



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

CLASIFICACIÓN DE ESTADOS DE CONCIENCIA CEREBRALES USANDO
MODELOS OCULTOS DE MARKOV

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA MENCIÓN ELÉCTRICA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

MATIAS DAVID GODOY PORTIOLI

PROFESOR GUÍA:
MARCOS ORCHARD CONCHA

PROFESOR CO-GUÍA:
JORGE SILVA SANCHEZ

MIEMBROS DE LA COMISIÓN:
RODRIGO COFRÉ TORRES
FELIPE TOBAR HENRÍQUEZ

SANTIAGO DE CHILE
2023

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE
LA INGENIERÍA, MENCIÓN ELÉCTRICA Y MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

POR: **MATIAS GODOY PORTIOLI**

FECHA: 2023

PROF. GUÍA: MARCOS ORCHARD C.

PROF. CO-GUÍA: JORGE SILVA S.

**CLASIFICACIÓN DE ESTADOS DE CONCIENCIA CEREBRALES
USANDO MODELOS OCULTOS DE MARKOV**

Uno de los grandes problemas en neurociencia es lograr comprender más acerca de la conciencia. Se buscan métodos para lograr entender mejor este concepto a través de mecanismos asociados al cerebro. Hoy en día, gracias a los grandes datos y gran poder computacional, se logra obtener modelos de cerebro que, a través de ecuaciones diferenciales estocásticas, logran simular señales cerebrales asociadas a ciertos estados de conciencia mediante un ajuste de los parámetros de estas ecuaciones. En el presente documento se abordará el problema de clasificación de estados cerebrales mediante el procesamiento de señales simuladas por un modelo cerebral para el caso de dos estados específicos clasificados como: “placebo” y “LSD”. Para esto se utilizan métodos asociados al procesamiento de señales de la ingeniería como matrices de correlación, extracción de características con métodos de descomposición de matrices y clasificación mediante el proceso de aprendizaje no supervisado llamado modelos ocultos de Markov. Se logra una clasificación de estados por sobre el 95% de acierto y además se muestran métodos para mejorar los resultados obtenidos. Finalmente, se usa k-means para obtener información de las matrices observando patrones en los centroides de cada cluster y poder caracterizar la dinámica de estos estados.

A cada persona que siempre confió en que lo lograría.
Gracias.

Agradecimientos

Quiero agradecer a mis padres que desde pequeño me enseñaron que con esfuerzo y dedicación es posible alcanzar los sueños más lejanos. Porque estuvieron y estarán para mí en todo momento y en todo lugar siempre.

Agradezco a mis profesores guía Marcos y Jorge porque fueron excelentes docentes en todo el proceso de esta tesis y porque superaron mis expectativas en cada aspecto que tenía. Por haberme enseñado más que solo una metodología de pensamiento y trabajo críticos y por enseñarme compromiso y devoción.

Quiero agradecer a mis amigos que hice dentro de la universidad, porque me enseñaron a valorar las cosas extraacadémicas y me ayudaron a conectarme con mi inteligencia emocional. Por todos los viernes nefastos y también por todos los momentos reflexivos que vivimos juntos. Agradezco haberlos conocido y que me hayan recibido como uno más en esta familia.

Agradezco especialmente a mis amigos Bastian Sáez, Gabriela Toledo, Valeria Valdés, Marcela Herrera, Graciela Espina y Constanza Navarro por ayudarme y apoyarme en el avance y desarrollo de esta tesis. Ustedes estuvieron conmigo en algún momento del proceso de mi tesis y me motivaron a seguir adelante siempre. Gracias por todos los momentos que hemos pasado juntos, siempre estarán en mis recuerdos. Los quiero demasiado a cada uno.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Hipótesis de Investigación	2
1.3. Objetivos Generales	2
1.4. Objetivos Específicos	3
1.5. Estructura del informe	3
2. Marco Teórico	4
2.1. Modelo de cerebro completo: Dynamic Mean-Field	4
2.1.1. Glosario	4
2.1.2. Antecedentes y Contexto	5
2.1.3. Modelos de Cerebro Completo	5
2.1.4. Dynamic Mean-Field	6
2.2. Covarianza, Correlación y SVD	8
2.2.1. Covarianza	8
2.2.2. Matrices de Correlación	9
2.2.3. Singular Value Descomposition (SVD)	10
2.3. Modelos y algoritmos de procesamiento de señales asociados al proceso de clasificación	11
2.3.1. Cadenas de Markov	11
2.3.2. Modelos Ocultos de Markov (HMM)	12
2.3.3. Algoritmo k-means	14

2.3.4. Algoritmo de Viterbi	16
3. Metodología	18
3.1. Señales BOLD	19
3.2. Preprocesamiento	20
3.3. Matrices de correlación	22
3.4. Extracción de características principales	23
3.5. Método de clasificación	24
3.6. Mejoras de clasificación por bloques	26
3.7. Dinámica de cada estado cerebral	26
4. Resultados	28
4.1. Clasificación de Señales BOLD	28
4.1.1. Algoritmo de Clasificación	28
4.1.2. Histogramas	29
4.1.3. Matrices de Transición	29
4.1.4. Clasificación por bloques	30
4.2. Clasificación de Señales BOLD con datos de testeo	30
4.2.1. Algoritmo de clasificación con datos de testeo	31
4.2.2. Histogramas	31
4.2.3. Matrices de Transición	32
4.2.4. Tabla de comparación de desempeño	32
4.3. Dinámica	33
4.3.1. Distribución	33
4.3.2. Histogramas	34
4.3.3. Matrices Centroides	34
4.3.4. Matrices de Transición	35
4.3.5. Conectoma y Centroides de los 11 clusters	36

4.3.6. Histogramas de los 11 clusters	37
5. Discusión	38
5.1. Algoritmo de Clasificación	38
5.2. Prueba con datos de testeo	39
5.3. Dinámica	39
6. Conclusiones y trabajo futuro	41
Bibliografía	42
ANEXO	46
A.1. DMF representado en ecuaciones.	46
A.2. Algoritmo de Viterbi	47
A.3. Código MATLAB de Parámetros por Default	48
A.4. Métodos de extracción de características	49
A.5. Error de las distintas combinaciones de valores propios	50
A.6. Algoritmo de clasificación del <i>caso 1</i>	51
A.6.1. Algoritmo de Viterbi	51
A.6.2. Histogramas	51
A.6.3. Matrices de Transición	52
A.6.4. Clasificación por bloques	52
A.7. Algoritmo de clasificación con datos de testeo	53
A.7.1. Algoritmo de Viterbi	53
A.7.2. Histogramas	53
A.7.3. Matrices de Transición	54
A.7.4. Clasificación por bloques	54
A.7.5. Tabla de comparación de desempeño	55
A.7.6. Matrices de Transición para 11 clusters	55

Índice de Tablas

4.1. Comparación de porcentajes de error de clasificación de datos de entrenamiento y datos de testeo.	32
A.1. Porcentajes de error de clasificación para datos de testeo de ambos casos . .	55

Índice de Ilustraciones

2.1. Visión general de la integración de variables multimodales que incluyen la neuromodulación en un modelo neuronal de todo el cerebro. Se muestran los ingredientes básicos para la integración de datos de neuroimagen multimodal de tipo estructural (dMRI, arriba a la izquierda), funcional (fMRI, abajo a la izquierda) y de neurotransmisión (PET, arriba a la derecha) utilizando la misma parcelación para cada modalidad de neuroimagen (abajo a la derecha) para generar un modelo computacional de todo el cerebro (centro). Modificada de [4]	6
2.2. Matriz de conectividad anatómica.	7
2.3. Una cadena de Markov con 5 estados (etiquetados de S_1 , a S_5 ,) con transiciones de estado (etiquetados como a_{ij} que pueden estar definidos o pueden ser números no conocidos). Obtenida de [31].	11
3.1. 90 Señales BOLD de tipo “placebo” preprocesadas	21
3.2. 90 Señales BOLD de tipo “LSD” preprocesadas	21
3.3. 90 Señales BOLD de tipo “placebo” procesadas	21
3.4. 90 Señales BOLD de tipo “LSD” procesadas	22
3.5. Ilustración de la extracción de matrices de correlación obtenido desde señales BOLD. Obtenida de [4]	22
3.6. Ejemplo de matriz de correlación de señales BOLD.	23
3.7. Representación de cada etapa del procedimiento y clasificación de las señales BOLD.	25
3.8. Puntaje del desempeño del algoritmo k-means en función de la cantidad de clusters que tenga para clasificar.	26
4.1. Gráfico de clasificación de los vectores de características.	29
4.2. Histogramas de la clasificación de los vectores de características.	29

4.3.	Matrices de Transición de la clasificación.	30
4.4.	Desempeño de la clasificación del algoritmo de Viterbi en función del tamaño de los bloques de vectores.	30
4.5.	Clasificación del algoritmo de Viterbi con datos de testeo.	31
4.6.	Histogramas de la clasificación con los datos de testeo.	31
4.7.	Matrices de transición de los datos de testeo.	32
4.8.	Desempeño de clasificación en función del tamaño de los bloques de vectores con datos de testeo.	33
4.9.	Distribución de vectores de características del algoritmo <i>k-means</i>	33
4.10.	Histogramas de clasificación con 3 clusters.	34
4.11.	Matrices que representan los centroides de los 3 <i>clusters</i> con sus respectivas distribuciones de matrices “placebo” y “LSD”, ordenados por orden de semejanza con el conectoma.	35
4.12.	Matrices de transición del algoritmo <i>k-means</i> para 3 <i>clusters</i>	36
4.13.	Matrices que representan los centroides de los 11 <i>clusters</i> ordenados conforme a la semejanza con el conectoma.	36
4.14.	Histogramas de clasificación con 11 clusters.	37
A.1.	Gráfico de cajas del MSE normalizado de las matrices reconstruidas con respecto a las matrices originales de los distintos métodos que se utilizaron para extraer las características de las matrices. <i>Variational</i> , <i>Convolutional</i> y <i>dense</i> son redes neuronales <i>encoder-decoder</i> cuyo vector de <i>encoder</i> es de largo 20. <i>Wavelet</i> corresponde al método <i>Wavelet</i> para extraer características, sin embargo, se utilizó un largo de 2000 elementos para poder tener un desempeño similar al resto. SVD se utilizaron los primeros 10 valores y vectores propios para reconstruir las matrices.	49
A.2.	Errores del algoritmo de Viterbi para las distintas combinaciones de los valores propios asociados al conjunto de entrenamiento placebo y LSD.	50
A.3.	Suma de los errores del algoritmo de Viterbi para las distintas combinaciones de los valores propios asociados al conjunto de entrenamiento placebo y LSD.	50
A.4.	Gráfico de clasificación de muestras de tiempo para el <i>caso 1</i>	51
A.5.	Histogramas de la clasificación del <i>caso 1</i>	51
A.6.	Matrices de Transición de la clasificación del <i>caso 1</i>	52

A.7. Desempeño de la clasificación del algoritmo de Viterbi en función del tamaño de los bloques de vectores para el <i>caso 1</i>	52
A.8. Clasificación del algoritmo de Viterbi con datos de testeo.	53
A.9. Histogramas de la clasificación con los datos de testeo.	53
A.10. Matrices de transición de los datos de testeo.	54
A.11. Desempeño de clasificación en función del tamaño de los bloques de vectores con datos de testeo.	54
A.12. Matriz de Transición de la distribución de 11 <i>clusters</i> en los datos placebo	55
A.13. Matriz de Transición de la distribución de 11 <i>clusters</i> en los datos LSD	56

Capítulo 1

Introducción

1.1. Motivación

El cerebro es un órgano que centraliza la actividad del sistema nervioso y existe en la mayor parte de los animales. Durante las primeras etapas de la psicología, se creyó que la mente debía separarse del cerebro. Sin embargo, posteriormente, los científicos realizaron experimentos que llegaron a determinar que la mente era un componente en el funcionamiento cerebral [1]. A pesar de eso, los mecanismos por los cuales la actividad cerebral da lugar a la conciencia y al pensamiento son muy difíciles de comprender, y mucho acerca de cómo funciona el cerebro sigue siendo un misterio a pesar de los múltiples y rápidos avances científicos. En la actualidad, las operaciones de las células cerebrales individuales son comprendidas con más detalle, pero la forma en que cooperan entre los conjuntos de millones ha sido muy difícil de descifrar.

El hecho de que el cerebro no se pueda comprender de manera exacta es una de las razones por la cual, hoy en día, existen métodos subjetivos para reconocer algunos *estados de conciencia*. Un ejemplo útil es la forma más habitual de saber si un paciente de hospital está en estado de sedación o agitación. Los métodos utilizados para este fin son llamados escalas de valoración de la sedación, los cuales son instrumentos subjetivos que, en general, miden la respuesta de los pacientes a la estimulación auditiva o física, a través de la observación [2]. De manera similar, se podrían clasificar varios estados de conciencia como Alzheimer, depresión o cualquiera cuyo diagnóstico dependa del juicio de un médico o profesional de la salud.

Bajo estos problemas se han desarrollado trabajos que reconcilian las teorías de la conciencia con las observaciones de la correlación de largo alcance en el cerebro. Uno de estos trabajos es el de Barttfeld et al. [3] donde se miden los patrones de conectividad funcional en monos en 3 estados: Vigilia, Anestesia y Anestesia Profunda. Se presenta que, entre más presencia de anestesia en los monos, los patrones de conectividad funcional más frecuentes tienen semejanza con la estructura de la conectividad anatómica y que, por el contrario, en el estado de Vigilia se muestra un repertorio más rico de configuraciones funcionales. Estos resultados tienen potenciales implicaciones clínicas para la detección de la conciencia

en pacientes anestesiados y con lesiones cerebrales, además muestran que una rica dinámica funcional podría constituir una firma de la conciencia, lo cual va en la misma dirección de lo que propone esta tesis.

Otro de los trabajos relacionados con esta tesis es el de Deco et al. [4] el cual utiliza el *modelo de cerebro completo DMF* (Dinamic Mean-Field) para modelar el estado de reposo y logra explicar mecánicamente los efectos funcionales de la estimulación del 5- $HT_{2A}R^1$ con dietilamida de ácido lisérgico (LSD). Este trabajo es de especial interés, ya que contiene el modelo de cerebro completo que utiliza esta tesis para obtener los datos base para su desarrollo.

También, se han escrito otras investigaciones que usan el modelo DMF y que van en sintonía con esta tesis. En el trabajo de Herzog et al. [5] se propone abordar los estados alterados de conciencia con un modelo simplificado de cerebro (FastDMF) para proponer una explicación mecanicista del aumento de entropía neural gatillado por drogas psicodélicas serotoninérgicas (agonistas del receptor 5- HT_{2A}), conocido como la hipótesis del cerebro entrópico. Estos resultados ayudan a entender los mecanismos subyacentes al estado psicodélico desde reglas biofísicas de bajo nivel y, más generalmente, a entender la modulación farmacológica de los indicios globales de la conciencia.

Yendo en sintonía con lo anterior, el desafío de esta tesis consiste en crear un modelo, utilizando algoritmos asociados a modelos ocultos de Markov y propiedades matemáticas asociados al área de las matrices, con el propósito de clasificar señales cerebrales sintéticas de manera estadística y objetiva, los cuales se podrían asociar a determinados estados de conciencia. El alcance de este trabajo se limita a diferenciar señales cerebrales, simuladas con el modelo DMF, que se asocian a dos “estados” distintos, definiendo “estado” como las interacciones entre las regiones cerebrales asociados a una determinada experiencia subjetiva.

1.2. Hipótesis de Investigación

La hipótesis de esta tesis consiste en que el cerebro presenta señales que están ligadas a un estado de conciencia único y, por tanto, es posible saber cuál es el estado de conciencia que presentan los individuos al medir sus señales cerebrales.

1.3. Objetivos Generales

El objetivo principal de este proyecto, es lograr utilizar señales simuladas del cerebro asociadas a los estados “placebo” y “LSD”, y aplicar algoritmos asociados a modelos ocultos de Markov para poder clasificar, con un bajo porcentaje de error, cada serie de tiempo asociado a señales cerebrales condicionadas a que el estado cerebral puede estar en uno de los dos estados. Además, se quiere lograr obtener información de las matrices observando

¹Receptor de serotonina en el que la activación aguda por psicodélicos serotoninérgicos produce un estado alterado de conciencia transitorio.

patrones en estas y así poder caracterizar la dinámica de estos estados.

1.4. Objetivos Específicos

Los objetivos específicos del proyecto en cuestión, son:

- Procesar y filtrar datos obtenidos de la aplicación del modelo DMF en MATLAB.
- Obtener matrices de correlación de las señales BOLD² con un tiempo de muestreo representativo (ventana) y una diferencia de tiempo significativa entre matrices (slide).
- Identificar y extraer las características de las matrices de correlación que muestren un mejor desempeño de clasificación.
- Clasificar las características de las matrices de correlación utilizando el algoritmo de Viterbi obtenido a partir de Hidden Markov Model y que su clasificación sea mayor al 90 % de acierto.
- Utilizar k-means para encontrar patrones en las matrices de correlación que dan información relevante respecto a la dinámica cerebral de cada estado.

1.5. Estructura del informe

A continuación se mostrará el marco teórico donde describirá con detalle el modelo de cerebro completo DMF, la teoría detrás de las matrices de correlación, la propiedad de descomposición de las matrices y los modelos y algoritmos asociados al proceso de clasificación. En la metodología se explica el procedimiento utilizado para obtener y procesar los datos, obtener las matrices de correlación, sus características principales y finalmente se menciona la forma en como se clasifica cada serie de tiempo a cada estado. En los resultados se da una breve descripción de cada imagen y luego estas se analizan dando una explicación de ciertos resultados. Por último, en las conclusiones se describe el cumplimiento del objetivo general, las limitaciones del procedimiento y de los alcances propuestos a futuro.

²Blood oxygen level-dependent es una técnica usada en la imagen por resonancia magnética para medir niveles locales y transitorios de oxígeno en la sangre.

Capítulo 2

Marco Teórico

En esta sección se detallarán las bases teóricas, así como conceptos necesarios para entender la tesis realizada. El marco teórico se dividirá en 4 subsecciones: modelos de cerebro completo: DMF, donde se especificará al detalle las características fundamentales del funcionamiento de este y cómo se adquieren los datos usados en el algoritmo de clasificación; Covarianza, Correlación y SVD, donde se describirán los métodos usados para seccionar los datos y extraer las características de estos; Modelos y algoritmos de procesamiento de señales asociados al proceso de clasificación, sección en la cual se detallarán las cadenas de Markov, los modelos ocultos de Markov, el algoritmo k-means y el algoritmo de Viterbi el cual da las herramientas necesarias para clasificar los datos.

2.1. Modelo de cerebro completo: Dynamic Mean-Field

2.1.1. Glosario

- **Bottom-up:** Define la dinámica local de las unidades que interactúan (como las neuronas o los grupos de neuronas) para generar características lo más parecidas posible a las observadas en el cerebro durante diferentes condiciones experimentales
- **AAL:** Automated anatomical labeling
- **DTI:** Diffusion tensor imaging
- **PET:** Positron emission tomography
- **dMFI:** Diffusion magnetic resonance imaging
- **fMRI:** Functional magnetic resonance imaging

2.1.2. Antecedentes y Contexto

En neurociencia se ha intentado proporcionar respuestas a como surge la conciencia, uno de los enfoques más actuales es la propuesta de Crick y Koch [6, 7], el cual se entiende como el conjunto mínimo de eventos neuronales que se asocian a una determinada experiencia subjetiva. Si bien, esta propuesta apunta a descubrir dónde y cuándo ocurre la conciencia, existen otros esfuerzos [8, 9] sobre esta área que intentan predecir cómo surge la conciencia a partir de los procesos que dan lugar a estos eventos neuronales, los cuales son llamados normalmente *modelos teóricos de la conciencia*.

Existen varios enfoques de modelos de conciencia, pero para efectos de esta tesis, se utiliza el enfoque denominado *Bottom-up* [10], el cual se caracteriza principalmente por partir desde las reglas dinámicas de las unidades elementales (como las neuronas o los grupos de neuronas [11]) e intentan proporcionar predicciones cuantitativas explorando las consecuencias agregadas de estas reglas a través de varias escalas temporales y espaciales. Una de las ventajas que proporciona este tipo de enfoque es que se puede utilizar de manera práctica mediante una de las herramientas más útiles: Los *Modelos de Cerebro Completo*.

2.1.3. Modelos de Cerebro Completo

Los modelos de cerebro completo son conjuntos de ecuaciones que describen la dinámica y las interacciones entre las poblaciones neuronales de diferentes regiones del cerebro [12]. Estos modelos suelen centrarse en la evolución conjunta de un conjunto de variables biofísicas clave utilizando sistemas de ecuaciones diferenciales acopladas (aunque también se pueden utilizar modelos de pasos de tiempo discretos). Estas ecuaciones pueden construirse a partir del conocimiento de los mecanismos biofísicos que subyacen a las diferentes formas de actividad cerebral, o como modelos fenomenológicos elegidos por el tipo de dinámica que producen.

La mayoría de los modelos de cerebro completo se estructuran en torno a tres elementos básicos:

A. Parcelación del cerebro: La parcelación del cerebro determina el número de regiones y la resolución espacial en la que tiene lugar la dinámica cerebral. La parcelación puede incluir regiones corticales, subcorticales y cerebelosas. Ejemplos de parcelaciones bien conocidas son la parcelación de Hagmann [13], y el atlas de etiquetado anatómico automatizado (AAL) [14] siendo este último el utilizado en esta tesis.

B. Matriz de conectividad anatómica: Esta matriz define la red de conexiones entre las regiones del cerebro. La mayoría de los estudios se basan en el conectoma humano, obtenido mediante la estimación del número de fibras de materia blanca que conectan las áreas cerebrales a partir de datos DTI combinados con tractografía probabilística [15]. Esta matriz del conectoma se muestra de forma gráfica en la Figura 2.1.

C. Dinámica local: La actividad de cada región del cerebro suele estar determinada por la dinámica local elegida más los términos de interacción con otras regiones. Se han propuesto diversos enfoques para modelar la dinámica de todo el cerebro, como los autómatas celulares [16, 17], el modelo de espín de Ising [18, 19], los modelos autorregresivos [20], los modelos

lineales estocásticos [21], los osciladores no lineales [22, 23], la teoría del campo neural [24, 25], los modelos de masa neural [26, 27] y los modelos dinámicos de campo medio [28, 4, 29].

Los elementos A y B se guían por los datos experimentales disponibles. Por el contrario, la elección de la dinámica local suele estar guiada por los fenómenos que se estudian y el contexto epistemológico en el que tiene lugar el esfuerzo de modelización [12]. En este caso, la dinámica local que se utilizará es *Dynamic Mean-Field*, ya que ha sido capaz de modelar de manera precisa los estados cerebrales que le competen a esta tesis.

2.1.4. Dynamic Mean-Field

Dynamic Mean-Field es un modelo de cerebro completo que integra imágenes multimodales en participantes humanos sanos sometidos a la manipulación del sistema de la serotonina. Se combinan datos anatómicos de imágenes de resonancia magnética de difusión (dMRI) y de imágenes de resonancia magnética funcional (fMRI) con datos de neurotransmisores obtenidos con tomografía de emisión de positrones (PET) del mapa de densidad detallado del receptor de serotonina 2A ($5-HT_{2A}R$). En la Figura 2.1 se muestra, la descripción anterior.

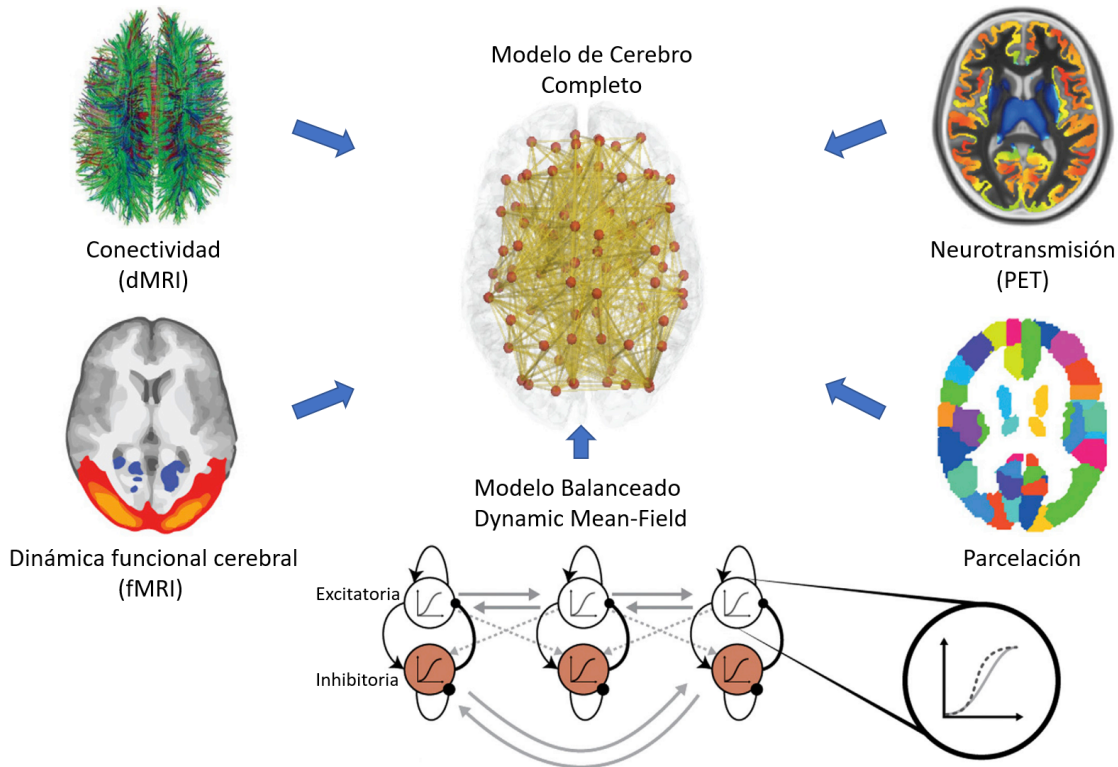


Figura 2.1: Visión general de la integración de variables multimodales que incluyen la neuro-modulación en un modelo neuronal de todo el cerebro. Se muestran los ingredientes básicos para la integración de datos de neuroimagen multimodal de tipo estructural (dMRI, arriba a la izquierda), funcional (fMRI, abajo a la izquierda) y de neurotransmisión (PET, arriba a la derecha) utilizando la misma parcelación para cada modalidad de neuroimagen (abajo a la derecha) para generar un modelo computacional de todo el cerebro (centro). Modificada de [4]

En este modelo, la actividad neuronal en cada región del cerebro se representa mediante un conjunto de ecuaciones diferenciales que describen la interacción entre **grupos de neuronas inhibitorias y excitatorias**. La variable de salida del modelo es la tasa de disparo de la población excitatoria que se incluye en un modelo hemodinámico no lineal para simular las **señales BOLD** regionales [12]. Las ecuaciones (mostradas en Anexo A.1) fueron modificadas por Deco et al. [4], básicamente se añadieron 2 parámetros a las ecuaciones que permiten modelar el estado de reposo (o “placebo”) y explicar mecánicamente los efectos funcionales de la estimulación del $5-HT_{2A}R$ con dietilamida de ácido lisérgico (LSD) en participantes sanos.

El primero de estos es el **parámetro de escalado de ganancia global** (sE), que se añadió a los parámetros de ganancia fijos originales (que eran los mismos para todas las regiones) y así se pudo escalar los valores regionales de $5-HT_{2A}R$ influyendo en los circuitos recursivos de neuronas excitatorias e inhibitorias. El segundo, es un parámetro neuronal que escala el **acoplamiento global** de las poblaciones neuronales llamado G , el cual se ajusta para explicar la condición “LSD”, cambiando selectivamente la ganancia neuronal de cada región según la densidad $5-HT_{2A}R$ medida empíricamente.

En la Figura 2.2 se muestra el mapa de conectividad anatómica que se utiliza en el modelo DMF.

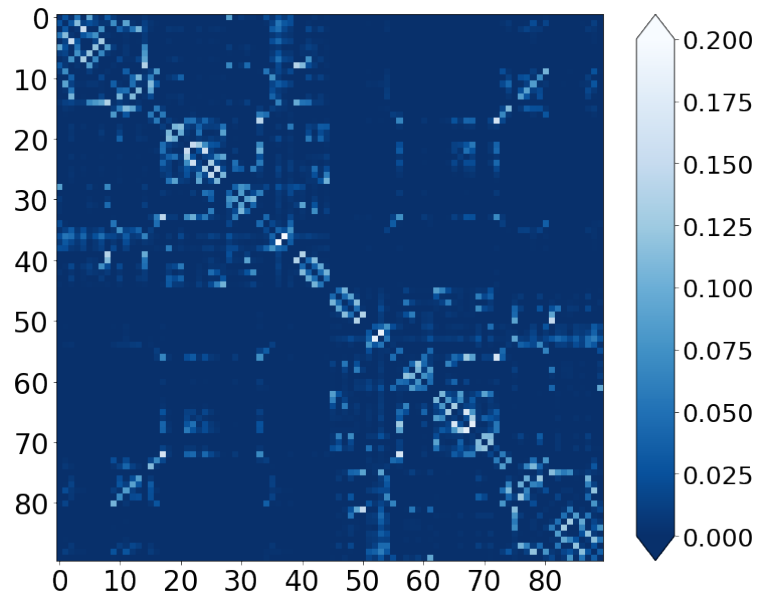


Figura 2.2: Matriz de conectividad anatómica.

En concreto, este modelo es de esencial importancia, ya que permite simular, de manera muy aproximada, las señales BOLD de los estados llamados “placebo” y estado “LSD”, los cuales son datos necesarios para el desarrollo de esta tesis.

2.2. Covarianza, Correlación y SVD

2.2.1. Covarianza

Sean X e Y dos variables aleatorias. La covarianza de X e Y , denotada por $Cov(X, Y)$, se define como:

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])]. \quad (2.1)$$

La covarianza de X e Y proporciona una medida de la asociación entre X e Y . Una covarianza positiva indica que las desviaciones de X e Y respecto a sus respectivas medias tienden a tener el mismo signo. Así, cuando una de ellas (X o Y) es alta, la otra tiende a ser alta, y cuando una de ellas es baja, la otra tiende a ser baja respecto a su media. También significa que, en promedio, tienden a aumentar o disminuir juntos o a moverse en la misma dirección. Por otro lado, una covarianza negativa indica que las desviaciones de X e Y respecto a sus respectivas medias tienden a tener signos opuestos. Cuando una de ellas es alta, la otra tiende a ser baja, y cuando una de ellas aumenta, la otra tiende a disminuir en promedio.

La covarianza de una variable aleatoria consigo misma es igual a su varianza:

$$Cov(X, X) = E[(X - E[X])^2] = VAR[X], \quad (2.2)$$

también se puede escribir la covarianza de otra forma. Si X es una variable aleatoria y $E[X]$ y $E[Y]$ son constantes, entonces, basándonos en la propiedad lineal de la esperanza, se tiene:

$$\begin{aligned} Cov(X, Y) &= E[(X - E[X])(Y - E[Y])] \\ &= E[XY - XE[Y] - YE[X] + E[X]E[Y]] \\ &= E[XY] - E[X]E[Y] - E[Y]E[X] + E[X]E[Y] \\ &= E[XY] - E[X]E[Y], \end{aligned} \quad (2.3)$$

ahora bien, si X e Y son independientes, se puede escribir:

$$\begin{aligned} Cov(X, Y) &= E[XY] - E[X]E[Y] \\ &= E[X]E[Y] - E[X]E[Y] \\ &= 0, \end{aligned} \quad (2.4)$$

así que dos variables aleatorias independientes tienen una covarianza cero.

2.2.2. Matrices de Correlación

El coeficiente de correlación entre X e Y , denotada por $\rho(X, Y)$, se define como sigue:

$$\rho(X, Y) = \frac{Cov(X, Y)}{\sigma_x \sigma_y}, \quad (2.5)$$

basándose en la desigualdad de CAUCHY-SCHWARZ se tiene:

$$(Cov(X, Y))^2 \leq \sigma_x^2 \sigma_y^2, \quad (2.6)$$

lo cual se puede escribir de la siguiente forma

$$- \sigma_x \sigma_y \leq Cov(X, Y) \leq \sigma_x \sigma_y, \quad (2.7)$$

dividiendo la desigualdad anterior por $\sigma_X \sigma_Y$ se obtiene:

$$- 1 \leq \rho(X, Y) \leq 1. \quad (2.8)$$

Basándonos en la Ecuación 2.5, la correlación y la covarianza tienen el mismo signo (ya que la desviación estándar es siempre positiva). Cuando $\rho(X, Y) > 0$, X e Y están positivamente correlacionados y tienden a moverse juntos, y cuando $\rho(X, Y) < 0$, están negativamente correlacionados y tienden a moverse en la dirección opuesta. Si $\rho(X, Y) = 0$, se dice que X e Y no están correlacionados.

Además, se puede demostrar que si existe una relación lineal entre X e Y

$$Y = aX + b, \quad (2.9)$$

donde a y b son constantes y $a \neq 0$, entonces se tiene:

$$(Cov(X, Y))^2 = \sigma_x^2 \sigma_y^2. \quad (2.10)$$

Así que se puede obtener que $|\rho(X, Y)| = 1$. Si $a > 0$, entonces la línea tiene una pendiente positiva y X e Y deben aumentar o disminuir juntos. Entonces se tiene una covarianza y correlación positivas y $\rho(X, Y) = 1$. Del mismo modo, si $a < 0$, X e Y se mueven en dirección opuesta, y $\rho(X, Y) = -1$. Por último, si existen dos variables aleatorias independientes, su covarianza es cero (Ec. 2.4), por lo que su coeficiente de correlación es cero y se dice que no están correlacionadas.

En resumen, el valor de $\rho(X, Y)$ proporciona una medida del grado en que dos variables aleatorias X e Y están relacionadas linealmente. Si los valores que toman X e Y están relativamente concentrados en torno a una línea recta con pendiente positiva, entonces $\rho(X, Y)$ será cercano a 1. Por otro lado, si estos valores están relativamente concentrados alrededor de una recta con pendiente negativa, entonces $\rho(X, Y)$ será cercano a -1.

En Python, el valor de retorno de la función `corrcoef(x, y)` es una matriz bidimensional que puede considerarse como una matriz de covarianza definida como sigue:

$$\begin{bmatrix} \rho(X, X) & \rho(X, Y) \\ \rho(Y, X) & \rho(Y, Y) \end{bmatrix},$$

usando las ecuaciones 2.2, 2.5 y la definición de $\sigma_X = \sqrt{Var(X)}$ se puede llegar a

$$\rho(X, X) = \frac{Cov(X, X)}{\sigma_x \sigma_x} = \frac{Var(X, X)}{Var(X, X)} = 1. \quad (2.11)$$

Así que los elementos diagonales de esta matriz son siempre iguales a 1. De la definición de covarianza, se puede ver que $Cov(X, Y) = Cov(Y, X)$ y $\rho(X, Y) = \rho(Y, X)$. Por lo tanto, los elementos fuera de diagonal de la matriz de covarianza son siempre iguales y cualquiera de ellos da la correlación entre X e Y .

2.2.3. Singular Value Decomposition (SVD)

Sea X una matriz de $m \times n$ de datos de valor real y rango r , donde sin pérdida de generalidad $m \geq n$ y, por tanto, $r \leq n$. La ecuación de descomposición del valor singular de X es la siguiente:

$$X = USV^T, \quad (2.12)$$

donde U es una matriz de $m \times n$, S es una matriz diagonal de $n \times n$, y V^T es también una matriz de $n \times n$. Las columnas de U se denominan vectores propios izquierdos, $\{\mathbf{u}_k\}$. Las filas de V^T contienen los elementos de los vectores propios de la derecha, $\{\mathbf{v}_k\}$. Los elementos de S solo son distintos de cero en la diagonal, y se llaman los **valores propios**. Por lo tanto, $S = \text{diag}(s_1, \dots, s_n)$. Además, $s_k > 0$ para $1 \leq k \leq r$, y $s_i = 0$ para $(r + 1) \leq k \leq n$ [30].

2.3. Modelos y algoritmos de procesamiento de señales asociados al proceso de clasificación

2.3.1. Cadenas de Markov

Un proceso de Markov es un proceso estocástico que satisface la propiedad de Markov (a veces caracterizada como “ausencia de memoria”), que básicamente establece que se pueden hacer predicciones para el futuro del proceso basándose únicamente en su estado actual, es decir, condicionado por el estado actual del sistema, sus estados futuros y pasados son independientes [31].

A modo de ejemplo, se puede considerar un sistema que puede describirse en cualquier momento como uno de un conjunto de N estados distintos, S_1, S_2, \dots, S_N , como se ilustra en la Figura 2.3 (donde $N = 5$ para simplificar).

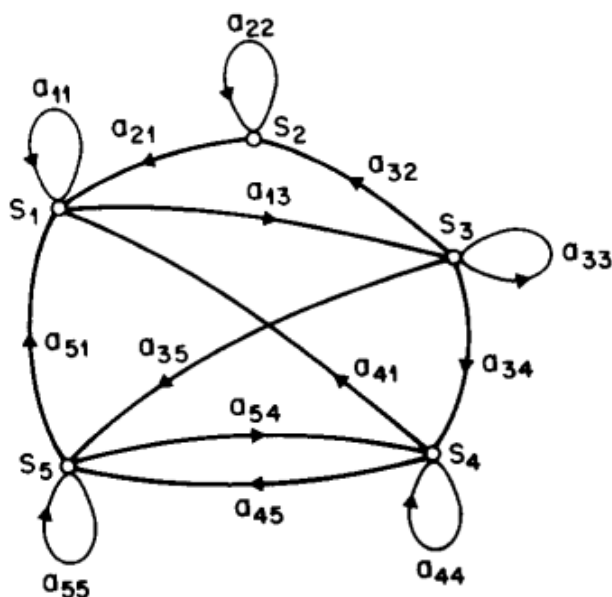


Figura 2.3: Una cadena de Markov con 5 estados (etiquetados de S_1 , a S_5 ,) con transiciones de estado (etiquetados como a_{ij} que pueden estar definidos o pueden ser números no conocidos). Obtenida de [31].

En momentos discretos regularmente espaciados, el sistema experimenta un cambio de estado (posiblemente de vuelta al mismo estado) según un conjunto de probabilidades asociadas al estado. Se denotan los instantes de tiempo asociados a los cambios de estado como $t = 1, 2, \dots$ y el estado real en el momento t se denomina q_t . Una descripción probabilística completa del sistema anterior requeriría, en general, la especificación del estado actual (en el momento t), así como de todos los estados predecesores. Para el caso especial de una cadena de Markov discreta y de primer orden, esta descripción probabilística se reduce al estado actual y al estado anterior, es decir:

$$\begin{aligned}
P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots] \\
= P[q_t = S_j | q_{t-1} = S_i].
\end{aligned}
\tag{2.13}$$

Además, solo se consideran aquellos procesos en los que el lado derecho de la Ecuación (2.13) es independiente del tiempo (Cadena de Markov homogéneas), lo que conduce al conjunto de probabilidades de transición de estado a_{ij} de la forma

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i], 1 \leq i, j \leq N, \tag{2.14}$$

a_{ij} son los coeficientes de transición de estado que cumplen:

$$\begin{aligned}
a_{ij} &\geq 0 \\
\sum_{j=1}^N a_{ij} &= 1,
\end{aligned}
\tag{2.15}$$

ya que obedecen a restricciones de modelos de probabilidad. El proceso estocástico anterior podría denominarse modelo de Markov observable, ya que la salida del proceso es el conjunto de estados en cada instante de tiempo, donde cada estado responde a un evento físico (observable). Luego, la probabilidad de observar una secuencia dada puede calcularse en este caso mediante el siguiente procedimiento sencillo:

$$\begin{aligned}
P(O = \{S_{n_0}, S_{n_1}, \dots, S_{n_l}\} \mid \text{Model}) \\
= P[S_{n_0}] \cdot P[S_{n_1} \mid S_{n_0}] \cdots P[S_{n_l} \mid S_{n_{l-1}}] \\
= \pi_0(n_0) \cdot a_{n_0 n_1} \cdots a_{n_{l-1} n_l},
\end{aligned}
\tag{2.16}$$

donde se utiliza la notación $\pi_0(n_0)$ como:

$$\pi_0(n_0) = P[q_0 = S_0]. \tag{2.17}$$

2.3.2. Modelos Ocultos de Markov (HMM)

En esta sección se amplía el concepto de modelos de Markov para incluir el caso en el que la observación es una función probabilística del estado, es decir, el modelo resultante (que se denomina modelo de Markov oculto) es un proceso estocástico totalmente integrado con un proceso estocástico subyacente que no es observable (está oculto), pero que solo puede observarse a través de otro conjunto de procesos estocásticos que producen la secuencia de observaciones [31].

El adjetivo “oculto” se refiere a la secuencia de estados por la que pasa el modelo, no a los parámetros del mismo; el modelo se sigue denominando modelo de Markov oculto, aunque se conozcan exactamente estos parámetros. Las mediciones están vinculadas a los estados del sistema a través de una función de densidad de probabilidad condicional.

Como consecuencia, el modelo resultante tiene dos fuentes de incertidumbre que afectan al problema de la inferencia: (i) la dinámica de los estados ocultos y (ii) el ruido de las mediciones [31].

Ahora se define formalmente los elementos de un HMM, y se explicará cómo el modelo genera secuencias de observación. Un HMM se caracteriza por lo siguiente:

- \mathbf{N} , el número de estados del modelo. Aunque los estados están ocultos, para muchas aplicaciones prácticas suele haber algún significado físico ligado a estos. Por lo general, los estados están interconectados de tal manera que se puede llegar a cualquiera desde cualquier otro. Se denotan los estados individuales como $S = \{S_1, S_2, \dots, S_N\}$, y el estado en el momento t como q_t [31].
- \mathbf{M} , el número de símbolos de observación distintos por estado. Los símbolos de observación corresponden a la salida física del sistema modelado. Se denotan los símbolos individuales como $V = \{v_1, v_2, \dots, v_M\}$ [31].
- La distribución de probabilidad de transición de estados $\mathbf{A} = \{a_{ij}\}$ donde

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N. \quad (2.18)$$

Para el caso especial en el que cualquier estado puede alcanzar cualquier otro en un solo paso, se tiene que $a_{ij} > 0$ para todo i, j . Para otros tipos de HMM, se tendría que $a_{ij} = 0$ para uno o más pares (i, j) .

- La distribución de probabilidad del símbolo de observación en estado j , $\mathbf{B} = \{b_j(k)\}$, donde

$$b_j(k) = P[v_k \text{ dado } t | q_t = S_j], 1 \leq j \leq N \\ 1 \leq k \leq M. \quad (2.19)$$

- La distribución del estado inicial $\pi = \{\pi_i\}$ donde

$$\pi_i = P[q_i = S_i], 1 \leq i \leq N. \quad (2.20)$$

Dados los valores adecuados de N , M , A , B y π , el HMM puede utilizarse como generador para obtener una secuencia de observación.

$$O = O_1 O_2 \dots O_T, \quad (2.21)$$

donde cada observación O_t es uno de los símbolos de V y T es el número de observaciones de la secuencia.

De lo anterior se desprende que la implementación completa de un HMM requiere la especificación de dos parámetros del modelo (N y M), de la observación y de las tres medidas de probabilidad A , B y π . Por comodidad, se utiliza la notación compacta para indicar el conjunto completo de parámetros del modelo.

$$\lambda = (A, B, \pi). \quad (2.22)$$

2.3.3. Algoritmo k-means

k-means es un algoritmo no supervisado de Clustering. Se utiliza cuando se tiene un conjunto de datos sin etiquetar. El objetivo de este algoritmo es el de encontrar “k” grupos (clusters) entre los datos crudos.

Cómo funciona k-means

El algoritmo trabaja iterativamente para asignar a cada “punto” (las filas del conjunto de entrada forman una coordenada) uno de los “k” grupos basado en sus características. Son agrupados basándose en la similitud de sus features (las columnas). Como resultado de ejecutar el algoritmo tendremos:

- Los **centroides** de cada grupo que serán unas coordenadas de cada uno de los “k” conjuntos que se utilizarán para poder etiquetar nuevas muestras.
- Etiquetas para el conjunto de datos de entrenamiento. Cada etiqueta perteneciente a uno de los “k” grupos formados.

Los grupos se van definiendo de manera orgánica, es decir, que se va ajustando su posición en cada iteración del proceso, hasta que converge el algoritmo. Una vez hallados los centroides se deben analizar para ver cuáles son sus características únicas, frente a la de los otros grupos. Estos grupos son las etiquetas que genera el algoritmo.

Casos de Uso de k-means

El algoritmo de clustering k-means es uno de los más usados para encontrar grupos ocultos sobre un conjunto de datos no etiquetado. Esto puede servir para confirmar (o desechar) alguna teoría que se tiene sobre los datos, y también puede ayudar a descubrir relaciones interesantes entre conjuntos de datos que, de manera manual, sería complicado de reconocer.

Una vez que el algoritmo ha ejecutado y obtenido las etiquetas, será fácil clasificar nuevos valores o muestras entre los grupos obtenidos.

Algunos casos de uso son:

- Segmentación por Comportamiento, como por ejemplo, relacionar el carrito de compras de un usuario, sus tiempos de acción e información del perfil.
- Categorización de Inventario: agrupar productos por actividad en sus ventas
- Detectar anomalías o actividades sospechosas: según el comportamiento en una web, reconocer un bot de un usuario normal

Datos de Entrada para k-means

Las features o características que se utilizan como entradas para aplicar el algoritmo k-means deben ser de valores numéricos, continuos en lo posible. En caso de valores categóricos (por ej. Hombre/Mujer o Ciencia Ficción, Terror, Novela, etc.) se puede intentar pasarlo a valor numérico, pero no es recomendable, pues no hay una “distancia real” (como en el caso de géneros de película o libros). Además, es recomendable que los valores utilizados estén normalizados, manteniendo una misma escala. En algunos casos también funcionan mejor datos porcentuales en vez de absolutos. No conviene utilizar features que estén correlacionados o que sean escalares de otros.

El procedimiento del Algoritmo k-means

El algoritmo utiliza un proceso iterativo en el que se van ajustando los grupos para producir el resultado final. Para ejecutar el algoritmo, se entrega como entrada el conjunto de datos y un valor de “k”. El conjunto de datos serán las características o features para cada punto. Las posiciones iniciales de los “k” centroides serán asignadas de manera aleatoria de cualquier punto del conjunto de datos de entrada. Luego se itera en dos pasos:

1- Paso de Asignación de datos

En este paso, cada “fila” de nuestro conjunto de datos se asigna al centroide más cercano basado en la distancia cuadrada Euclidiana. Se utiliza la siguiente fórmula (donde $\text{dist}()$ es la distancia Euclidiana estándar):

$$\arg \min_{c_i \in C} \text{dist}(c_i, x)^2.$$

2- Paso de actualización de Centroide

En este paso los centroides de cada grupo son recalculados. Esto se hace tomando una media de todos los puntos asignados en el paso anterior.

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i. \quad (2.23)$$

El algoritmo itera entre estos pasos hasta cumplir un criterio de detención:

- si no hay cambios en los puntos asignados a los grupos.
- o si la suma de las distancias se minimiza.
- o se alcanza un número máximo de iteraciones.

Es importante destacar que el algoritmo converge a un resultado que puede ser el óptimo local, por lo que es conveniente volver a ejecutar más de una vez con puntos iniciales aleatorios para confirmar si hay una salida mejor.

Elegir el valor del parámetro “k” en algoritmo de k-means

Este algoritmo funciona preseleccionando un valor de “k”. Para encontrar el número de clusters en los datos, se debe ejecutar el algoritmo para un rango de valores “k”, ver los resultados y comparar características de los grupos obtenidos. En general no existe un modo exacto de determinar el valor “k”, pero se puede estimar con aceptable precisión siguiendo la técnica de la distancia media entre los puntos de datos y su centroide. Como el valor de la media disminuirá a medida que aumenta el valor de “k”, se debe utilizar la distancia media al centroide en función de “k” y encontrar el punto de inflexión de la curva.

2.3.4. Algoritmo de Viterbi

El algoritmo de Viterbi tiene como objetivo encontrar la estimación óptima de una secuencia de estados ocultos (denominada trayectoria de Viterbi) en un HMM, condicionada a un conjunto de medidas del sistema. Esta tarea se realiza mediante una formulación de programación dinámica, en la que el problema de inferencia se divide en una serie de pequeñas etapas (indexadas por el tiempo asociado a cada observación). En cada etapa, el algoritmo encuentra el valor óptimo para el estado dentro de la secuencia, y continúa el análisis hacia la siguiente etapa de manera inductiva. Formalmente, para encontrar la secuencia óptima de estados ocultos $Q^* = \{q_1^* q_2^* \dots q_T^*\}$ en una realización de un HMM, condicionada a una secuencia de medidas del sistema $O = \{O_1 O_2 \dots O_T\}$, se define la siguiente variable [31, 32]:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_{t-1}, O_1 O_2 \dots O_T | \lambda], \quad (2.24)$$

donde $\delta_t(i)$ es el camino más probable para el HMM en el tiempo t , considerando las primeras t observaciones y el estado S_i como condiciones terminales. Por inducción, es posible escribir

$$\delta_{t+1}(i) = \left[\max_j \delta_t(j) a_{ij} \right] \cdot B_j(O_{t+1}). \quad (2.25)$$

Como resultado, el problema de inferencia se resuelve utilizando el pseudocódigo que se muestra en el Anexo A.2.

Densidades de observación continuas en HMMs

Hasta este punto, se ha considerado solo el caso en el que las observaciones se caracterizaban como símbolos discretos elegidos de un alfabeto finito y, por lo tanto, se podría utilizar una densidad de probabilidad discreta dentro de cada estado de este modelo [31]. El problema de este enfoque, al menos para efectos de esta tesis, es que las observaciones son señales continuas (vectores). Por lo tanto, se deben utilizar HMMs con densidades de observación continuas.

Para utilizar una densidad de observación continua, hay que imponer algunas restricciones a la forma de la función de densidad de probabilidad del modelo para garantizar que los parámetros de esta puedan recalcularse de forma coherente. Para efectos de esta tesis no se desarrollará mayor detalle sobre la forma en que se pueden extender los HMM hacia un espacio continuo, no obstante es importante saber que es factible (para mayor detalle revisar la Ref. [31]).

Capítulo 3

Metodología

En esta sección se detallarán los procedimientos necesarios para entender los resultados obtenidos que se mostrarán en la siguiente sección. Existen 2 metodologías de procesamiento de los datos que se analizan de forma separada: el primero (desde ahora llamado caso 1) consta de dos señales, cada uno en un estado distinto, con una larga duración, representando a una sola persona con los efectos de LSD por largo tiempo y a la misma persona, pero con en el estado “placebo” por la misma cantidad de tiempo. El segundo procedimiento (desde ahora llamado caso 2) consta de 100 señales para “LSD” y 100 señales para “placebo”, pero con menor duración. En el caso 2, cada señal en un estado representa una persona distinta, sin embargo, cada persona tiene una representación tanto en “LSD” como en “placebo”. Cabe destacar que la sección de resultados se centrará en aquel caso que logra generar mayores certezas respecto a la calidad de los resultados obtenidos.

La metodología se dividirá en 7 etapas que detallarán cada uno de los procedimientos mencionados:

- Señales BOLD; en esta etapa se especificará al detalle las características de la función DMF en código, las modificaciones de los parámetros correspondientes para obtener un estado u otro, y los filtros necesarios obtener de manera definitiva las señales BOLD.
- Preprocesamiento; en esta etapa se explican las modificaciones a las señales BOLD para obtener señales lo más depuradas posibles.
- Matrices de correlación; aquí se detalla la manera en que se obtiene información relevante de las señales.
- Extracción de características principales; en esta etapa se mencionan los distintos métodos para obtener las características más relevantes y el método de comparación de estos.
- Método de clasificación; acá se explica como se obtiene el vector de características final y se diseña el conjunto de entrenamiento del algoritmo de Viterbi.
- Mejoras de clasificación por bloques; en esta etapa se describe un procedimiento con el cual se obtienen mejores resultados de clasificación.
- Dinámica de cada estado cerebral; en esta etapa se menciona el uso del algoritmo que ayuda a obtener las características intrínsecas de cada estado.

3.1. Señales BOLD

La señal BOLD es una medida indirecta de la actividad neuronal que parte del supuesto de la existencia de un acoplamiento neurovascular ante una demanda funcional. El análisis de la naturaleza de la actividad neuronal a través de la señal BOLD es, por tanto, una inferencia a partir de la respuesta hemodinámica, y no de la respuesta neuronal en sí misma [33]. En resumen, la señal BOLD es una buena aproximación medible para inferir la actividad neuronal en el cerebro, y como tal, es usada en esta tesis para obtener información del cerebro en un estado dado.

Las señales BOLD de las distintas zonas del cerebro se obtienen a través de simulaciones computacionales del modelo cerebral DMF utilizado en Deco et al. [4]. El código que se presenta en el trabajo de Gustavo Deco está escrito en lenguaje Python y sirve para simular las señales sintéticas “LSD” y “placebo”. Sin embargo, tanto el código, los parámetros y la matriz de conectividad se obtuvieron a través del mismo código escrito en lenguaje MATLAB. Este contiene, a modo de función, el modelo DMF descrito en el marco teórico, cuyas entradas corresponden a: **los parámetros** del modelo, el número de pasos (**steps**) y el **modo de uso** del modelo.

Con respecto a los parámetros del modelo, existe un objeto del tipo clase que tiene todas las características por defecto de los parámetros del modelo (descritos con detalle en el Anexo A.3). Para adaptar el modelo a un estado “LSD” se dejan intactos la mayoría de los parámetros a excepción de algunos, de igual manera para el estado “placebo”. Los parámetros cambiados son los siguientes:

- **C**: Es la matriz del conectoma, el cual se normaliza y se pondera por 0.2
- **receptors**: Es el mapa regional de receptores de fármacos. Este se normaliza dividiéndolo por el valor máximo del mismo.
- **G**: Parámetro que escala el acoplamiento global de las poblaciones neuronales. Se impone como 2.1, ya que este es el valor óptimo para efectos del experimento [4].
- **seed**: Se utiliza para no perder las condiciones iniciales azarosas del modelo y poder comparar un estado con otro.
- **wgain**: Parámetro neuromodulador que escala los efectos del neurotransmisor en la función de ganancia neuronal ponderada por la densidad de recepción regional empírica llamado sE. El valor debe ser 0.2 para simular el estado “LSD” y 0 para simular el estado “placebo” [4].

Es importante mencionar que para el caso 2, al momento de simular señales para una persona diferente, el parámetro **seed** debe cambiar. Esta es una buena aproximación para simular señales de distintas personas, sin embargo, tanto para el caso 1 como para el caso 2, este parámetro debe mantenerse igual para simular ambos estados de una misma persona, esto con el fin de poder compararlos.

El parámetro **steps** tiene el propósito de determinar la cantidad de **unidades de tiempo** que tendrá la señal de salida. Cada 2000 steps se obtiene una unidad de tiempo y cada unidad de tiempo equivale a una muestra de señal tomada cada 2 segundos en tiempo real. Por limitaciones computacionales, se estableció que la duración total de cada señal sea de, 27100 unidades de tiempo para el caso 1 y para el caso 2 se estableció que la duración de cada persona sea de 220 muestras de tiempo, con un total de 100 personas por estado.

Y por último, los modos de operación se pueden establecer de 3 modos diferentes, los cuales son: **bold**, **rate** y **both**. Para efectos de esta tesis solo es de interés el modo **bold** para ambos casos.

Una vez obtenidas las señales, estas se filtran de acuerdo a las especificaciones del estudio [4]. A continuación se muestra la extracción de código para calcular la matriz BOLD en condiciones de estado “LSD” y su respectivo filtro:

```

1  % se calcula la matriz bold en estado LSD
2  bold_LSD = DMF(params_lsd , nb_steps , 'bold ');
3  [B, A] = butter(2, [0.01, 0.1]*2*params.TR);
4  BOLDLSD = filter(B, A, bold_LSD ');

```

Donde `params_lsd` corresponde a la clase con parámetros modificados para simular señales “LSD”. El caso es análogo para las señales en estado “placebo”, pero cambiando el parámetro `wgain` como se mencionó anteriormente. Luego estas señales se guardan como archivos “.mat” para luego ser importadas en código Python con el fin de lograr una mayor manejo del código.

3.2. Preprocesamiento

El preprocesamiento de las señales es una parte esencial para limpiar datos de sus impurezas o información innecesaria que no aporta al objetivo final. Como tal, consiste en la revisión, visualización y análisis de los datos para diagnosticar cuáles son los datos ruidosos que no aportan información y quitarlos. En este caso en particular los datos corresponden a 90 señales en el tiempo, por lo que se debe observar el comportamiento que estos tienen e identificar que parte de las señales son las que corresponden a un comportamiento de común de señales BOLD.

Debido a los efectos propios del modelo computacional del cerebro en las primeras 30 unidades de tiempo aproximadamente, las señales muestran un comportamiento transitorio indeseado, el cual tiende a estabilizarse y pasar a un estado permanente donde las señales oscilan en torno a 0. Este último estado muestra el comportamiento estacionario de cada estado cerebral y, por tanto, solo se utilizan las señales desde el segundo 60 aproximadamente en adelante. El tiempo de corte elegido no tiene un respaldo matemático detrás, solo se buscó un tiempo desde el cual las señales BOLD se encuentren en estado permanente.

En las imágenes 3.1 y 3.2 se muestra una comparación de las señales “placebo” y “LSD” preprocesadas respectivamente. De igual manera, en las Figuras 3.3 y 3.4 se muestra una

comparación de las mismas señales recortadas. Este recorte de señal provoca que la cantidad de unidades de tiempo de las señales del caso 1 se reduzcan a 27070 muestras. Por otro lado, en el caso 2, las señales obtenidas fueron en un principio de largo 250 unidades de tiempo, de manera que, al preprocesarlas, quedaran de largo 220 unidades de tiempo.

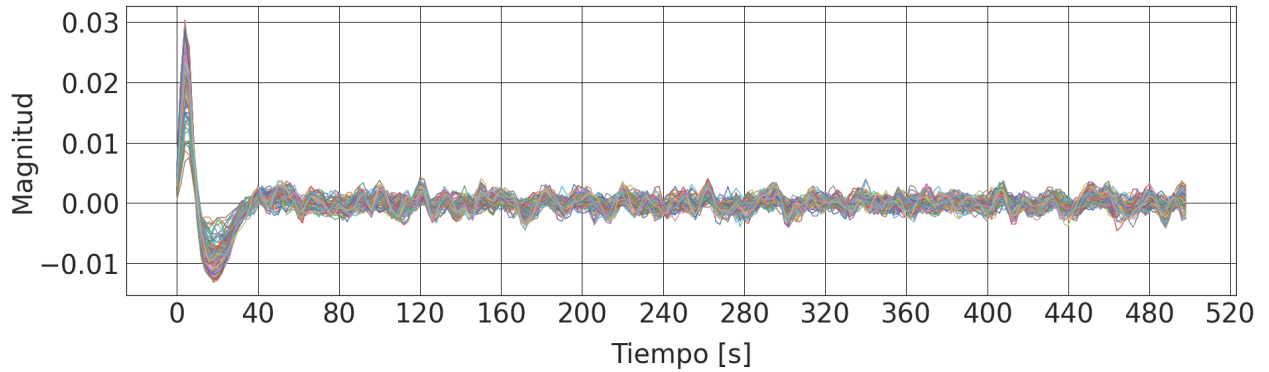


Figura 3.1: 90 Señales BOLD de tipo “placebo” preprocesadas

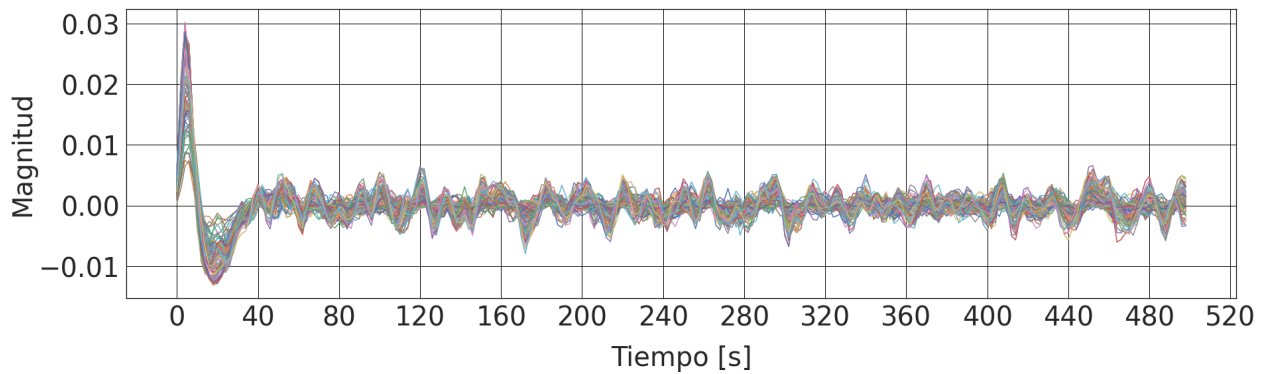


Figura 3.2: 90 Señales BOLD de tipo “LSD” preprocesadas

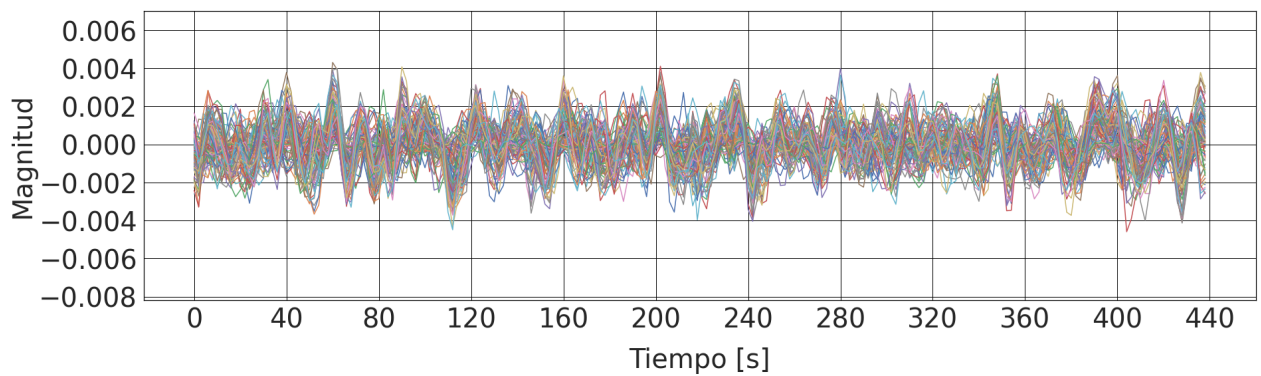


Figura 3.3: 90 Señales BOLD de tipo “placebo” procesadas

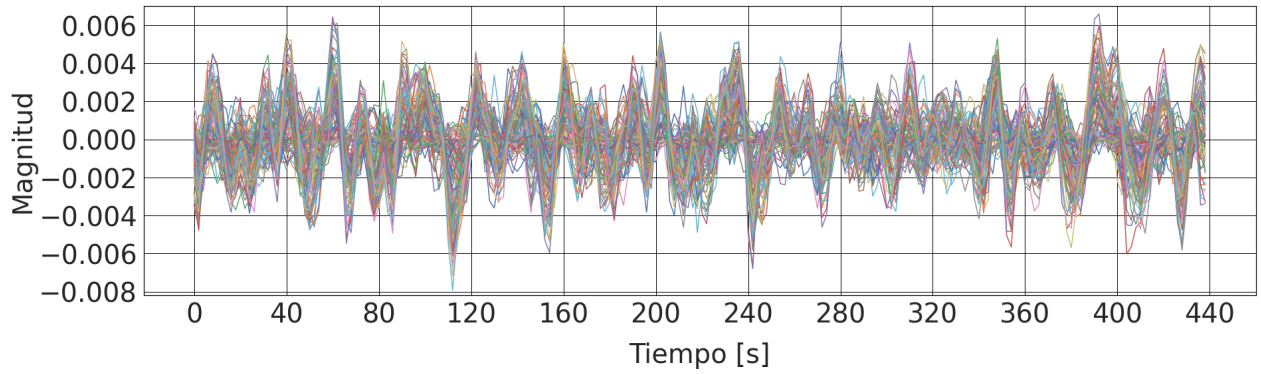


Figura 3.4: 90 Señales BOLD de tipo “LSD” procesadas

3.3. Matrices de correlación

Como se observó en las imágenes 3.3 y 3.4, la parte estacionaria por si sola no muestra información relevante y distintiva de un estado con respecto al otro, por lo que se utilizan las matrices de correlación con el fin de obtener información de manera visual de las señales BOLD procesadas, ya que estas muestran la interacción en el tiempo de las distintas zonas cerebrales. Para esto, se calcula una serie de matrices de correlación entre las 90 señales cerebrales en un tiempo (desde ahora **ventana**) de 60 muestras de tiempo cada una y una distancia temporal (desde ahora **slide**) de 5 muestras de tiempo de diferencia entre una ventana y otra. Las medidas de tiempo anteriormente señaladas se impusieron con el fin de tener una ventana lo suficientemente duradera como para que la correlación estimada entre señales sea representativa y el **slide** se eligió con el fin de que las ventanas estuvieran lo suficientemente separadas pero sin perder mucha información entre una y otra.

La Figura 3.5 representa de manera visual el procedimiento descrito anteriormente.

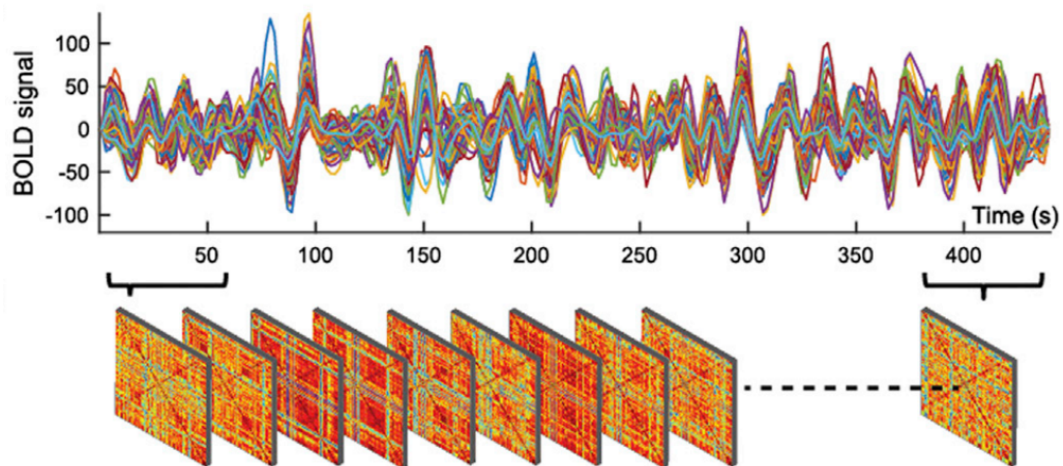


Figura 3.5: Ilustración de la extracción de matrices de correlación obtenido desde señales BOLD. Obtenida de [4]

Cada valor (i, j) de la matriz t , representa la correlación que tuvo la zona i con la zona j del cerebro en el intervalo de tiempo t . Por tanto, cada matriz es simétrica, con dimensiones de 90×90 y su diagonal tiene valores iguales a 1. En total son 5403 matrices por estado cerebral para el caso 1 y 33 matrices por persona, dando un total de 3300 matrices para el caso 2 por estado. Cabe mencionar que, para el caso 2, solo se calcularon la matrices de correlación para una misma persona y no entre personas distintas, ya que esto último carece de sentido físico. En la Figura 3.6 se muestra un ejemplo visual de como se ve una de estas matrices de correlación.

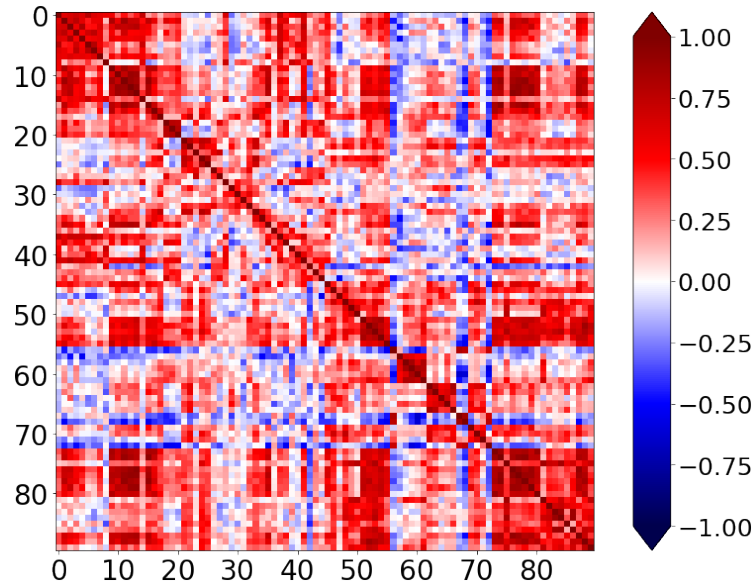


Figura 3.6: Ejemplo de matriz de correlación de señales BOLD.

3.4. Extracción de características principales

Si bien una matriz de correlación muestra información acerca de la sincronización entre las regiones cerebrales, lo cierto es que dado que son simétricas y con diagonal igual a 1, cada matriz aporta 4950 datos no repetidos y no triviales (valores de la parte triangular superior), lo cual es demasiada información para ser considerada como una extracción de las características principales de las señales. Por lo tanto, es necesario incluir un método de extracción de características de las matrices que, a través de un vector de no más de 20 valores, represente de buena manera las características principales de cada matriz. Para esto se probaron 5 métodos de extracción de características:

- Singular Value Descomposition [30]: cuyo vector de características corresponde a los principales valores propios de cada matriz.
- *Transformada Wavalet*[34]: cuyo vector de características corresponde a los componentes en el dominio Wavalet más grandes de la transformada de cada matriz.
- *Red Neuronal Variational Autoencoder Densa* [35]

- *Red Neuronal Variational Autoencoder Convolutional* [36]
- *Red Neuronal Autoencoder Convolutional* [37]

Para los últimos 3 métodos el vector de características corresponde al sector del encoder de cada red. Es importante señalar que cada matriz de correlación se normalizó para que los valores varíen entre 0 y 1, esta transformación lineal se hizo porque estos 3 últimos métodos mejoraban considerablemente sus resultados.

A cada método se le evaluó la capacidad de reconstrucción para todas las matrices utilizando el **NMSE** (Normalized Mean Square Error) entre las matrices originales y las reconstruidas. El resultado, presentado en el Anexo A.1, mostró que el mejor método de extracción de características es el Singular Value Descomposition y, por tanto, el vector de características de cada matriz corresponde a los valores propios de cada matriz.

3.5. Método de clasificación

Una vez obtenido el método de extracción de características, es necesario notar que, si bien los valores propios más grandes reconstruyen (junto a sus vectores propios) bastante bien las matrices y, por tanto, son valores representativos de estas, desde el punto de vista de la clasificación de matrices (“placebo” o “LSD”) esta elección de los principales valores propios es arbitrario, es más, es posible que existan algunos valores propios que entorpezcan el procedimiento de clasificación. Es por esta razón que se determinó encontrar la combinación de valores propios que dé el mejor desempeño de clasificación. Para esto, se probaron distintas combinaciones de valores propios con el fin de encontrar (por fuerza bruta) la combinación óptima que muestre el porcentaje de error de clasificación más bajo para ambos estados. Dado que la combinación óptima para “LSD” no es necesariamente el mismo óptimo para “placebo”, se determinó que el óptimo en común es la combinación que da la suma mínima de los errores de cada estado. Esta selección asegura que para una misma combinación de valores propios el error de clasificación de ambos estados será relativamente bajo.

Sin embargo, al tener 90 valores propios, la cantidad de combinaciones posibles para determinar un vector de características óptimo es intratable computacionalmente. Por lo que se decidió probar las distintas combinaciones con un vector desde un 1 hasta máximo de 10 elementos y que este vector además no pierda demasiado su capacidad de reconstrucción de la matriz de origen, es decir, se probaron todas las combinaciones posibles de los primeros 10 valores propios y a cada combinación se le aplicó el método de clasificación para obtener el error asociado a dicho vector. En los Anexos A.2(a) y A.2(b) se muestran los errores individuales de cada estado para las distintas combinaciones del caso 1 y caso 2 respectivamente. A su vez, en los Anexos A.3(a) y A.3(b) se muestran las sumas de los errores de los gráficos anteriores respectivamente. Es importante señalar que si bien se obtiene una combinación con error mínimo cuyos valores propios corresponden a los valores 0, 1, 2, 3, 6 y 8 (desde ahora llamado **vector de características**), este vector tiene un valor de error muy similar de otras combinaciones, por lo que se podrían obtener resultados similares con distintas combinaciones de valores propios.

El problema de clasificar las señales BOLD en los dos estados cerebrales observados, puede verse como un problema en el que la secuencia temporal de los vectores de características son **observaciones** asociadas a uno de los dos **estados ocultos**. Esta forma de ver el problema sugiere que el método de clasificación por excelencia sea el **algoritmo de Viterbi** asociado a un **HMM** (de valores continuos). Por tanto, el método de clasificación se define como un HMM donde:

- **N**, el número de estados del modelo, es decir, 2 ya que se están modelando solos 2 estados.
- **M**, el número de símbolos de observación distintos por estado, es decir, el largo del vector de características.
- La distribución de probabilidad de transición de estados **A**, se define como una matriz de valores azarosos de $N \times N$.
- Las matrices de **medias** y **covarianzas** para cada estado, se definen como unas matrices de valores azarosos.
- La distribución del estado inicial π , es un vector de valores azaroso de largo 2 que suman 1.

Este modelo es entrenado con una secuencia temporal de los vectores de características consecutivos de matrices “placebo” concatenado a otra secuencia de vectores de características consecutivos de matrices “LSD” del mismo largo. Esta concatenación de vectores, simula de manera forzosa un cambio de estado de “placebo” a “LSD”. El algoritmo de Viterbi entrena, de manera no supervisada, con estos datos y no solo clasifica cada uno de estos de manera individual en un estado u otro dependiendo de las características de ese dato, sino que también encuentra un óptimo estadístico para la distribución de probabilidad de transición de estados, las matrices de medias, covarianzas y la distribución del estado inicial π inicializadas azarosamente.

A continuación, se presenta en la Figura 3.7 una representación de cada etapa del procedimiento y clasificación de los datos para ilustrar de manera gráfica y resumida lo que se explicó anteriormente.

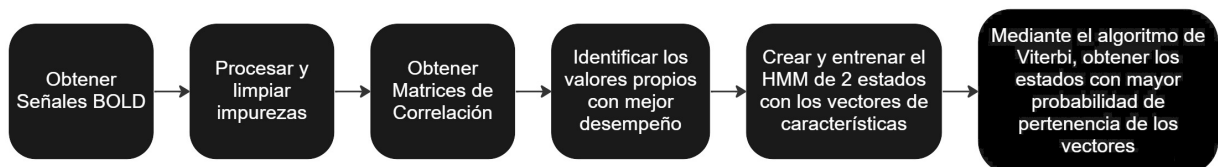


Figura 3.7: Representación de cada etapa del procedimiento y clasificación de las señales BOLD.

3.6. Mejoras de clasificación por bloques

Una de las formas para mejorar la clasificación de una señal, ya sea “placebo” o “LSD”, es notar en primer lugar que la clasificación del algoritmo de Viterbi se hace de forma **individual** para cada vector de características dependiendo de sus valores. Sin embargo, ¿qué pasaría si se agruparan 100 vectores seguidos en un **bloque** y se clasifica ese bloque entero dependiendo del estado donde pertenezca la mayoría de los vectores? Si se contaran todos los bloques de 100 vectores seguidos que existen en cada estado cerebral y se calcula el porcentaje de **error de clasificación de los bloques**, este sería muy probablemente menor al porcentaje de error de clasificación individual. Esta idea hace surgir la siguiente pregunta: ¿Cómo disminuye el error de clasificación de los bloques en función del tamaño de estos bloques?, o lo que es equivalente, ¿cómo aumenta el accuracy de clasificación por bloque en función del tamaño de los bloques? Este método de clasificación por bloques necesariamente permite conocer el largo de la muestra de una señal a partir del cual no hay error de clasificación de ese estado cerebral, lo cual es muy útil.

3.7. Dinámica de cada estado cerebral

Con el fin de detallar cada estado y darle una explicación o justificación a la clasificación anteriormente desarrollada, es necesario describir la dinámica de cada estado, es decir, mostrar o seccionar las características que los identifican.

Una de las formas en las que se propone captar la dinámica de cada estado cerebral es mediante el algoritmo k-means. El procedimiento a seguir consiste en transformar en vectores de 1×4950 cada matriz, tomando solo los valores de la parte triangular superior de estas. Una vez hecha la transformación de cada matriz, se obtiene una lista de vectores secuenciales temporalmente de tamaño $L \times 4950$ (donde “L” es la cantidad de matrices de ambos estados) y se les aplica el algoritmo k-means para clasificarlos. Es importante destacar que se debe imponer un número de clusters, ya que el algoritmo no es capaz de encontrar la cantidad de clusters óptima. Sin más detalle, se puede ver en la Figura 3.8 como el algoritmo muestra un mayor desempeño de clasificación entre más clusters tenga para clasificar.

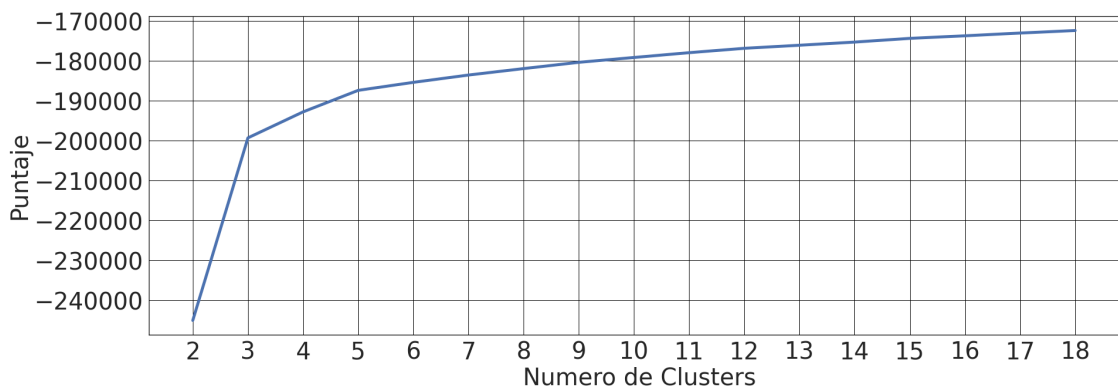


Figura 3.8: Puntaje del desempeño del algoritmo k-means en función de la cantidad de clusters que tenga para clasificar.

Por tanto, para efectos de este trabajo, se considerarán 3 y 11 clusters para mostrar de manera particular y general respectivamente el comportamiento de clasificación de las matrices y su dinámica asociada. Cada cluster tiene asociado una matriz centroide y la cantidad de matrices “placebo” y “LSD” que pertenecen al cluster, y se mostrarán en orden según el MSE entre el valor absoluto de los centroides y la matriz del conectoma normalizada, donde el centroide N^o0 es el que más semejanza tiene con el conectoma y N^o10 el que menos. Es importante aclarar que estos clusters no representan algo físico en el cerebro ni tampoco un concepto como es un estado cerebral, sino que es más bien una manera de agrupar matrices que tengan características parecidas. De este modo, los estados que tengan mayor diversidad de características tenderán a dividirse en más clusters que las características que sean menos diversas.

Capítulo 4

Resultados

En esta sección se mostrarán los resultados obtenidos del procedimiento descrito en la metodología. Solo se mostrarán los resultados del *caso 2* debido a que es el que tiene mejor desempeño de los dos casos. Esta sección se divide en 3 partes: *Clasificación de Señales BOLD*, donde se muestran resultados de la clasificación del algoritmo de Viterbi junto a sus respectivos histogramas, matrices de transición y clasificación por bloques; *Datos de testeo*, donde se muestran los mismos resultados anteriores, pero probando el modelo con los datos de testeo; *Dinámica de los datos*, donde se muestra la distribución de los datos (entrenamiento y test juntos) en 3 y 11 clusters, los centroides de estos, los respectivos histogramas y matrices de transición. En Anexos, en la sección A.6 se muestran los resultados de entrenamiento y testeo de los datos del *caso 1* junto con una tabla de comparación del desempeño entre ambos casos.

4.1. Clasificación de Señales BOLD

En esta parte de los resultados se mostrará el desempeño del algoritmo con los datos de entrenamiento que corresponden al 70% de los datos “placebo” y “LSD” en conjunto, es decir, a las primeras 70 personas en ambos estados.

4.1.1. Algoritmo de Clasificación

En la Figura 4.1 se muestra de manera gráfica el resultado de la clasificación del algoritmo de Viterbi, cuyo conjunto de entrenamiento, corresponde una concatenación de los vectores de características de “placebo” y “LSD” respectivamente, donde cada vector se representa en la imagen con una X y cada subconjunto de vectores está ordenado de manera cronológica. Cabe destacar que el algoritmo, al entrenar, clasifica cada dato en uno de los dos conjuntos distintos (0 o 1), pero el estado “placebo” no necesariamente se asociará siempre al conjunto 0 y es por eso que en la Figura 4.1 el estado “placebo” corresponde al conjunto 1. Para evitar confusiones, se decidió que tanto el caso 1 como el 2 se asocie el estado “placebo” al conjunto 1 tal y como se muestra también en la Figura A.4 en Anexo.

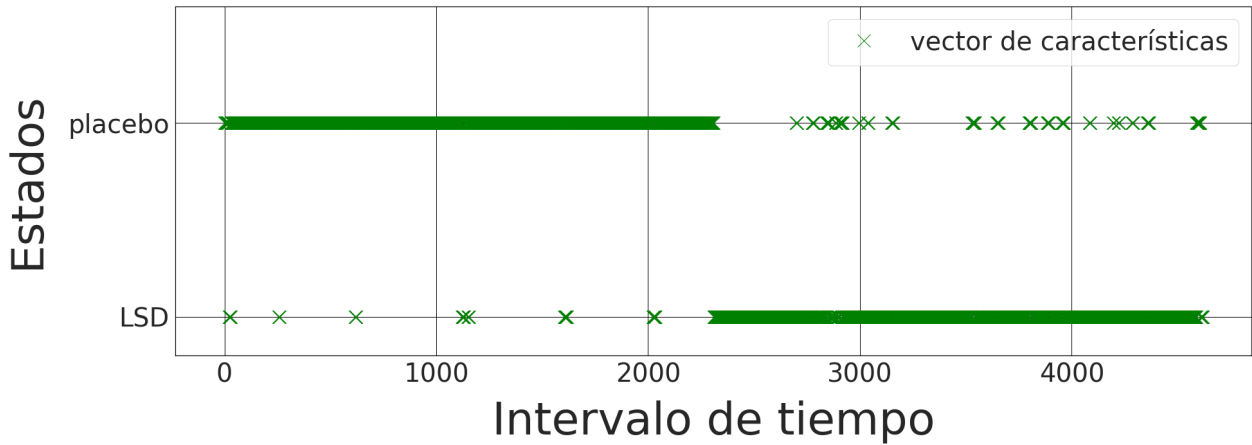
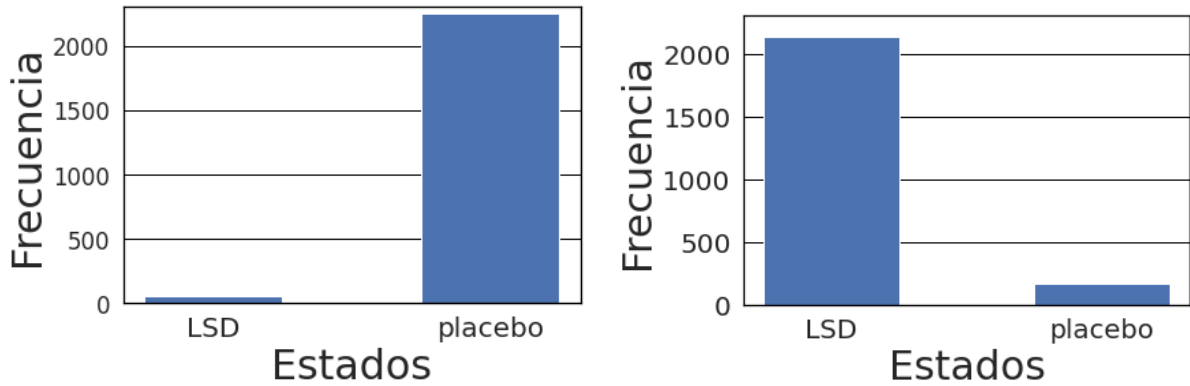


Figura 4.1: Gráfico de clasificación de los vectores de características.

4.1.2. Histogramas

Las imágenes de la Figura 4.2 se obtienen al medir la frecuencia de clasificación, en ambos estados, de los vectores de características correspondientes al estado placebo (Figura 4.2(a)) y “LSD” (Figura 4.2(b)). Cada conjunto de vectores tiene 2310 elementos en total.



(a) Histograma de clasificación de los datos “placebo”

(b) Histograma de clasificación los datos “LSD”

Figura 4.2: Histogramas de la clasificación de los vectores de características.

4.1.3. Matrices de Transición

Las Figuras 4.3(a) y 4.3(b) muestran las matrices de transición de los datos correspondientes a los estados “placebo” y “LSD” respectivamente. En cada matriz, cada celda (i, j) muestra la cantidad, en porcentaje, de transiciones de un estado i a un estado j , de manera que cada fila de las matrices suma 100%. Además, a la derecha de cada matriz se muestra la cantidad, en porcentaje, de vectores clasificados como pertenecientes a un determinado

estado.

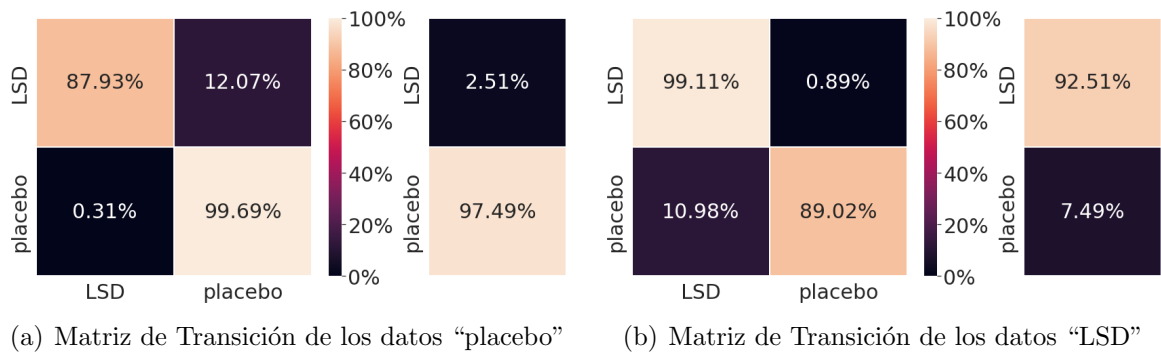


Figura 4.3: Matrices de Transición de la clasificación.

4.1.4. Clasificación por bloques

En la Figura 4.4 se muestra la evolución del accuracy de los estados "placebo" y "LSD" en función del tamaño de los bloques definidos en la metodología. Es importante destacar que un bloque de tamaño 1 (un vector de características) representa 120 segundos de tiempo, sin embargo, para obtener 2 vectores solo son necesarios 130 segundos de señal.

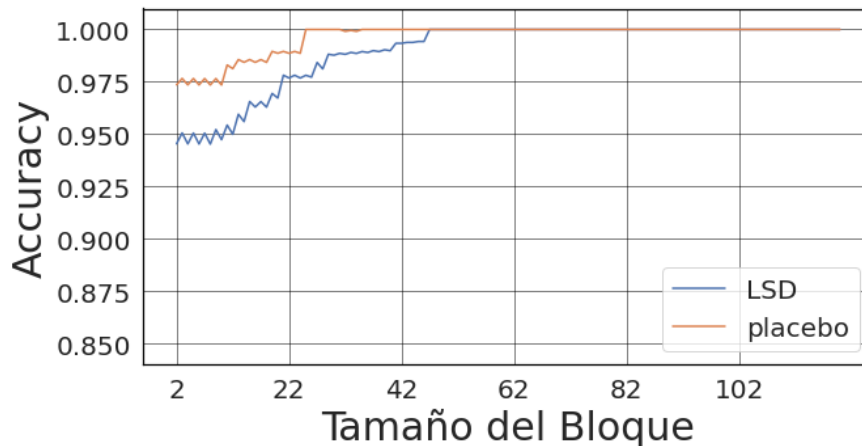


Figura 4.4: Desempeño de la clasificación del algoritmo de Viterbi en función del tamaño de los bloques de vectores.

4.2. Clasificación de Señales BOLD con datos de testeo

En esta parte de los resultados se mostrará el desempeño del algoritmo ya entrenado, pero con los datos de testeo que corresponden al 30% de los datos "placebo" y "LSD" en conjunto, es decir, a las últimas 30 personas en ambos estados.

4.2.1. Algoritmo de clasificación con datos de testeo

En la Figura 4.5 se muestra de manera gráfica el resultado de la clasificación del algoritmo de Viterbi probado con datos de testeo. Su composición, orden y representación en el gráfico es análogo a los datos de entrenamiento. Sin embargo, es importante destacar que una vez entrenado el algoritmo de Viterbi, clasificará cada dato de testeo según el mismo criterio sobre el cual clasificó el conjunto de entrenamiento, es decir, los datos que el algoritmo clasifique como “placebo” pertenecerán al conjunto 1 tal y como se hizo en la Figura 4.1.

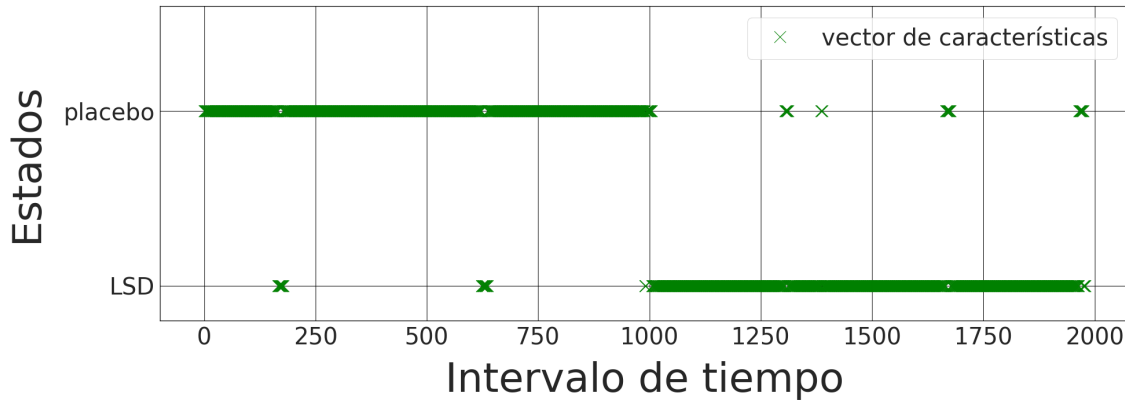
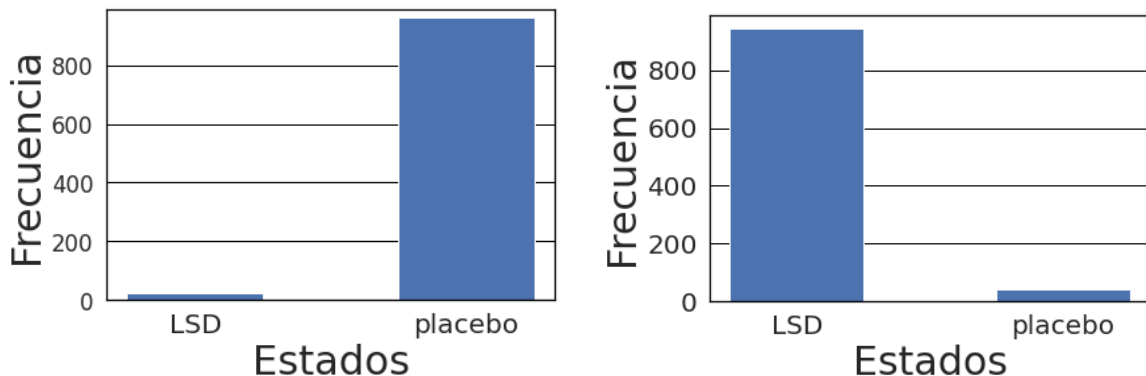


Figura 4.5: Clasificación del algoritmo de Viterbi con datos de testeo.

4.2.2. Histogramas

Las imágenes de la Figura 4.6 se obtienen al medir con qué frecuencia el algoritmo de Viterbi clasifica, en ambos estados, los vectores de características correspondientes al estado placebo (Figura 4.6(a)) y al estado “LSD” (Figura 4.6(b)). Cada conjunto de vectores tiene 990 elementos en total.

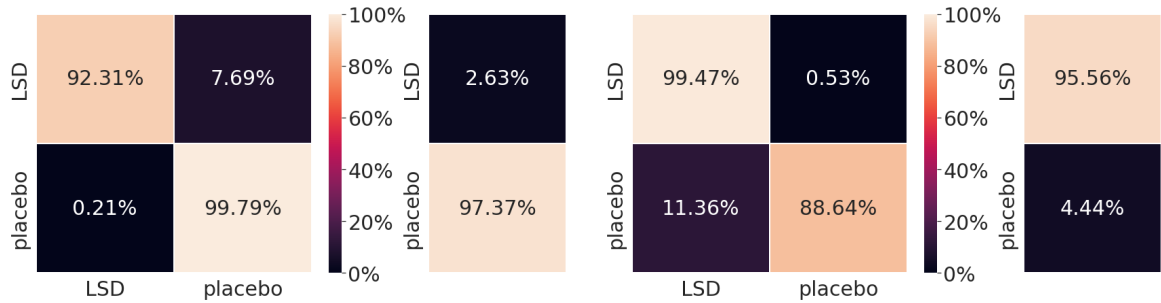


(a) Histograma “placebo” de los datos de testeo (b) Histograma “LSD” de los datos de testeo

Figura 4.6: Histogramas de la clasificación con los datos de testeo.

4.2.3. Matrices de Transición

Las Figuras 4.7(a) y 4.7(b) muestran las matrices de transición de los datos de testeo correspondientes a los estados “placebo” y “LSD” respectivamente. Las características de estas imágenes son iguales a las descritas con anterioridad en la Figura 4.3.



(a) Matriz de Transición “placebo” con datos de testeo (b) Matriz de Transición “LSD” con datos de testeo

Figura 4.7: Matrices de transición de los datos de testeo.

4.2.4. Tabla de comparación de desempeño

En la Tabla 4.1 se muestra el error de clasificación, en porcentaje del algoritmo de Viterbi para ambos estados con datos de entrenamiento y testeo.

Tabla 4.1: Comparación de porcentajes de error de clasificación de datos de entrenamiento y datos de testeo.

	Datos <i>Entrenamiento</i>		Datos <i>Testeo</i>	
	Placebo	LSD	Placebo	LSD
Error de Clasificación(%)	2.51	7.49	2.63	4.44

En la Figura 4.8 se muestra la evolución del *accuracy* de los estados “placebo” y “LSD” en función del tamaño de los bloques para los datos de testeo.

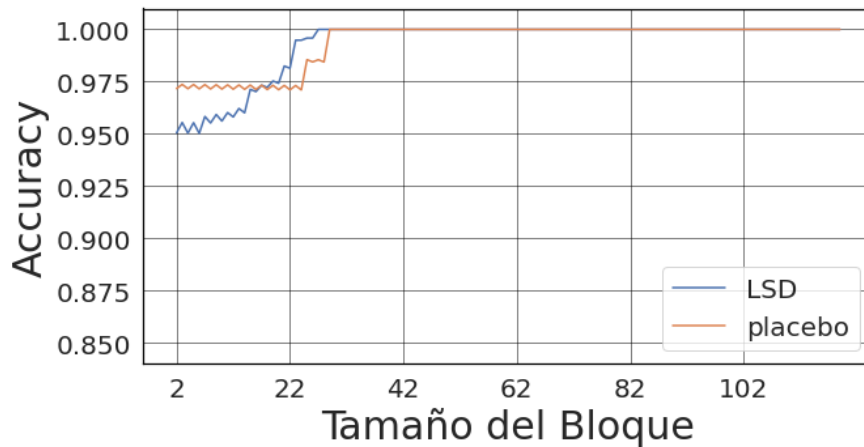


Figura 4.8: Desempeño de clasificación en función del tamaño de los bloques de vectores con datos de testeo.

4.3. Dinámica

En esta parte de los resultados se mostrará la distribución, histogramas y transición que muestra el algoritmo *k-means* con el 100 % de los datos “placebo” y “LSD” en conjunto, para 3 y 11 clusters.

4.3.1. Distribución

En la Figura 4.9 se muestra de manera gráfica el resultado de la distribución del algoritmo *k-means* para 3 clusters, cuyo conjunto de clasificación corresponde una concatenación de los vectores de características de “placebo” y “LSD” en ese orden, donde cada vector se representa en la imagen con una *X* y cada subconjunto de vectores está ordenado de manera cronológica.

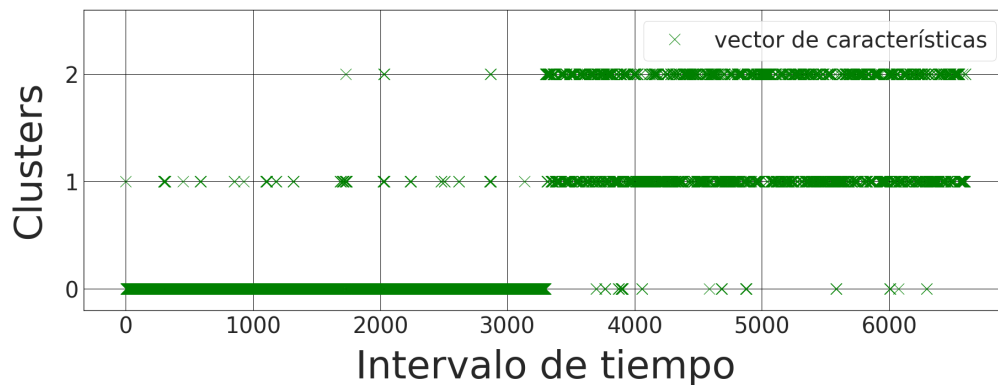


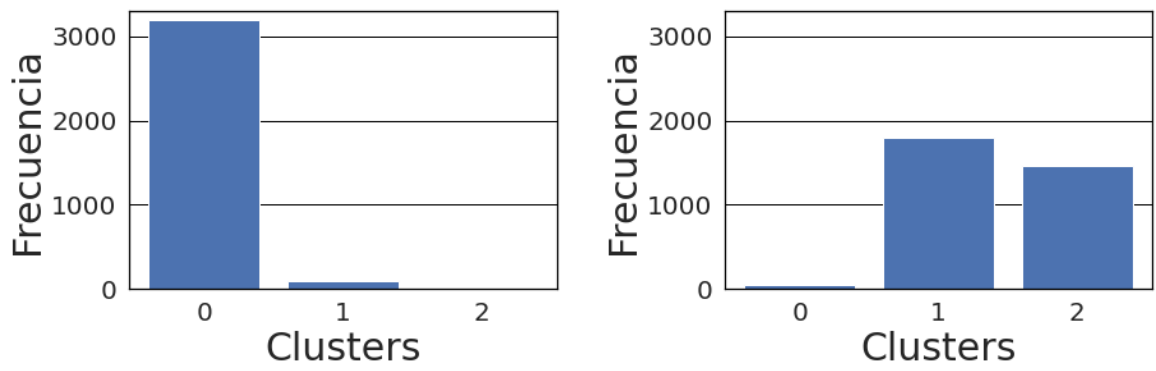
Figura 4.9: Distribución de vectores de características del algoritmo *k-means*

Es importante destacar que el objetivo del algoritmo *k-means* es clasificar los datos en *k*

clusters distintos, sin embargo, el nombre de estos clusters (0, 1 y 2) se imponen de manera azarosa. No obstante, se decidió ordenar estos clusters comparando sus respectivos centroides respecto a la matriz del conectoma, de modo que el cluster 0 tiene el centroide menos parecido con el conectoma y el centroide 2 es el que más similitud tiene.

4.3.2. Histogramas

Las imágenes de la Figura 4.10 se obtienen al medir con qué frecuencia el algoritmo k-means clasifica, en los 3 clusters, las matrices correspondientes al estado placebo (Figura 4.6(a)) y al estado “LSD” (Figura 4.6(b)). Cada conjunto de matrices tiene 3300 elementos en total, los cuales corresponden al 100% de cada estado.

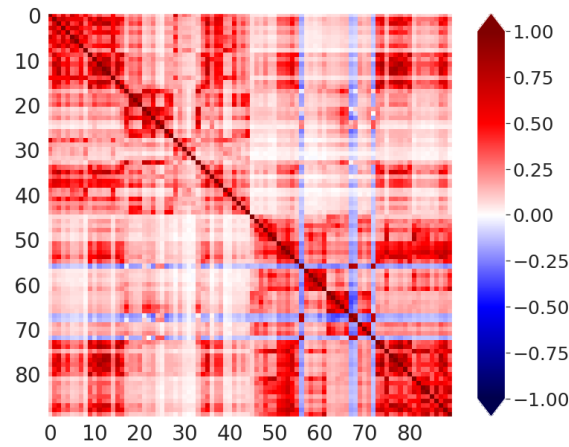


(a) Histograma de la distribución de *clusters* de los datos “placebo” (b) Histograma de la distribución de *clusters* de los datos “LSD”

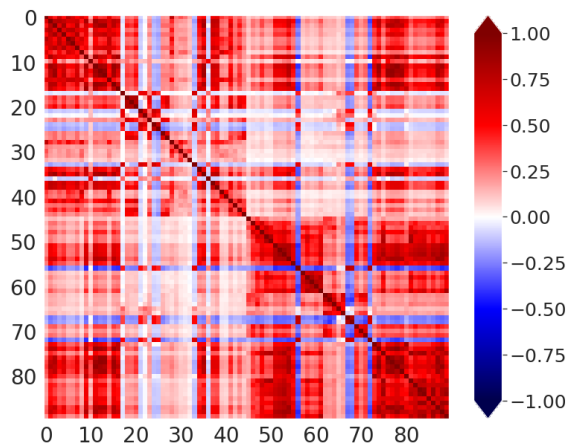
Figura 4.10: Histogramas de clasificación con 3 clusters.

4.3.3. Matrices Centroides

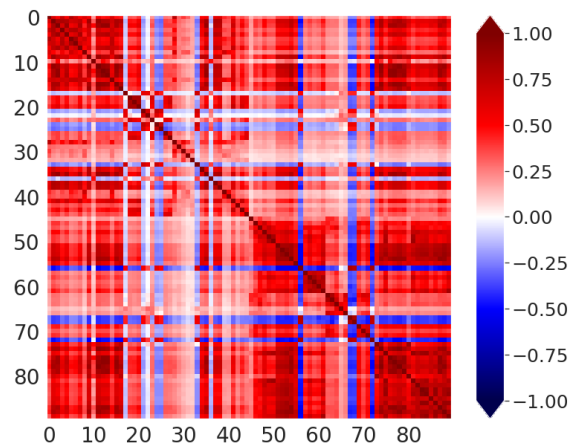
En la Figura 4.11 se muestran las matrices correspondientes a los centroides de cada cluster. En las descripciones de las Figuras 4.11(a), 4.11(b) y 4.11(c) se muestran las concentraciones, en porcentaje, de matrices “placebo” y “LSD” que existen en total.



(a) Centroide del cluster $N^{\circ}0$ con 96.94% de matrices “placebo” y 1.39% de matrices “LSD”



(b) Centroide del cluster $N^{\circ}1$ con 2.85% de matrices “placebo” y 54.39% de matrices “LSD”

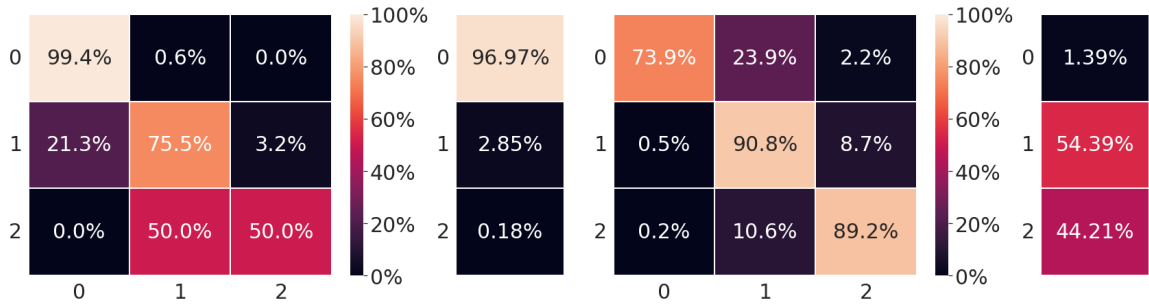


(c) Centroide del cluster $N^{\circ}2$ con 0.18% de matrices “placebo” y 44.21% de matrices “LSD”

Figura 4.11: Matrices que representan los centroides de los 3 *clusters* con sus respectivas distribuciones de matrices “placebo” y “LSD”, ordenados por orden de semejanza con el conectoma.

4.3.4. Matrices de Transición

Las Figuras 4.12(a) y 4.12(b) muestran las matrices de transición de los datos correspondientes a los estados “placebo” y “LSD” respectivamente para la distribución del algoritmo *k-means*. En cada matriz, cada celda (i, j) muestra la cantidad, en porcentaje, de transiciones de un cluster i a un cluster j , de manera que cada columna de las matrices suma 100%. Además, a la derecha de cada matriz, se muestra la cantidad, en porcentaje, de vectores clasificados como pertenecientes a un determinado cluster.



(a) Matriz de Transición de la distribución de clusters en los datos "placebo" (b) Matriz de Transición de la distribución de clusters en los datos "LSD"

Figura 4.12: Matrices de transición del algoritmo *k-means* para 3 clusters.

4.3.5. Conectoma y Centroides de los 11 clusters

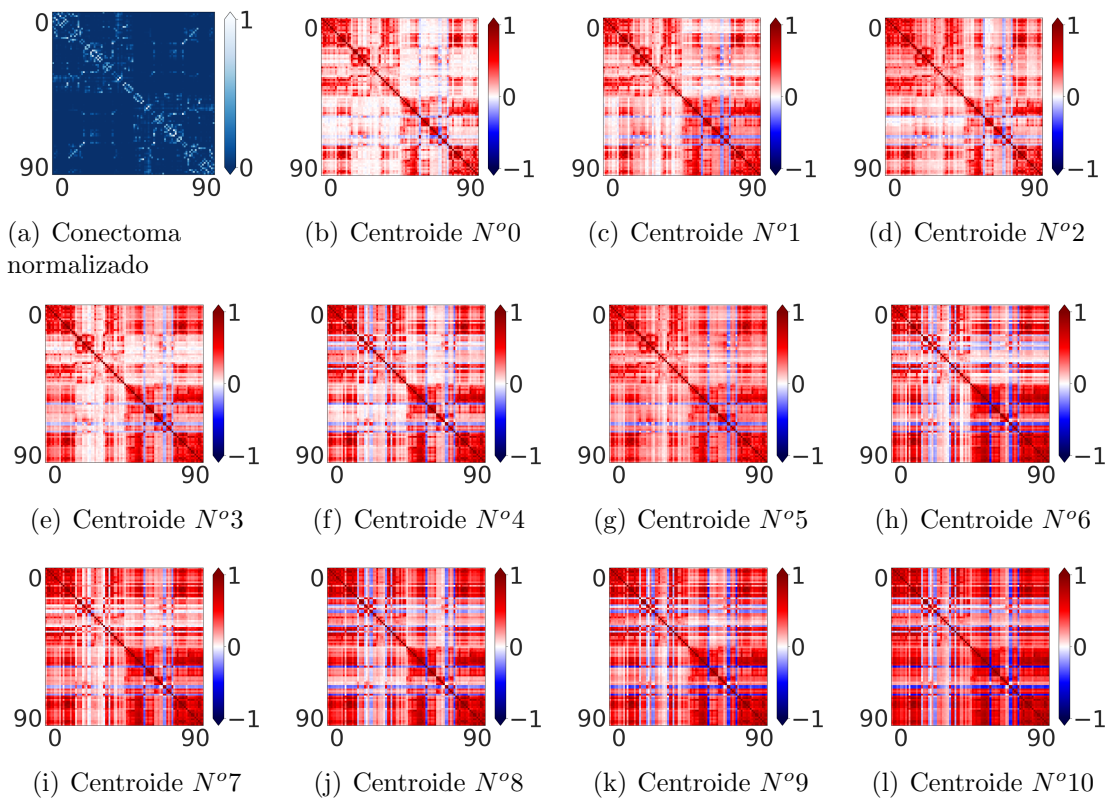


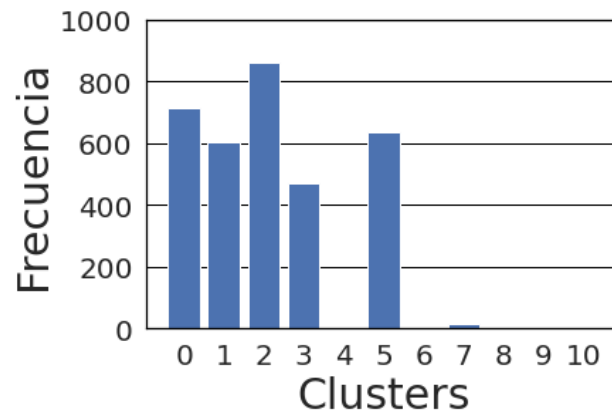
Figura 4.13: Matrices que representan los centroides de los 11 clusters ordenados conforme a la semejanza con el conectoma.

La Figura 4.13 muestra el conectoma normalizado del cerebro y los centroides de los 11 clusters del algoritmo *k-means* ordenados, de manera que la sub Figura 4.13(b) es el centroide

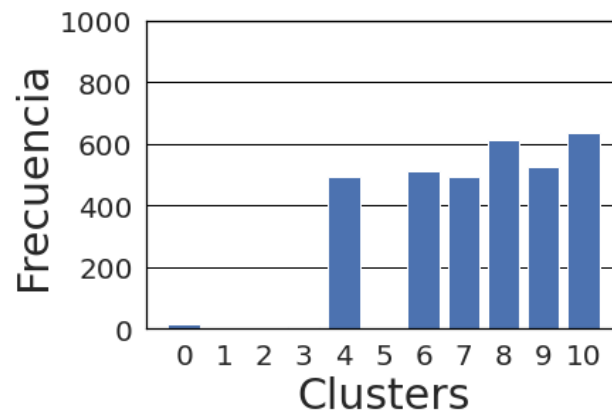
que más semejanza tiene con el conectoma y la sub Figura 4.13(1) es el centroide que menos semejanza tiene.

4.3.6. Histogramas de los 11 clusters

Las imágenes de la Figura 4.14 se obtienen al medir con qué frecuencia el algoritmo k-means clasifica, en los 11 clusters, las matrices correspondientes al estado placebo (Figura 4.14(a)) y al estado “LSD” (Figura 4.14(b)). Cada conjunto de matrices tiene 3300 elementos en total, los cuales corresponden al 100 % de cada estado.



(a) Histograma de la distribución de *clusters* de los datos “placebo”



(b) Histograma de la distribución de *clusters* de los datos “LSD”

Figura 4.14: Histogramas de clasificación con 11 clusters.

Capítulo 5

Discusión

El texto siguiente constituye una introducción a la sección de debate. En él se analiza el algoritmo de clasificación y la dinámica observada en los datos. Se examina el rendimiento del algoritmo de Viterbi en la clasificación de los estados placebo y LSD, mostrando una alta precisión y una tendencia a que los estados no cambien. El rendimiento del algoritmo se evalúa además utilizando datos de prueba, demostrando coherencia y precisión. La distribución de los datos en grupos revela características distinguibles entre los dos estados, lo que indica una diferenciación potencial entre los estados placebo y LSD. Además, la dinámica del cerebro en el estado LSD muestra una mayor sincronización entre las distintas regiones cerebrales. Estos hallazgos arrojan luz sobre la complejidad de los estados jerárquicos del cerebro y las posibles implicaciones para comprender la actividad cerebral.

5.1. Algoritmo de Clasificación

Partiendo por los resultados de la clasificación del algoritmo de Viterbi, se puede observar en la Figura 4.1 que existe una alta concentración de vectores de características asociados a los estados que pertenecen, siendo más alto el desempeño de clasificación de los vectores “placebo”. Esto se puede apreciar de mejor manera en los histogramas de las Figuras 4.2(a) y 4.2(b) donde la frecuencia de vectores bien clasificados supera en el 90 % de los datos en ambos casos, teniendo un 97,49 % de acierto en los datos “placebo” y un 92,51 % en los datos LSD. Si bien estos resultados corresponden a datos de entrenamiento, las cifras muestran un desempeño bastante alentador y afín a los objetivos que se lograban buscar, mostrando que el algoritmo creado es capaz de clasificar con bajo porcentaje de error las señales de estos dos estados con señales simuladas.

La Figura 4.3, que muestra las matrices de transición de ambos casos, señala que la tendencia de cambios entre un estado y otro no es alta, de hecho se observa que lo más frecuente es la permanencia en el mismo estado. Esta tendencia de permanencia crece significativamente cuando el vector es clasificado en el estado al cual verdaderamente pertenece, llegando a valores por sobre el 99 % de permanencia en el estado correcto para ambos datos. Esta

particular característica puede aprovecharse con el fin de mejorar de manera significativa la clasificación del algoritmo usando la *clasificación por bloques*. Como se puede apreciar en la Figura 4.4, al usar la *clasificación por bloques*, se ve una tendencia a converger al 100% a partir de un tamaño cercano de los 25 y 45 vectores de características para el estado “placebo” y el estado LSD respectivamente. Recordando que cada vector representa un tiempo de 120 segundos reales y la distancia entre cada vector es de 10 segundos, se puede calcular que, para los datos de entrenamiento, el tiempo de acierto total es cercano a 6 y 9.5 minutos de señal para los datos “placebo” y LSD respectivamente.

5.2. Prueba con datos de testeo

Con el fin de comprobar que el algoritmo de Viterbi no se sobre ajustó a los datos de entrenamiento, se probó el algoritmo ya entrenado con los datos de testeo. Los resultados en la Figura 4.5 muestran una consistencia clara en la clasificación de datos de testeo con respecto a la Figura 4.1, demostrando que el algoritmo muestra buenos resultados con datos de testeo y, con esto, que no está sobre ajustado a los datos de entrenamiento. Además, en las imágenes de la Figura 4.6 no solo reafirman lo dicho anteriormente, sino que denotan una buena mejora de clasificación para los datos LSD, llegando a un 95,56% de acierto. La tabla 4.1 muestra lo descrito con más detalle en función del error.

Las matrices de transición de las Figuras 4.7(a) y 4.7(b) muestran la misma tendencia de permanencia en los estados que en las matrices de los datos de entrenamiento, por lo que es factible mejorar la clasificación de los estados a través del agrupamiento por bloques. Tal y como se muestra en la Figura 4.8, se ve una tendencia a converger al 100% a partir de un tamaño cercano de los 30 vectores de características aproximadamente tanto para el estado “placebo” como para el estado LSD. Haciendo el mismo cálculo anterior, el tiempo de acierto total para ambos estados es cercano a 7 minutos. Resultado que muestra una notable mejora en comparación con los datos de entrenamiento.

5.3. Dinámica

En la Figura 4.9 se observa un comportamiento bastante peculiar en la distribución de los datos. Por un lado, los datos del estado “placebo” se concentran, en su mayoría, en un solo *cluster*. Sin embargo, las Figuras 4.10(a) y 4.10(b) no solo muestran que los datos del estado LSD se distribuyen de manera similar en 2 *clusters*, sino que estos últimos son distintos al *cluster* donde se encuentran los datos de “placebo”. Esto confirma, de una manera distinta al algoritmo de Viterbi, que las matrices “placebo” y LSD son **separables y distinguibles**, ya que ambos algoritmos apuntan hacia un mismo resultado. Por otro lado, el hecho de que los datos LSD se distribuyan de manera similar en 2 *clusters*, significa que las características de las matrices LSD son más **“diversas”** que las matrices “placebo”, y que su diversidad no tiene una preferencia marcada, es decir, que las matrices que el algoritmo separa no tienen una preferencia notoria por algún *cluster*. Las Figuras 4.11(b) y 4.11(c) muestran de

manera visual estas características a través de las matrices centroides en los *clusters* 1 y 2 respectivamente, donde se puede notar que tanto la Figura 4.11(b) como la 4.11(c) muestran, en comparación a la Figura 4.11(a), una tendencia a tener valores con mayor magnitud, es decir, que las regiones del cerebro parecen estar más correlacionados en el estado LSD que en el estado “placebo” donde los valores tienden a tener una magnitud menor. Se puede concluir, entonces, que la dinámica que presenta el cerebro cuando se encuentra en un estado LSD es una tendencia a que las distintas zonas del cerebro se comportan de una manera más sincronizada, a pesar de no tener conectores físicos que los relacionen.

Las matrices de transición mostradas en la Figura 4.12 muestran que la dinámica de transición entre el *cluster* 1 y 2 es bastante baja en comparación a la permanencia en el propio *cluster*, esto también afirma los resultados del algoritmo de Viterbi. Se observa que para los estados donde no hay mucho porcentaje de datos, la dinámica de transición no tiene necesariamente una tendencia clara a quedarse en ese estado. En el Anexo A.12 y A.13 se puede ver esto con mayor claridad.

Al aumentar la cantidad de *clusters* en el algoritmo *k-means*, se puede observar en la Figura 4.13 que la diferencia entre los centroides vecinos no es mucha, es más, se observa una tendencia continua de cambio a colores más intensos a medida que se aumenta de N^o de centroide. Esto muestra que, a pesar de que los datos están hechos como la unión forzada de un estado con otro aparentemente distinto, solo se presenta un **cambio de la magnitud de la correlación** de las zonas cerebrales. En decir, no existe una zona donde se presente notoriamente un cambio de correlación negativa a una positiva o viceversa, solo se muestran cambios progresivos de intensidad de la correlación entre las zonas. Por otro lado, se puede observar en la Figura 4.14 que al aumentar la cantidad de *clusters*, no aumenta necesariamente en la misma proporción la cantidad de *clusters* LSD sobre “placebo”, solo se mantiene la diferencia absoluta de *clusters*, por lo que se puede concluir que las matrices LSD si bien son más ricas en “**diversidad**” no es mucha la diferencia.

Capítulo 6

Conclusiones y trabajo futuro

El foco de este trabajo fue clasificar entre dos estados cerebrales: “*placebo*” y “*LSD*”. Observando los resultados se puede concluir que se logró clasificar con un porcentaje de error menor al 5% cada señal, demostrando la veracidad de la hipótesis señalada en un principio, en que el cerebro presenta señales que están ligadas a un estado de conciencia único y, por tanto, es posible saber cuál es el estado de conciencia que presentan los individuos al saber sus señales cerebrales.

Con respecto a los resultados de la dinámica, se aprecia que las matrices “*placebo*” tienen un comportamiento menos dinámico que las matrices “*LSD*”, es decir, las características que definen las matrices “*placebo*” son menos diversas que las matrices “*LSD*”. Estos resultados presentan conclusiones que van en la misma dirección que los presentados por Barttfeld et al. [3], ya que ambas reconcilian las teorías de la conciencia con las observaciones de las matrices de correlación del cerebro. En Barttfeld et al. [3] se presenta que los estados asociados a mayor anestesia (menor actividad cerebral) tienden a estar en mejor sintonía con las conexiones anatómicas del cerebro y en esta tesis se muestra que en estado “*LSD*” (mayor actividad cerebral) tiende a estar en peor sintonía con las conectividades anatómicas del cerebro. En ambas se concluye que una rica dinámica funcional podría constituir una firma de la conciencia.

Una de las mejoras que se le podrían hacer al clasificador es que cuantifique la intensidad con que el estado está presente en cada sujeto, es decir, que no solo sean 0 o 1, sino que poder encontrar cuando existen puntos intermedios entre estos estados. Para lograr este trabajo a futuro, una de las recomendaciones es poder tener una base de datos que tenga una transición real de “*placebo*” a “*LSD*”, en vez de juntar dos estados a la fuerza. Otro objetivo interesante sería probar este algoritmo clasificador con datos “*placebo*” y “*LSD*” de personas reales y poder medir el verdadero potencial del algoritmo. Por último, tener un base de datos que integra más estados sería un desafío interesante, ya que podrían existir estados que se parezcan mucho entre sí y que el algoritmo no sea capaz de diferenciarlos con precisión. Esto podría llevar a la idea de que la jerarquía de los estados cerebrales es mucho más compleja de lo que parece.

Finalmente, se logran los objetivos de manera exitosa y se presenta una nueva metodología que utiliza algoritmos de HMM con datos de neurociencia, obteniendo buenos resultados.

Bibliografía

- [1] Peter B Rosenberger y Heather R Adams. *Big brain/smart brain*. 2011. URL: <https://n.neurology.org/content/77/16/1504.short>.
- [2] M.J. Frade Mera y col. “Análisis de 4 escalas de valoración de la sedación en el paciente crítico”. En: *Enfermería Intensiva* 20.3 (2009), págs. 88-94. ISSN: 1130-2399. DOI: [https://doi.org/10.1016/S1130-2399\(09\)72588-X](https://doi.org/10.1016/S1130-2399(09)72588-X). URL: <https://www.sciencedirect.com/science/article/pii/S113023990972588X>.
- [3] Pablo Barttfeld y col. “Signature of consciousness in the dynamics of resting-state brain activity”. En: *Proceedings of the National Academy of Sciences* 112.3 (2015), págs. 887-892. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1418031112>.
- [4] Gustavo Deco y col. “Whole-brain multimodal neuroimaging model using serotonin receptor maps explains non-linear functional effects of LSD”. En: *Current biology* 28.19 (2018), págs. 3065-3074. URL: <https://www.sciencedirect.com/science/article/pii/S0960982218310455>.
- [5] Patricio Orio Álvarez Rubén Herzog Amunátegui Rodrigo Cofré Torres. *Whole-brain models and consciousness: a novel framework for a mechanistic understanding of different states of consciousness*. 2022. URL: <http://repositoriobibliotecas.uv.cl/handle/uvsc1/8411>.
- [6] Francis Crick y Christof Koch. “Towards a neurobiological theory of consciousness”. En: *Seminars in the Neurosciences*. Vol. 2. Saunders Scientific Publications. 1990, págs. 263-275. URL: <https://resolver.caltech.edu/CaltechAUTHORS:20130816-103136937>.
- [7] Francis Crick y Christof Koch. “A framework for consciousness”. En: *Nature neuroscience* 6.2 (2003), págs. 119-126. URL: <https://www.nature.com/articles/nn0203-119>.
- [8] Anil Seth. “Models of consciousness”. En: *Scholarpedia* 2.1 (2007), pág. 1328. URL: http://var.scholarpedia.org/article/Models_of_consciousness.
- [9] Claire Sergent y Lionel Naccache. “Imaging neural signatures of consciousness: ‘What’, ‘When’, ‘Where’ and ‘How’ does it work?”. En: *Archives italiennes de biologie* 150.2/3 (2012), págs. 91-106. URL: <http://www.architalbiol.org/index.php/aib/article/view/15091>.

- [10] Catherine Stinson y Jacqueline Sullivan. “Mechanistic explanation in neuroscience 1”. En: *The Routledge handbook of mechanisms and mechanical philosophy*. Routledge, 2017, págs. 375-388. URL: <https://www.taylorfrancis.com/chapters/edit/10.4324/9781315731544-28/mechanistic-explanation-neuroscience-1-catherine-stinson-jacqueline-sullivan>.
- [11] Gustavo Deco y col. “The dynamic brain: from spiking neurons to neural masses and cortical fields”. En: *PLoS computational biology* 4.8 (2008), e1000092. URL: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000092>.
- [12] Rodrigo Cofré y col. “Whole-brain models to explore altered states of consciousness from the bottom up”. En: *Brain Sciences* 10.9 (2020), pág. 626. URL: <https://www.mdpi.com/2076-3425/10/9/626>.
- [13] Patric Hagmann y col. “Mapping the structural core of human cerebral cortex”. En: *PLoS biology* 6.7 (2008), e159. URL: <https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.0060159>.
- [14] Nathalie Tzourio-Mazoyer y col. “Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain”. En: *Neuroimage* 15.1 (2002), págs. 273-289. URL: <https://www.sciencedirect.com/science/article/abs/pii/S1053811901909784>.
- [15] Olaf Sporns, Giulio Tononi y Rolf Kötter. “The human connectome: a structural description of the human brain”. En: *PLoS computational biology* 1.4 (2005), e42. URL: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.0010042>.
- [16] Enzo Tagliazucchi y col. “Large-scale signatures of unconsciousness are consistent with a departure from critical dynamics”. En: *Journal of The Royal Society Interface* 13.114 (2016), pág. 20151027. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsif.2015.1027>.
- [17] Ariel Haimovici y col. “Brain organization into resting state networks emerges at criticality on a model of the human connectome”. En: *Physical review letters* 110.17 (2013), pág. 178101. URL: <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.110.178101>.
- [18] Gustavo Deco y Viktor K Jirsa. “Ongoing cortical activity at rest: criticality, multistability, and ghost attractors”. En: *Journal of Neuroscience* 32.10 (2012), págs. 3366-3375. URL: <https://www.jneurosci.org/content/32/10/3366.short>.
- [19] Pubuditha M Abeyasinghe y col. “Consciousness and the dimensionality of DOC patients via the generalized ising model”. En: *Journal of clinical medicine* 9.5 (2020), pág. 1342. URL: <https://www.mdpi.com/707776>.
- [20] Arnaud Messé y col. “Relating structure and function in the human brain: relative contributions of anatomy, stationary dynamics, and non-stationarities”. En: *PLoS computational biology* 10.3 (2014), e1003530. URL: <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003530>.
- [21] Maria Luisa Saggio, Petra Ritter y Viktor K Jirsa. “Analytical operations relate structural and functional connectivity in the brain”. En: *PloS one* 11.8 (2016), e0157292. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0157292>.

- [22] Joana Cabral, Morten L Kringelbach y Gustavo Deco. “Exploring the network dynamics underlying brain activity during rest”. En: *Progress in neurobiology* 114 (2014), págs. 102-131. URL: <https://www.sciencedirect.com/science/article/pii/S0301008213001457>.
- [23] Beatrice M Jobst y col. “Increased stability and breakdown of brain effective connectivity during slow-wave sleep: mechanistic insights from whole-brain computational modelling”. En: *Scientific reports* 7.1 (2017), págs. 1-16. URL: <https://www.nature.com/articles/s41598-017-04522-x>.
- [24] PA Robinson y N Roy. “Neural field theory of nonlinear wave-wave and wave-neuron processes”. En: *Physical Review E* 91.6 (2015), pág. 062719. URL: <https://journals.aps.org/pre/abstract/10.1103/PhysRevE.91.062719>.
- [25] Tahereh Babaie Janvier y Peter A Robinson. “Neural field theory of corticothalamic prediction with control systems analysis”. En: *Frontiers in Human Neuroscience* 12 (2018), pág. 334. URL: <https://www.frontiersin.org/articles/10.3389/fnhum.2018.00334/full>.
- [26] Michael Breakspear, John R Terry y Karl J Friston. “Modulation of excitatory synaptic coupling facilitates synchronization and complex dynamics in a biophysical model of neuronal dynamics”. En: *Network: Computation in Neural Systems* 14.4 (2003), pág. 703. URL: <https://iopscience.iop.org/article/10.1088/0954-898X/14/4/305/meta>.
- [27] Christopher J Honey y col. “Predicting human resting-state functional connectivity from structural connectivity”. En: *Proceedings of the National Academy of Sciences* 106.6 (2009), págs. 2035-2040. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.0811168106>.
- [28] Gustavo Deco y col. “How local excitation–inhibition ratio impacts the whole brain dynamics”. En: *Journal of Neuroscience* 34.23 (2014), págs. 7886-7898. URL: <https://www.jneurosci.org/content/34/23/7886.short>.
- [29] Morten L Kringelbach y col. “Dynamic coupling of whole-brain neuronal and neurotransmitter systems”. En: *Proceedings of the National Academy of Sciences* 117.17 (2020), págs. 9566-9576. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.1921475117>.
- [30] Michael E Wall, Andreas Rechtsteiner y Luis M Rocha. “Singular value decomposition and principal component analysis”. En: *A practical approach to microarray data analysis*. Springer, 2003, págs. 91-109. URL: https://link.springer.com/content/pdf/10.1007/0-306-47815-3_5.pdf.
- [31] Lawrence R Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. En: *Proceedings of the IEEE* 77.2 (1989), págs. 257-286. URL: <https://ieeexplore.ieee.org/abstract/document/18626/>.
- [32] Andrew Viterbi. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. En: *IEEE transactions on Information Theory* 13.2 (1967), págs. 260-269. URL: <https://ieeexplore.ieee.org/abstract/document/1054010/>.
- [33] Helene Girouard y Costantino Iadecola. “Neurovascular coupling in the normal brain and in hypertension, stroke, and Alzheimer disease”. En: *Journal of applied physiology* 100.1 (2006), págs. 328-335.

- [34] Dengsheng Zhang. “Wavelet transform”. En: *Fundamentals of image data mining*. Springer, 2019, págs. 35-44.
- [35] Diederik P Kingma y Max Welling. “Auto-encoding variational bayes”. En: *arXiv preprint arXiv:1312.6114* (2013). URL: <https://arxiv.org/abs/1312.6114>.
- [36] Johannes Ballé y col. “Variational image compression with a scale hyperprior”. En: *arXiv preprint arXiv:1802.01436* (2018).
- [37] Yifei Zhang. “A better autoencoder for image: Convolutional autoencoder”. En: 2018. URL: http://users.cecs.anu.edu.au/~Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf.
- [38] Daniele Marinazzo y col. “Information transfer and criticality in the Ising model on the human connectome”. En: *PloS one* 9.4 (2014), e93616. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0093616>.
- [39] Vincent Beliveau y col. “A high-resolution in vivo atlas of the human brain’s serotonin system”. En: *Journal of Neuroscience* 37.1 (2017), págs. 120-128. URL: <https://www.jneurosci.org/content/37/1/120.abstract>.

ANEXO

A.1. DMF representado en ecuaciones.

Se utilizó un modelo de red para simular la actividad cerebral espontánea a nivel de todo el cerebro, donde cada nodo representa un área cerebral y los enlaces representan las conexiones de materia blanca entre ellas. La actividad en cada área cerebral está representada por el modelo DMF, que reduce la actividad de grandes conjuntos de neuronas excitadoras e inhibitoras interconectadas a un conjunto reducido de ecuaciones dinámicas que describen la actividad de grupos de neuronas excitadoras (E) e inhibitoras (I) acopladas. En esta reducción DMF, las corrientes sinápticas excitatorias, $I^{(E)}$, están mediadas por receptores $NMDA$ y las corrientes inhibitoras, $I^{(I)}$, están mediadas por receptores $GABA_A$. Dentro de cada área cerebral n , los conjuntos neuronales E e I están conectados recíprocamente, mientras que el acoplamiento inter área entre dos áreas n y p se produce solo a nivel de E a E y está escalado por la conectividad estructural C_{pn} .

Más concretamente, el modelo DMF a nivel de todo el cerebro se expresa mediante el siguiente sistema de ecuaciones diferenciales acopladas:

$$I_n^{(E)} = W_E I_0 + W_+ J_{NMDA} S_n^{(E)} + G J_{NMDA} \sum_p C_{np} S_p^{(E)} - J_n S_n^{(I)}, \quad (\text{A.1})$$

$$I_n^{(I)} = W_I I_0 + J_{NMDA} S_n^{(E)} - S_n^{(I)} \quad (\text{A.2})$$

$$r_n^{(E)} = H^{(E)}(I_n^{(E)}) = \frac{g_E \left(I_n^{(E)} - l_{thr}^{(E)} \right)}{1 - \exp \left(-d_E g_E \left(I_n^{(E)} - l_{thr}^{(E)} \right) \right)} \quad (\text{A.3})$$

$$r_n^{(I)} = H^{(I)}(I_n^{(I)}) = \frac{g_I \left(I_n^{(I)} - l_{thr}^{(I)} \right)}{1 - \exp \left(-d_I g_I \left(I_n^{(I)} - l_{thr}^{(I)} \right) \right)} \quad (\text{A.4})$$

$$\frac{dS_n^{(E)}(t)}{dt} = -\frac{S_n^{(E)}}{\tau_{NMDA}} + (1 - S_n^{(E)}) \gamma r_n^{(E)} + \sigma v_n(t), \quad (\text{A.5})$$

$$\frac{dS_n^{(I)}(t)}{dt} = -\frac{S_n^{(I)}}{\tau_{GABA}} + r_n^{(I)} + \sigma v_n(t). \quad (\text{A.6})$$

Aquí, para cada grupo excitatorio (E) o inhibitorio (I) de neuronas en cada área cerebral n , $I_n^{(E,I)}$ (en nA) representa la corriente de entrada total, $r_n^{(E,I)}$ (en Hz) denota la tasa de disparo y $S_i^{(E,I)}$ denota la variable de compuerta sináptica. Las funciones de respuesta neuronal, $H^{(E,I)}$, convierten las corrientes de entrada totales recibidas por los grupos E e I en tasas de disparo, $r_i^{(E,I)}$, siguiendo la función de entrada-salida de Abbott y Chance [80], donde $g_E = 310nC^{-1}$ y $g_I = 615nC^{-1}$ son factores de ganancia que determinan la pendiente de H , $Ithr(E) = 0403nA$, e $Ithr(I) = 0,288nA$ son las corrientes umbral por encima de las cuales las tasas de disparo aumentan linealmente con las corrientes de entrada, y $dE = 0,16$ y $dI = 0,087$ son constantes que determinan la forma de la curvatura de H alrededor de $Ithr$.

A.2. Algoritmo de Viterbi

El pseudocódigo para la implementación del algoritmo de Viterbi es:

Algorithm 1 Viterbi Algorithm ($\lambda = (A, B, \pi), O$)

Inputs: $\lambda = (A, B, \pi), O$

Output: $Q^* = \{q_1^*, q_2^*, \dots, q_T^*\}$

1: **for** $i = 1, \dots, N$ **do**

2: $\delta_1(i) = \pi_i b_i(O_1)$

3: $\psi_1(i) = 0$

4: **for** $j = 1, \dots, N, t = 2, \dots, T$ **do**

5: $\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \cdot B_j(O_t)$

6: $\psi_t(i) = \operatorname{argmax}_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}]$

7: $P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$

8: $q_T^* = \operatorname{argmax}_{1 \leq i \leq N} [\delta_T(i)]$

9: **for** $t = T - 1, T - 2, \dots, 1$ **do**

10: $q_t^* = \psi_{t+1}(q_{t+1}^*)$

11: **return** Q^*

A.3. Código MATLAB de Parámetros por Default

```

1 function [ params ] = DefaultParams(varargin)
2 %% DEFAULTPARAMS Default parameter settings for DMF simulation
3 %
4 % P = DEFAULTPARAMS() yields a struct with default values for all necessary
5 % parameters for the DMF model. The default structural connectivity is the
6 % DTI fiber consensus matrix obtained from the HCP dataset, using a
7 % Schaeffer–100 parcellation.
8 %
9 % P = DEFAULTPARAMS('key', 'value', ...) adds (or replaces) field 'key' in
10 % P with given value.
11 %
12 % Pedro Mediano, Feb 2022
13
14 params = [];
15
16 % Connectivity matrix
17 if any(strcmp(varargin, 'C'))
18     C = [];
19
20 else
21     try
22         p = strrep(mfilename('fullpath'), 'DefaultParams', '');
23         C = dlmread([p, '/../data/DTI_fiber_consensus_HCP.csv'], ',');
24         % C = C/max(C(:));
25         C = C/max(C(:)).*0.2; % Deco's offset
26     catch
27         error('No connectivity matrix provided, and default matrix not found. ');
28     end
29
30 end
31
32
33 DMF parameters
34 params.C = C;
35 params.receptors = 0;
36 params.dt = 0.1; % ms
37 params.taon = 100; % NMDA tau ms
38 params.taog = 10; % GABA tau ms
39 params.gamma = 0.641; % Kinetic Parameter of Excitation
40 params.sigma = 0.01; % Noise SD nA
41 params.JN = 0.15; % excitatory synaptic coupling nA
42 params.I0 = 0.382; % effective external input nA
43 params.Jexte = 1.; % external → E coupling
44 params.Jexti = 0.7; % external → I coupling
45 params.w = 1.4; % local excitatory recurrence
46 params.g_e = 0.16; % excitatory conductance
47 params.Ie = 125/310; % excitatory threshold for nonlinearity
48 params.ce = 310.; % excitatory non linear shape parameter
49 params.g_i = 0.087; % inhibitory conductance
50 params.Ii = 177/615; % inhibitory threshold for nonlinearity
51 params.ci = 615.; % inhibitory non linear shape parameter
52 params.wgain = 0; % neuromodulatory gain
53 params.wgaini = 0; % neuromodulatory gain
54 params.G = 2; % Global Coupling Parameter

```

```

55
56 % Balloon-Windkessel parameters (from firing rates to BOLD signal)
57 params.TR = 2; % number of seconds to sample bold signal
58 params.dtt = 0.001; % BW integration step, in seconds
59
60 % Parallel computation parameters
61 params.batch_size = 5000;
62
63 % Add/replace remaining parameters
64 for i=1:2:length(varargin)
65     params.(varargin{i}) = varargin{i+1};
66 end
67
68 % If feedback inhibitory control not provided, use heuristic
69 if ~any(strcmp(varargin, 'J'))
70     %params.J = 1.5*params.G*sum(params.C, 1)' + 1;
71     params.J = 0.75.*params.G.*sum(params.C, 1)' + 1;
72 end
73
74 end

```

A.4. Métodos de extracción de características

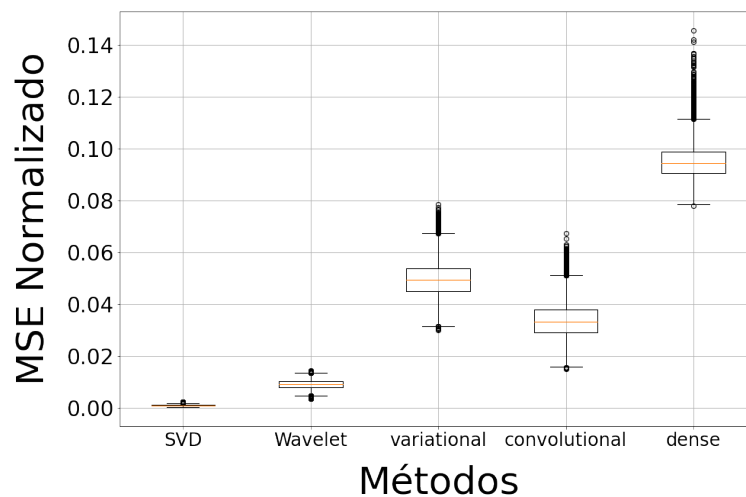
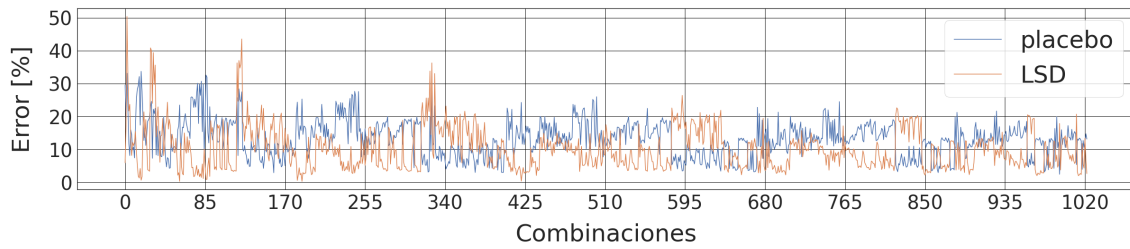
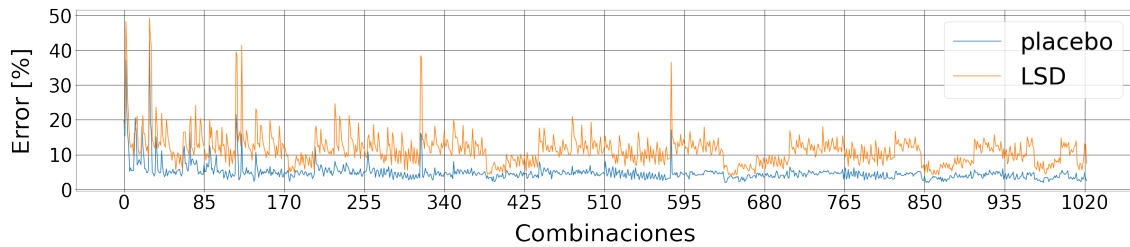


Figura A.1: Gráfico de cajas del MSE normalizado de las matrices reconstruidas con respecto a las matrices originales de los distintos métodos que se utilizaron para extraer las características de las matrices. *Variational*, *Convolutional* y *dense* son redes neuronales *encoder-decoder* cuyo vector de *encoder* es de largo 20. *Wavelet* corresponde al método *Wavelet* para extraer características, sin embargo, se utilizó un largo de 2000 elementos para poder tener un desempeño similar al resto. SVD se utilizaron los primeros 10 valores y vectores propios para reconstruir las matrices.

A.5. Error de las distintas combinaciones de valores propios

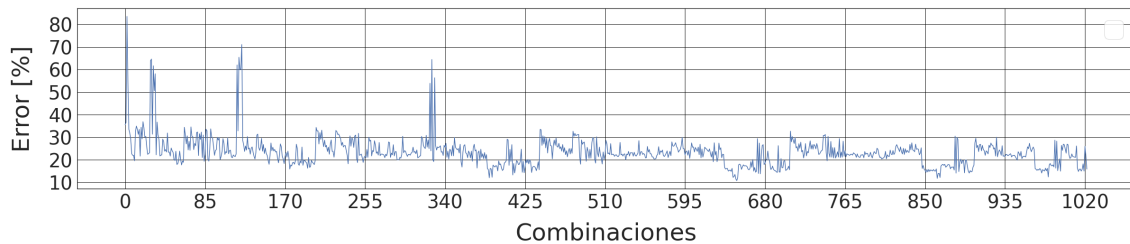


(a) Error de las combinaciones de los valores propios *caso 1*

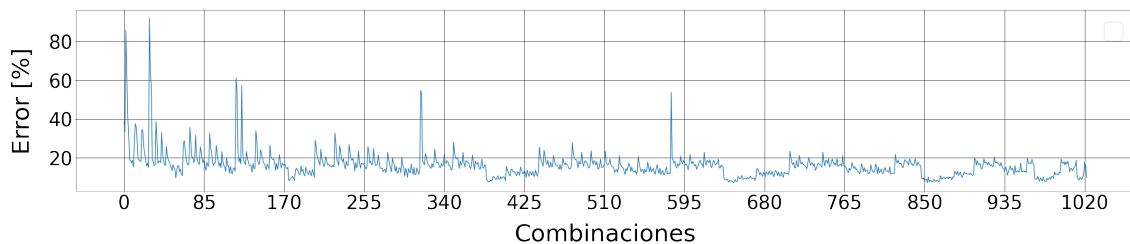


(b) Error de las combinaciones de los valores propios *caso 2*

Figura A.2: Errores del algoritmo de Viterbi para las distintas combinaciones de los valores propios asociados al conjunto de entrenamiento placebo y LSD.



(a) Suma de errores de las combinaciones de los valores propios *caso 1*



(b) Suma de errores de las combinaciones de los valores propios *caso 2*

Figura A.3: Suma de los errores del algoritmo de Viterbi para las distintas combinaciones de los valores propios asociados al conjunto de entrenamiento placebo y LSD.

A.6. Algoritmo de clasificación del *caso 1*

A.6.1. Algoritmo de Viterbi

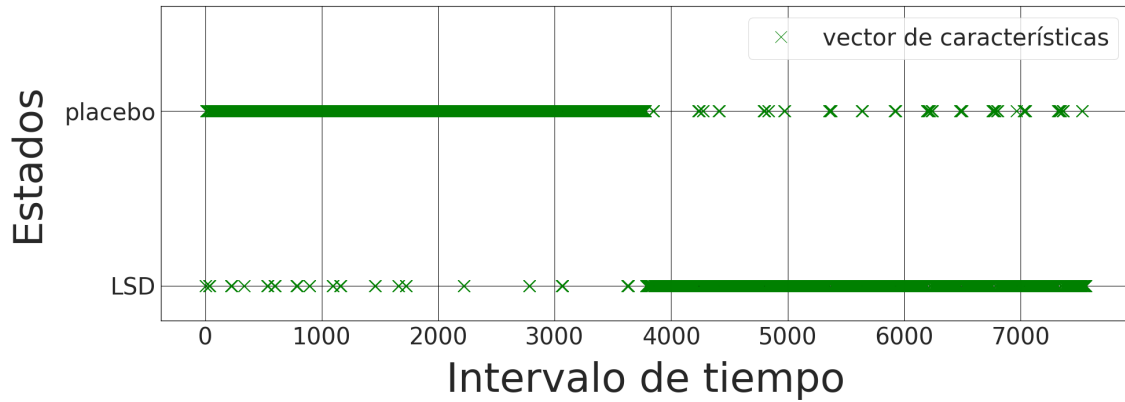
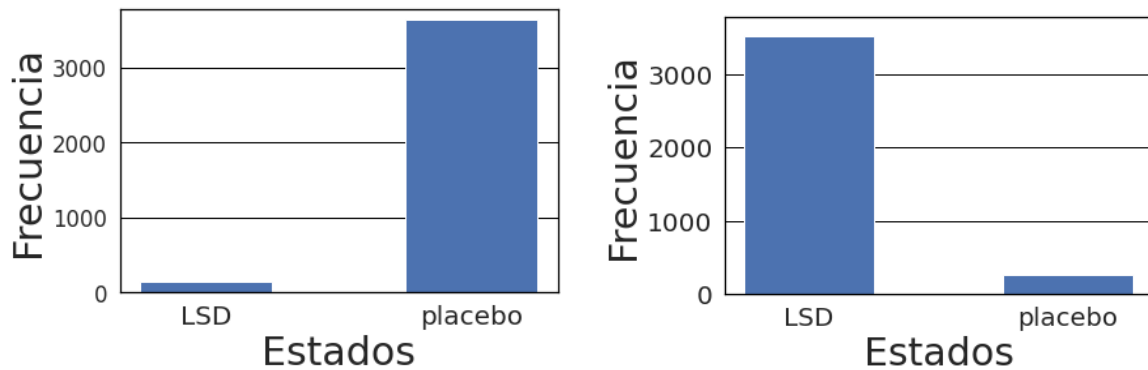


Figura A.4: Gráfico de clasificación de muestras de tiempo para el *caso 1*.

A.6.2. Histogramas



(a) Histograma de los datos Placebo

(b) Histograma de los datos LSD

Figura A.5: Histogramas de la clasificación del *caso 1*

A.6.3. Matrices de Transición

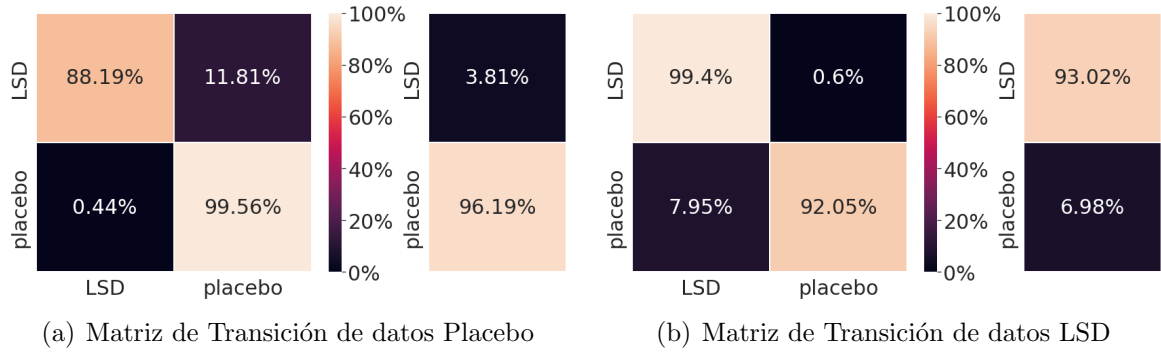


Figura A.6: Matrices de Transición de la clasificación del *caso 1*.

A.6.4. Clasificación por bloques

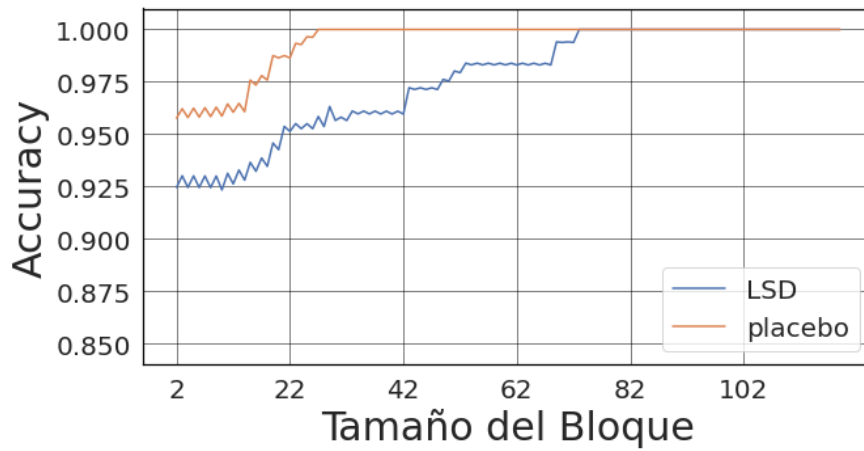


Figura A.7: Desempeño de la clasificación del algoritmo de Viterbi en función del tamaño de los bloques de vectores para el *caso 1*.

A.7. Algoritmo de clasificación con datos de testeo

A.7.1. Algoritmo de Viterbi

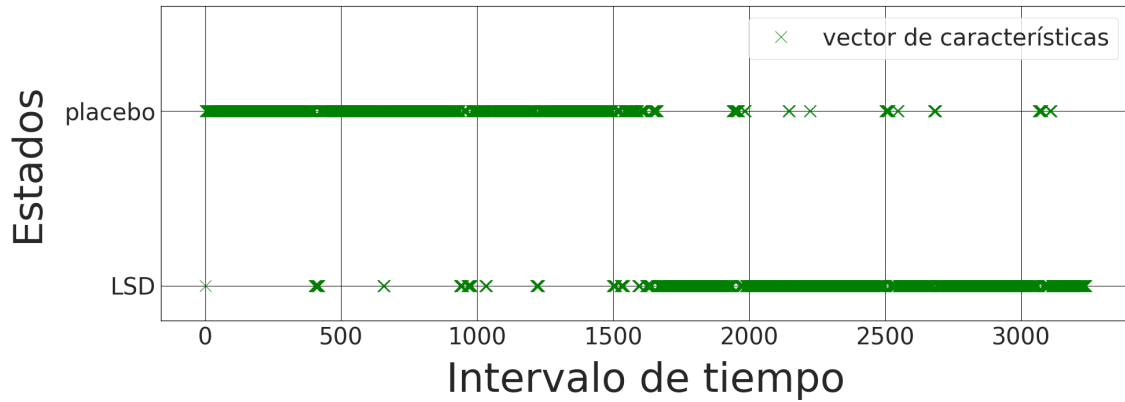
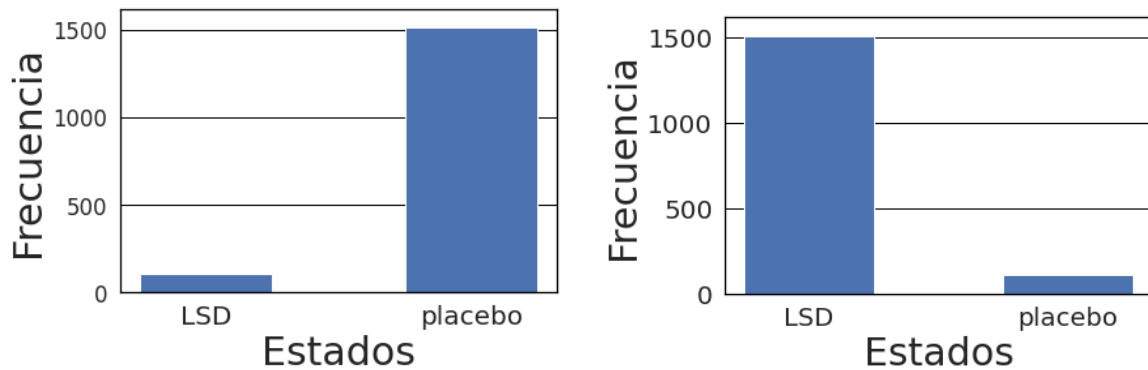


Figura A.8: Clasificación del algoritmo de Viterbi con datos de testeo.

A.7.2. Histogramas

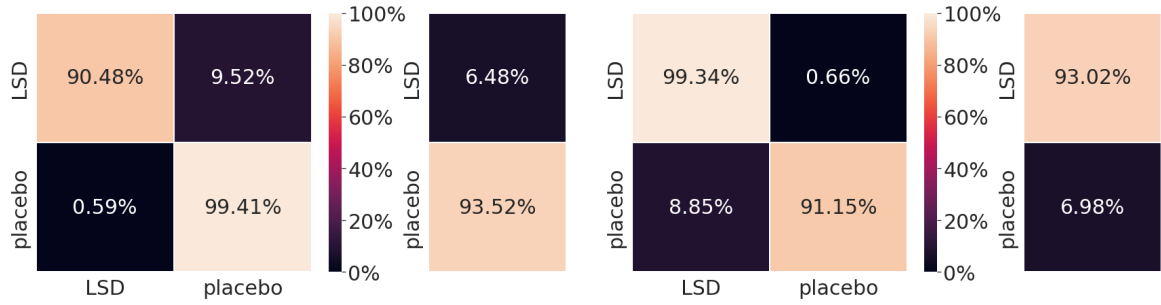


(a) Histograma placebo de los datos de testeo

(b) Histograma LSD de los datos de testeo

Figura A.9: Histogramas de la clasificación con los datos de testeo.

A.7.3. Matrices de Transición



(a) Matriz de Transición placebo con datos de tes- (b) Matriz de Transición LSD con datos de teste-
teo

Figura A.10: Matrices de transición de los datos de testeo.

A.7.4. Clasificación por bloques

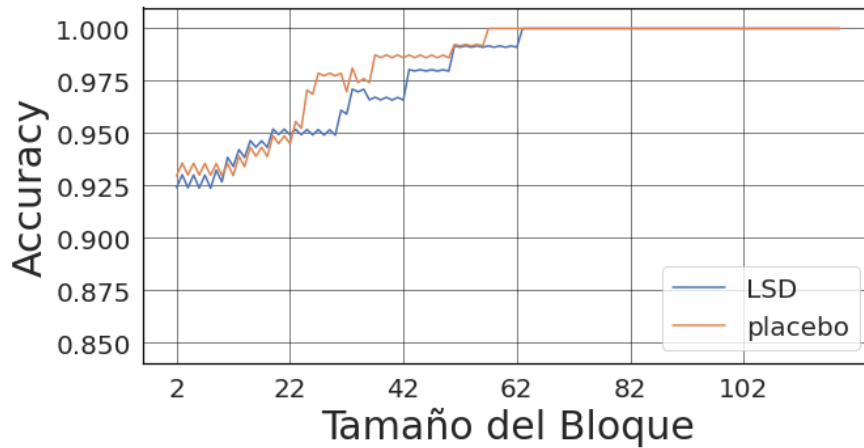


Figura A.11: Desempeño de clasificación en función del tamaño de los bloques de vectores con datos de testeo.

A.7.5. Tabla de comparación de desempeño

Tabla A.1: Porcentajes de error de clasificación para datos de testeo de ambos casos

	Datos <i>Caso 1</i>		Datos <i>Caso 2</i>	
	Placebo	LSD	Placebo	LSD
Error de Clasificación(%)	12.2	3.33	2.63	4.44

A.7.6. Matrices de Transición para 11 clusters

Las Figuras A.12 y A.13 muestran las matrices de transición de los datos correspondientes a los estados placebo y LSD respectivamente. Las características de estas imágenes son iguales a las descritas con anterioridad en la Figura 4.3.

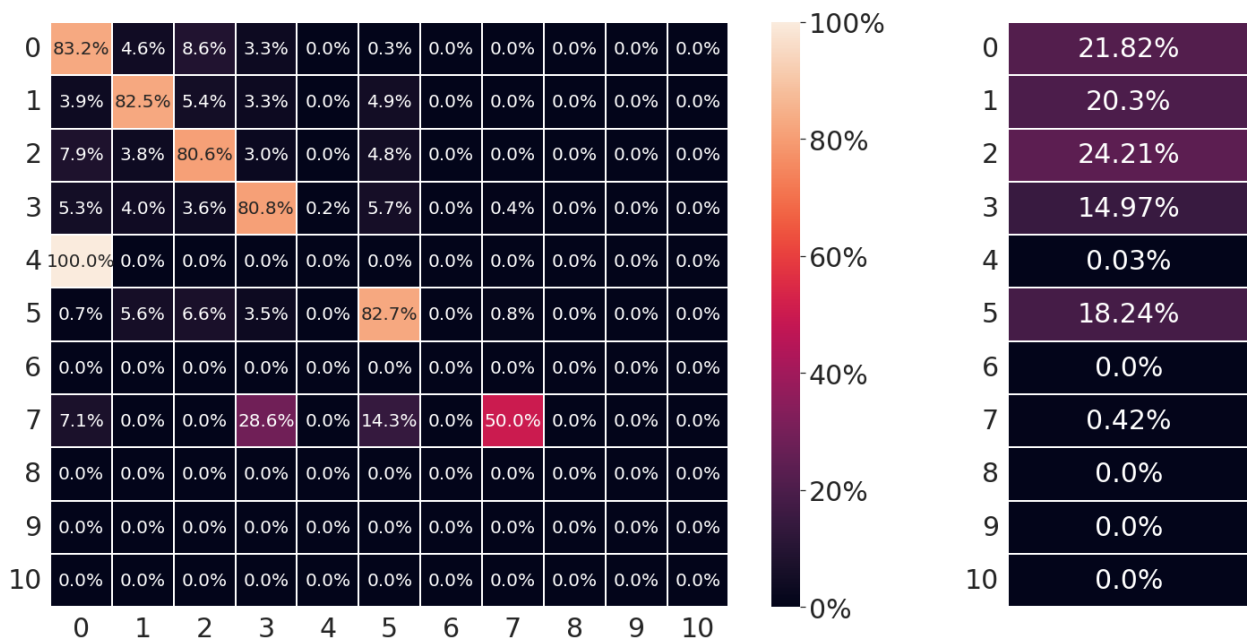


Figura A.12: Matriz de Transición de la distribución de 11 clusters en los datos placebo

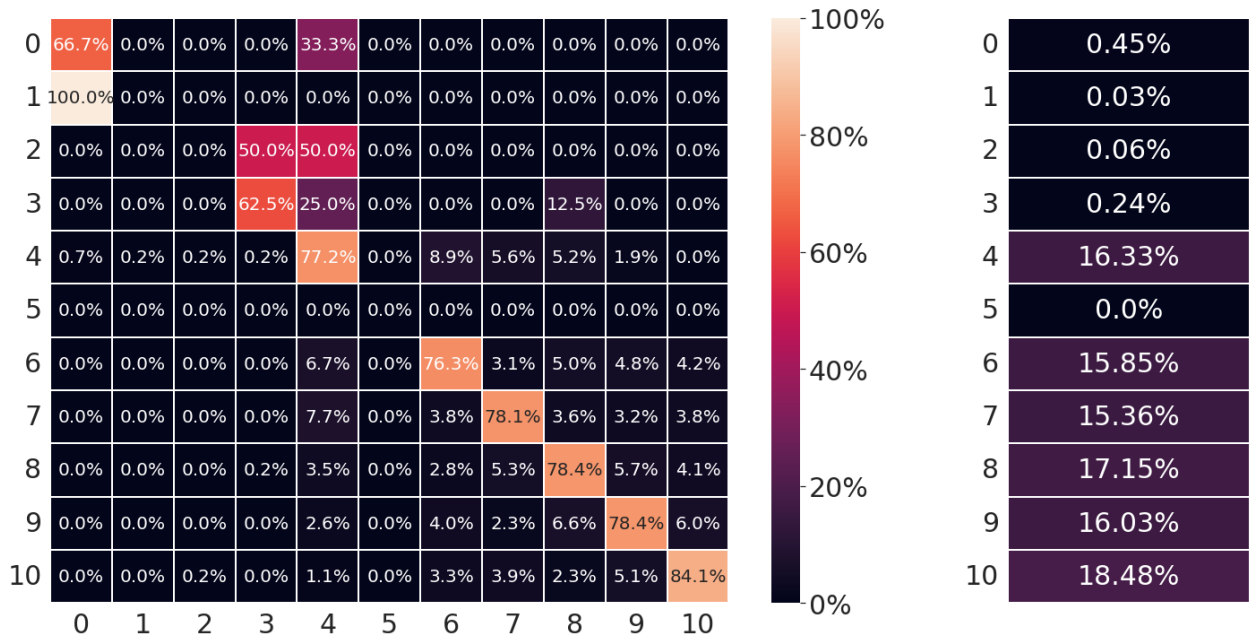


Figura A.13: Matriz de Transición de la distribución de 11 *clusters* en los datos LSD