir movimientos oculares poco correlacionados con el objetivo), y (2) donde un objeto en movimiento aleatorio indicaba los objetivos (lo que inducía movimientos oculares no correlacionados al objetivo). De los sujetos *al* y *aw* se registraron dos sesiones de ambos tipos, mientras que de los demás sujetos se registraron tres sesiones del tipo (2) y una sesión del tipo (1). Para este trabajo se utilizaron las clases **derecha** (R) y **pie** (F).

3.1.3. Dataset C: Data set IVb <motor imagery, uncued classifier application> [2]

Este dataset corresponde al registro de EEG de un sujeto sano con 118 electrodos según una versión extendida del sistema 10-20. Se presentan estimulaciones visuales por 3.5 segundos que indican 3 acciones que el sujeto debe imaginar realizar: Mano izquierda (L), pie derecho (F), lengua (Z). Entre señales, había un periodo en el que el sujeto permanecía relajado. También se tienen más registros en el que no se utilizaba un estímulo visual, pero sí uno auditivo, en el que se tenían las clases **left** (izquierda), **foot** (pie) y **relax** (estado de relajación). Para este trabajo se utilizaron las clases **left** y **foot**.

3.1.4. Dataset D: dataset for simultaneous EEG and NIRS brain-computer interface (BCI) [3]

Este dataset corresponde al registro de EEG de 29 sujetos sanos con 30 electrodos según el sistema 10-5. El EEG también fue utilizado para medir electrooculograma (EoG), electrocardiograma (ECG) y respiración con un piezo basado en un cinturón de respiración. Se tienen dos experimentos: Motor Imagery (A), donde los sujetos debieron imaginar la sensación de abrir y cerrar sus manos (izquierda y derecha), según una instrucción visual, y Mental Arithmetic (B), donde les presentaba una resta entre números de 3 dígitos y 1 dígito (384-8 por ejemplo) en el centro de una pantalla y los sujetos debían memorizar los números mientras la operación era mostrada en pantalla, y cuando esta desaparecía, debían realizar la operación, y en contraste se tenía un estado de relajo en el que los sujetos no realizaban ninguna acción pero si miraban a la pantalla (baseline).

Para el experimento A se tienen las clases **izquierda** y **derecha**, mientras que para el experimento B se tienen las clases **aritmética** y **baseline**.

3.2. Preprocesamiento de datos EEG

3.2.1. Segmentación en épocas

Para la extracción de la actividad cerebral de interés, los registros continuos de EEG fueron segmentados en ventanas temporales denominadas épocas. Cada ventana se encuentra delimitada por t_{min} y t_{max} , usando como referencia $t = 0$ el instante exacto en que se presenta el estímulo visual o auditivo (cue) al sujeto. El tiempo mínimo t_{min} puede ser negativo y representa los segundos previos al evento, lo que puede ser una práctica estándar en casos que se requiera capturar la actividad previa al estímulo y al evento motor. En la Tabla 3.1 se muestran los tiempos utilizados para cada dataset.

Dataset	t_{min} (s)	t_{max} (s)
A	-1.0	4.0
B	0.5	2.5
C	0.5	2.5
D	0.5	3.0

Tabla 3.1: Tiempos mínimos y máximos para segmentación de épocas

3.2.2. Filtrado y eliminación de artefactos mediante ICA

EN la etapa inicial de preprocesamiento, se aplicó un filtro pasabanda IIR Butterworth de tercer orden, con frecuencias de corte entre 0.5 y 45 Hz, con el objetivo de acondicionar las señales de manera óptima para aplicar Análisis de Componentes Independientes (ICA).

Posteriormente, se procedió a remover los artefactos oculares (principalmente parpadeos) mediante ICA para los conjuntos de datos B, C y D. En el caso de los datasets B y C, la identificación y eliminación de componentes que generen ruido se realizó de forma manual mediante la selección de estas fuentes de ruido. El dataset D incluye canales de electrooculografía (EOG), lo que permitió utilizar estas señales de referencia para aislar y eliminar las componentes correspondientes a parpadeos de una manera mucho más precisa y eficiente. Por otro lado, en el dataset A se optó por omitir la eliminación de artefactos mediante ICA debido a que existe una gran variabilidad de las componentes ruidosas entre sujetos y el elevado tamaño de la muestra (más de 100 sujetos) hace que esta tarea manual se vuelva muy ineficiente.

Luego, a todas las épocas se les vuelve a aplicar el mismo filtro pasabanda pero ahora en el rango de 8 a 30 Hz. Tal como se mencionó en los fundamentos teóricos (2.4.1), esta restricción permite aislar las frecuencias contienen actividad relevante para tareas de BCI. En las figuras 3.2 y 3.3 se ilustran 30 componentes independientes extraídos de los datos *raw* (sin preprocesar) del dataset C. Un análisis topográfico revela que los componentes 0,

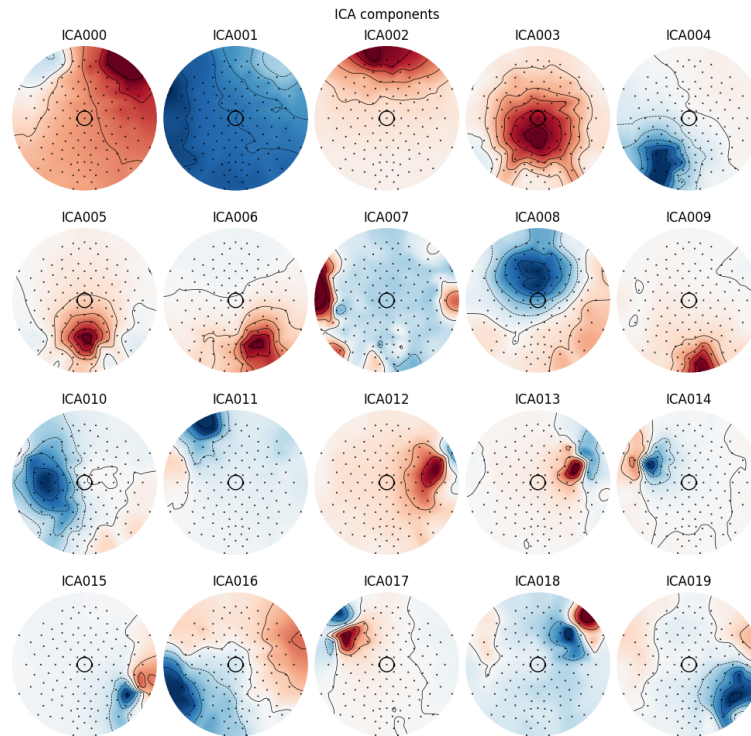


Figura 3.2: Primero componentes ICA en dataset C.

2, 11 registran una fuerte actividad en la zona ocular, de forma aislada y sin actividades que destaquen en otras áreas. Al coincidir con la topología típica del ruido de parpadeo y movimiento ocular, estos los componentes proceden a ser eliminados.

Una mayor cantidad de componentes, aísla de mejor manera las fuentes de la señal, y por lo tanto, los artefactos son más simples de identificar, y más importante aún, no mezclarlos con la actividad neuronal necesaria para BCI. Por esta razón, en los datasets donde se aplica ICA (B, C y D), se realiza una descomposición de 30 componentes. MNE al realizar ICA, por defecto utiliza el algoritmo *fastICA*, cuya implementación está basada en [23]

Una vez el preprocesamiento está completo, el conjunto de datos está listo para ser utilizada en el entrenamiento y clasificación. En las Figuras 3.4a y 3.4b se observa el antes y después en una señal dataset C luego de ser preprocesada. El efecto del preprocesamiento es notable, pues aquellos peaks en 3.4a (señal raw) son artefactos, los cuales desaparecen en la señal preprocesada, logrando un rango de $\pm 40\mu V$, normal para una señal EEG.

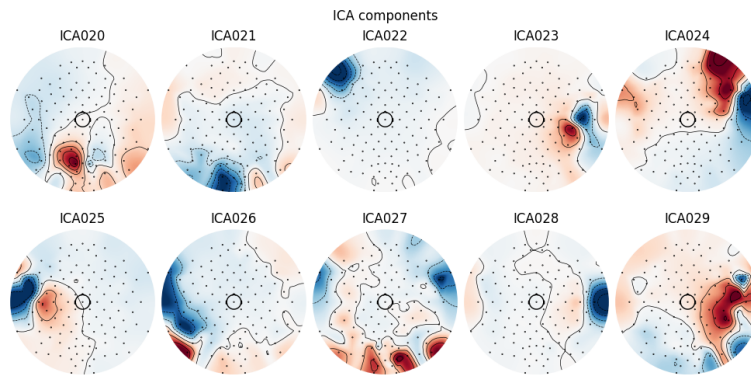
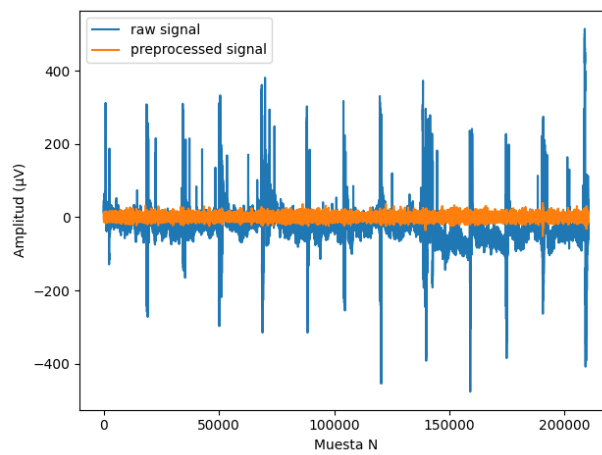
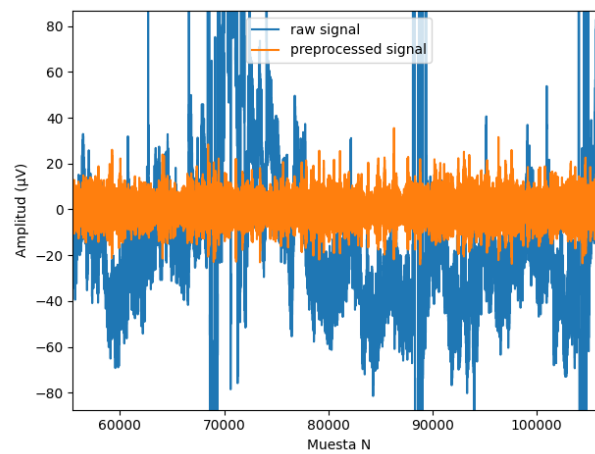


Figura 3.3: Componentes finales ICA en dataset C.



(a) Señal antes y después del preprocesamiento.



(b) Zoom a señal antes y después del preprocesamiento.

3.3. Construcción de grafos

La selección de parámetros en cada dataset fue determinada de manera manual, ajustándolos empíricamente con el objetivo de generar distintas configuraciones de grafos lo suficientemente distintos entre sí. Siendo específico, se busca variar la cantidad de aristas (o edges) en los grafos, permitiendo analizar el efecto de la conectividad sobre el desempeño de los algoritmos de selección de electrodos.

Cada grafo $\mathcal{G} = (\mathcal{V}, E)$ fue construido considerando los canales de EEG como el conjunto de vértices \mathcal{V} . La conectividad entre cada par de electrodos, que representa los pesos del grafo se calculó usando un Kernel Gaussiano basado en la distancia espacial entre los sensores, según 2.1. Es necesario destacar que los datasets B y C entregan las coordenadas de sus electrodos en 2 dimensiones.

Grafos dataset A

En la Figura 3.5 se pueden apreciar los grafos generados a partir de los parámetros utilizados en la Tabla 3.2 para el dataset A.

Grafo	κ	θ	\mathcal{V}	\mathcal{E}
1	0.2	0.1	64	117
2	0.25	0.1	64	187
3	0.27	0.1	64	210

Tabla 3.2: Grafos generados para dataset A.

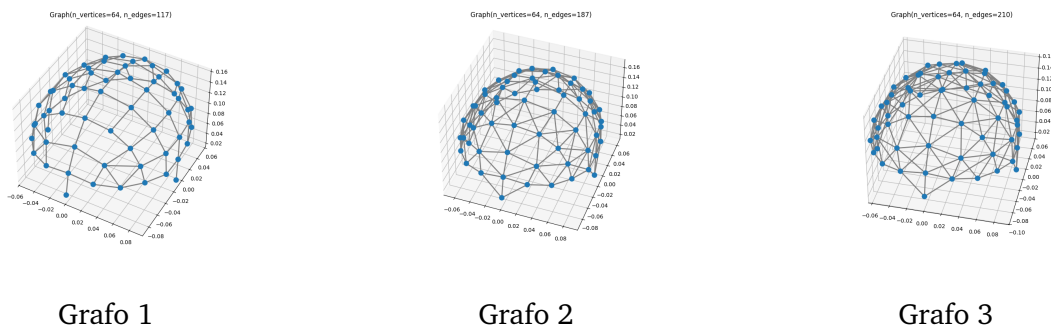


Figura 3.5: Topologías de los grafos generados para el dataset A al variar empíricamente sus parámetros de conectividad.

Grafos dataset B y C

En las Figuras 3.6 se pueden apreciar los grafos generados a partir de los parámetros utilizados en la Tabla 3.3 para los datasets B y C.

Grafo	κ	θ	\mathcal{V}	\mathcal{E}
1	0.1	0.1	118	278
2	0.15	0.1	118	596
3	0.2	0.1	118	1053

Tabla 3.3: Grafos generados para dataset B y C.

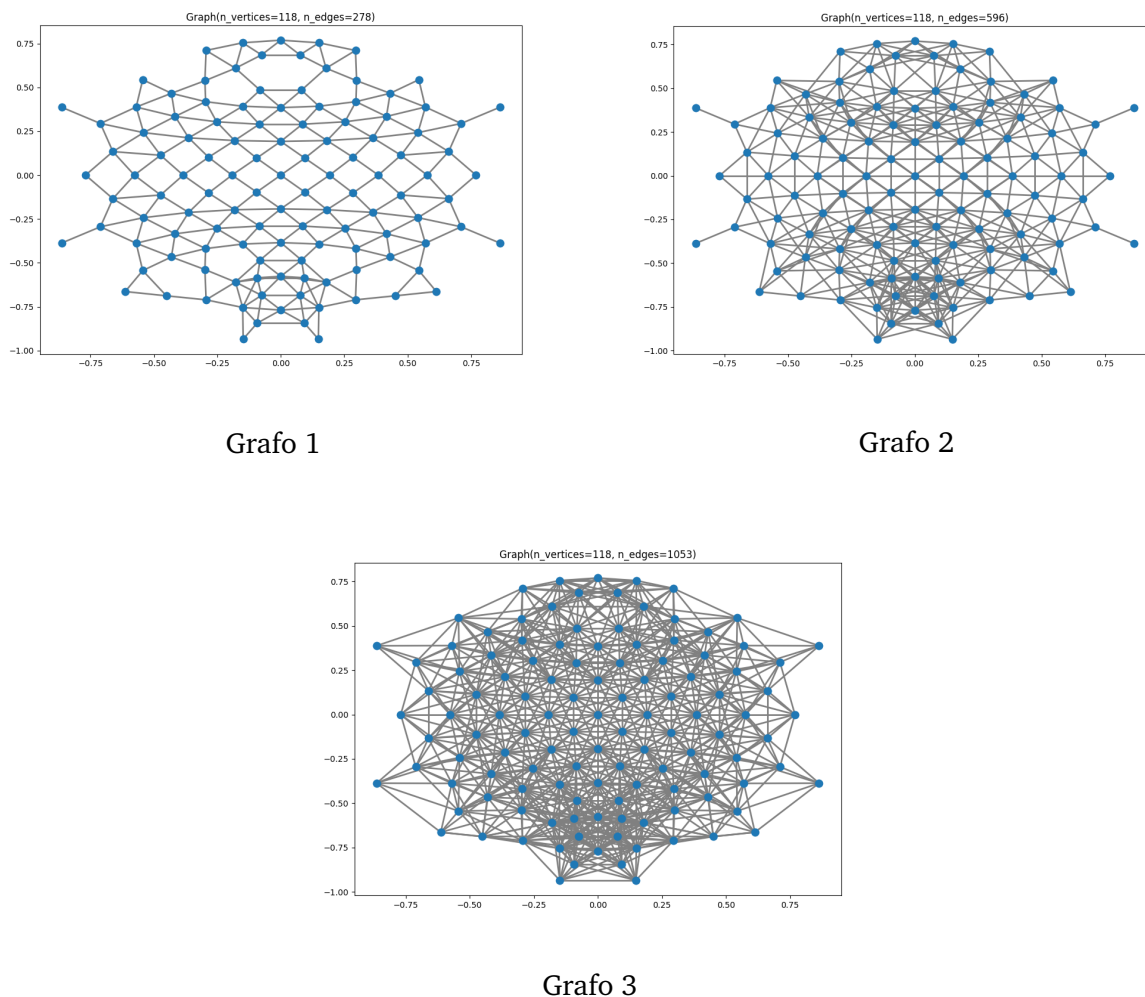


Figura 3.6: Topologías de los grafos generados para los datasets B y C.

Grafos dataset D

En la Figura 3.7 se pueden apreciar los grafos generados a partir de los parámetros utilizados en la Tabla 3.4 para el dataset D.

Grafo	κ	θ	\mathcal{V}	\mathcal{E}
1	0.4	0.4	30	89
2	0.45	0.1	30	105
3	0.5	0.1	30	153

Tabla 3.4: Grafos generados para dataset D.

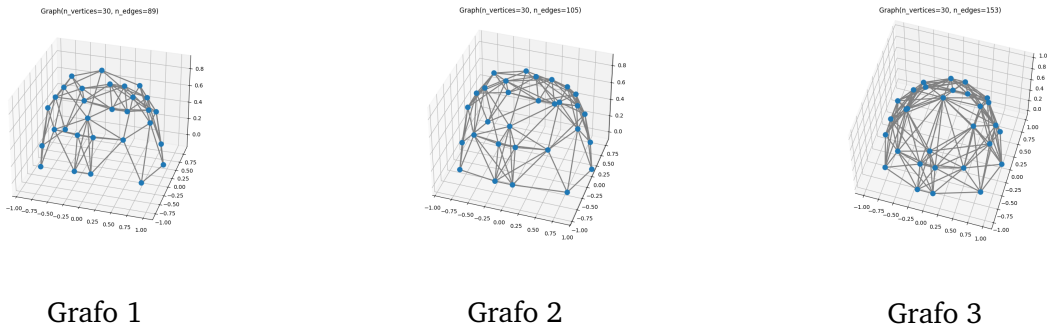


Figura 3.7: Topologías de los grafos generados para el dataset D.

3.4. Aplicación de algoritmos de selección y reducción de electrodos

La etapa de selección de canales se llevó a cabo implementando los tres algoritmos de muestreo de grafos descritos en 2.5. El objetivo principal de esta fase fue reducir la cantidad de electrodos en cada EEG, seleccionando aquellos que brindan la mayor cantidad de información espacial relevante.

Dado el enfoque del trabajo, la sincronización de parámetros para cada algoritmo se realizó de manera empírica, por lo que se obtienen distintos subconjuntos de electrodos lo suficientemente reducidos para garantizar una topología representativa del grafo completo.

- **Algoritmo 1 (Estimación Heurística):** La formulación matemática de este método no permite definir el número de nodos resultantes como un parámetro directo de entrada, ya que el tamaño del conjunto de unicidad depende de la frecuencia de corte objetivo. Por lo tanto, se procedió a variar iterativamente sus parámetros hasta converger en subconjuntos de electrodos lo suficientemente reducidos.
- **Algoritmos 2 y 3:** A diferencia del primer algoritmo, estos métodos sí permiten establecer explícitamente el tamaño del conjunto de muestreo deseado en su inicialización. Para estos algoritmos se mantienen constante sus hiperparámetros, variando solo el número de nodos en la salida.

Esta decisión del ajuste empírico en cada algoritmo permitió generar diversas configuraciones de electrodos. Al contar con grafos de distintas densidades, es posible evaluar el impacto en la reducción de sus electrodos en cada algoritmo, específicamente en tareas de BCI.

3.5. Extracción de características y clasificación

Para la extracción de características de las señales EEG, se utilizó el algoritmo de CSP [6], que calcula filtros espaciales óptimos que maximizan la varianza de la señal filtrada para una clase, mientras minimiza la varianza de la clase opuesta.

Dado que el número óptimo de componentes o filtros espaciales que se retienen en CSP varía según el conjunto de datos, se implementó una fase de calibración para este valor. Para cada dataset se evalúa el rendimiento del clasificador en el escenario en que se tienen todos los electrodos, seleccionando el número de componentes que entregó mayor accuracy. En el caso de datasets donde se analiza más de un sujeto, se escoge $n_components$ donde el promedio de la clasificación para todos los sujetos con dicho $n_components$ fue mayor.

Una vez determinado $n_components$ en cada caso, el parámetro se mantiene constante para las clasificaciones posteriores donde varía el subconjunto de electrodos generados por los algoritmos en 3.4. Esta estrategia garantiza que cualquier variación observada en el rendimiento final de la clasificación sea una consecuencia exclusiva de los electrodos utilizados. Finalmente, las características espaciales extraídas por CSP fueron introducidas a un clasificador de LDA. En la Tabla 3.5 se muestra el número de componentes utilizado en cada dataset.

Dataset	n_components
A	6
B	12
C	16
D	4 (task A) y 9 (task B)

Tabla 3.5: $n_components$ utilizado en cada dataset.

3.5.1. Diseño Experimental y Validación

Para evaluar la capacidad de generalización del modelo y garantizar la robustez de los resultados, se implementa un esquema de validación cruzada de 10 pliegues.

En cada iteración, del conjunto de datos se selecciona un nuevo conjunto de entrenamiento y otro de prueba. Cada conjunto de prueba corresponde al 20% del total de datos. Esta estrategia es fundamental para asegurar que el clasificador no se sesgue hacia alguna clase

en particular durante el entrenamiento, reduciendo las posibilidades de que se generen sobreajustes.

La métrica para cuantificar el desempeño del modelo de clasificación fue la accuracy. Dado que las épocas extraídas de los conjuntos de datos representan clases balanceadas, la accuracy refleja de manera directa la proporción de intenciones motoras clasificadas correctamente en el total de los intentos. El valor final de rendimiento para cada subconjunto de electrodos corresponde al promedio del accuracy obtenida en cada una de las 10 iteraciones. Esto garantiza que las comparaciones entre los distintos algoritmos de selección de nodos se basen en una métrica estable.

CAPÍTULO



RESULTADOS Y ANÁLISIS

4.1. Resultados por dataset

En este capítulo se presentan y analizan los resultados obtenidos al aplicar las distintas configuraciones de selección de electrodos sobre las señales de electroencefalografía. El análisis se estructura evaluando en primer lugar el desempeño referencial del sistema con la totalidad de los electrodos, para luego compararlo con la accuracy obtenida al aplicar los algoritmos de reducción de nodos en grafos. Así, se explorará tanto el impacto matemático como sus implicatorias fisiológicas en tareas de BCI.

4.2. Desempeño Base (Linebase performance) y Panorama General

Para cuantificar objetivamente el impacto y la eficacia de los algoritmos de muestreo en grafos, resulta necesario establecer un marco de referencia. Por lo tanto, antes de aplicar cualquier técnica de reducción de dimensionalidad, se evaluó el rendimiento del sistema BCI utilizando todos los electrodos posibles. Para esto, se realiza la extracción de características mediante CSP y su clasificación lineal con LDA.

A continuación se presentan los resultados de accuracy para todos los datasets:

- **Dataset A (task 3):** 73,24 % de accuracy.
- **Dataset A (task 4):** 63,81 % de accuracy.
- **Dataset B:** 89,10 % de accuracy.
- **Dataset C:** 97,619 % de accuracy.

- **Dataset D (task 1):** 63,88 % de accuracy.
- **Dataset D (task 2):** 70,68 % de accuracy.

Estos valores se consideran entonces como una línea base (baseline) del estudio, y representa el rendimiento de referencia de los sistemas.

Las siguientes secciones se centrarán en evaluar el comportamiento del accuracy a medida que se disminuyen los canales utilizados. Todo esto con el objetivo de determinar si los subconjuntos de electrodos generados mediante procesamiento de señales en grafos logran acercarse o superar el rendimiento de referencia, demostrando que GSP ayuda a descartar electrodos redundantes sin sacrificar información crítica requerida por los modelos de clasificación.

4.3. Análisis de rendimiento: Dataset A

En el dataset se tienen distintos experimentos, y el accuracy obtenido en ellos es el promedio del accuracy entre todos los sujetos que participaron. Se proceden a analizar las tareas 3 y 4, según las Figuras 4.1 y 4.2.

4.3.1. tarea 3

Al analizar las curvas de la tarea 3, se muestra que la referencia base se mantiene similar al accuracy obtenido por todos los grafos y algoritmos hasta aproximadamente los 25 electrodos, donde se alcanza en promedio el 73 % de accuracy. De ahí en adelante, comienzan a disminuir su accuracy, especialmente a partir de los 15 y 20 electrodos. Destacan el grafo 1 con algoritmo 1, grafo 2 con algoritmo 1 y 3, y grafo 3 con algoritmo 3. El punto de inflexión, donde los modelos comienzan disminuir su accuracy de forma drástica es cerca de los 12 electrodos.

4.3.2. tarea 4

Al analizar los resultados de la tarea 4, se observa un comportamiento similar al encontrado en la tarea 3. A partir de los 12 electrodos, la accuracy comienza a disminuir de forma drástica. En este caso, destaca el grafo 1 con el algoritmo 1, y el grafo 2 con el algoritmo 1 y 3.

4.4. Análisis de rendimiento: Dataset B

En el dataset B se presentan cinco sujetos. La figura 4.3 muestra el promedio en el accuracy de todos los sujetos en cada subconjunto de electrodos para cada grafo y algoritmo.

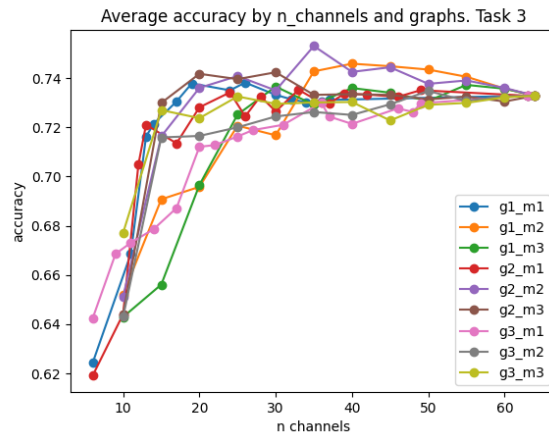


Figura 4.1: Accuracy tarea 3 dataset A.

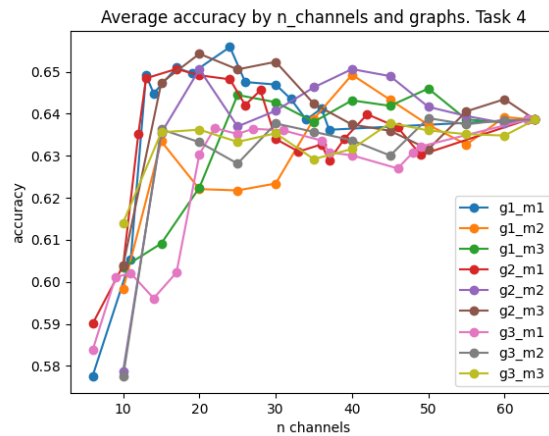


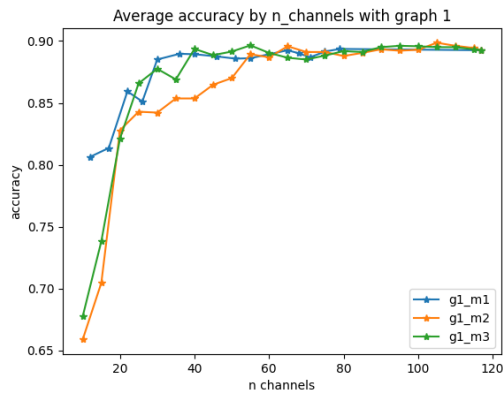
Figura 4.2: Accuracy tarea 4 dataset A.

Como es evidente, el algoritmo 2 es el que peor rendimiento tiene en todos los grafos. Por el contrario, el algoritmo 1 no solo tiene el mejor rendimiento en general, sino que también disminuye de forma suave en todos los casos.

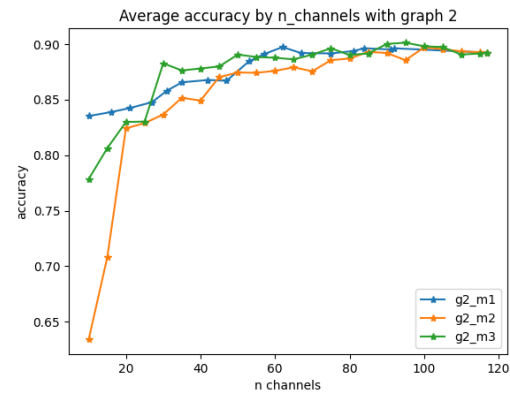
Para el grafo 1, todos los algoritmos tienen un accuracy similar a su referencia hasta que se llega a los 55 electrodos, donde comienza a disminuir de forma abrupta el rendimiento del algoritmo 3, cosa que no sucede hasta llegar a los 40 electrodos para el algoritmo 3 y a los 30 electrodos para el algoritmo 1. El algoritmo 1 es el que obtiene el mejor accuracy, de 80,40 %, con 12 electrodos.

Para el grafo 2, el algoritmo 1 se mantiene similar a la referencia hasta los 53 electrodos; para el algoritmo 2 se mantiene cercano a la referencia hasta 75 electrodos; y para el algoritmo 3 esto se mantiene hasta los 33 electrodos. El algoritmo 1 es el que obtiene el mejor accuracy, de 83,49 %, con 10 electrodos.

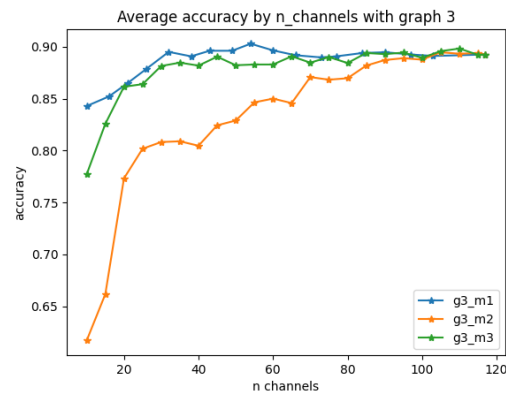
En el grafo 3, el algoritmo 2 definitivamente se destaca como el peor para cualquier densidad de electrodos, y los algoritmos 1 y 3 mantienen valores similares a la referencia hasta los 25 electrodos, donde comienzan a disminuir. Con 10 electrodos, el algoritmo 1 es el que tiene mejor accuracy, de 84,20 %, con la menor cantidad de información



(a) Resultados grafo 1 dataset B.



(b) Resultados grafo 2 dataset B.



(c) Resultados grafo 3 dataset B.

Figura 4.3: Accuracy en dataset B.

4.5. Análisis de rendimiento: Dataset C

En esta sección se evalúa el impacto de la reducción de dimensionalidad sobre las señales de imaginación motora del Dataset C. Para ello, se analizó la evolución de la exactitud de clasificación (*Accuracy*) a medida que se disminuyó progresivamente la cantidad de electrodos.

Este procedimiento se repitió para las tres topologías de grafos generadas empíricamente (ver Sección 3.3), con el fin de observar cómo la densidad de las conexiones afecta la selección de canales. Los resultados comparativos para los tres algoritmos evaluados se

presentan en la Figura 4.4.

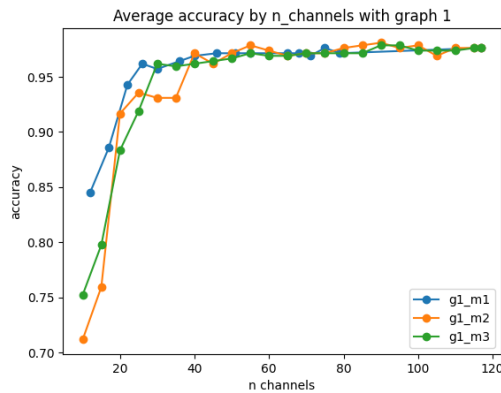
A partir de las curvas de rendimiento obtenidas, es posible extraer distintas observaciones. En primer lugar, se destaca que el algoritmo 1 demostró una mayor robustez frente a la eliminación de canales en los tres grafos utilizados (ver Figura 3.6). Por el contrario, el algoritmo 2 es el que tiene peor rendimiento, situación que se repite en todos los grafos. El algoritmo 3 tiene un rendimiento mejor que el algoritmo 2, pero no mejor que el algoritmo 1, en todos los grafos.

En cuanto al grafo 1, los tres algoritmos tienen un rendimiento similar, sin perder prácticamente accuracy hasta que el número de electrodos llega a 40, punto en el que el accuracy de los 3 algoritmos disminuye de forma pronunciada. En su mínimo de electrodos, los algoritmos tienen un rendimiento de aproximado de 84,52%, 71,18% y 75,23%, respectivamente.

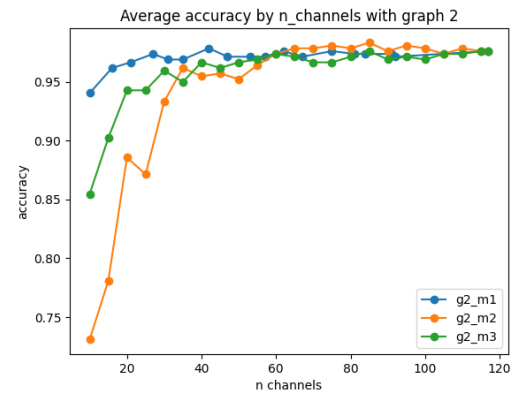
Para el grafo 2, de forma similar, se mantiene un rendimiento similar y casi sin cambiar para los tres algoritmos hasta los 40 electrodos aproximadamente. De ahí en adelante es cuando el rendimiento de los algoritmos cambia. En el caso del algoritmo 1, su accuracy disminuye muy lento y muy poco, hasta un valor de 94,05%, mientras que el algoritmo 2 llega a 73,01% y el algoritmo 3 llega a 85,45%.

Para el grafo 3, los tres algoritmos tienen un rendimiento similar hasta los 50 electrodos. De ahí en adelante el algoritmo 3 comienza a disminuir más rápido que el algoritmo 1 y 2. Finalmente alcanzan un accuracy de 88,40%, 69,69% y 80,68%, respectivamente.

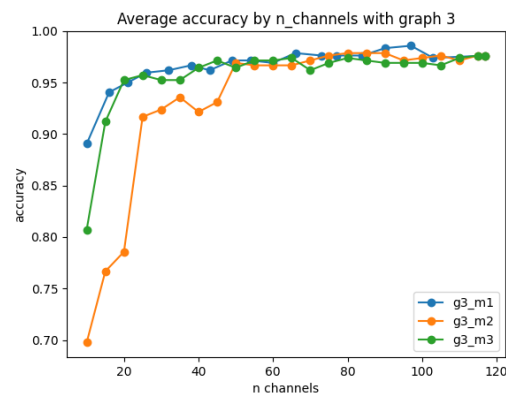
Como se observa en las Figuras y en los resultados, el algoritmo 1 es tremendamente superior a los algoritmos 2 y 3, logrando mantener niveles de accuracy competitivos al baseline, de 97,619%. En los grafos 1, 2 y 3, este algoritmo disminuyó su accuracy en un 13,42%, 3,65% y 9,44%, respectivamente, con tan solo 10 electrodos de un total de 118, que corresponde al 8,47% de la información total.



(a) Resultados grafo 1 dataset C.



(b) Resultados grafo 2 dataset C.



(c) Resultados grafo 3 dataset C.

Figura 4.4: Accuracy dataset C.

4.6. Análisis de rendimiento: Dataset D

El dataset D contiene dos experimentos distintos. En *task 1* la tarea a clasificar es de imaginación motora, y en el caso de *task 2*, es una tarea aritmética. Por esto, el análisis del rendimiento del dataset D se divide según la tarea.

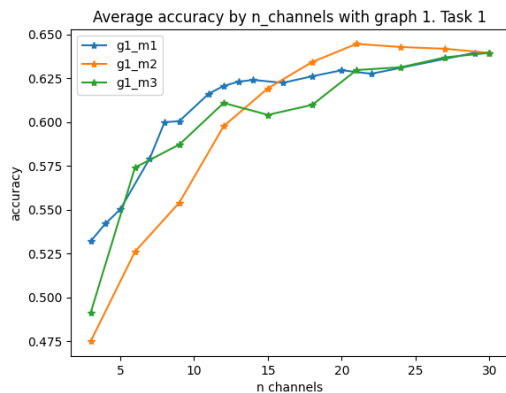
4.6.1. task 1

A partir de las curvas obtenidas para la clasificación de la tarea de imaginación motora del dataset D, se tienen sus rendimientos en la Figura 4.5. Se puede observar que en los tres grafos, el algoritmo 1 y 3 tienen rendimientos similares, siendo superados por el algoritmo 2 hasta que se disminuye a 15 electrodos. A partir de ese punto, el algoritmo 3 comienza a disminuir de forma abrupta su accuracy, a diferencia de los algoritmos 1 y 3 que pese a disminuir, su caída no es tan pronunciada con la del algoritmo 2.

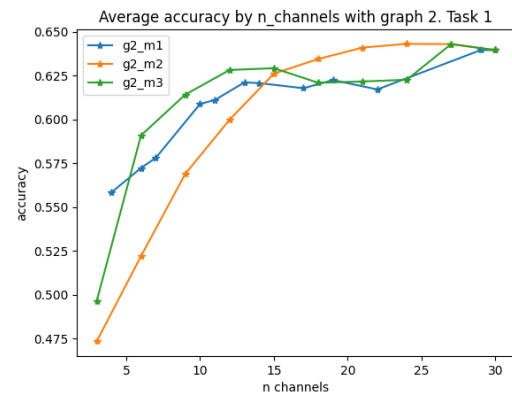
En los tres grafos, el algoritmo 1 es el que su accuracy es menos sensible a la disminución de electrodos, especialmente cuando se tiene 15 electrodos o menos, siendo superior al algoritmo 3. Mientras la cantidad de electrodos más disminuye, el accuracy del algoritmo 3 disminuye más que la del 2.

Hasta el uso de 15 electrodos, independiente del grafo, el algoritmo 3 tiene resultados prácticamente iguales. De 15 electrodos hacia abajo, el grafo 3 con el algoritmo 3 tienen la mejor accuracy.

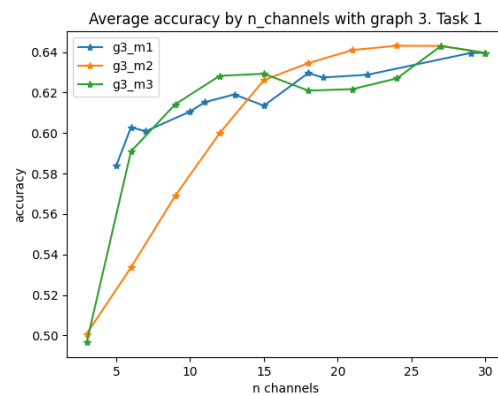
Con 3 electrodos, el algoritmo 2 es el que peor accuracy tiene (47,35%), mientras que el algoritmo 1 es el mejor con un accuracy de 53,09%. Respecto a la referencia con el total de electrodos (63,88%), el peor subconjunto tiene un 25,87% en la disminución de accuracy, mientras que en el mejor subconjunto de 3 electrodos, hay una disminución de 16,89% en el accuracy. Eso último, significa que pese a disminuir sus electrodos a aproximadamente 10% del total, hay como máximo una disminución en la accuracy de 25,87% y un mínimo de 16,89%.



(a) Resultados grafo 1 dataset D, task 1.



(b) Resultados grafo 2 dataset D, task 1.



(c) Resultados grafo 3 dataset D, task 1.

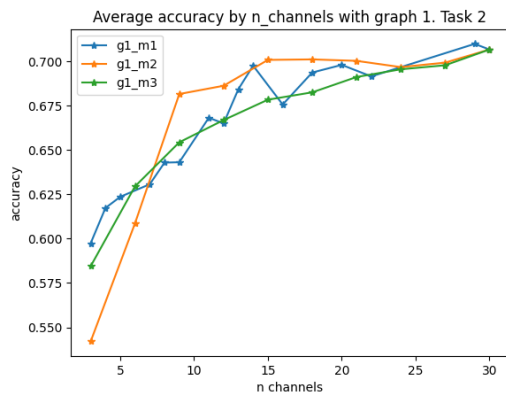
Figura 4.5: Accuracy dataset D para task 1.

4.6.2. task 2

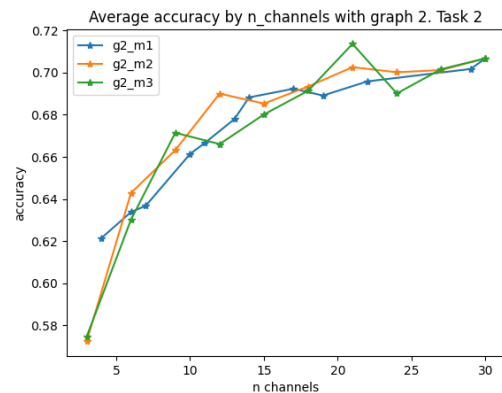
Siguiendo con el análisis del dataset D, pero ahora en la clasificación de una tarea aritmética, que se tiene en la Figura 4.6, se nota una mejora en el accuracy en comparación a la tarea de imaginación motora. La referencia ahora es de 70,68 %.

Al igual que en task 1, el algoritmo 3 es el que tiene peor rendimiento cuando se tienen menos electrodos, independiente del grafo. El algoritmo 1, también independiente del grafo, tiene una disminución menos abrupta que el resto de los algoritmos.

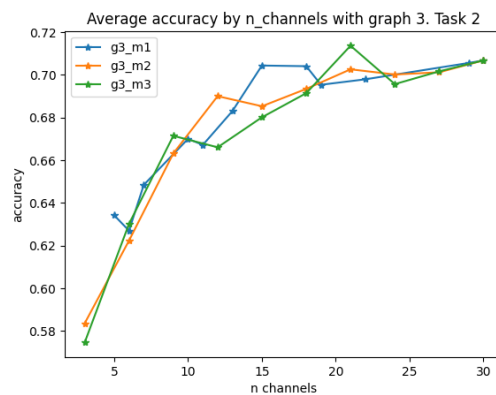
El grafo 1 con el algoritmo 2 tienen el peor accuracy, de 54,18 %, mientras que el grafo 1 y el algoritmo 1 tiene en el mejor accuracy, de 59,69 %. Esto ocurre cuando se tiene la menor densidad de electrodos, en este caso 3. En el mejor caso, significa una disminución de 15,55 % de la referencia, mientras que en el peor caso significa una disminución de 23,34 %, utilizando aproximadamente un 10 % de todo el conjunto de datos.



(a) Resultados grafo 1 dataset D, task 2.



(b) Resultados grafo 2 dataset D, task 2.



(c) Resultados grafo 3 dataset D, task 2.

Figura 4.6: Accuracy dataset D para task 2.

4.7. Resultados generales

A partir de los resultados obtenidos, es posible notar las diferencias, principalmente, entre la densidad de electrodos y los algoritmos utilizados.

El algoritmo que tiene el mejor rendimiento, es por lejos, el algoritmo 1. Mientras que el que tiene, en general, un peor rendimiento es el algoritmo 3.

Al trabajar con distintos grafos para un mismo problema, variando el número de conexiones entre sus nodos, es posible notar que no hay una relación entre el aumento del número de conexiones y una mejora (o disminución) del rendimiento del sistema BCI. Podría sonar poco intuitivo, pero pese a que un grafo tenga una cantidad exagerada de conexiones, no es algo que afecte de manera positiva o negativa al resultado.

Luego de analizar a fondo el rendimiento positivo de los distintos algoritmos, como lo es el caso del algoritmo 1, es necesario decir que el uso de técnicas de procesamiento de señales en grafo si tiene aplicaciones útiles y apropiadas en el contexto de electroencefalogramas y tareas de BCI.

4.8. Visualización de subconjuntos de electrodos

En las Figuras 4.7, 4.8 y 4.9 se puede apreciar cómo va variando la selección de electrodos para el grafo 1. De esta manera, el rendimiento de clasificación obtenido es un reflejo directo de la disposición espacial de los canales seleccionados. El algoritmo 1 demuestra que su selección, a diferencia de los algoritmos 2 y 3, evita seleccionar nodos periféricos. Ahora, viéndolo desde el punto de vista fisiológico, tiene todo el sentido que sea el que tiene mejor rendimiento en general, ya que en las tareas de imaginación motora, la actividad neuronal se concentra en la corteza motora y el área motora suplementaria, ubicadas en la región central-parietal del cuero cabelludo.

graph 1, metodo 1

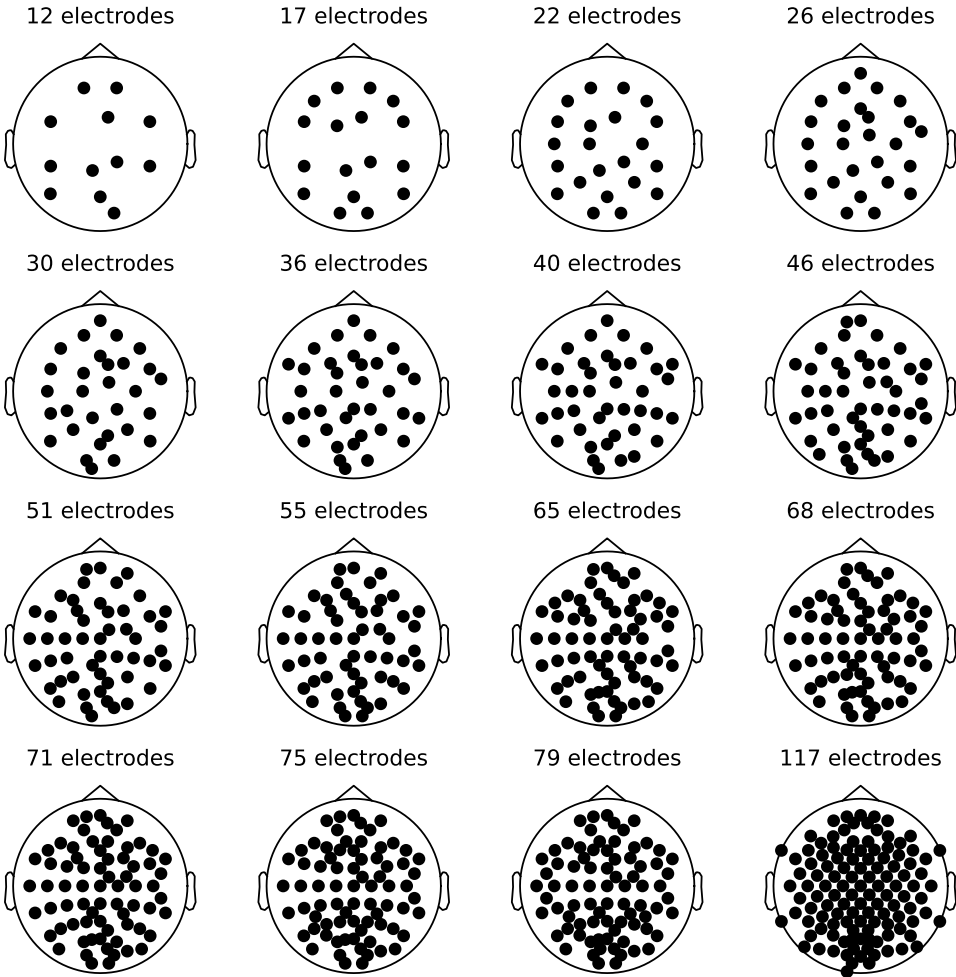


Figura 4.7: Selección espacial de electrodos para grafo 1 algoritmo 1 en dataset B y C.

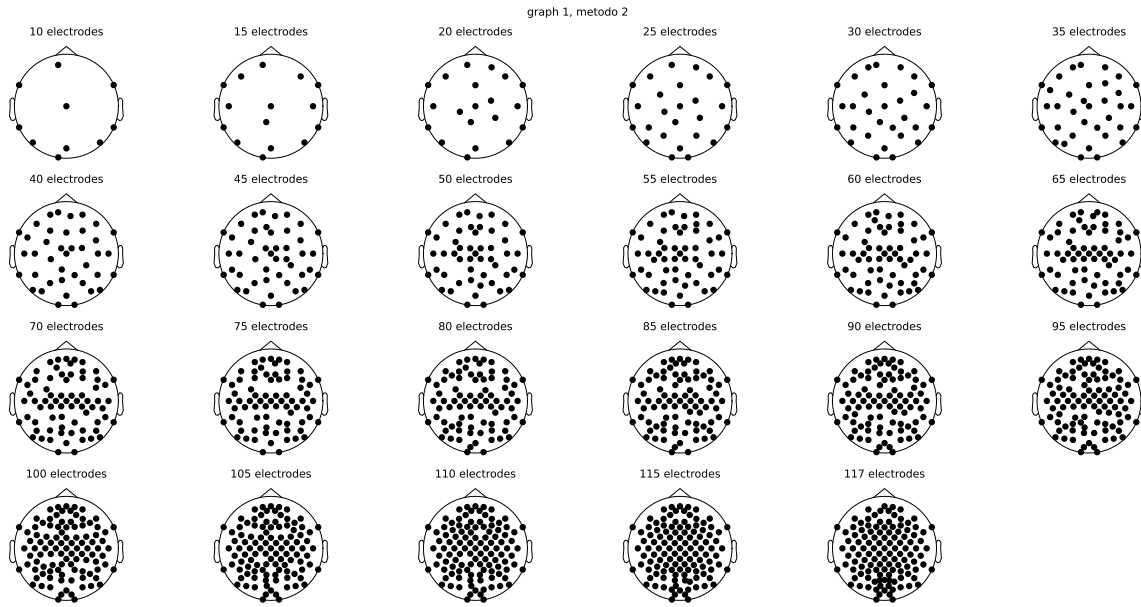


Figura 4.8: Selección espacial de electodos para grafo 1 algoritmo 2 en dataset B y C.

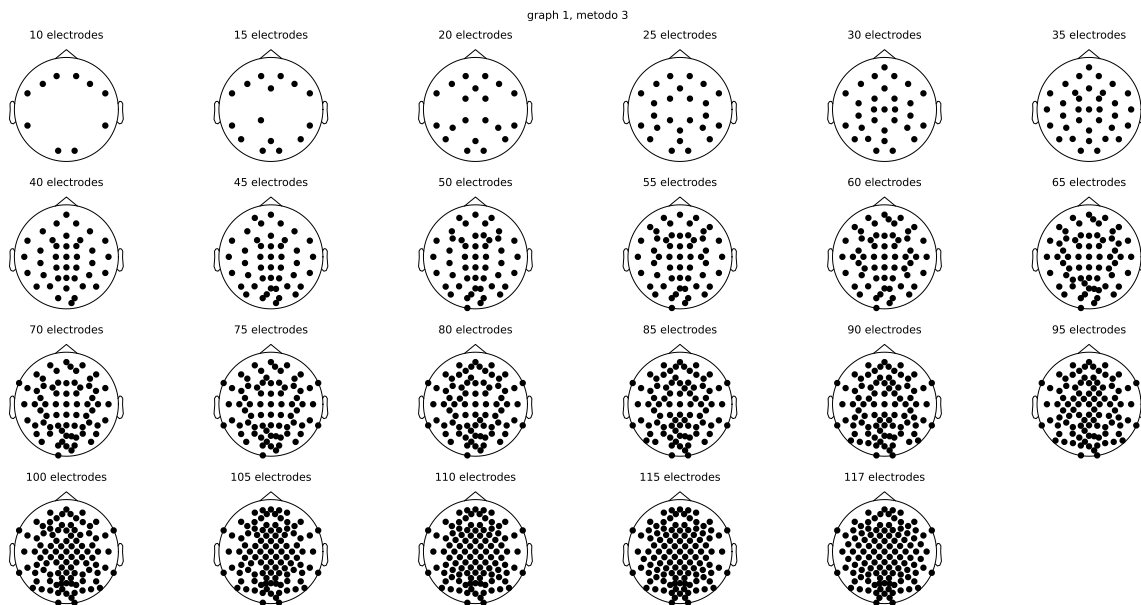


Figura 4.9: Selección espacial de electodos para grafo 1 algoritmo 3 en dataset B y C.

CAPÍTULO

5

CONCLUSIONES Y TRABAJO FUTURO

5.1. Conclusiones Generales

El presente trabajo tuvo como objetivo principal evaluar la viabilidad y el impacto de las técnicas de procesamiento de señales en Grafos (GSP) aplicadas a la reducción de dimensionalidad en Interfaces Cerebro-Computador (BCI). A partir del diseño experimental, la implementación algorítmica y el análisis de los resultados obtenidos, se derivan las siguientes conclusiones:

- **Validación de GSP en el dominio BCI:** Se demostró empíricamente que el uso de técnicas de procesamientos de señales en grafos constituye una herramienta altamente efectiva para el análisis de electroencefalogramas. Modelar el EEG mediante grafos asegura la preservación de características espaciales, que son esenciales para clasificar actividades motoras.
- **Superioridad y robustez algorítmica:** Al trabajar bajo la restricción de una baja densidad de electrodos, el algoritmo 1 demostró un rendimiento categóricamente superior a las alternativas. Logró aislar las características espaciales más relevantes, manteniendo niveles de exactitud competitivos frente a la baseline de referencia, incluso descartando una fracción mayoritaria de los canales originales. Por el contrario, el Algoritmo 3 evidenció una mayor vulnerabilidad ante la reducción de dimensiones.
- **Independencia de la densidad de electrodos:** Se encontró una ausencia en la correlación entre la cantidad de aristas de un grafo, con su rendimiento en la clasificación. Así, se comprobó que realizar muchas conexiones entre nodos del grafo

no aporta ventajas frente a un grafo cuya topología tiene pocas o significativamente menos conexiones. Esto indica que el éxito de los modelos no va enfocado en la saturación de sus conexiones, sino en su selección de forma inteligente.

- **Optimización de recursos:** La implementación de esta metodología abre puertas al diseño de interfaces más ligeras, posiblemente ergonómicas y computacionalmente eficientes, demostrando que se puede operar un sistema BCI con una fracción del volumen de datos original, reduciendo tiempos de procesamiento.

5.2. Trabajo Futuro

Pese a los prometedores resultados obtenidos, la intersección entre GSP y las interfaces neuronales sigue siendo un campo abierto. A partir de las limitaciones y hallazgos de este trabajo, se proponen las siguientes líneas de investigación futura:

- **Implementación en tiempo real (Online BCI):** Este estudio se realizó en un entorno offline, utilizando conjuntos de datos previamente obtenidos. El paso siguiente es utilizar estos algoritmos en un entorno de clasificación en tiempo real, evaluando cómo afecta la latencia el tema de los filtros CSP y el clasificador LDA.
- **Explorar con otros conjuntos de datos:** Se utilizaron datos de tareas de imaginación motora, un próximo paso sería utilizar conjuntos de datos con tareas BCI más complejas o en otro contexto, para ver cómo afectaría una señal completamente distinta en el problema de reducción de dimensionalidad.

BIBLIOGRAFÍA

- [1] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw, “Bci2000: a general-purpose brain-computer interface (bci) system,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1034–1043, 2004.
- [2] B. Blankertz, K.-R. Müller, D. J. Krusienski, G. Schalk, J. R. Wolpaw, A. Schlögl, G. Pfurtscheller, J. d. R. Millan, M. Schroder, and N. Birbaumer, “The bci competition iii: Validating alternative approaches to actual bci problems,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 153–159, 2006.
- [3] J. Shin, A. von Lühmann, B. Blankertz, D.-W. Kim, J. Jeong, H.-J. Hwang, and K.-R. Müller, “Open access dataset for eeg+nirs single-trial classification,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1735–1745, 2017.
- [4] J. Wolpaw and E. Wolpaw, *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, USA, 2012. [Online]. Available: <https://books.google.es/books?id=0dST2Lg4KVYC>
- [5] A. Ortega, *Introduction to Graph Signal Processing*. Cambridge University Press, 2022.
- [6] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K.-r. Müller, “Optimizing spatial filters for robust eeg single-trial analysis,” *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 41–56, 2008.
- [7] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [8] V. Jurcak, D. Tsuzuki, and I. Dan, “10/20, 10/10, and 10/5 systems revisited: their validity as relative head-surface-based positioning systems,” *Neuroimage*, vol. 34, no. 4, pp. 1600–1611, 2007.

- [9] M. M. N. Mannan, M. A. Kamran, and M. Y. Jeong, "Identification and removal of physiological artifacts from electroencephalogram signals: A review," *IEEE Access*, vol. 6, pp. 30 630–30 652, 2018.
- [10] M. Sazgar and M. G. Young, *EEG Artifacts*. Cham: Springer International Publishing, 2019, pp. 149–162. [Online]. Available: <https://doi.org/10.1007/978-3-030-03511-2.8>
- [11] T.-W. Lee, *Independent Component Analysis*. Boston, MA: Springer US, 1998, pp. 27–66. [Online]. Available: <https://doi.org/10.1007/978-1-4757-2851-4.2>
- [12] E. Musk and Neuralink, "An integrated brain-machine interface platform with thousands of channels," *Journal of Medical Internet Research*, vol. 21, no. 10, p. e16194, 2019.
- [13] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [14] M. Hersche, T. Rellstab, P. D. Schiavone, L. Cavigelli, and L. Benini, "An accurate eegnet-based motor-imagery brain–computer interface for low-power edge computing," in *2020 IEEE International Symposium on Medical Information and Communication Technology (ISMICT)*. IEEE, 2020, pp. 1–6.
- [15] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3864–3868.
- [16] D. E. Olivier Tzamarias, P. Akyazi, and P. Frossard, "A novel method for sampling bandlimited graph signals," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 126–130.
- [17] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, "Eigendecomposition-free sampling set selection for graph signals," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2679–2692, 2019.
- [18] W. Ren, M. Han, J. Wang, D. Wang, and T. Li, "Efficient feature extraction framework for eeg signals classification," in *2016 Seventh International Conference on Intelligent Control and Information Processing (ICICIP)*, 2016, pp. 167–172.
- [19] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. O'Reilly Media, Inc., 2019.

- [20] Q.-S. Xu and Y.-Z. Liang, "Monte carlo cross validation," *Chemometrics and Intelligent Laboratory Systems*, vol. 56, no. 1, pp. 1–11, 2001.
- [21] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. S. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Frontiers in Neuroscience*, vol. 7, no. 267, pp. 1–13, 2013.
- [22] M. Defferrard, L. Martin, R. Pena, and N. Perraudin, "Pygsp: Graph signal processing in python." [Online]. Available: <https://github.com/epfl-lts2/pygsp/>
- [23] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4, pp. 411–430, 2000. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608000000265>