



*ugr* | Universidad  
de **Granada**

TRABAJO FIN DE GRADO  
GRADO EN INGENIERÍA DE TECNOLOGÍAS DE  
TELECOMUNICACIÓN

# Simulador de imágenes cerebrales para el incremento del tamaño muestral en estudios funcionales

---

**Autor**  
Ulises Vidal Sanz

**Directores**  
Francisco Jesús Martínez Murcia  
Juan Manuel Górriz Sáez  
Javier Ramírez Pérez de Inestrosa



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

—  
Granada, Junio de 2016







# **Simulador de imágenes cerebrales para el incremento del tamaño muestral en estudios funcionales**

Ulises Vidal Sanz

**Palabras clave:** Alzheimer's Disease, Principal Component Analysis Kernel Density Estimation, Support Vector Machines, Simulated Brains, Small Sample Size Problem

## **Resumen**

Este proyecto aborda, desde un nuevo punto de vista, el problema del pequeño tamaño muestral, conocido como “*small sample size problem*”. Este es uno de los problemas más comunes en el diagnóstico de imágenes cerebrales funcionales, donde el número de características suele ser mucho mayor en comparación con el tamaño muestral, dando lugar a sistemas de diagnóstico poco generalizables. Para ello, hemos desarrollado un algoritmo novel capaz de generar una base de datos de imágenes funcionales con características similares a la original, en nuestro caso formada por imágenes PET. Este proceso incluye el modelado de la base de datos original mediante PCA, proyectando las imágenes en el espacio de *Eingenbrains*. Mediante KDE modelamos la distribución el set de coordenadas de dichas imágenes, generando nuevas coordenadas en el espacio de *Eingenbrains* que nos permitirá reconstruir las imágenes cerebrales simuladas, dando lugar a una nueva base de datos cuyos resultados son analizados.



# **Simulating brain images to increase the sample size in functional studies**

Ulises Vidal Sanz

**Keywords:** Alzheimer's Disease, Principal Component Analysis Kernel Density Estimation, Support Vector Machines, Simulated Brains, Small Sample Size Problem

## **Abstract**

In this work, we propose the simulation of new functional brain imaging samples as a solution to the small sample size problem. The system takes existing functional imaging datasets, and extracts commons features using Principal Component Analylis (PCA). Afterwards, the statistical distribution of the coordinates in the eingenbrain space is estimated usign Kernel Density Stimation (KDE) and new coordinates that match that distribution can be used to generate new images with similar properties to the original ones. Our strategy is intended to serve as a computational aid to reduce bias in the evaluation of new Computer Aided Diagnosis systems.



---

Yo, **Ulises Vidal Sanz**, alumno de la titulación GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 09057361F, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Ulises Vidal Sanz

Granada a 11 de Julio de 2016.



---

D. **Francisco Jesús Martínez Murcia**, Profesor del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Telecomunicaciones de la Universidad de Granada.

D. **Juan Manuel Górriz Sáez**, Profesor del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Telecomunicaciones de la Universidad de Granada.

D. **Javier Ramírez Pérez de Inestrosa**, Profesor del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Telecomunicaciones de la Universidad de Granada.

**Informan:**

Que el presente trabajo, titulado *Simulador de imágenes cerebrales para el incremento del tamaño muestral en estudios funcionales*, ha sido realizado bajo su supervisión por **Ulises Vidal Sanz**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 11 de Julio de 2016.

**Los directores:**

**Francisco Jesús Martínez Murcia**      **Juan Manuel Górriz Sáez**

**Javier Ramírez Pérez de Inestrosa**



# Agradecimientos

Mi más sincero agradecimiento a Paco, Juanma y Javier, por haberme brindado la oportunidad de realizar este proyecto con ellos. En especial a Paco, por su dedicación y generosidad, y a Juanma por abrirme las puertas al mundo de la investigación.

A mis padres y mi hermana por si ayuda desinteresada durante todos estos años, sin ellos nada de esto tendría sentido.

A mi Anita, que ha sabido aguantarme y darme ese empujón cuando más lo necesitaba, gracias de corazón.



# Índice general

<b>I INTRODUCCIÓN GENERAL</b>	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Motivación . . . . .	3
1.2. Enfermedad de Alzheimer . . . . .	3
1.2.1. Síntomas externos . . . . .	5
1.2.2. Diagnóstico . . . . .	6
1.3. Tomografía por Emisión de Positrones . . . . .	7
1.3.1. Imágenes PET . . . . .	7
1.3.2. Obtención de la imagen . . . . .	8
1.3.3. Reconstrucción . . . . .	9
1.4. Estado del arte . . . . .	11
1.5. Objetivos . . . . .	13
1.6. Publicaciones . . . . .	13
<b>II FUNDAMENTOS</b>	<b>15</b>
<b>2. Base de Datos</b>	<b>17</b>
2.1. ADNI . . . . .	17
2.1.1. Protocolo de adquisición de las imágenes . . . . .	18
2.1.2. Preprocesado . . . . .	18
2.1.3. Criterios de Etiquetado . . . . .	21
<b>3. Análisis de Componentes Principales</b>	<b>25</b>
3.1. PCA . . . . .	25
3.1.1. Eigenbrains . . . . .	29
3.1.2. Descomposición en Valores Singulares . . . . .	30
3.2. Interpretación de PCA . . . . .	31
<b>4. Estimación de la Función de Densidad</b>	<b>33</b>
4.1. Estimación de la Función de Densidad . . . . .	33
4.2. Modelos paramétricos y no paramétricos . . . . .	34
4.3. Histograma . . . . .	36

4.4. Estimador Tipo Núcleo . . . . .	37
4.4.1. KDE vía difusión . . . . .	38
<b>5. Aprendizaje Estadístico Supervisado</b>	<b>43</b>
5.1. Clasificación . . . . .	43
5.2. Máquinas de Soporte Vectorial . . . . .	45
5.2.1. SVM Lineal . . . . .	45
5.3. Validación . . . . .	52
5.3.1. Parámetros de evaluación del rendimiento de un clasificador . . . . .	52
5.3.2. Validación Cruzada . . . . .	53
5.4. El problema del pequeño tamaño muestral . . . . .	56
<b>III Simulación de Imágenes Funcionales Cerebrales</b>	<b>59</b>
<b>6. Desarrollo Experimental</b>	<b>61</b>
6.1. Introducción . . . . .	61
6.2. Base de datos utilizada . . . . .	62
6.2.1. Normalización en intensidad . . . . .	63
6.3. Extracción de características . . . . .	63
6.3.1. Selección de vóxeles basado en umbral . . . . .	63
6.3.2. PCA . . . . .	65
6.4. Modelado y simulación de Componentes Principales . . . . .	68
6.4.1. KDE vía difusión . . . . .	68
6.4.2. Interpolación de coordenadas . . . . .	71
6.5. Reconstrucción de imágenes funcionales . . . . .	73
<b>7. Validación y Resultados</b>	<b>75</b>
7.1. Experimento 1 . . . . .	75
7.2. Experimento 2 . . . . .	79
7.3. Experimento 3 . . . . .	82
7.4. Evolución de las PDFs . . . . .	84
<b>8. Planificación y estimación de costes</b>	<b>87</b>
8.1. Fases del desarrollo . . . . .	87
8.1.1. Revisión del estado del arte . . . . .	87
8.1.2. Solución propuesta y desarrollo . . . . .	88
8.1.3. Resultados experimentales . . . . .	88
8.1.4. Documentación . . . . .	88
8.1.5. Diagrama de Gantt . . . . .	88
8.2. Recursos . . . . .	89
8.3. Coste asociado . . . . .	90

<b>9. Conclusiones y Trabajo futuro</b>	<b>93</b>
9.1. Conclusiones . . . . .	93
9.2. Trabajo futuro . . . . .	96
<b>Bibliografía</b>	<b>111</b>
<b>A. Resultados Detallados</b>	<b>113</b>



# Índice de figuras

1.1. Placas de amiloide y neurofibrillas [Fuente: stanfordhealthcare.org]	5
1.2. Imágenes PET-FDG.	7
1.3. Obtención de la imagen PET.	8
1.4. Toshiba Celesteion PET/CT Scanner [Fuente: www.medgadget.com].	9
1.5. Geometría para la construcción de las proyecciones.	10
1.6. Representación del Teorema de Fourier de los Cortes.	10
2.1. Línea AC-PC [Fuente: iranjradiol.com]	20
2.2. Proceso de normalización espacial de imágenes PET. <i>Columna izq.:</i> Imagen original. <i>Columna central:</i> Plantilla. <i>Columna der.:</i> Imagen registrada.	22
3.1. Contribución a la varianza de las 10 primeras Componentes Principales.	29
3.2. Representación en términos de eigenbrains de la imagen de media cero $\Phi$ , mostrando tres cortes transaxiales representativos.	31
4.1. Histograma de la primera Componente Principal para un sujeto NC.	37
4.2. Histograma y estimación tipo núcleo mostrando los núcleos individuales. [Fuente: Wikipedia]	39
5.1. Representación de los planos que separan dos clases y hacen máximo el margen entre clases.[Fuente: Wikipedia]	47
5.2. Clasificador con <i>overfitting</i> y sin <i>overfitting</i> . Círculos y triángulos rellenos datos de entrenamiento y vacíos, datos de test. [Fuente: csie.ntu.edu.tw]	51
5.3. Representación del espacio ROC.	54
5.4. Ejemplo de validación cruzada K-Fold con 4 iteraciones.[Fuente: Wikipedia]	55

5.5. Probabilidad de agrupar $n$ puntos en en 2 clases linealmente separables cuando la dimensión del espacio de características es $m$ . . . . .	57
6.1. Dos ejemplos de máscara construidas para un umbral: a) $t=1$ y b) $t=60$ . . . . .	63
6.2. Imagen formada por la media de todas las imágenes de tipo NC,AD y MCI. . . . .	64
6.3. Algunas secciones transversales representativas de imágenes originales a) NC, b) MCI y c) AD, d) la media de dichas imágenes y e) la máscara aplicada formada a partir de la imagen media. . . . .	66
6.4. Representación de los 4 primeros eigenbrains en orden descentente trás aplicar PCA al Grupo 1 (AD+NC). . . . .	68
6.5. Distribuciones de la primera componente principal para las 3 clases, estimadas mediante KDE sobre la base de datos completa: NC+AD+MCI. . . . .	69
6.6. Distribuciones estimadas sobre la base de datos original de la primera CP para cada grupo a) Grupo 1, b) Grupo 2, c) Grupo 3 y comparativa de las distribuciones NC y AD. . . . .	70
6.7. Generación de coordenadas en el espacio de eigenbrains. . . . .	71
6.8. Componentes Principales originales y generadas mediante interpolación para clase a) NC, b) AD y c) MCI. En rojo las CPs originales y en azul las CPs generadas. . . . .	72
6.9. Imagen de Control Normal a) original y b) simulada. . . . .	74
6.10. Imagen de clase MCI a) original y b) simulada. . . . .	74
6.11. Imagen de clase AD a) original y b) simulada. . . . .	74
7.1. Evolución de la precisión en función del tamaño del conjunto generado correspondiente a la clasificación del Grupo 1 (AD vs NC). Experimento 1. . . . .	77
7.2. Espacio ROC experimento 1: Grupo 1 (AD vs NC). . . . .	78
7.3. Espacio ROC experimento 1: Grupo 2 (MCI vs NC). . . . .	79
7.4. Espacio ROC experimento 1: Grupo 3 (AD vs MCI). . . . .	80
7.5. Evolución de la precisión en función del tamaño del conjunto generado correspondiente a la clasificación del Grupo 1 (Experimento 2). . . . .	82
7.6. Espacio ROC experimento 2: Grupo 1 (AD vs NC). . . . .	83
7.7. Espacio ROC experimento 2: Grupo 2 (MCI vs NC). . . . .	84
7.8. Espacio ROC experimento 2: Grupo 3 (AD vs MCI). . . . .	85
7.9. Evolución de la PDF correspondiente a la primera Componente Principal para distintos tamaños de sets generados, para la clases NC y AD del Grupo 1. . . . .	86

8.1. Diagrama de Gantt asociado. . . . .	89
A.1. Evolución de la precisión, especificidad y sensibilidad para los 3 grupos del experimento 1. Columna 1: ADvsNC. Columna 2: MCIvsNC. Columna 3: ADvsMCI. . . . .	115
A.2. Evolución de la precisión, especificidad y sensibilidad para los 3 grupos del experimento 2. Columna 1: ADvsNC. Columna 2: MCIvsNC. Columna 3: ADvsMCI. . . . .	116
A.3. Evolución de la precisión, especificidad y sensibilidad para los 3 grupos del experimento 2. Columna 1: MCIasNC. Columna 2: MCIasAD. . . . .	117
A.4. Resultados de la clasificación del experimento 1 en el espacio ROC. . . . .	118
A.5. Resultados de la clasificación del experimento 2 en el espacio ROC. . . . .	119



# Índice de cuadros

4.1. Algunas de las funciones núcleo más comunes. . . . .	38
5.1. Posibles resultados de la clasificación binaria. . . . .	52
7.1. Resultados Experimento 1. . . . .	76
7.2. Resultados Experimento 2. . . . .	81
7.3. Resultados Experimento 3. . . . .	83
8.1. Coste asociado a los recursos humanos. . . . .	90
8.2. Coste asociado a los recursos software. . . . .	90
8.3. Coste asociado a los recursos hardware. . . . .	90
8.4. Coste total asociado. . . . .	91
A.1. Resultados Experimento 1. . . . .	113
A.2. Resultados Experimento 2. . . . .	114
A.3. Resultados Experimento 3. . . . .	114



**Parte I**

**INTRODUCCIÓN  
GENERAL**



# Capítulo 1

## Introducción

### 1.1. Motivación

El avance tecnológico en las últimas décadas, en cuanto a imágenes médicas se refiere, ha hecho posible que médicos y especialistas tengan a su alcance numerosas herramientas para el estudio del cerebro. Se abre así un nuevo enfoque en el estudio de enfermedades neurodegenerativas, permitiendo la caracterización de diferentes patrones anatómicos y funcionales asociados a éstas.

En esta línea, se han realizado numerosos esfuerzos para el diagnóstico precoz de dichas enfermedades, tratando de encontrar un sistema objetivo capaz de realizar un análisis semi-automático de las imágenes funcionales cerebrales. Se han desarrollado así un conjunto de algoritmos y herramientas estadísticas para tal fin, dando lugar a los sistemas de Diagnóstico Asistido por Ordenador, CAD (del inglés, *Computer Aided Diagnosis*).

Sin embargo, debido a que las imágenes funcionales representan una gran cantidad de datos (voxels, el equivalente tridimensional de los pixels) en comparación con el número de imágenes de que se dispone, nos encontramos con un problema fundamental a la hora de evaluar nuevos algoritmos CAD: el problema del pequeño tamaño muestral o como se conoce generalmente en la literatura, *the small sample size problem*.

### 1.2. Enfermedad de Alzheimer

La enfermedad de Alzheimer, AD (del inglés, *Alzheimer's Disease*) es la forma más común de demencia entre las personas mayores [1]. El término demencia deriva del latín *demens, dementatus* que significa “sin mente”. Definida según la Organización Mundial de la Salud, en la Clasificación In-

ternacional de las enfermedades - Décima Edición (CIE-10, 1992), “como un síndrome debido a una enfermedad del cerebro, generalmente de naturaleza crónica o progresiva en la que hay déficits de múltiples funciones corticales superiores que repercuten en la actividad cotidiana del enfermo”.

Existen numerosas enfermedades similares al Alzheimer que pueden causar demencia, como el Parkinson, la demencia de cuerpos de Lewy o la enfermedad de Huntington. Todas ellas son enfermedades neurodegenerativas que presentan un factor común, la destrucción progresiva de células cerebrales. Por ello, no solo es importante diagnosticar un paciente con demencia, sino también ser capaces de distinguir entre los diferentes tipos de demencia existentes.

El Alzheimer involucra las partes del cerebro que controlan el pensamiento, la memoria y el lenguaje, mediante un deterioro progresivo de la corteza cerebral, que en su primera etapa se caracteriza por leves pérdidas de memoria y que finalmente causa la muerte.

Se trata así, de una enfermedad neurodegenerativa lenta que va asociada a la edad, de hecho, este el factor de riesgo conocido más importante hoy en día. Tanto es así, que el número de personas que lo sufren se duplica cada 5 años entre las personas mayores de 65 años y alcanza el 30 de los mayores de 85 años. De acuerdo con la Fundación Alzheimer España, cerca de 700.000 personas sufren la enfermedad en España y se manifiestan más de 100.000 nuevos enfermos cada año. Se estima que en Europa 8 millones de personas padecen la enfermedad de Alzheimer, alcanzando la cifra de 30 millones en todo el mundo.

El curso de la enfermedad varía en función de la persona, pero en media los pacientes que sufren de esta enfermedad viven de 8 a 10 años desde que son diagnosticados, alargándose hasta los 20 años en algunos casos.

La enfermedad de Alzheimer es llamada así por su descubridor Alois Alzheimer, que en 1906 detectó cambios en el tejido cerebral de una de sus pacientes, encontrando aglutinaciones anormales (conocidas actualmente como “placas de amiloide”) y una masa confusa de fibras, ahora denominadas “enredo de neurofibrillas”[2]. Ambas siguen siendo consideradas un sello característico de la enfermedad, aunque hoy en día los científicos han encontrado otros cambios en el cerebro de las personas que la padecen. Entre estos cambios está la pérdida de células nerviosas en áreas del cerebro, provocando una disminución cortical severa y un encogimiento del hipocampo, vital para la memoria. Por otro lado, se produce una disminución de sustancias químicas responsables de la transmisión de mensajes entre células nerviosas.

Debido al hecho de que actualmente no existe cura para la enfermedad de Alzheimer, entender las causas y obtener un diagnóstico precoz de la enfermedad, se ha convertido en uno de los retos más importantes del siglo

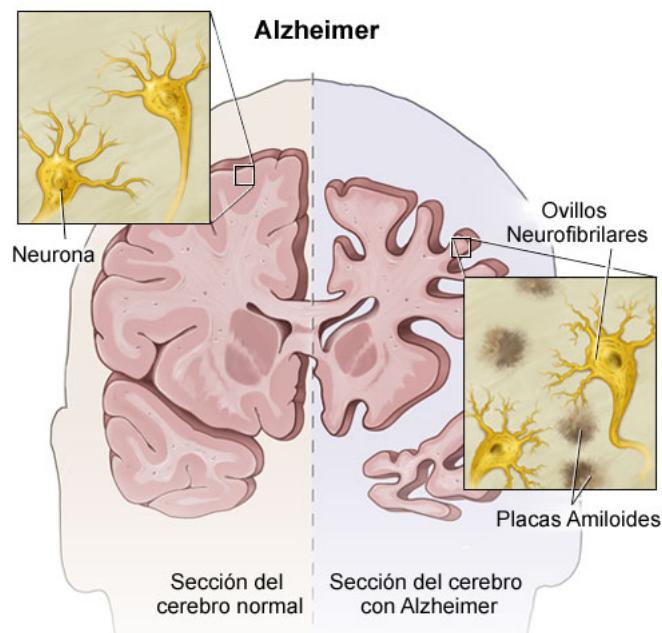


Figura 1.1: Placas de amiloide y neurofibrillas [Fuente: stanfordhealthcare.org]

XXI.

### 1.2.1. Síntomas externos

La principal característica de la enfermedad de Alzheimer es la pérdida de memoria. No obstante se trata de un proceso lento y progresivo. La fase más temprana se caracteriza por pequeños olvidos, con problemas para recordar nombres de objetos cotidianos, eventos recientes o nombres de personas, que pueden causar irritabilidad en las personas que la padecen, pero que no son tan serios como para causar alarma. De hecho, puede pasar un periodo de tiempo largo, hablamos de años, hasta que se manifiesta de forma notoria.

A medida que la enfermedad avanza, esta interfiere en las actividades cotidianas, olvidando la forma de realizar tareas sencillas como cepillarse los dientes o peinarse. Además, favorece la desorientación en espacios no familiares y la pérdida de habilidad para asimilar nueva información. Los problemas de conducta en esta fase se acentúan, ya no pueden pensar con claridad y pueden volverse más ansiosas o agresivas.

En una etapa avanzada, las personas que padecen la enfermedad sufren de un severo deterioro de las funciones cognitivas y corporales, perdiendo por completo el sentido del tiempo y el espacio. En esta fase es frecuen-

te que no sean capaces de reconocer cosas y situaciones que anteriormente les resultaban muy familiares. Asimismo, la comunicación verbal disminuye hasta el punto de repetir frases, palabras o simplemente sílabas constantemente y aparecen a su vez problemas físicos, con limitaciones de movilidad y coordinación que reducen drásticamente la independencia de los pacientes.

Por tanto, los principales síntomas clínicos usados para el diagnóstico del AD son problemas de memoria (amnesia), problemas en el lenguaje, apraxia y agnosia.

### 1.2.2. Diagnóstico

Actualmente, el diagnóstico de la enfermedad de Alzheimer se basa fundamentalmente en la evaluación clínica, lo que requiere un exhaustivo examen neuropsicológico.

No existe un test *pre-mortem* que pueda diagnosticar concluyentemente la enfermedad de Alzheimer, por lo que la evaluación neuropsicológica se basa en una serie de test que permiten cuantificar el grado de demencia del paciente. Entre los más usados se encuentran test sencillos como el Mini-Examen del Estado Mental (MMSE por sus siglas en inglés) y otras escalas más complejas en las que se establecen los dominios de la función cognitiva afectadas como la Clasificación Clínica de la Demencia de Hughes (CDR, del inglés *Clinical Dementia Rating*) o la Escala de Deterioro Global (GDS por sus siglas en inglés). Mientras que el MMSE es un test sencillo que evalúa principalmente el lenguaje, la atención y la memoria inmediata, la escala GDS, establece en siete estadios posibles la progresión del AD.

Existen dos posibles diagnósticos para la enfermedad de Alzheimer: posible o probable.

**Possible** Diagnóstico basado en la observación de síntomas clínicos y el deterioro de dos o más funciones cognitivas (como la memoria, el lenguaje o la atención) en presencia de una enfermedad secundaria, que no siendo causante de la demencia, hace el diagnóstico del Alzheimer menos certero.

**Probable** Diagnóstico basado en los mismos criterios usados para diagnosticar la enfermedad de Alzheimer como posible, pero en ausencia de una enfermedad secundaria.

Un diagnóstico que confirme la enfermedad de Alzheimer solo es posible tras un análisis *post-mortem*, mediante una autopsia neuropatológica que confirme la existencia de placas de amiloide y enredos de neurofibrillas en el cerebro (veáse Figura 1.1).

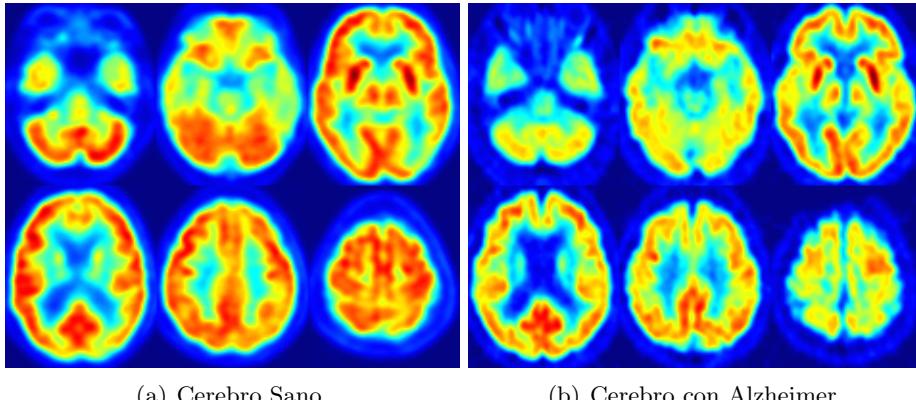


Figura 1.2: Imágenes PET-FDG.

El uso de test neuropsicológicos, como diagnóstico clínico, es frecuentemente poco preciso para detectar la enfermedad de Alzheimer. Sin embargo, el desarrollo de la medicina nuclear en las últimas décadas, ha favorecido la aparición de nuevas técnicas menos invasivas para la obtención de imágenes funcionales cerebrales, como apoyo a la detección del AD.

### 1.3. Tomografía por Emisión de Positrones

En esta sección introducimos la Tomografía por Emisión de Positrones, PET (del inglés, *Positron Emission Tomography*), como técnica ampliamente usada para el estudio de las propiedades funcionales del cerebro [3], siendo además el origen de las imágenes que usaremos para el desarrollo y evaluación del algoritmo propuesto en el Capítulo 6.

#### 1.3.1. Imágenes PET

La Tomografía por Emisión de Positrones es una técnica no invasiva de diagnóstico por imagen *in vivo*, capaz de medir la actividad metabólica del cuerpo humano, mediante la ingestión de un radiofármaco de vida media ultracorta administrado a través de una inyección intravenosa.

El proceso de obtención de estas imágenes se inicia con la ingestión de un trazador de la actividad metabólica. Se trata de una molécula que contiene un isótopo radiactivo emisor de positrones (por ejemplo,  $^{11}\text{C}$ ,  $^{13}\text{N}$ ,  $^{15}\text{O}$  ó  $^{18}\text{F}$ ). Pasados unos minutos, el isótopo se acumula en áreas con las que la molécula tiene cierta afinidad. Por ejemplo uno de los más importantes, el flúor de la molécula de fluorodesoxiglucosa (FDG por sus siglas en inglés) es seleccionado para ser convertido en flúor-18 ( $^{18}\text{F}$ ), un isótopo radiactivo

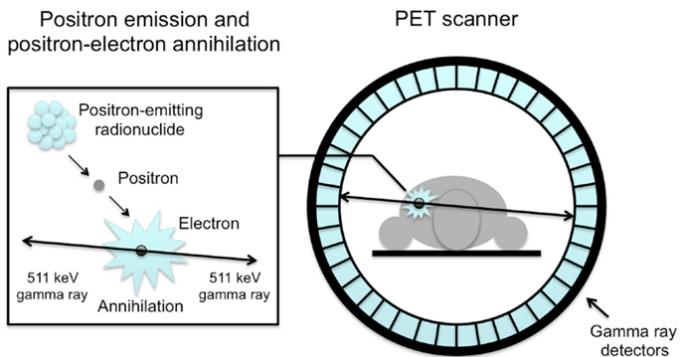


Figura 1.3: Obtención de la imagen PET.

emisor de positrones, dando así lugar a una molécula  $^{18}\text{F}$ -FDG. La glucosa etiquetada con  $^{18}\text{F}$  es incorporada por las células que están desarrollando actividad metabólica, como neuronas activas o células cancerosas.

En nuestro caso, las imágenes PET ( $^{18}\text{F}$ -FDG) utilizadas (ver Figura 1.2), representan de manera tridimensional la tasa de metabolismo de glucosa en el cerebro. Numerosos estudios han relacionado esta disminución del metabolismo de glucosa en regiones características con la enfermedad de Alzheimer. Concretamente regiones bilaterales en los lóbulos temporal y parietal, cíngulo posterior y precunei, así como en el cortex frontal y el conjunto global del cerebro en casos de afección severa [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14].

### 1.3.2. Obtención de la imagen

La tomografía por emisión de positrones PET se remonta a los años 50, cuando investigadores estadounidenses demostraron que los fotones de alta energía producidos por aniquilación de positrones podrían emplearse para describir tridimensionalmente la distribución fisiológica de compuestos químicos. En los años 80 ya se había consolidado la técnica PET como técnica tomográfica para el estudio del cuerpo humano.

Un positrón es la antimateria del electrón, ambos tienen misma masa pero distinta carga. Los radioisotopos usados en PET son isótopos ricos en protones que se descomponen a través de la emisión de positrones. Estos positrones viajan a través de los tejidos durante un corto periodo de tiempo, interactuando con los electrones presentes en el tejido. En cada interacción el positrón pierde energía cinética y finalmente choca con un electrón produciéndose la aniquilación positrón-electrón. De esta aniquilación mutua se emiten dos fotones gamma, de 511 keV cada uno.

Estos fotones que viajan en direcciones opuestas son registrados por los

detectores PET. Para que estos fotones acaben por conformar la imagen, deben detectarse “en coincidencia”, en una ventana de tiempo adecuada (nanosegundos) y además su energía debe superar un umbral mínimo que certifique que no ha sufrido dispersiones energéticas de importancia en su trayecto (fenómeno de scatter) hasta los detectores.

Los detectores de un tomógrafo PET (ver Figura 1.4) están dispuestos en anillo alrededor del paciente y debido a que los fotones son detectados “en coindicencia” se conforma la imagen. Previamente estos fotones se convierten en señales eléctricas, que tras un proceso de filtrado y reconstrucción, permiten obtener una imagen funcional tridimensional.

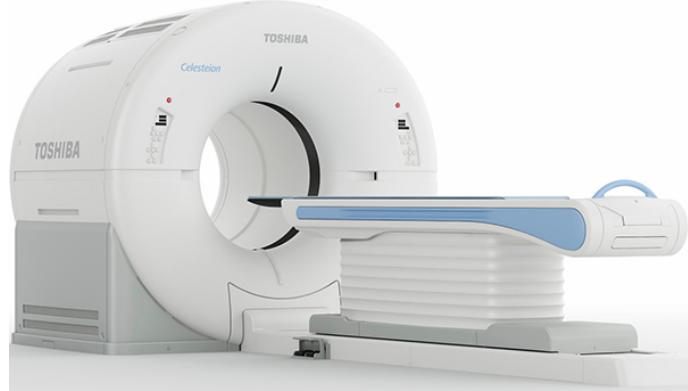


Figura 1.4: Toshiba Celesteion PET/CT Scanner [Fuente: [www.medgadget.com](http://www.medgadget.com)].

### 1.3.3. Reconstrucción

Se ha demostrado que las imágenes de sección se pueden obtener a partir de las proyecciones obtenidas [15, 16, 17, 18]. Observando la figura 6.5 podemos ver como el vector de proyección para un determinado corte es la colección de integrales de linea (también denominadas LOR) que pasan a través del objeto a un determinado ángulo  $\theta$

De esta manera, si representamos el objeto como una función bidimensional  $f(x, y)$ , la proyección dada para todos los ángulos  $0 < \theta < \pi$  y muestras radiales,  $-\infty < s < \infty$ , entonces:

$$P(\theta, s) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - s) dx dy \quad (1.1)$$

donde  $P(\theta, s)$  es conocida como la Transformada de Radon de la función  $f(x, y)$ .

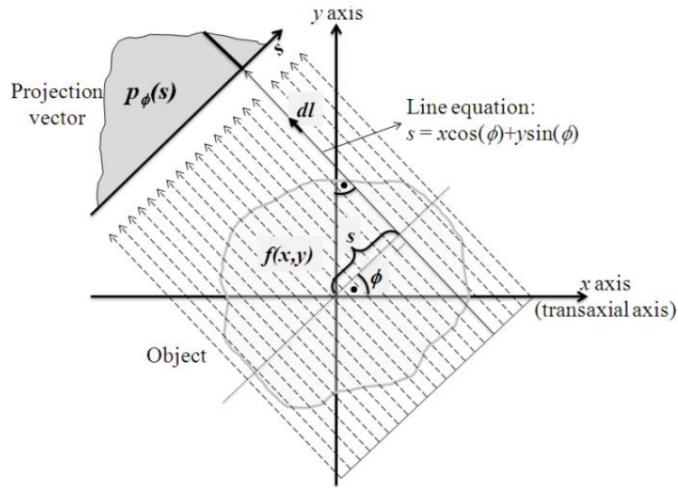


Figura 1.5: Geometría para la construcción de las proyecciones.

La reconstrucción de la imagen tomográfica está basada en el teorema de la Transformada de Fourier de los Cortes (véase 1.6), el cual establece que la transformada de Fourier 1D del vector de proyección, es equivalente al corte de la transformada de Fourier 2D del objeto en el mismo ángulo.

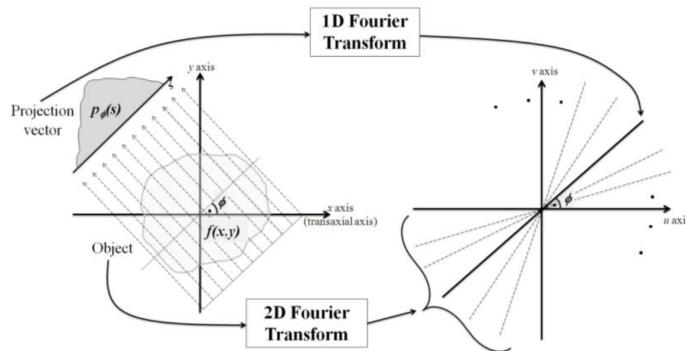


Figura 1.6: Representación del Teorema de Fourier de los Cortes.

### Teorema 1.1 *Teorema de Fourier de Cortes*

Se define la transformada de Fourier  $S(\theta, w)$  de la proyección paralela en un determinado ángulo,  $P_\theta$  de una imagen  $f(x, y)$ , como:

$$S(\theta, w) = \int_{-\infty}^{+\infty} P(\theta, s) \exp(-j2\pi ws) ds \quad (1.2)$$

Esta transformada proporciona una corte de la transformada de Fourier 2D:

$$F(u, v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \exp(-j2\pi(ux + vy)) dx dy \quad (1.3)$$

de forma que:

$$S(\theta, w) = F(u = w \cos \theta, v = w \sin \theta). \quad (1.4)$$

En este estudio, las proyecciones se han obtenido usando el método de reconstrucción filtrada hacia atrás, FBP (del inglés, *Filtered Back Projection*). Así, un corte de un objeto se puede recuperar de la siguiente manera:

$$f(x, y) = \int_0^{\pi} Q_{\theta}(x \cos \theta + y \sin \theta) d\theta, \quad (1.5)$$

donde

$$Q_{\theta} = \int_{-\infty}^{+\infty} S_{\theta}(w) |w| \exp(j2\pi ws) dw. \quad (1.6)$$

Es importante notar que un gran inconveniente de este algoritmo radica en que amplifica de forma indeseada el ruido de altas frecuencias, lo que afecta a la imagen. Este efecto se produce en la ecuación 1.6, al multiplicar  $S_{\theta}(w)$  por  $|w|$ . Para paliar el problema y atenuar el ruido de altas frecuencias se ha propuesto el uso de funciones tipo ventana:

- Ventana de Hamming
- Ventana de Hanning
- Sinc (filtro de Shepp-Logan)
- Coseno

#### 1.4. Estado del arte

Como hemos visto, el diagnóstico de la enfermedad de Alzheimer aún representa un reto en la actualidad, especialmente en la etapa precoz. Aunque en la práctica el diagnóstico se basa fundamentalmente entrevistas, test neurológicos como el MMSE y juicios clínicos, en las últimas décadas se ha incrementado notablemente el uso de técnicas de neuroimagen como ayuda al diagnóstico de enfermedades neurodegenerativas, como la enfermedad del Alzheimer o el Parkinson.

De esta manera, la capacidad de predictiva de la medicina nuclear con respecto a la enfermedad de Alzheimer y otras demencias, ha sido y sigue siendo, objeto de estudio en la literatura [10, 19, 20]. Así las modalidades

SPECT y PET han sido ampliamente usadas en investigación biomédica y en la práctica clínica en las últimas décadas. Sin embargo, debido a que los scáneres tridimensionales PET tienen mayor sensibilidad que los SPECT de alta resolución, los primeros son preferibles en estudios funcionales donde se requiere analizar estructuras pequeñas como el hipocampo (región típicamente afectada por AD) [21, 22, 23].

En esta línea se han desarrollado sistemas para el análisis de imágenes a nivel de voxel mediante el uso de sistemas CAD, que permiten hacer comparaciones estadísticas voxel a voxel, creando así mapas paramétricos de efectos significativos. Estas técnicas univariadas permiten un examen global del cerebro a nivel de voxel y no requieren ninguna hipótesis regional previa. Ejemplos de estos sistemas son SPM [24] o NEUROSTAT & 3D-SSP [25]. Sin embargo, para el diagnóstico, la información importante tiene carácter regional y no puntual, por lo que también se han desarrollado técnicas multivariadas para obtener datos a analizar con otras herramientas estadísticas, como ANOVA, MANCOVA o NEUROSTAT & 3D-SSP [26, 27, 28, 29, 30, 31, 32, 33].

La última tendencia consiste en nuevas técnicas para el procesado, modelado y clasificación de imágenes basadas en la *teoría del aprendizaje estadístico por computador*, que veremos con más detalle en el Capítulo 5, y que tienen como objetivo entrenar un clasificador a partir de una base de datos conocida para predecir nuevos sujetos. Estudios basados en estos métodos de clasificación incluyen, la Enfermedad de Alzheimer [34, 35, 36]; Trastorno Depresivo Mayor, MDD (del inglés *Major Depressive Disorder*) [37, 38]; Trastorno del Espectro Autista, ASD (del inglés *Autism Spectrum Disorder*) [39, 40]; Esquizofrenia [41], Trastorno Cognitivo Leve, MCI (del inglés *Mild Cognitive Impairment*) [42] y Trastorno por Déficit de Atención e Hiperactividad [43, 44, 45].

En neuroimagen, sin embargo, los métodos basados en clasificación estadística se encuentran con un problema fundamental, el problema del pequeño tamaño muestral [46, 27, 47, 48], fenómeno que se da cuando el tamaño de la población (número de pacientes) es pequeño en comparación con la dimensión del espacio de características (vóxeles). Normalmente se disponen de bases de datos de menos de 1000 sujetos (50-100 es un número realista), en comparación con las imágenes funcionales cerebrales, que una vez pre-procesadas contienen ( $> 100.000$ ) vóxeles. En la sección 5.4, veremos los fundamentos de este problema con mayor profundidad.

Consecuentemente, si no se seleccionan las características más relevantes, el modelo predictivo está sujeto a padecer un sobreajuste o *overfitting* [49, 50], que tiene como efecto principal la disminución de la capacidad de generalización, entendida ésta como la capacidad de hacer predicciones precisas sobre nuevas bases de datos, dando lugar a modelos con una menor

potencia estadística [49, 50, 51]. Se ha afirmado y demostrado, que como consecuencia de ello, muchas de las conclusiones arrojadas en el campo de la investigación biomédicas sean probablemente falsas o erróneas [52].

Es por ello que se han propuesto numerosas soluciones en la bibliografía tratando de paliar este efecto de manera parcial, como el uso de clasificadores con mayor capacidad de generalización mediante la optimización de distintos parámetros [48, 53, 54, 55] o métodos de extracción de características para la reducción de la dimensionalidad [31, 56, 57]. Otras soluciones que están cogiendo fuerza están basadas en bases de datos adquiridas en múltiples centros, cuyo objetivo es conseguir un mayor número de imágenes en un periodo de tiempo menor. Sin embargo, estos estudios se ven afectados por diversos problemas técnicos y logísticos, como aprobaciones en su regularización o el uso de distintos escáneres entre otros.

## 1.5. Objetivos

En este trabajo, enfocamos el problema del pequeño tamaño muestral (véase 5.4) desde un nuevo punto de vista. Así, en lugar de tratar de optimizar el cómputo de distintos parámetros para hacer los sistemas más generalizables o reducir la dimensionalidad del espacio de características, proponemos un algoritmo novel para la simulación de imágenes funcionales del cerebro, lo que nos permite ampliar el número de muestras que formarán parte del conjunto de entrenamiento de los clasificadores.

El objetivo final es poder reducir el sesgo en la evaluación de nuevos sistemas CAD, de manera que aumente la veracidad de los análisis y se mejoren los métodos que hasta ahora han sido desarrollados sobre bases de datos afectadas por el problema del tamaño muestral, y en un futuro próximo poder incluir nuevos sistemas automáticos en la práctica clínica para el diagnóstico de enfermedades neurodegenerativas, como el Alzheimer o el Parkinson entre otras.

## 1.6. Publicaciones

Durante el desarrollo de este proyecto hemos presentando el algoritmo propuesto a la Conferencia de Imágenes Médicas, IEEE MIC que ha sido aceptado para su publicación:

- U. Vidal-Sanz, F.J. Martínez-Murcia, J.M. Górriz, J. Ramírez, I.A. Illán, F. Segovia, D. Salas-González. *Simulating Functional Brain Images in Alzheimer's Disease*.



## **Parte II**

# **FUNDAMENTOS**



## Capítulo 2

# Base de Datos

La base de datos usada para el desarrollo y evaluación del algoritmo presentado en este trabajo está compuesta por 403 imágenes FDG-PET, las cuales han sido obtenidas de la Iniciativa en Neuroimagen para la Enfermedad del Alzheimer, ADNI (Alzheimer's Disease Neuroimaging Initiative).

### 2.1. ADNI

EL proyecto ADNI [58] se inicia en el año 2003 gracias a organizaciones de distinta índole como el Instituto Nacional sobre el Envejecimiento (NIA por sus siglas en inglés), el Instituto Nacional de Imágenes Biomédicas y Bioingeniería (NIBIB pos sus siglas en inglés) o la Administración de Alimentos y Medicamentos (FDA por sus siglas en inglés), así como compañías farmacéuticas privadas y organizaciones sin ánimo de lucro, como un proyecto conjunto con financiación público-privada que ascendía a \$60 millones de dólares.

El primer objetivo de este proyecto, dirigido por el investigador principal Michael W. Weiner<sup>1</sup>, es probar si se pueden combinar las técnicas de Imagen por Resonancia Magnética (MRI por sus siglas en inglés) y PET [59], así como otros marcadores biológicos y evaluaciones neuropsicológicas y clínicas, para evaluar la evolución del AD en sus primeras etapas.

Inicialmente se quería conseguir que alrededor de 800 pacientes participen en el proyecto, con edades comprendidas entre los 55 y 90 años, aproximadamente unos 200 personas sin enfermedad pero de edades avanzadas para un seguimiento 3 años, alrededor de 200 personas con síntomas de AD con un seguimiento de 2 años y unas 400 personas con deterioro cognitivo

---

<sup>1</sup>Michael W. Weiner, M.D., VA Medical Center and University of California - San Francisco.

leve<sup>2</sup>.

### 2.1.1. Protocolo de adquisición de las imágenes

Debido al gran número de centros participantes en el proyecto y a la variabilidad de los recursos materiales de que disponían para realizar las exploraciones, se establecieron tres protocolos estandarizados para la adquisición de imágenes FDG-PET:

- **Estáticas:** una toma de 30 minutos en las mismas condiciones.
- **Dinámicas:** 6 tomas de 5 minutos, adquiridas de 30 a 60 minutos tras la inyección intravenosa compuesta por  $5.0 \pm 0.5$  mCi del trazador  $^{18}F$ -FDG<sup>3</sup>.
- **Cuantitativas:** basadas en una exploración dinámica de doble duración, consistente en 33 tomas desde el momento de la inyección hasta transcurridos una hora.

### 2.1.2. Preprocesado

Aunque la mayoría de las exploraciones fueron tomadas siguiendo el protocolo de adquisición dinámico, la variabilidad en los escáneres usados por los diferentes centros resulta en diferentes tipos de imágenes. De esta manera, la iniciativa ADNI pretende minimizar, en lo posible, estas diferencias, proporcionando diferentes conjuntos de datos ordenados según el tratamiento recibido.

A continuación detallamos el preprocesado de las imágenes ADNI:

1. **Co-registro dinámico:** todas las imágenes PET originales las procede la Universidad de Míchigan para el control de calidad, convirtiéndolas a un formato estándar. Se extraen entonces las distintas secuencias de la imagen para registralas. Normalmente se toman 6 secuencias que son registradas. Estas secuencias se co-registran de nuevo a una imagen conjunto. Estas imágenes tienen el mismo tamaño y dimensión de vox y además, mantienen la misma orientación espacial que la imagen original PET. Se dice que estas imágenes están en el espacio ‘nativo’. Estos archivos se cargan en el Laboratorio de Neuroimagen (LONI)<sup>4</sup>

---

<sup>2</sup>Detalles del proyecto ADNI en [www.adni-info.org](http://www.adni-info.org).

<sup>3</sup>El curio (abreviación Ci) es una unidad de radiactividad, nombrada así en homenaje a los físicos y químicos Pierre y Marie Curie. milicurio (mCi) =  $10^{-3}$  Ci

<sup>4</sup>Laboratory of Neuro Imaging (LONI) es un laboratorio de neuroinvestigación fundado junto el Departamento de Neurología en la Escuela de Medicina Geffen.

en formato DICOM (Digital Imaging and Communication in Medicine)<sup>5</sup>. Este paso sólo será necesario las imágenes adquiridas con el primer o tercer protocolo. Una vez realizado este paso se obtienen:

- secuencias separadas co-registradas entre sí para paliar los efectos de movimiento del paciente.
- imágenes en formato DICOM.

2. **Promediado:** Se promedian las imágenes resultado del paso anterior, creando una única imagen PET, todavía en el espacio ‘nativo’.

3. **Imagen y tamaño de voxel estandarizado:** Las imágenes anteriores están definidas a un estándar de 160x160x96 voxels y cada voxel con un tamaño cúbico de lado 1.5mm. Además las imágenes están orientadas de tal manera que el eje anterior-posterior se mantenga paralelo a la recta desde la comisura anterior a la comisura posterior, llamada línea AC-PC (véase Figura 2.1).

Gracias a esta normalización de la imagen, sirve como referencia para las exploraciones PET futuras correspondientes al mismo paciente. Así las secuencias individuales de cada exploración se co-registran teniendo la línea AC-PC como referencia.

4. **Resolución uniforme:** El objetivo de esta etapa es producir imágenes con una resolución uniforme de 8mm de Anchura a Media Altura, FWHM (*Full Width at Half Maximum*)<sup>6</sup>, aproximándose a la resolución más baja dada por ciertos escáneres usados en el proyecto.

5. **Normalización espacial:** Dada la complejidad de las estructuras cerebrales y las diferencias existentes entre cerebros de distintos sujetos, hace necesaria la normalización de las imágenes con respecto a una plantilla común. De esta manera, podemos comparar las intensidades de los voxels de las imágenes cerebrales de pacientes diferentes. Así, mediante SPM [24] las imágenes son normalizadas espacialmente, de modo que conseguimos que los voxels de distintas imágenes se refieran a la misma posición anatómica del cerebro. Esto se logra con una deformación de la imagen de tal manera que quede encajada con una plantilla que finalmente se usará como un patrón. El patrón

---

<sup>5</sup>DICOM (Digital Imaging and Communication in Medicine) es el estándar reconocido mundialmente para el intercambio de pruebas médicas, pensado para su manejo, visualización, almacenamiento, impresión y transmisión. Permite la integración de escáneres, servidores, estaciones de trabajo, impresoras y hardware de red de múltiples proveedores dentro de un sistema de almacenamiento y comunicación de imágenes.

<sup>6</sup>FWHM es una medida de la extensión de una función, que viene dada por la diferencia entre los dos valores extremos de la variable independiente en los que la variable dependiente es igual a la mitad de su valor máximo.

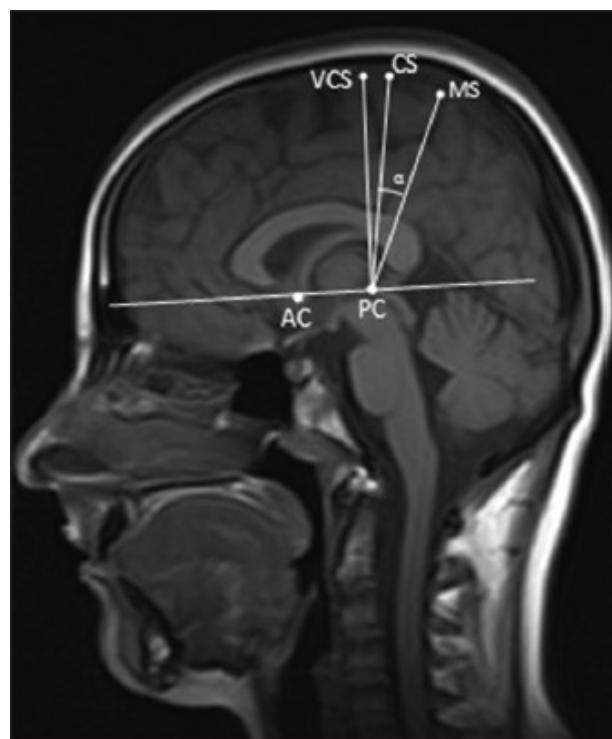


Figura 2.1: Línea AC-PC [Fuente: iranjradiol.com]

usado obtiene como un promedio del conjuntos de imágenes de control normal. De este modo, las regiones del cerebro de cada sujeto se corresponden con su localización en el espacio estándar.

Este método de normalización asume un modelo genérico afín con 12 parámetros [60] y una función de coste que presenta un valor extremo cuando la plantilla y la imagen corresponden una con la otra [61]. Así la función a optimizar es la diferencia cuadrática media de la imagen usada como plantilla y la imagen usada como fuente:

$$CF = \sum_i (f(\mathbf{Mx}_i) - g(\mathbf{x}_i))^2, \quad (2.1)$$

En esta ecuación  $g$  hace referencia a la plantilla, mientras que  $f$  hace referencia a la imagen original. De este modo, para cada voxel  $\mathbf{x} = (x_1, x_2, x_3)$ , las transformación afín de  $\mathbf{y} = (y_1, y_2, y_3)$  será por tanto  $\mathbf{y} = \mathbf{Mx}$ :

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} \quad (2.2)$$

Una vez aplicada la transformación afín, las imagen resultante se registra ahora usando un modelo de transformación no rígido más complejo. De esta manera, las deformaciones no lineales se ajustan a una plantilla de Montreal Neurological Imaging (MNI), parametrizándose por una combinación lineal de componentes de frecuencias más bajas de las bases de la transformadas coseno tridimensional [62].

Tras las normalización espacial, se obtienen imágenes de 95x69x79 voxels y cada voxel representa un volumen cerebral de 2.18x2.18x3.56 mm<sup>3</sup>. Ahora es posible hacer comparaciones entre imágenes, ya que hemos asegurado la correspondencia espacial de los voxels entre distintas imágenes, minimizando así los efectos producidos por las diferencias individuales de cada sujeto, sin modificar las diferencias relacionadas con la enfermedad.

En la Figura 2.2 podemos observar el resultado del proceso de normalización espacial de imágenes PET. La columna de la izquierda muestra la imagen en el espacio nativo, la columna central representa la plantilla de normalización usada y por último, la tercera columna muestra la imagen original registrada, que se ajusta a la plantilla.

### 2.1.3. Criterios de Etiquetado

Con el objetivo se hacer una selección de los participantes en el proyecto ADNI, estos eran sometidos a una serie de entrevistas y test que permitía

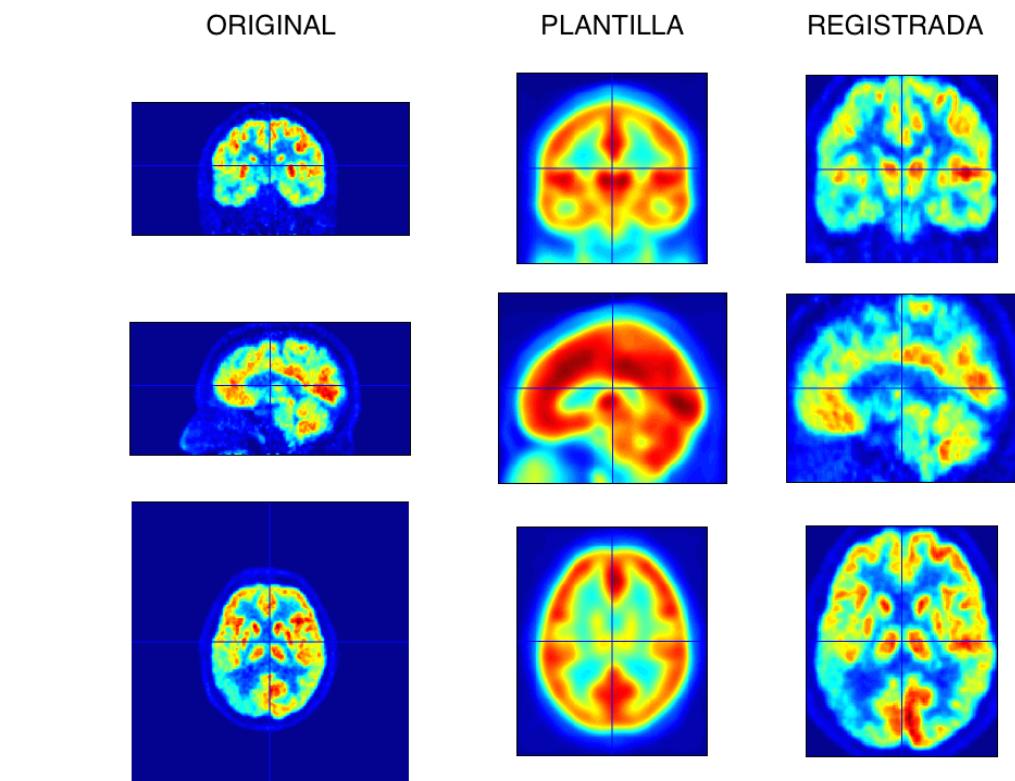


Figura 2.2: Proceso de normalización espacial de imágenes PET. *Columna izq.*: Imagen original. *Columna central*: Plantilla. *Columna der.*: Imagen registrada.

etiquetarlos adecuadamente. Así estos eran asignados a las siguientes clases en función de la edad y en función de las puntuaciones obtenidas en los test MMSE y CDR:

- **Pacientes AD (Alzheimer Disease):** pacientes con un resultado MMSE entre 20-26 (inclusive), un CDR entre 0.5-1 y tener un diagnóstico de AD probable según los criterios NINCDS/ADRDA.
- **Pacientes MCI (Mild Cognitive Impairment):** pacientes con deterioro cognitivo leve, un resultado en el test MMSE entre 24-30 (inclusive), síntomas de leves pérdidas de memoria y un CDR de 0.5.
- **Pacientes NC (Normal Control):** pacientes sin demencia con un resultado MMSE entre 24-30 (inclusive), CDR de 0, y sin depresión.

Aunque el etiquetado basado en tests neurológicos no es completamente fiable [52], se asume como *gold standard* para el desarrollo de este trabajo.

Es importante destacar que dentro del grupo MCI distinguimos dos subgrupos: los que finalmente se convierten al grupo AD (MCI converters) y los que finalmente no padecen AD (MCI no converters). Sin embargo, aunque no haremos una distinción específica a la hora de evaluar los resultados, veremos que el hecho de existir ambos grupos tendrá implicaciones tanto en la estimación de la distribución de la población MCI (véase 6.5), así como en la clasificación.



## Capítulo 3

# Análisis de Componentes Principales

Las imágenes cerebrales usadas en neuroimagen presentan un número muy elevado de características, del orden de 500.000 véxeles. Sin embargo, no serán características individuales, sino conjuntos de ellas, las que gobiernen el comportamiento del sistema.

El Análisis de Componentes Principales, PCA (del inglés, *Principal Component Analysis*) como técnica estadística [63], aprovecha esta ventaja de los sistemas multivariados, para extraer las características más significativas de un conjunto de datos, permitiendo reducir así la dimensionalidad de dicho conjunto. Es por ello que este método es usado ampliamente en la literatura en diversos campos, como la detección de AD [64], reconocimiento de emociones [65], reconocimiento facial [66], clasificación de electrocardiogramas [67] o escoliosis [68]. PCA se basa en la acción de una transformación lineal (también conocida como la transformación *Karhunen-Loéve*) sobre un conjuntos de datos de media nula, que diagonaliza su matriz de covarianza. Este método genera un nuevo conjunto de variables, llamadas Componentes Principales, donde cada componente es una combinación lineal de las variables originales y además son ortogonales entre sí, por lo que no hay redundancia de información.

### 3.1. PCA

Como ya hemos introducido, el algoritmo PCA tiene como objetivo transformar un conjunto de observaciones  $\Gamma$  de variables posiblemente correlacionadas, en un nuevo conjunto de variables linealmente no correladas, denominadas Componentes Principales (PC por sus siglas en inglés).

Nuestro conjunto de datos está formado por  $n$  imágenes cerebrales 3D  $\boldsymbol{\Gamma}_i$ , cuyo tamaño típico es de  $m = 79 \times 95 \times 69 \sim 5 \cdot 10^5$  voxels y se entenderá en este contexto como un conjunto de vectores columna  $\boldsymbol{\Gamma}_i \in \mathbb{R}^m, i = 1, 2, \dots, n$ , formados por la concatenación de los voxels de la imagen. Así,  $\boldsymbol{\Gamma}_i^T = (\Gamma_1, \Gamma_2, \dots, \Gamma_m)_i$ , donde  $\Gamma_j$  representa el valor de intensidad correspondiente al voxel  $j$ .

Para entender el fundamento sobre el que se construye PCA, empezaremos viendo la transformación de Karhunen-Loéve:

Sea  $\boldsymbol{\Gamma}_i \in \mathbb{R}^m$  un vector  $m$ -dimensional, existe una representación exacta de este a través de un conjunto de  $m$  vectores linealmente independientes  $\mathbf{e}_i \in \mathbb{R}^m$  como:

$$\boldsymbol{\Gamma} = \sum_{i=1}^m z_i \mathbf{e}_i \quad (3.1)$$

donde se asume que los vectores  $\mathbf{e}_i$  están sujetos a la condición de ortogonalidad:

$$\mathbf{e}_i^T \mathbf{e}_j = \delta_{ij} \quad (3.2)$$

donde  $\delta_{ij}$  es la delta de Kronecker. De esta manera, la ecuación (3.1) no describe otra cosa que un cambio de coordenadas a una nueva base ortonormal de  $\mathbb{R}^m$ , donde las coordenadas del vector  $\boldsymbol{\Gamma}$  en la nueva base vendrán dadas por:

$$z_i = \mathbf{e}_i^T \boldsymbol{\Gamma} \quad (3.3)$$

De modo que llamaremos a esta coordenada  $z_i$ , la componente  $i$ -ésima en el nuevo espacio engendrado por la base  $\{\mathbf{e}_i\}$ .

Sin embargo, supongamos que estamos interesados en aproximar  $\boldsymbol{\Gamma}$  usando un número reducido ( $p < m$ ) de vectores de la base  $\{\mathbf{e}_i\}$ . Así, podemos sustituir algunas componentes  $z_i$  por constantes arbitrarias  $b_i$ , llegando a la siguiente aproximación de  $\boldsymbol{\Gamma}$ :

$$\hat{\boldsymbol{\Gamma}} = \sum_{i=1}^p z_i \mathbf{e}_i + \sum_{i=p+1}^m b_i \mathbf{e}_i \quad (3.4)$$

Por tanto el error que se comete al aproximar  $\boldsymbol{\Gamma}$  por  $\hat{\boldsymbol{\Gamma}}$  viene dado por:

$$\Delta \boldsymbol{\Gamma} = \boldsymbol{\Gamma} - \hat{\boldsymbol{\Gamma}} = \sum_{i=1}^m z_i \mathbf{e}_i - \sum_{i=1}^p z_i \mathbf{e}_i - \sum_{i=p+1}^m b_i \mathbf{e}_i = \sum_{i=p+1}^m (z_i - b_i) \mathbf{e}_i \quad (3.5)$$

Con el fin de obtener una solución óptima al problema planteado sobre la aproximación, vamos a seguir el criterio de mínimos cuadrados, para encontrar el valor de las constantes  $b_i$  que minimiz el error cuadrado de la media, mse, dado por:

$$mse = E \{ \Delta \Gamma^2 \} = \sum_{i=p+1}^m E \{ (z_i - b_i)^2 \} \quad (3.6)$$

Lo que equivale a encontrar una solución a:

$$\frac{\partial}{\partial b_i} E \{ (z_i - b_i)^2 \} = -2(E \{ z_i \} - b_i) = 0 \quad (3.7)$$

lo que conduce a:

$$b_i = E \{ z_i \} = \mathbf{e}_i^T E \{ \Gamma \} \quad (3.8)$$

quedando así especificadas las constantes  $b_i$  a el valor esperado de las componentes  $z_i$ . Podemos entonces reescribir el error en (8.6) como:

$$\begin{aligned} mse &= \sum_{i=p+1}^m E \{ (z_i - E \{ z_i \})^2 \} \\ &= \sum_{i=p+1}^m \mathbf{e}_i^T E \{ (\Gamma - E \{ \Gamma \}) \} E \{ (\Gamma - E \{ \Gamma \}) \}^T \mathbf{e}_i \\ &= \sum_{i=p+1}^m \mathbf{e}_i^T \Sigma_{\Gamma} \mathbf{e}_i \end{aligned} \quad (3.9)$$

donde  $\Sigma_{\Gamma}$  es la matriz de covarianza de  $\Gamma$ . Se puede demostrar (ver [69, 70]) que la elección óptima para  $\mathbf{e}_i$  es aquella que cumple:

$$\Sigma_{\Gamma} \mathbf{e}_i = \lambda_i \mathbf{e}_i \quad (3.10)$$

o dicho de otra forma, aquella en la que  $\mathbf{e}_i$  y  $\lambda_i$  son los autovectores y autovalores de la matriz de covarianza.

Antes de continuar, es importante notar de la ecuación (3.8) que la transformación de Karhunen-Loéve de los datos originales, se simplifica si se centran los datos extrayéndose la media muestral, de manera que se elimina el segundo término en la ecuación (3.4) que no contiene información relevante sobre la varianza de los datos. Se ha demostrado que, más que una simplificación, trabajar con datos de media nula es una condición necesaria para la obtención de las componentes principales [4].

Además, tanto el valor esperado como la matriz de covarianza, han de ser estimados con la media muestral de los datos:

$$\bar{\Gamma} = \frac{1}{n} \sum_{i=1}^n \Gamma_i \quad (3.11)$$

y la covarianza muestral:

$$\mathbf{C} = \frac{1}{n-1} \sum_{i=1}^n (\boldsymbol{\Gamma}_i - \bar{\boldsymbol{\Gamma}})(\boldsymbol{\Gamma}_i - \bar{\boldsymbol{\Gamma}})^T \quad (3.12)$$

Por tanto la transformación PCA actuará sobre el conjunto:

$$\boldsymbol{\Phi}_i = \boldsymbol{\Gamma}_i - \bar{\boldsymbol{\Gamma}} \quad i = 1, 2, \dots, n \quad (3.13)$$

y estará compuesta por un conjunto de  $m$  autovectores ortogonales  $\mathbf{e}_i$  de la matriz de covarianza muestral:

$$\mathbf{C} = \frac{1}{n-1} \sum_{i=1}^n \boldsymbol{\Phi}_i \boldsymbol{\Phi}_i^T = \frac{1}{n-1} \mathbf{A} \mathbf{A}^T \quad (3.14)$$

donde  $\mathbf{A} = [\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \dots, \boldsymbol{\Phi}_m]$ , tales que:

$$\mathbf{z}_i = \mathbf{e}_i^T \mathbf{A} \quad (3.15)$$

Este vector fila  $\mathbf{z}_i \in \mathbb{R}^n$ , será la  $i$ -ésima Componente Principal. Existirán por tanto  $m$  componentes principales, entre las que no existirá correlación ya que su matriz de covarianza será diagonal.

Ademas, los autovalores  $\lambda_i$  de la matriz de covarianza nos darán la varianza de las componentes principales puesto que:

$$\mathbf{E}^T \Sigma_{\Phi} \mathbf{E} = \Sigma_{\mathbf{Z}} = \text{diag}(\lambda_1^2, \lambda_2^2, \dots, \lambda_m^2) \quad (3.16)$$

donde  $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m]$  y  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m]$ . De las propiedades de la traza se deduce también que:

$$\text{tr}(\Sigma_{\Phi}) = \text{tr}(\Sigma_{\mathbf{Z}}) = \sum_{i=1}^n \lambda_i^2 \quad (3.17)$$

Esta propiedad nos permite además darle un interpretación a la ecuación de minimización del error (3.9):

$$\text{MMSE} = \sum_{i=p+1}^m \mathbf{e}_i^T \Sigma_{\Phi} \mathbf{e}_i = \sum_{i=p+1}^m \Sigma_{\mathbf{Z}} = \sum_{i=p+1}^m \lambda_i^2 \quad (3.18)$$

donde ahora MMSE es el error cuadrático medio mínimo.

El proceso de optimización buscara por tanto, los  $m-p$  valores de  $\lambda_i$  cuya suma sea mínima, o lo que es lo mismo, seleccionará aquellas  $p$  componentes principales cuya varianza sea máxima. En la figura 3.1 podemos ver representada esta idea. En ella aparece representada la contribución a la varianza de las 10 primeras comonentes principales, que suponen casi un 40 % de la

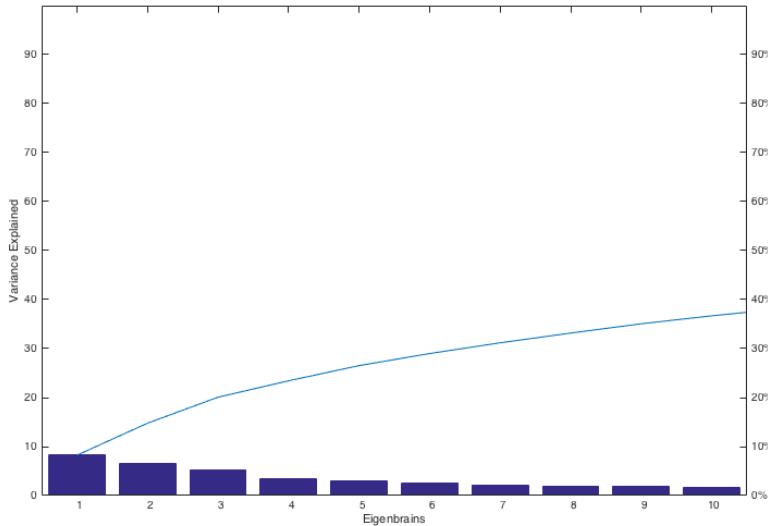


Figura 3.1: Contribución a la varianza de las 10 primeras Componentes Principales.

varianza total. Así, la aproximación (3.4) vendrá dada por la combinación lineal de los  $p$  autovectores  $\mathbf{e}_i$  cuyas componentes principales tengan mayor varianza, recogiéndose las características de mayor variabilidad de los datos en un número de variables  $p < m$ .

PCA se reduce la diagonalización de la matriz de covarianza de los datos centrados, mediante un conjunto de autovectores ortonormales, lo cual siempre es posible debido al teorema de descomposición espectral, ya que es una matriz simétrica definida positiva.

### 3.1.1. Eigenbrains

De aquí en adelante, usaremos el término *eigenbrains* para referirnos a los autovectores  $\mathbf{e}_i$ , por su apariencia de imagen cerebral, refiriéndonos también al espacio que engendran como '*espacio de eigenbrains*'. Usamos esta terminología en correspondencia con la usada en el campo de reconocimiento facial, dónde se usa el término *eigenfaces* para denominar a los autovalores de la matriz de covarianza [71].

Para obtener los *eigenbrains*, será necesario diagonalizar una matriz  $m \times m$ , que en el caso de imágenes cerebrales se convertirá en un matriz  $\sim 5 \cdot 10^5 \times 5 \cdot 10^5$ . Sin embargo, podemos reducir la complejidad computacional considerablemente si se ataca el problema de diagonalizar la matriz  $\hat{\mathbf{C}} =$

$\mathbf{A}^T \mathbf{A}$ , cuyo tamaño es  $n \times n$ , normalmente con  $n < m$ . Si llamamos  $\mathbf{v}_j \in \mathbb{R}^n$  a los autovectores de  $\hat{\mathbf{C}}$ :

$$\mathbf{A}^T \mathbf{A} \mathbf{v}_j = \lambda_j \mathbf{v}_j, \quad i = 1, 2, \dots, n \quad (3.19)$$

Multiplicando esta ecuación por la izquierda por  $\mathbf{A}$ , obtenemos:

$$\mathbf{A} \mathbf{A}^T \mathbf{A} \mathbf{v}_j = \lambda_j \mathbf{A} \mathbf{v}_j \quad (3.20)$$

de manera que  $\mathbf{A} \mathbf{v}_j$  son autovectores de  $\mathbf{A} \mathbf{A}^T$ . Esto nos permite obtener  $n$  de los  $m$  autovectores  $\mathbf{e}_i$  de  $\mathbf{C}$ , de los autovectores  $\mathbf{v}_j$  de  $\hat{\mathbf{C}}$  como:

$$\mathbf{e}_j = \mathbf{A} \mathbf{v}_j, \quad j = 1, 2, \dots, n \quad (3.21)$$

Por lo general, solo será necesario un número reducido de eigenbrains obtener la mayor parte de la varianza muestral, por lo que solo un pequeño número  $p$  será necesario para describir apropiadamente un conjunto de datos.

### 3.1.2. Descomposición en Valores Singulares

Hasta ahora no ha quedado demostrado que los  $n$  eigenbrains obtenidos a través de  $\hat{\mathbf{C}}$  sean suficientes o adecuados para representar bien la matriz de datos original. Sin embargo, existe un justificación que demuestra que en lo son y está basado en la relación de PCA con la Descomposición de Valor Singular, SVD (del inglés, *Singular Value Descomposition*).

La transformación (3.15) es equivalente a la SVD de la matriz de datos  $\mathbf{A}$ , que viene dada por:

$$\mathbf{A} = \mathbf{E} \mathbf{D} \mathbf{V}^T \quad (3.22)$$

donde  $\mathbf{E}$  y  $\mathbf{V}$  son matrices ortogonales  $m \times m$  y  $n \times n$  respectivamente y  $\mathbf{D}$  es una matriz diagonal  $m \times n$ . La SVD garantiza que la matrix  $\mathbf{D}$  queda únicamente determinada por el valor de  $\mathbf{A}$  si los valores de  $\mathbf{D}$  están organizados en orden decreciente, mientras que las matrices  $\mathbf{E}$  y  $\mathbf{V}$  no quedan completamente determinadas. Podemos expresar la matriz  $\mathbf{C}$  en términos de la descomposición singular como:

$$\mathbf{C} = \mathbf{A} \mathbf{A}^T = \mathbf{E} \mathbf{D} \mathbf{D}^T \mathbf{E}^T \quad (3.23)$$

que no es otra cosa que la ecuación (3.16) con:

$$\mathbf{D} \mathbf{D}^T = \Sigma_{\mathbf{Z}} = \text{diag}(\lambda_1^2, \lambda_2^2, \dots, \lambda_m^2) \quad (3.24)$$

Como antes, hacemos lo mismo con  $\hat{\mathbf{C}}$ , llegando a:

$$\hat{\mathbf{C}} = \mathbf{A}^T \mathbf{A} = \mathbf{V} \mathbf{D}^T \mathbf{D} \mathbf{V}^T \quad (3.25)$$

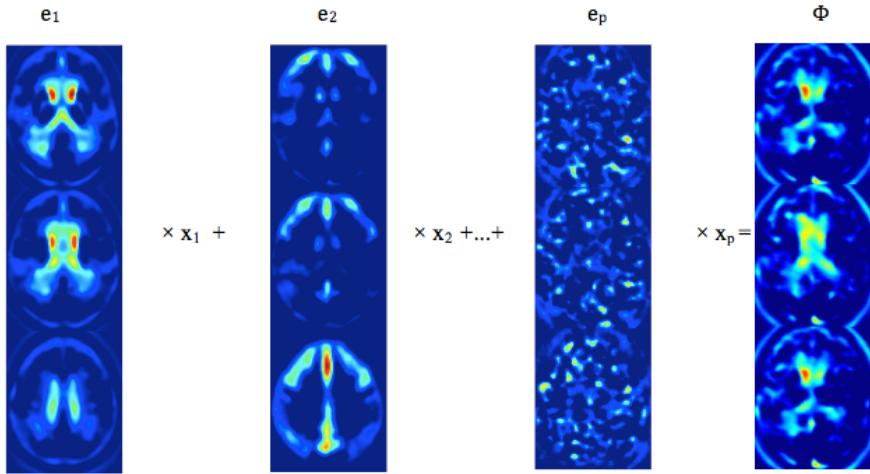


Figura 3.2: Representación en términos de eigenbrains de la imagen de media cero  $\Phi$ , mostrando tres cortes transaxiales representativos.

donde ahora:

$$\mathbf{D}^T \mathbf{D} = \text{diag}(\lambda_1^2, \lambda_2^2, \dots, \lambda_n^2) \quad (3.26)$$

Puesto que  $\mathbf{D}$  es diagonal la única forma que puede tomar es:

$$D = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 & & \vdots \\ 0 & 0 & \dots & \lambda_n & \dots & 0 \end{pmatrix} \quad (3.27)$$

lo que obliga a que al menos  $\lambda_k = 0$  con  $k = n+1, \dots, m$ , garantizando así que sólo es necesario calcular los  $n$  eigenbrains obtenidos a través de  $\hat{\mathbf{C}}$ . En nuestro caso y por la implementación en Matlab, usaremos los  $n-1$  primeros eigenbrains, de forma que  $p = n-1$ .

### 3.2. Interpretación de PCA

Por lo tanto, sabiendo que usaremos un número  $p$  de eigenbrains para representar nuestras imágenes originales, podemos ver la ecuación la ecuación 3.15 como:

$$\mathbf{x}_n = \Phi_j^T \tilde{\mathbf{E}} \quad (3.28)$$

donde ahora  $\tilde{\mathbf{E}} = [\mathbf{e}_1, \dots, \mathbf{e}_p]$ . Este vector  $\mathbf{x}_n \in \mathbb{R}^p$  no es más que la proyección de la imagen original de media cero sobre el espacio de eigenbrains, obtenido mediante una reorganización de las componentes principales.

Esto se puede apreciar en la figura 3.2, donde se representa la imagen  $\Phi_i$  de media cero, en términos de los  $p$  eigenbrains y de las componentes principales.

Resumiendo, con PCA podemos trasladar las coordenadas de nuestra base de datos original, formada por 403 imágenes cerebrales cuyas variables están posiblemente correladas, a un nuevo espacio de eigenbrains. Así, proyectando cada observación en la nueva base de eigenbrains, obtenemos nuevas variables decorreladas, las componentes principales, que representan de manera más fiel el comportamiento del sistema, reduciéndose además de manera considerable el número de variables que lo describen. PCA nos permite por tanto reducir la dimensionalidad del espacio de características, pasando de  $\mathbb{R}^m$  a  $\mathbb{R}^p$ , donde  $m \sim 10^6$ ,  $n \sim 10^2$  y  $p < n$ .

## Capítulo 4

# Estimación de la Función de Densidad

La Estadística es la ciencia de los datos, entendiendo como “datos” un conjunto de observaciones generalmente numéricas, obtenidas mediante la observación reiterada de un experimento de interés [72]. Hoy en día, la estadística no se concibe sin el concepto de distribución de probabilidad de una variable aleatoria, entendido este como un modelo matemático que describe el comportamiento probabilístico de la misma [73]. La forma más común de la distribución de una variable aleatoria se corresponde con las denominadas funciones de distribución y de densidad de probabilidad de la variable aleatoria. Conocer la función de densidad de un variable aleatoria, nos da un descripción completa de la misma, por lo que su estimación es un problema fundamental de la estadística.

En el capítulo anterior usábamos PCA para modelar nuestra base de datos, extrayendo nuevas variables denominadas las Componentes Principales, que representaban fielmente el comportamiento de nuestro sistema. No obstante, necesitamos conocer la distribución que siguen las CP que gobiernan el comportamiento de dicho sistema, para poder simular nuevas imágenes cerebrales que cumplan las características de las originales.

### 4.1. Estimacion de la Función de Densidad

Generalmente, la estadística inferencial se aplica al estudio de una cierta magnitud aleatoria o “variable aleatoria” (v.a.). Normalmente, esta variable corresponde a la observación de una cierta característica en los individuos de una gran población y por eso se emplean a veces las palabras población y “poblacional” para referirnos a la variable aleatoria y a sus características. La teoría de la probabilidad proporciona las herramientas para identificar y

definir las características de esa variable que interesa estudiar (distribución, media, varianza,...) y proporciona algunos modelos típicos que aparecen con frecuencia en las aplicaciones.

En nuestro caso, las variables a estudiar serán cada una de las componentes principales,  $\mathbf{z}_i$ , obtenidas mediante PCA pertenecientes a tres poblaciones diferentes: sujetos AD, MCI y NC. De estas componentes estamos interesados en estimar su función de densidad, que una vez conocida, nos permitirá generar nuevas coordenadas en el espacio de eigenbrains.

Estimar una densidad consiste en la construcción de una estimación, basada en las observaciones, de una función de densidad de probabilidad subyacente no observable. La función de densidad no observable se considera como la densidad según la cual se distribuye la población; los datos son generalmente considerados como una muestra al azar de esa población.

Si  $\mathbf{X}$  es la variable aleatoria bajo estudio se define su función de distribución en un punto  $x_0$  como la probabilidad de que  $\mathbf{X}$  tome un valor menor o igual que  $x_0$ , es decir:

$$F(x_0) = P(X \leq x_0) \quad (4.1)$$

Por tanto  $P(a < X \leq b) = F(b) - F(a)$ .

Se dice que la distribución de  $\mathbf{X}$  es “(absolutamente) continua” cuando la probabilidad  $P(a < X \leq b)$  puede calcularse como la integral de una cierta función  $f$ , llamada función de densidad:

$$P(a < X \leq b) = \int_a^b f(x)dx \quad (4.2)$$

Por las propiedades de la integral,  $F'(x) = f(x)$  y la distribución de  $\mathbf{X}$  viene caracterizada indistintamente por  $f$  o por  $F$ , en el sentido de que cualquiera de las dos funciones permite calcular la probabilidad de que  $\mathbf{X}$  tome valores en cualquier intervalo prefijado.

## 4.2. Modelos paramétricos y no paramétricos

A la hora de estimar la densidad de una variable aleatoria, podemos optar por usar modelos paramétricos o por el contrario modelos no paramétricos. La palabra “paramétrico” alude al hecho de que estos modelos fijan completamente la distribución excepto por el valor de uno o varios parámetros reales que deben ser estimados.

Mientras que en los modelos paramétricos es necesaria una idea a priori y se supone que la densidad de  $\mathbf{X}$  pertenece a una familia paramétrica conocida (Normal, lognormal, Pareto, Gamma,...), los modelos no paramétricos no

requieren ninguna suposición previa sobre la distribución de la variable bajo estudio. En este sentido los métodos no paramétricos ofrecen mayor flexibilidad y son capaces de estimar una distribución más ajustada a la realidad, permitiendo así que la función de densidad pueda adoptar cualquier forma, sin más límites que los impuestos por las propiedades que se exigen a las funciones de densidad para ser consideradas como tales. Esta es la principal razón por la que basaremos la estimación de la función de densidad de las Componentes Principales en un enfoque no paramétrico. Además, a diferencia de la aproximación paramétrica clásica, no se ve afectado por el sesgo de especificación<sup>1</sup>[74].

La estimación no paramétrica de la función de densidad tiene uno de sus orígenes más comúnmente aceptados en los trabajos de Fix y Hodges [75], que buscaban una alternativa a las técnicas clásicas de análisis discriminante, que permitiera liberarse de las rígidas restricciones sobre la distribución de las variables implicadas. Sin embargo, la elección en el uso de estimadores paramétricos o no paramétricos sigue siendo objeto de controversia hoy en día. Así, a la eficiencia que ofrece la estimación paramétrica se contrapone el riesgo que suponen desviaciones de las suposiciones que determinan el modelo y que pueden conducir por tanto a errores de interpretación, suponiendo mayor pérdida que la ganancia proporcionada por la eficacia estimadora.

A continuación describimos las dos propiedades para evaluar el rendimiento de un estimador:

#### **Definición 4.1 *Sesgo***

*Un estimador  $\hat{f}$  de una función de densidad  $f$  es insesgado para  $f$  si  $\forall x \in \mathbb{R}, E_f[\hat{f}(x)] = f(x)$ . Aunque existen estimadores insesgados conocidos para muchas densidades paramétricas, Rosenblatt [76] demuestra que no es posible la existencia de un estimador de la densidad que verifique  $\int f(x)dx = 1$  y que sea insesgado para todas las densidades continuas. Esto ha motivado que se centre la atención en secuencias de estimadores no paramétricos  $\hat{f}_n$  que sean asintóticamente insesgadas, es decir*

$$E_f[\hat{f}(x)] \rightarrow f(x) \quad \text{si } n \rightarrow \infty \quad (4.3)$$

#### **Definición 4.2 *Consistencia***

*La noción intuitiva de consistencia es que el estimador se aproxime a la verdadera densidad cuando aumenta el tamaño de la muestra. Diremos que un estimador de la densidad  $\hat{f}$  es débilmente consistente en forma puntual si  $\hat{f}(x) \rightarrow f(x)$  en probabilidad para todo  $x \in \mathbb{R}$ , y es fuertemente consistente en forma puntual si la convergencia es casi segura.*

---

<sup>1</sup>El error de especificación se produce cuando una variable independiente se correlaciona con el término de error. Hay varias causas de error de especificación, como variables omitidas o irrelevantes.

### 4.3. Histograma

El histograma es considerado como el estimador no paramétrico de densidad más sencillo. Supongamos que  $f$  tiene soporte en  $[a, b]$  generalmente deducido de los datos, realizamos una partición en  $k$  intervalos no solapados  $B_i = [t_i, t_{i+1})$   $i = 1, \dots, k$  donde  $a = t_1 < t_2 < \dots < t_{k+1} = b$  el histograma viene definido entonces como:

$$\hat{f}(x) = \sum_{i=1}^k \frac{N_i/n}{t_{i+1} - t_i} I_{B_i}(x) \quad (4.4)$$

donde  $N_i$  es el número de datos dentro de  $B_i$ . Si consideramos ahora que la longitud de los intervalos es siempre la misma  $h_n = t_{i+1} - t_i$ , valor que denominamos *ancho de ventana*

$$\hat{f}(x) = \frac{1}{nh_n} \sum_{i=1}^k N_i I_{B_i}(x) \quad (4.5)$$

Así si el ancho de ventana o longitud del intervalo  $h_n$  tiende a cero cuando el número de datos tiende a infinito, cabe esperar que  $\hat{f}_n(x)$  tienda a hacia la densidad instantánea en el punto  $x$  que es precisamente la función de densidad. De hecho, la condición que se requiere para que se produzca la convergencia, cuando  $n \rightarrow \infty$  de  $\hat{f}_n(x)$  hacia la verdadera función de densidad  $f(t)$ , es que  $nh_n \rightarrow \infty$ , además de  $h_n \rightarrow 0$ .

En general, los estimadores no paramétricos dependen paradójicamente de un parámetro de suavizado, en el caso del histograma es el ancho del intervalo. Variando este, obtendremos distinto número de intervalos y por tanto distintas formas de histograma para los mismos datos, como podemos comprobar en la figura 4.1, donde representamos el histograma de la primera componente principal para un sujeto AD. Sin embargo la elección adecuada del parámetro de suavizado es uno de los problemas más delicados de la estadística no paramétrica y ha sido objeto de una investigación intensiva a lo largo de los años 80 y 90. Aunque los histogramas pueden resultar muy ilustrativos para determinados propósitos, son por naturaleza funciones discontinuas, aunque la función de densidad de la variable que se está estimando sea realmente continua. En este sentido, se puede decir que los histogramas son estimadores insatisfactorios. Por otro lado, los estimadores no paramétricos requieren de muestras grandes, pues realizan estimaciones locales basadas en las proximidades del punto local.

A continuación veremos los estimadores tipo núcleo (o kernel), como una versión más sofisticada de los histogramas que fueron diseñados para superar estas dificultades, obteniendo estimaciones más realistas (menos sesgadas y más consistentes).

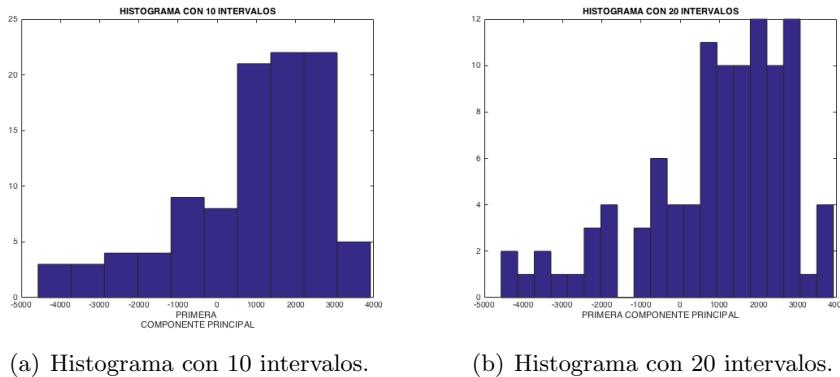


Figura 4.1: Histograma de la primera Componente Principal para un sujeto NC.

#### 4.4. Estimador Tipo Núcleo

Dada la muestra de  $n$  observaciones reales  $X_1, \dots, X_n$  definiremos la estimación tipo Núcleo de función núcleo  $K$  como:

$$\hat{f}_n(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \quad (4.6)$$

donde  $K(x)$  es una función, denominada función Kernel o función núcleo, que satisface ciertas condiciones de regularidad, generalmente es un función de densidad simétrica, y donde  $h > 0$  es un parámetro de suavizado, generalmente denominado ancho de ventana o *bandwidth*.

El estimador núcleo puede interpretarse como una suma de protuberancias (del inglés *bump*) situadas en las observaciones. La función núcleo  $K$  determina la forma de estas protuberancias mientras que el parámetro  $h$  determina su anchura. Al igual que el histograma,  $h$  también determina la cantidad de suavizado de la estimación, siendo el límite, cuando esta tiende a cero, una suma de funciones delta de Dirac en los puntos de las observaciones. También puede interpretarse como una transformación en continua de la función de distribución empírica de acuerdo a la función  $K(x)$  que se encarga de redistribuir la masa de probabilidad  $1/n$  en la vecindad de cada punto muestral [73]. En la Tabla 4.1 mostramos algunas de las funciones univariantes más comunes, mientras que en la Figura 4.2 podemos observar un ejemplo estimación tipo nucleo mostrando los núcleos individuales y su comparación con el histograma.

Por sus ventajas, KDE se ha estudiado ampliamente en la bibliografía [77, 78], siendo el método de estimación no paramétrica más usado en la literatura [79, 80]. Usado también en diversas aplicaciones como visión arti-

Núcleo	$K(t)$	Rango
Epanechnikov	$\frac{3}{4}(1 - t^2)$	$ t  < 1$
Gauss	$\frac{1}{\sqrt{2\pi}}e^{-(1/2)t^2}$	$ t  < \infty$
Triangular	$1 -  t $	$ t  < 1$
Rectangular	$\frac{1}{2}$	$ t  < 1$
Biweight	$\frac{15}{16}(1 - t^2)^2$	$ t  < 1$
Triweight	$\frac{35}{32}(1 - t^2)^3$	$ t  < 1$
Arco coseno	$\frac{\pi}{4} \cos \frac{\pi}{2}t$	$ t  < 1$

Tabla 4.1: Algunas de las funciones núcleo más comunes.

ficial [81, 82, 83], radares en tráfico aéreo [84, 85, 86], filtrado de partículas [87] o domótica [88].

Es evidente que el uso de una ventana de ancho fijo para todas las muestras, puede ser no óptimo para la estimación causando además distorsiones en los extremos. Es por ello que en este trabajo usaremos la estimación por núcleos adaptables a la hora de estimar la densidad de las componentes principales. Concretamente para el desarrollo del algoritmo hemos usado KDE vía difusión [74], que presenta como característica principal la estimación automática del ancho de banda basada en los datos.

#### 4.4.1. KDE vía difusión

El método KDE vía difusión es un método de estimación de densidad mediante el uso de un kernel adaptativo basado en las propiedades de suavizado de procesos de difusión lineales. Además, no asume ningún modelo paramétrico para los datos y es inmune a los problemas de suavizado causado por poblaciones multimodales.

Para el desarrollo de nuestro algoritmo, usaremos la implementación presentada en [74], que asumirá un kernel Gaussiano y una estimación automática del ancho de banda.

A pesar de la gran cantidad de literatura sobre el tema, todavía hay muchas cuestiones polémicas relativas a la implementación y funcionamiento práctico de los estimadores de densidad tipo kernel. Por una lado el método más popular para estimar el ancho de banda automáticamente sobre los datos, los denominados métodos *plug-in* [89, 78], se ven afectados por la regla de referencia normal [90, 91] que es básicamente la construcción de modelo normal preliminar de los datos del que depende el método de elección de ancho de banda. A pesar de que estos métodos funcionan bien cuando la asunción del modelo es aproximada, a nivel conceptual el uso de un modelo de referencia normal invalida nuestra motivación original de aplicar un

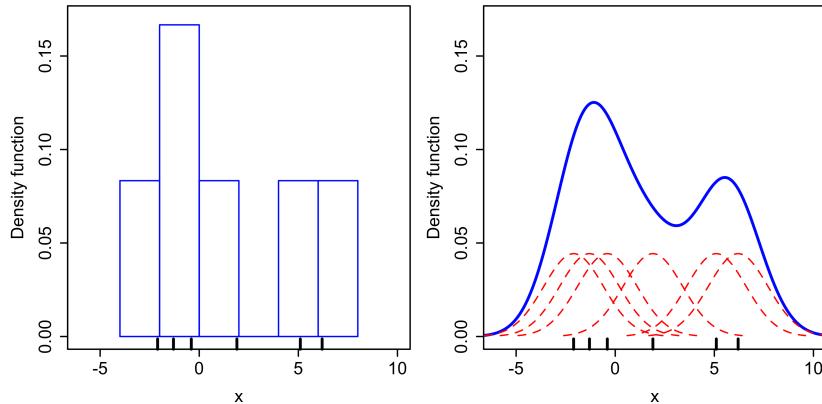


Figura 4.2: Histograma y estimación tipo núcleo mostrando los núcleos individuales. [Fuente: Wikipedia]

método no paramétrico. Otros métodos hacen estimaciones basados en una simple regla empírica [92].

En segundo lugar, el uso del kernel Gaussiano [93] carece de adaptabilidad local, lo que suele resultar en una gran sensibilidad para valores extremos, así como la presencia de protuberancias esporádicas y en general un rendimiento general insatisfactorio en cuanto al sesgo, que además tiene tendencia a aplanar los picos y los valles de la función de densidad [94]. Y en tercer lugar, las mayoría de los kernels sufren de sesgo en los extremos debido a que no suelen tener en cuenta el dominio de los datos [95, 96].

Estos problemas se han solventado parcialmente mediante el uso de kernels más sofisticados. Así se han aplicado kernels de mayor orden para solventar el problema de la adaptabilidad local y reducción del sesgo [97], pero dan lugar a estimaciones negativas pobres y requieren de grandes muestras para un rendimiento óptimo [93]. La falta de adaptabilidad local también se ha abordado introduciendo kernels adaptativos [98, 99, 100, 101].

Como indicábamos, esta variante de KDE introduce un kernel adaptativo basado en las propiedades de suavizado de los procesos lineales de difusión. La idea fundamental es ver el kernel a partir del cual se construye el estimador, como un transición de densidad de un proceso de difusión, usando para ello el proceso de difusión lineal más general, que tiene una densidad de probabilidad acotada y estacionaria. Así, esta densidad estacionaria es seleccionada como estimación de densidad de referencia o como una densidad que los estadísticos creen representa la información acerca de los datos previamente a la observación empírica disponible. Esta aproximación da lugar a un estimador tipo kernel sencillo e intuitivo que reduce sustancialmente el

seso asintótico<sup>2</sup> y el error cuadrático medio en comparación con otros estimadores kernel, dando lugar a estimaciones más acordes con la distribución subyacente.

Así, dadas  $N$  realizaciones independientes ( $X_1, X_2, \dots, X_N$  pertenecientes a una función densidad de probabilidad (PDF) continua desconocida  $f$ , definimos el estimador de densidad con kernel Gaussiano como:

$$\hat{f}(x; t) = \frac{1}{N} \sum_{i=1}^N \phi(x, X_i; t), \quad \forall x \in \mathbb{R} \quad (4.7)$$

donde

$$\phi(x, X_i; t) = \frac{1}{\sqrt{2\pi t}} e^{-(x-X_i)^2/(2t)} \quad (4.8)$$

es un Kernel Gaussiano localizado en  $X_i$  y factor de escala  $\sqrt{t}$ , que antes denominábamos ancho de banda. Como veímos, la elección de este parámetro se ha discutido ampliamente en la literatura, pues el rendimiento de  $\hat{f}$  como estimador de  $f$  tiene una fuerte dependencia con su valor [89, 78]. Un criterio comúnmente usado para la elección óptima de  $t$  es el Error Cuadrático Medio Integrado (MISE, por sus siglas en inglés),

$$MISE\hat{f}(t) = E \int [\hat{f}(x; t) - f(x)]^2 dx \quad (4.9)$$

que convenientemente descompuesto en términos de sesgo y varianza:

$$MISE\hat{f}(t) = \int (E[\hat{f}(x; t)] - f(x))^2 dx + \int Var[\hat{f}(x; t)] dx \quad (4.10)$$

donde el primer término corresponde al sesgo puntual de  $f$  y el segundo término a la varianza puntual de  $f$ . El error MISE depende del ancho de banda  $\sqrt{t}$  y de forma más compleja de  $f$ . Aunque el análisis se simplifica asumiendo la aproximación asintótica de MISE, denominada AMISE, bajo los requisitos de consistencia de manera que  $t = t_N$  depende del tamaño muestral  $N$ , de modo que  $t_N \downarrow 0$  y  $N\sqrt{t_N} \rightarrow \infty$  cuando  $N \rightarrow \infty$ , siendo  $f$  continuamente diferenciable [78]. De este modo, el ancho de banda óptimo se obtiene minimizando AMISE.

Una observación importante a tener en cuenta sobre el estimador kernel de densidad Gaussiano 4.7 es que es solución única a la ecuación parcialmente diferencial de disusión (PDE, por sus siglas en inglés):

$$\frac{\partial}{\partial t} \hat{f}(x; t) = \frac{1}{2} \frac{\partial^2}{\partial x^2} \hat{f}(x; t) \quad x \in X, t > 0 \quad (4.11)$$

---

<sup>2</sup>Que el  $\lim_{n \rightarrow \infty} (E[\hat{\theta}_n] - \theta)$  tienda a cero. Siendo  $\hat{\theta}_n$  un estimador basado en una muestra de tamaño  $n$ .

with  $X \equiv \mathbb{R}$  y condiciones iniciales  $\hat{f}(x; 0) = \Delta(x)$ , donde  $\Delta(x) = \frac{1}{N} \sum_{i=1}^N \delta(x - X_i)$  es la densidad empírica de las muestras  $X_N$  ( $\delta(x - X_i)$  es la delta de Dirac medida en  $X_i$ ). La ecuación 4.11 es bien conocida como la ecuación del calor de Fourier<sup>3</sup> [102]. Esta relación entre la estimador de densidad Gaussiano y la ecuación del calor ha sido discutida en [103]. En la interpretación de la ecuación del calor, el kernel Gaussiano 4.8 es comúnmente conocido como la ecuación verde (en inglés, *Green's function*) del proceso de difusión PDE 4.11. Entonces, el estimador kernel Gaussiano  $\hat{f}(x; t)$  se puede obtener desarrollando la solución parabólica PDE 4.11 hasta el momento  $t$ . En [74] demuestran que el estimador obtenido como solución de PDE es:

$$\hat{f}(x; t) = \frac{1}{N} \sum_{i=1}^N \kappa(x, X_i; t), \quad x \in [0, 1] \quad (4.12)$$

donde el kernel  $\kappa$  viene dado por:

$$\kappa(x, X_i; t) = \sum_{k=-\infty}^{\infty} \phi(x, 2k + X_i; t) + \phi(x, 2k - X_i; t), \quad x \in [0, 1] \quad (4.13)$$

En [74] se demuestran que, para valores pequeños de  $t$ , el estimador 4.12 se comporta como el estimador con kernel Gaussiano 4.7 en  $[0, 1]$ . Por otro lado, cerca de los extremos  $x = 0, 1$ , el nuevo estimador es consistente, mientras que el estimador con Kernel Gaussiano no lo es.

---

<sup>3</sup>La ecuación del calor es una importante ecuación diferencial en derivadas parciales del tipo parabólica que describe la distribución del calor (o variaciones de la temperatura) en una región a lo largo del transcurso del tiempo.



## Capítulo 5

# Aprendizaje Estadístico Supervisado

Podemos decir que cualquier método que incorpore información de un conjunto de muestras con el fin de diseñar una función clasificadora, emplea aprendizaje, refiriéndonos a aprendizaje como un procedimiento algorítmico para reducir el error en la clasificación de un conjunto de muestras. En este contexto, los métodos de clasificación consisten en la extracción de cierta información a partir de un conjunto de imágenes (aprendizaje automático), que es empleado posteriormente para hacer un diagnóstico de nuevos pacientes. El término *supervisado* implica que se disponen de unos datos, llamado conjunto de entrenamiento, de los cuales se conoce previamente su etiqueta. En el caso que nos ocupa en este trabajo, usaremos la clasificación como método de validación, clasificando las imágenes funcionales simuladas, con el objetivo de validar la veracidad de estas.

### 5.1. Clasificación

En el contexto de diagnóstico de enfermedades neurodegenerativas, el objetivo final de la clasificación consiste en el diagnóstico certero de nuevos pacientes. En el aprendizaje supervisado, el proceso de clasificación comprende dos etapas, el entrenamiento y el test.

Como hemos visto, las imágenes están discretizadas y formadas por tanto por un conjunto finito de  $m$  voxels. Estos representan la intensidad en los distintos puntos del cerebro (en nuestro caso al ser imágenes FDG-PET, representan la tasa de consumo de glucosa).

De esta manera, podemos construir vectores a partir de estos valores y formar así los vectores de características para realizar la clasificación.

$$\mathbf{x} = (\phi_1, \phi_2, \dots, \phi_m) \quad (5.1)$$

De esta manera los vectores de características serán objeto de la clasificación y pertenecerán al espacio de características de dimensión  $m$ :  $\mathbf{x}_i \in \mathbb{R}^m, i = 1, \dots, n$ . Estos vectores se pueden dividir en 2 grupos de vectores:

- Vectores pertenecientes al conjunto de entrenamiento:  $\mathbf{x}_i \in X$
- Vectores pertenecientes al conjunto de test:  $\hat{\mathbf{x}}_j \in Y$

donde tanto  $X$  como  $Y$  son dos subconjuntos del espacio  $\mathbb{R}^m$ .

De este modo, construiremos los vectores de características a partir de los datos originales. En nuestro caso, y como veremos más adelante, los vectores de características estarán formados por todos los voxels.

Una vez obtenidos los vectores de características, es posible definir un clasificador:

**Definición 5.1** *Un clasificador se define como una función  $f = f(\mathbf{x}_i, w)$ , que depende de unos parámetros  $w$  y que se construye desde los datos de entrenamiento  $n$ -dimensionales  $\mathbf{x}_i \in X$  y sus etiquetas  $y_i$ , de tal forma que clasifica un nuevo vector de test  $\mathbf{x}_i \in Y$ , asignándole un valor  $z_j \in \pm 1$  correspondiente a cada clase:*

$$\begin{aligned} f : \mathbb{R}^n &\longrightarrow \{\pm 1\} \\ \mathbf{x}_j \in Y &\longmapsto f(\mathbf{x}_j, w) = z_j \end{aligned} \quad (5.2)$$

En resumen, el objetivo final es la construcción de un clasificador que entrenado a partir de un conjunto de vectores de entrenamiento, cuyas etiquetas se conocen, sea capaz de predecir la clase de nuevos vectores. Además se pretende, como es obvio, que el error cometido en esta tarea sea mínimo, optimizando así los parámetros  $w$  del clasificador.

Como indicábamos al principio del capítulo, con el fin de evaluar los resultados derivados de un análisis estadístico se usa lo que se conoce como validación y que veremos en profundidad en la Sección 5.3. De esta manera seremos capaces de medir el rendimiento del clasificador usado y evaluar al mismo tiempo la capacidad de generalización, entendida esta como la capacidad del clasificador para evaluar nuevos conjuntos de datos independientes.

En la clasificación intervienen por tanto varios factores: extracción de características, definición del clasificador en función de las características, aprendizaje o entrenamiento, y test. En la tarea de clasificar imágenes médicas como ayuda al diagnóstico, existe un problema fundamental que afecta a la hora de construir el clasificador: el problema del pequeño tamaño

muestral. Es por ello que en las tareas de diagnóstico, la *extracción de características* se ha convertido en un paso previo fundamental, como solución parcial para obtener clasificadores con mayor capacidad de generalización. Es aquí donde nuestro trabajo ofrece un punto de inflexión, como solución la problema del pequeño tamaño muestral. En la Sección 5.4 veremos los fundamentos de este problema y sus implicaciones a la hora de desarrollar la clasificación sobre un conjunto de datos.

A continuación introduciremos los fundamentos matemáticos de los clasificadores basados máquinas de soporte vectorial, que usamos para la evaluación de nuestro simulador de imágenes cerebrales funcionales y posteriormente veremos la Validación como método para evaluar el rendimiento del clasificador.

## 5.2. Máquinas de Soporte Vectorial

Las máquinas de soporte vectorial, máquinas de vectores de soporte o SVM (del inglés, *Support Vector Machine*) son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T. Desde su introducción a finales de los años 70 [104], las maquinas de soporte vectorial marcaron el inicio de una nueva era en el paradigma del aprendizaje a partir de datos experimentales [105, 106]. Su exitoso uso en el campo de reconocimiento de patrones, ha hecho que estas técnicas se hayan aplicado en numerosas aplicaciones incluyendo la recuperación de imágenes basadas en contenido [107], clasificación de texturas [108] y diagnosis de imágenes médicas [53, 109, 110, 111] o la detección de actividad vocal [112, 113, 114, 115, 116, 117]. Uno de los puntos fuertes de SVM es que proporciona resultados mas ajustados que otros clasificadores convencionales.

### 5.2.1. SVM Lineal

En este trabajo haremos uso de un clasificador lineal, debido a que se trata de un clasificador simple y que posee grandes ventajas computacionales. Abordaremos los fundamentos de este clasificador para el caso en que las clases sean linealmente separables y continuaremos con el caso de clases linealmente no separables, que es en sí una generalización del primer tipo.

#### Clases linealmente separables

Los hiperplanos de decisión o hipersuperficies definidos por los clasificadores lineales vienen dados por:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_o = 0 \quad (5.3)$$

donde  $\mathbf{w}$  se conoce como vector de peso y  $w_0$  como el umbral. El vector de pesos  $\mathbf{w}$  es ortogonal al hiperplano de decisión y la tarea de optimización consiste en encontrar el conjunto de parámetros  $w_i, i = 1, \dots, n$ , que definen el hiperplano de decisión.

Dados los vectores de características pertenecientes al conjunto de entrenamiento, estos pertenecerán a una de las dos clases,  $C^1, C^2$ . Partiendo de que estas son linealmente separables, el objetivo es encontrar el hiperplano que sea capaz de separar todos los vectores del conjunto de entrenamiento. Existirán varios hiperplanos que cumplan este requisito, pero nos interesa aquel con mayor capacidad de generalización, es decir, esto se traduce en encontrar el hiperplano que haga máxima la separación entre clases, o lo que es lo mismo, que tenga un margen máximo entre estas.

Dado que la distancia desde el hiperplano a cualquier punto  $\mathbf{x}$  se define como:

$$z = \frac{|g(\mathbf{x})|}{\|\mathbf{w}\|} \quad (5.4)$$

escalamos  $\mathbf{w}$  y  $w_0$  de modo que el valor de  $g(\mathbf{x})$  en los puntos más cercanos sea  $+1$  para el punto más cercano en  $C^1$  y  $-1$  para el punto más cercano en  $C^2$ . Esto es equivalente a:

- Tener un margen de  $\frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$
- Con las condiciones:

$$\begin{aligned} \mathbf{w}^T \mathbf{x} + w_0 &\geq 1, & \forall \mathbf{x} \in C^1, \\ \mathbf{w}^T \mathbf{x} + w_0 &\leq -1, & \forall \mathbf{x} \in C^2 \end{aligned} \quad (5.5)$$

En la figura 5.1 podemos ver representados estos conceptos.

Si para cada  $\mathbf{x}_i$  asignamos el indicador de clase correspondiente  $y_i$  ( $+1$  para  $C^1$  y  $-1$  para  $C^2$ ), el objetivo es encontrar  $\mathbf{w}$  y  $w_0$  tal que se minimice esta expresión:

$$\mathbf{J}(\mathbf{w}) \equiv \frac{1}{2} \|\mathbf{w}\|^2 \quad (5.6)$$

condicionado a

$$y_i(\mathbf{w}^T \mathbf{x}_i + w_0) \geq 1 \quad \text{para } i = 1, 2, \dots, N \quad (5.7)$$

Este proceso de optimización se rige por las condiciones de Karush-Kuhn-Tucker (KKT):

$$\frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}, w_0, \lambda) = 0 \quad (5.8)$$

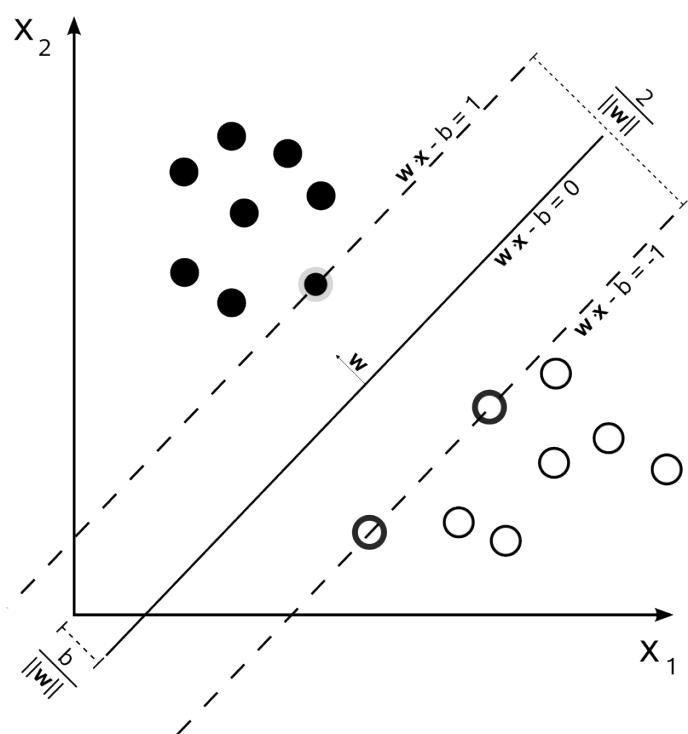


Figura 5.1: Representación de los planos que separan dos clases y hacen máximo el margen entre clases.[Fuente: Wikipedia]

$$\frac{\partial}{\partial w_0} \mathcal{L}(\mathbf{w}, w_0, \lambda) = 0 \quad (5.9)$$

$$\lambda_i \geq 0 \quad \text{para } i = 1, 2, \dots, N \quad (5.10)$$

$$\lambda_i[y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1] = 0 \quad \text{para } i = 1, 2, \dots, N \quad (5.11)$$

donde  $\mathcal{L}(\mathbf{w}, w_0, \lambda)$  es la función Lagrangiana definida como:

$$\mathcal{L}(\mathbf{w}, w_0, \lambda) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \lambda_i [y_i(\mathbf{w}^T \mathbf{x}_i + w_0) - 1] \quad (5.12)$$

Así la ecuación 5.14 en combinación con 5.10 y 5.11 resulta en:

$$\mathbf{w} = \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i \quad (5.13)$$

$$\sum_{i=1}^N \lambda_i y_i = 0 \quad (5.14)$$

Puesto que los multiplicadores de Lagrange serán positivos o cero, el vector paramétrico  $\mathbf{w}$  cuya solución es óptima, dará lugar a una combinación de  $N_s \leq N$  vectores de características asociados a  $\lambda_i \neq 0$ :

$$\mathbf{w} = \sum_{i=1}^{N_s} \lambda_i y_i \mathbf{x}_i \quad (5.15)$$

Conocidos como *vectores de soporte*. De ahí que al hiperplano óptimo que separa ambas clases se denomine máquina de vectores de soporte. Así, para  $\lambda_i \neq 0$ :

$$\mathbf{w}^T \mathbf{x} + w_0 = \pm 1 \quad (5.16)$$

El cálculo de los parámetros se resuelve mediante dualidad Lagrangiana:

Maximizar

$$\mathcal{L}(\mathbf{w}, w_0, \lambda) \quad (5.17)$$

sujeto a:

$$\mathbf{w} = \sum_{i=1}^{N_s} \lambda_i y_i \mathbf{x}_i \quad (5.18)$$

$$\sum_{i=1}^{N_s} \lambda_i y_i = 0 \quad (5.19)$$

$$\lambda \geq 0 \quad (5.20)$$

Si sustituimos 5.20 y 5.21 en 5.19 se puede llegar al problema equivalente

$$\max_{\lambda} \left( \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right) \quad (5.21)$$

sujeto a

$$\sum_{i=1}^{N_s} \lambda_i y_i = 0 \quad (5.22)$$

$$\lambda \geq 0 \quad (5.23)$$

Una vez los multiplicadores de Lagrange han sido calculados, maximizando 5.23, el hiperplano óptimo se calcula con 5.20 y  $w_0$  como antes. Además, es importante notar que la función a optimizar no depende explícitamente de la dimensionalidad, lo que da lugar a una generalización para el caso de clases que no son linealmente separables.

### Clases linealmente no separables

Cuando las clases no son linealmente separables, no existirá una hiperplano entre clases que consiga separar ambas clases sin dejar puntos dentro de la banda.

Puesto que el margen se define como la distancia entre los dos hiperplanos:

$$\mathbf{w}^T \mathbf{x} + w_0 = \pm 1 \quad (5.24)$$

los vectorres de características de entrenamiento pertenecen ahora a una de las siguientes categorías:

- Vectores que cayendo fuera de la banda, son correctamente clasificados. Estos cumplen las restricciones en 5.9.
- Vectores que caen dentro de la banda y son correctamente clasificados. Satisfacen la inecuación:

$$0 \leq y_i(\mathbf{w}^T \mathbf{x} + w_0) < 1 \quad (5.25)$$

- Vectores que son clasificados erróneamente y cumplen la inecuación:

$$y_i(\mathbf{w}^T \mathbf{x} + w_0) < 0 \quad (5.26)$$

Estos casos se pueden tratar como un solo tipo de restricción, introduciendo nuevas variables:

$$y_i(\mathbf{w}^T \mathbf{x} + w_0) \geq 1 - \xi_i \quad (5.27)$$

Así los tres casos anteriores quedan recogidos en esta inecuación:

- El primer grupo se corresponde con  $\xi_i = 0$
- El segundo con  $0 < \xi_i \leq 1$
- El tercero con  $\xi_i > 1$

Luego como decíamos, esto es una generalización del caso de clases linealmente separables.

Por tanto, ahora se quiere conseguir un margen tan grande como sea posible, pero de manera que la cantidad de puntos  $\xi_i > 0$  sea lo menor posible con el objetivo de clasificar correctamente nuevas muestras. En términos matemáticos equivale a minimizar la función de coste:

$$J(\mathbf{w}, w_0, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \mathbf{I}(\xi_i) \quad (5.28)$$

donde  $\xi$  es el vector de parámetros  $\xi_i$  y

$$\mathbf{I}(\xi_i) = \begin{cases} 1 & \xi_i > 0 \\ 0 & \xi_i = 0 \end{cases} \quad (5.29)$$

El parámetro  $C > 0$  es un parámetro de penalización que controla el término de error. Sin embargo, la optimización de 5.30 es difícil puesto que incluye una función discontinua. De esta forma, se elige optimizar una función estrechamente relacionada, y el nuevo problema se traduce en:

Minimizar

$$J(\mathbf{w}, w_0, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i \quad (5.30)$$

sujeto a

$$\begin{aligned} y_i(\mathbf{w}^T \mathbf{x} + w_0) &\geq 1 - \xi_i & i = 1, \dots, N \\ \xi_i &> 0 \end{aligned} \quad (5.31)$$

Realizando pasos similares al caso de clases linealmente separables se llega al problema de optimización equivalente:

$$\max_{\lambda} \left( \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \right) \quad (5.32)$$

sujeto a

$$0 \leq \lambda_i \leq C, \quad \text{para } i = 1, 2, \dots, N \quad (5.33)$$

$$\sum_{i=1}^N \lambda_i y_i = 0 \quad (5.34)$$

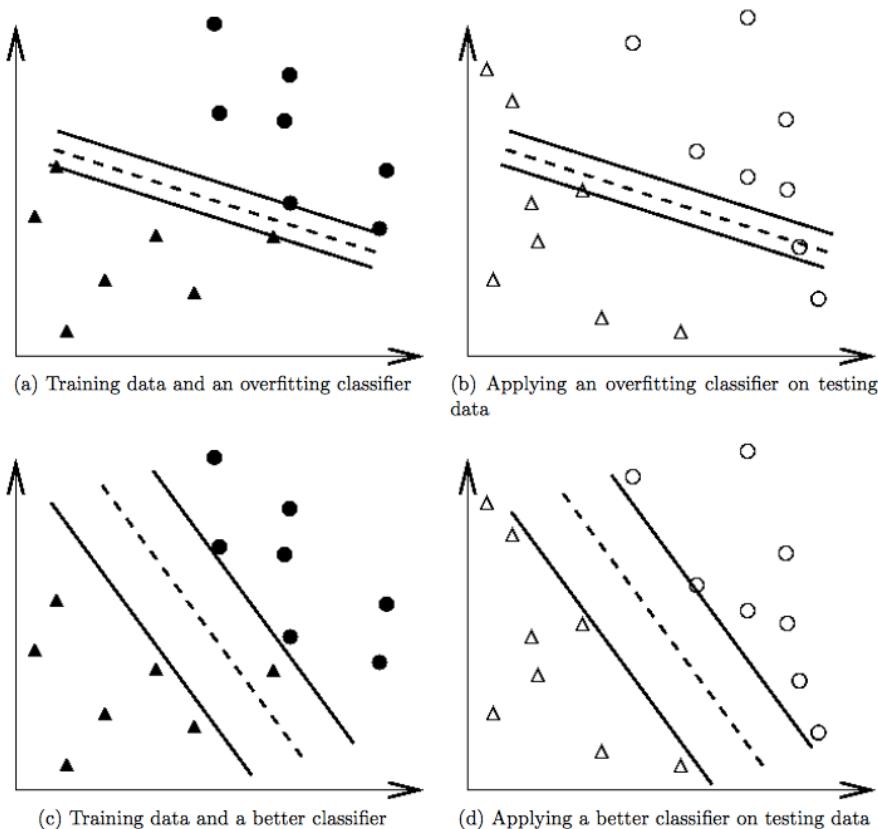


Figura 5.2: Clasificador con *overfitting* y sin *overfitting*. Círculos y triángulos rellenos datos de entrenamiento y vacíos, datos de test. [Fuente: csie.ntu.edu.tw]

Se puede comprobar que la única diferencia con el caso en el que considera clases linealmente separables, está en la primera restricción, donde es necesario limitar por arriba los multiplicadores de Lagrange por  $C$ . El caso linealmente separable corresponde con  $C \rightarrow \infty$ .

En la implementación práctica de SVM, se tratará de minimizar la función de coste 5.28. El parámetro  $C$  se estimará mediante el método de validación cruzada *k-fold* (véase 5.3.2) para evitar el sobreajuste del clasificador. En la Figura 5.2 se representa esta idea. Los círculos y triángulos llenos representan los datos de entrenamiento, mientras que los círculos y triángulos vacíos representan los datos de test. Así, podemos ver en 5.2a) y b) que el clasificador tiene un sobreajuste y por tanto no será un clasificador general, mientras que en 5.2 c) y d) el clasificador no está sobreajustado, por lo que su capacidad de generalización será mayor.

		Etiqueta	
		Positiva	Negativa
Test	Positivo	PV	PF
	Negativo	NF	NV

Tabla 5.1: Posibles resultados de la clasificación binaria.

### 5.3. Validación

Como ya hemos indicado anteriormente, mediante la validación podemos evaluar los resultados del análisis estadístico. Mientras que el capítulo anterior veíamos como construir un clasificador, en este capítulo veremos como usar el método de validación para evaluar el comportamiento de estos.

#### 5.3.1. Parámetros de evaluación del rendimiento de un clasificador

Teniendo en cuenta que tratamos con una clasificación binaria ( $y = \pm 1$ ) el clasificador podrá cometer dos errores. Los posibles resultados de la clasificación se muestran en la tabla 5.1 y puede dar lugar a:

- **Positivo verdadero, TP (True Positive)**, cuando los dos resultados coinciden con valor positivo.
- **Negativo verdadero, TN (True Negative)**, cuando los dos resultados coinciden con valor negativo.
- **Positivo falso, FP (False Positive)**, siendo la clase original negativa, da positivo.
- **Negativo falso, FN (False Negative)**, siendo la clase original positiva, da negativo.

En función del número de TP,TN, FP y FN, se definen 3 parámetros para la evaluación del rendimiento de un clasificador:

- **Sensibilidad**: capacidad del clasificador para detectar positivos verdaderos.

$$Sensibilidad = \frac{TP}{TP + FN} \quad (5.35)$$

- **Especificidad**: capacidad de detectar negativos verdaderos.

$$Especificidad = \frac{TN}{TN + FP} \quad (5.36)$$

- **Precisión:** capacidad de realizar clasificaciones correctas.

$$\text{Precisión} = \frac{TP + TN}{TP + FP + TN + FN} \quad (5.37)$$

De este modo un valor alto de sensibilidad (idealmente un 100 %) significa que será capaz de detectar todos los pacientes positivos, mientras que un valor alto de especificidad (idealmente 100 %) dará lugar a un clasificador que es capaz de detectar todos los pacientes negativos. Sin embargo, estos valores por separado no ofrecen información acerca de la fiabilidad del clasificador. Así, si un clasificador tuviese mucha sensibilidad y muy poca especificidad, tomará como positivo la mayor parte de las clases, dando lugar a una clasificación al azar. Lo ideal es que tanto la sensibilidad como la especificidad toman valores muy altos, dando lugar a clasificadores muy precisos y fiables.

### Espacio ROC

El espacio ROC es una representación gráfica de la sensibilidad frente a 1-especificidad, o lo que es lo mismo, la tasa de positivos verdaderos representada en función de la tasa de positivos falsos, y nos da por tanto una idea del rendimiento del clasificador.

Observando la figura 5.3 se comprueba que la clasificación perfecta corresponde con la esquina superior izquierda, que daría lugar a una precisión del 100 % (ningún falso positivo y ningún falso negativo), que se corresponde con el punto D de la gráfica 5.3. Por el contrario, el resultado de una clasificación totalmente aleatoria, estaría situada sobre la diagonal, también conocida como línea de no discriminación (punto C). Por debajo de esta línea (punto E) se encontraría los resultados de la clasificación peor que el azar.

Lo normal es obtener clasificadores adecuados representados por los puntos A y B. En el primer caso más especificidad que sensibilidad y en el segundo caso más sensibilidad que especificidad.

Usaremos también esta representación gráfica para evaluar los resultados del algoritmo propuesto.

#### 5.3.2. Validación Cruzada

La validación cruzada o es un método muy usado para evaluar el rendimiento de un análisis estadístico y proporciona una medida sobre la generalización de este sobre conjuntos de datos independientes [118, 119].

Los fundamentos de la validación cruzada fueron desarrollados por Seymour Geisser y esta estrechamente relacionado con el error Tipo III, que

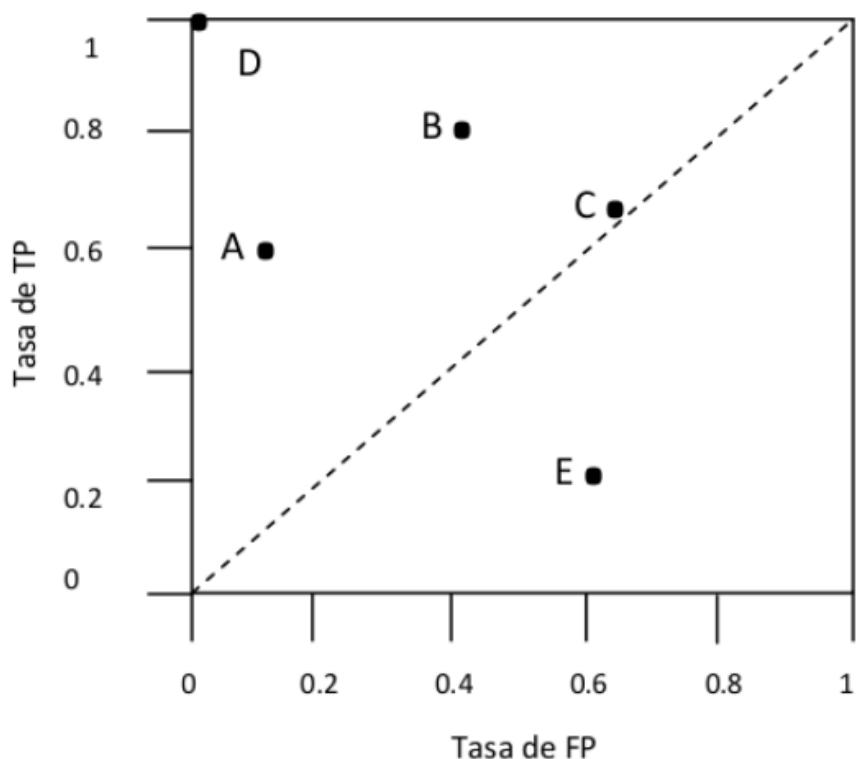


Figura 5.3: Representación del espacio ROC.

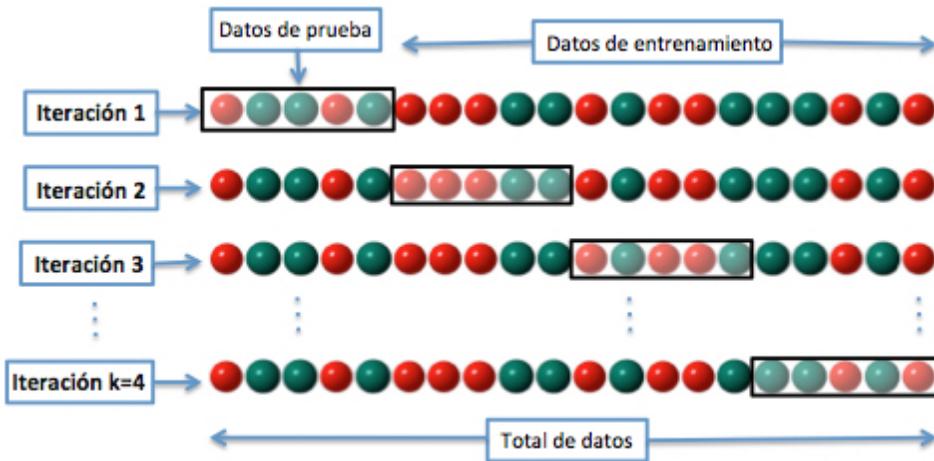


Figura 5.4: Ejemplo de validación cruzada K-Fold con 4 iteraciones.[Fuente: Wikipedia]

se basa en rechazar la hipótesis nula correctamente pero debido a causas erróneas. Además esto tiene una alta relación con el problema que fundamenta este proyecto, cuando se realizan análisis estadísticos con un número de muestras reducido.

El método se basa en dividir el conjunto de datos en subgrupos complementarios, de manera que se realiza el análisis en un conjunto (set entrenamiento) y se valida en otro (set de test). Esto se lleva a cabo varias veces con el objetivo de reducir la variabilidad de los resultados, promediando los resultados de las diferentes iteraciones

A pesar de que existen diferentes métodos de validación cruzada, solo abordaremos la validación Cruzada K-pliegues (o K-Fold), por ser el método usado para la validación de nuestra base de datos simulada.

### Validación Cruzada K-fold

Este método de validación cruzada divide el conjunto original en K subconjuntos, de manera que una de ellos se usa para test y el resto para entrenamiento. Esto se lleva a cabo K veces con cada uno de los subconjuntos definidos. Finalmente se promedian los resultados y devuelven una estimación.

En la figura 5.4 se puede observar un ejemplo de validación cruzada de 4 iteraciones. Nosotros usaremos 10 iteraciones.

## 5.4. El problema del pequeño tamaño muestral

En la sección de clasificación describíamos las bases para la construcción de un clasificador y veíamos que una de las cuestiones de vital importancia era su capacidad de generalización, entendida esta como la habilidad de hacer predicciones precisas sobre nuevos conjuntos de datos independientes, que es el objetivo final.

La capacidad de generalización del clasificador no solo depende del clasificador elegido, sino en mayor medida de la construcción del vector de características. Los datos originales se pueden organizar de diferentes formas para tal fin, pero primero es necesario abordar la relación entre el tamaño del conjunto de muestras de entrenamiento  $n$  y la dimensión del espacio de características  $m$ .

La Teoría del Aprendizaje Estadístico establece la habilidad de un clasificador para poder agrupar en dos clases diferentes y de forma efectiva,  $n$  puntos pertenecientes a un espacio de gran dimensión. De esta manera el número de agrupaciones que se pueden construir por hiperplanos de dimensión  $m - 1$  con el fin de separar los  $n$  puntos en dos clases es [120]:

$$O(n, m) = 2 \sum_{i=1}^m \binom{n-1}{i} \quad (5.38)$$

donde

$$\binom{n-1}{i} = \frac{(n-1)!}{(n-1-i)!i!} \quad (5.39)$$

De manera que la probabilidad de agrupar estos  $n$  puntos en el espacio mencionado en 2 clases linealmente separables viene dada por:

$$P_n^m = \frac{O(n, m)}{2^n} = \begin{cases} \frac{1}{2^{n-1}} \sum_{i=0}^m \binom{n-1}{i} & n > m + 1 \\ 1 & n \leq m + 1 \end{cases} \quad (5.40)$$

En la Figura 5.5 podemos ver la probabilidad  $P_n^m$  como función de  $m/n$ . Así para espacios de pequeña dimensionalidad, la probabilidad de separación de clases es casi nula. Aumentando ahora la dimensión del espacio de características, la probabilidad crece acercándose a 1.

A la vista de la Figura 5.5 y de la ecuación 5.40 es natural pensar que el aumento de la dimensión del espacio de características será siempre beneficioso para discriminar entre dos clases. Sin embargo, ocurre lo contrario, dando lugar a lo que introducimos en el primer capítulo como *la maldición de la dimensionalidad*, que no es más que una reducción de la capacidad generalizadora del clasificador cuando el número de características es muy grande y se tiene una base muestral pequeña. En neuroimagen normalmente

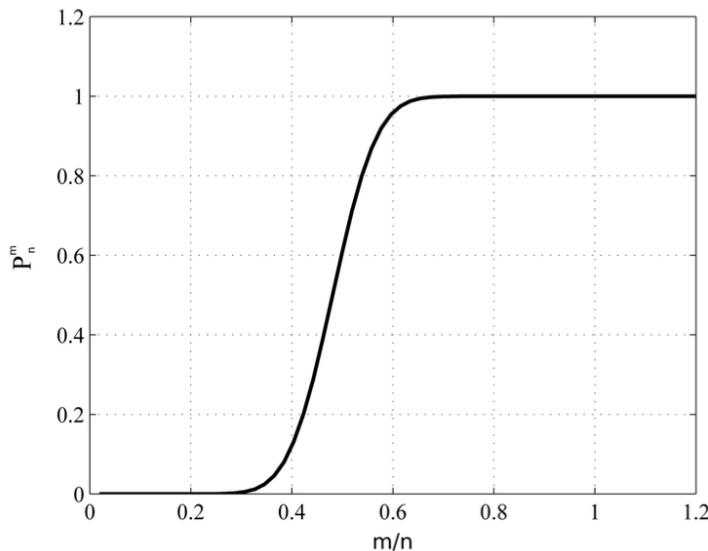


Figura 5.5: Probabilidad de agrupar  $n$  puntos en 2 clases linealmente separables cuando la dimensión del espacio de características es  $m$ .

se disponen de bases de datos de menos de 1000 sujetos (del orden de 100) en comparación con las imágenes funcionales que una vez procesadas contienen  $> 100.000 (\sim 5 \cdot 10^5)$  voxels.

Es por ello que, aunque se realice una buena clasificación, será baja la fiabilidad del clasificador a la hora de estimar los parámetros del clasificador, porque se tienen que estimar con pocos vectores de entrenamiento. Bajo estas condiciones se da una pobre capacidad de generalización.

Debido a este fenómeno, decíamos que la extracción de características era por tanto un paso previo fundamental a la construcción del clasificador con el objetivo de reducir la dimensionalidad del espacio de características, con el objetivo de evitar la maldición de la dimensionalidad. Aunque este es el principal motivo, existen motivaciones adicionales como:

- la reducción del coste computacional
- eliminación de correlación entre características
- selección de características más relevantes

Es por ello que nuestro objetivo está basado en obtener mejores sistemas de clasificación, aumentando el número de muestras en lugar de reducir la dimensionalidad del espacio de características, con el fin de obtener clasificadores con alta capacidad de generalización.



## **Parte III**

# **Simulación de Imágenes Funcionales Cerebrales**



# Capítulo 6

## Desarrollo Experimental

### 6.1. Introducción

Con el objetivo de paliar uno de los problemas fundamentales en neuroimagen, el problema del pequeño tamaño muestral 5.4, en este proyecto proponemos un algoritmo capaz de simular una base de datos de imágenes funcionales cerebrales con características similares a la base de datos original. En esta última parte del trabajo abordamos los detalles en la implementación de dicho algoritmo y simulamos un conjunto de imágenes funcionales FDG-PET de tipo AD, MCI y NOR.

En primer lugar, extraeremos las características de la base de datos original mediante el Análisis de Componentes Principales (ver Capítulo 3), lo que nos permitirá proyectar el conjunto de imágenes originales sobre el espacio de eigenbrains, de modo que cada imagen se podrá considerar como un conjunto de coordenadas en dicho espacio, llamadas Componentes Principales. Así, mediante KDE (ver 4.4) modelaremos la distribución estadística de las Componentes Principales, con el fin generar nuevas coordenadas en el espacio de eigenbrains, simulando de esta manera nuevas imágenes cerebrales funcionales con características similares a la base de datos original.

Una vez obtenida la base de datos simulada, estudiaremos mediante el uso Máquinas de Soporte Vectorial (ver 5.2), la capacidad discriminatoria de la base de datos obtenida frente a la original, en pos de evaluar la capacidad de generación de nuestro algoritmo.

Finalmente, discutiremos los resultados obtenidos, arrojando algunas conclusiones y abriendo una nueva línea de trabajo futuro.

## 6.2. Base de datos utilizada

Como veíamos en el capítulo 2, la base de datos usada para el desarrollo y testeo del algoritmo presentado en este trabajo, está compuesta por 403 imágenes  $^{18}F$ -FDG PET, las cuales han sido obtenidas de la base de datos ADNI (Alzheimer's Disease Neuroimaging Initiative).

La adquisición de estas imágenes se produjo tras administrar a los pacientes  $5.0 \pm 0.5$  mCi de  $^{18}F$ -FDG, vía intravenosa. Esta molécula es incorporada principalmente por aquellas células con elevadas tasas de consumo de glucosa, como el cerebro, permitiendo observar de forma tridimensional el metabolismo de glucosa en él.

Estas imágenes han sido etiquetadas por ADNI en tres clases, siguiendo los criterios mencionados en 2.1.3:

- Controles NC (Normal Control)
- Controles MCI (Mild Cognitive Impairment)
- Pacientes AD (Alzheimer Disease)

De esta manera, la base de datos estará formada por 101 imágenes de pacientes NC, 207 imágenes de pacientes MCI y 95 imágenes de pacientes AD.

Además, con el objetivo de poder separar las clases, etiquetamos cada una de ellas de la siguiente manera:

- Clase NC = '0'
- Clase MCI = '1'
- Clase AD = '2'

Por otro lado, dado que el algoritmo en sí consiste en la generación de una base de datos a partir de una existente, agrupamos las clases en 3 grupos, de manera que tendremos 3 subconjuntos sobre los que aplicaremos la extracción de características, modelado de componentes, clasificación y análisis:

- Grupo 1: NC+AD ( $n_1 = 196$ )
- Grupo 2: NC+MCI ( $n_2 = 308$ )
- Grupo 3: MCI+AD ( $n_3 = 302$ )

donde  $n_i$  representa el número de imágenes asociado a cada grupo.

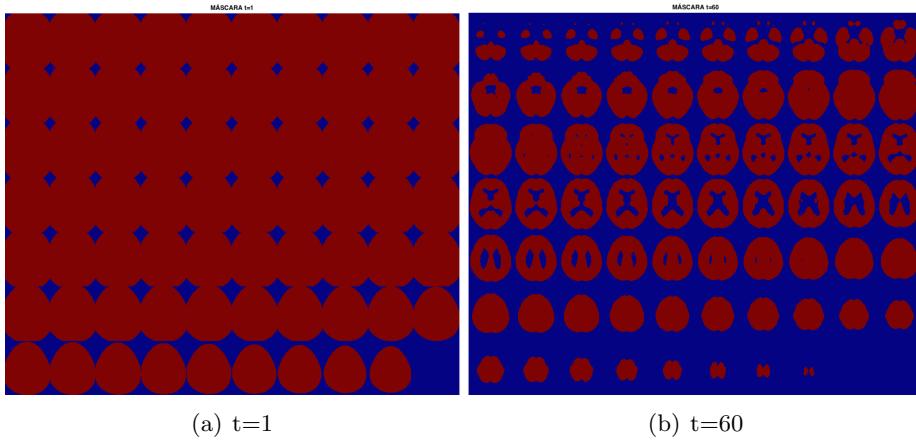


Figura 6.1: Dos ejemplos de máscara construidas para un umbral: a)  $t=1$  y b)  $t=60$ .

### 6.2.1. Normalización en intensidad

Tanto la base de datos original como la simulada son normalizadas en intensidad, para ajustar el rango de intensidad y mejorar el rendimiento del algoritmo. Usaremos una estrategia de normalización al máximo, que ha sido aplicada en otros trabajos obteniendo buenos resultados [121, 122, 123]. Para ello, se ha normalizado cada imagen al valor máximo de intensidad, dado por la media del 3 % de los voxels de mayor intensidad. Con ello evitamos incluir en el cálculo, picos de intensidad no relevantes o esporádicos.

## 6.3. Extracción de características

Con el fin de modelar la base de datos original, necesitamos extraer características comunes del conjunto de  $n$  imágenes que modelen el comportamiento del sistema. Primero, aplicamos una máscara al conjunto de imágenes originales para eliminar todos aquellos voxels que no formen parte del cerebro, para después aplicar PCA al conjunto resultante proyectando las imágenes en el espacio de eigenrains.

### 6.3.1. Selección de voxels basado en umbral

Para conseguir mejores resultados y modelar las imágenes con la máxima y mejor información posible, construimos una máscara binaria que extraiga de las imágenes aquellos voxels cuyos intensidad sobrepase un límite prefijado. Una razón más que evidente para usar este método, a parte de vectorizar

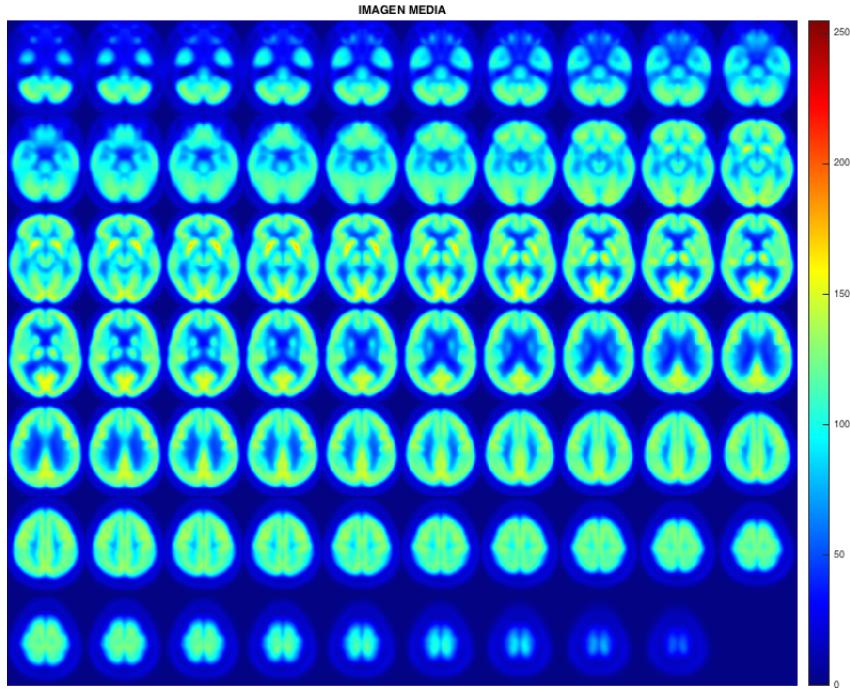


Figura 6.2: Imagen formada por la media de todas las imágenes de tipo NC,AD y MCI.

lás imágenes en 3D, es que, fijando un umbral de selección adecuadamente, nos permite seleccionar aquellos vóxeles de la imagen 3D que pertenezcan al cerebro, descartando todas aquellas regiones que quedan fuera del cerebro cuya intensidad es muy baja y no contienen información útil para la caracterización y modelado del cerebro.

Considerando que tenemos un conjunto de  $n$  imágenes cerebrales  $\Gamma_1, \Gamma_2, \dots, \Gamma_n$ , definimos la máscara como la media de todas las imágenes, que viene dada por:

$$\bar{\Gamma} = \frac{1}{n} \sum_{i=1}^n \Gamma_i \quad (6.1)$$

donde  $\Gamma_i \in \mathbb{R}^m$  es la imagen  $i$  muestral. En la Figura 6.2 podemos observar la imagen media  $\bar{\Gamma}$  de todas las imágenes tipo NC, MCI y AD. Definimos por tanto las máscara  $\mathbf{E}$  binaria a partir de los valores  $\epsilon_j$ :

$$\epsilon_j = \begin{cases} 1 & \text{si } (\mu_j > t) \\ 0 & \text{si } (\mu_j \leq t) \end{cases} \quad j = 1, \dots, m \quad (6.2)$$

donde  $t$  es el umbral de intensidad fijado. Con estos valores se puede construir

la máscara  $\mathbf{E}$  como:

$$\mathbf{E} = \begin{pmatrix} \epsilon_1 & 0 & \dots & 0 \\ 0 & \epsilon_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \epsilon_n \end{pmatrix} \quad (6.3)$$

De esta manera, aplicando la máscara  $\mathbf{E}$  sobre una imagen  $\mathbf{\Gamma}_i$ :

$$\hat{\mathbf{\Gamma}}_i^T = \mathbf{\Gamma}_i^T \mathbf{E} \quad (6.4)$$

dará lugar a un vector columna  $\hat{\mathbf{\Gamma}}_i$ , formado solo por aquellos voxels cuya intensidad media supere el valor  $t$ .

Así, para un umbral bajo conseguimos una máscara que solo seleccionará la parte de la imagen correspondiente al cerebro, y al aumentar  $t$ , se descartarán progresivamente todas las zonas en las cuales la intensidad promedio no sea relativamente baja. En la figura 6.1 podemos observar un ejemplo de ambos casos.

Se ha comprobado que las regiones de baja intensidad, tanto en los pacientes afectados por AD como en los pacientes de control, no son regiones de interés para el diagnóstico de la enfermedad de Alzheimer [124], debido a que en las imágenes AD presentan unos niveles de intensidad menores que los normales, por lo que su valor promedio será intermedio pero no bajo.

Sin embargo, dado que nuestro objetivo primero es caracterizar la base de datos original, un valor muy alto de  $t$  descartaría regiones del cerebro a la hora de extraer las características generales, que aún no siendo de interés para el diagnóstico, si lo son para el modelado de la base de datos original. En la figura 6.1 podemos ver muestra en a) la máscara aplicada en este trabajo  $t=1$ , mientras que en b) se puede observar como un valor umbral mayor  $t=60$ , selecciona regiones más concretas del cerebro que son usadas principalmente para el diagnóstico.

Es por ello que aplicamos una máscara muy permisiva, con valor de  $t$  bajo ( $t = 1$ ), que se limita a seleccionar aquellos voxels de la imagen tomográfica que pertenezca al cerebro, incluso una pequeña región fuera de él como se aprecia en la figura 6.3.

### 6.3.2. PCA

Con el objetivo de caracterizar la base de datos original, haremos uso de PCA (ver Capítulo 3) para extraer la características comunes. Vimos que una de las principales ventajas de este método, más allá de la reducción de dimensionalidad del espacio de características, residía en que la proyección de las imágenes originales en el espacio de eigenbrains, proporcionaba un nuevo conjunto de variables (que llamábamos Componentes Principales)

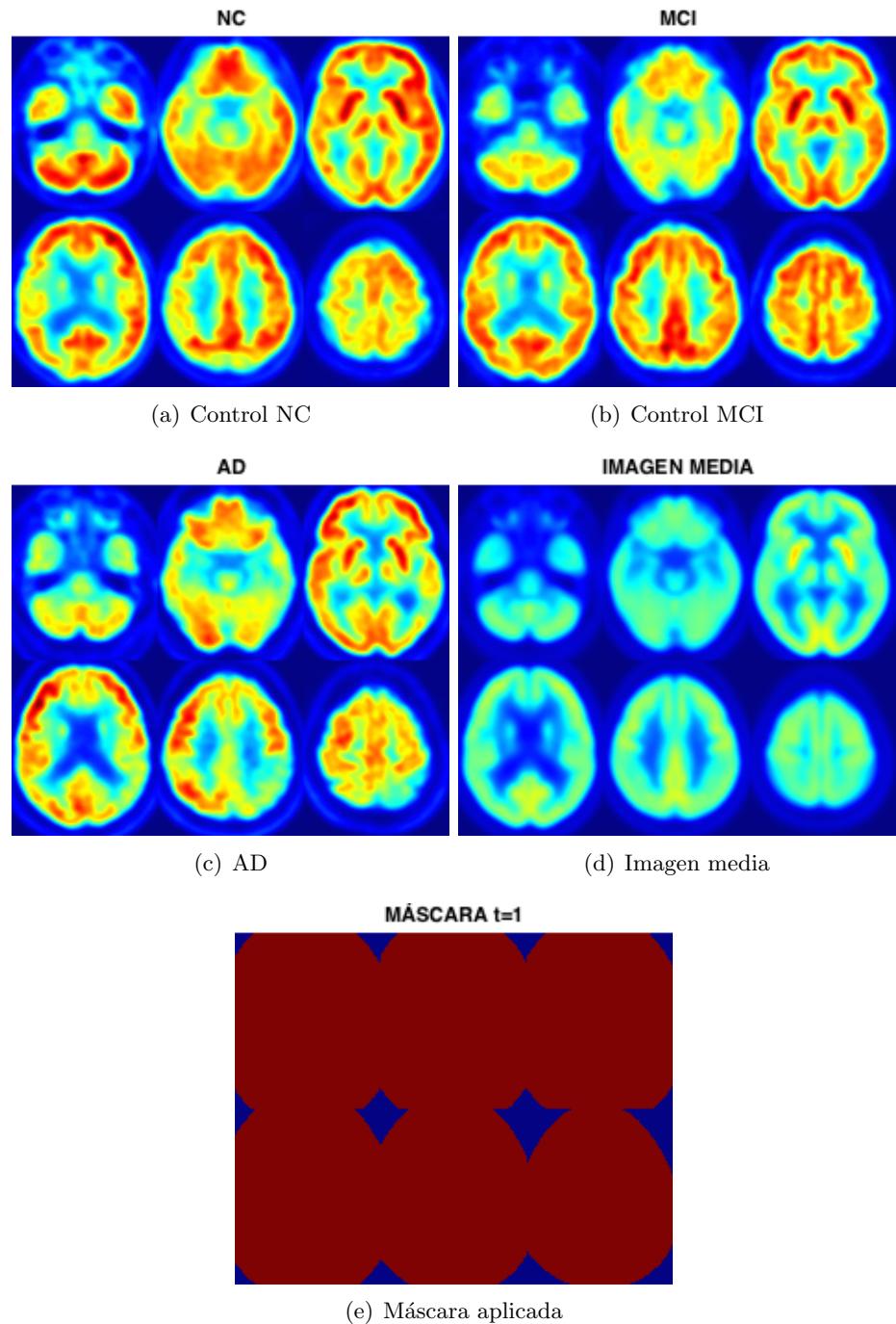


Figura 6.3: Algunas secciones transversales representativas de imágenes originales a) NC, b) MCI y c) AD, d) la media de dichas imágenes y e) la máscara aplicada formada a partir de la imagen media.

agrupando los datos con mayor contribución a la varianza (véase Figura 3.1).

Denotamos nuestras base de datos como tres conjuntos de  $n_i \times m$   $i = 1, 2, 3$  imágenes cerebrales, donde  $n_i$  es el tamaño de cada grupo y  $m$  es el número de características. Puesto que disponemos de 403 imágenes de tamaño  $79 \times 95 \times 69 \sim 5 \times 10^5$  vóxeles, tras aplicar la máscara se traduce en una matriz de  $403 \times 469615$ . Siguiendo la notación previa, denotamos el conjunto de imágenes como  $n_i$  vectores  $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \dots, \mathbf{\Gamma}_n$  de dimensión  $m$  cada uno.

PCA actuará por tanto sobre cada uno de los conjuntos especificados en 6.2:

$$\mathbf{\Phi}_j = \hat{\mathbf{\Gamma}} - \bar{\mathbf{\Gamma}} \quad j = 1, 2, \dots, n_i \quad (6.5)$$

donde  $\bar{\mathbf{\Gamma}}$  es la media muestral de las  $n_i$  imágenes.

Como demostramos en 3.1.1 dado que  $n_i \ll m$ , solo los  $n_i - 1$  Componentes Principales y autovalores serán distintos de cero<sup>1</sup>. De este modo obtenemos: 195 Componentes Principales para el Grupo 1 (NC+AD), 307 para el Grupo 2 (NC+MCI) y 301 para el Grupo 3 (MCI+AD).

Normalmente, en los algoritmos para el diagnóstico asistido por ordenador, se selecciona un número reducido de Componentes Principales con el objetivo de extraer aquellas que realmente infieran capacidad discriminatoria para la clasificación, además de reducir considerablemente el número de características como solución parcial al problema del pequeño tamaño muestral (véase 5.4). Sin embargo, en este trabajo nuestro objetivo es simular imágenes cerebrales con las mismas características que la base de datos original, por lo que haremos uso de todas las Componentes Principales, en este caso las  $n_i - 1$  primeras.

En resumen, de la ecuación 3.28 se deduce que tras aplicar PCA, obtendremos por un lado la matriz de coordenadas en el espacio de eigenbrains,  $\mathbf{x}_j$  y la matriz formada por los autovectores  $\mathbf{e}_p$  o eigenbrains.

En la Figura 3.1.1 representamos algunas secciones relevantes correspondientes a los 4 primeros eigenbrains, obtenidos tras aplicar PCA al Grupo 1. Se observa que estos no solo modelan la estructura anatómica del cerebro, sino también patrones característicos de AD [48, 57]. En particular podemos observar como el primero de ellos, modela las diferencias entre el giro cingulado y el precúneo, mientras las diferencias en el metabolismo de glucosa entre el lóbulo principal y occipital son modeladas por el tercer eigenbrain.

---

<sup>1</sup>En Matlab implementamos PCA mediante la función *pca* con la opción ‘econ’, que acelera el cómputo cuando  $n \ll m$  diagonalizando la matriz  $\hat{\mathbf{C}} = \mathbf{A}^T \mathbf{A}$ , donde  $\mathbf{A} = [\mathbf{\Phi}_1, \dots, \mathbf{\Phi}_n]$ , en lugar de  $\hat{\mathbf{C}} = \mathbf{A} \mathbf{A}^T$

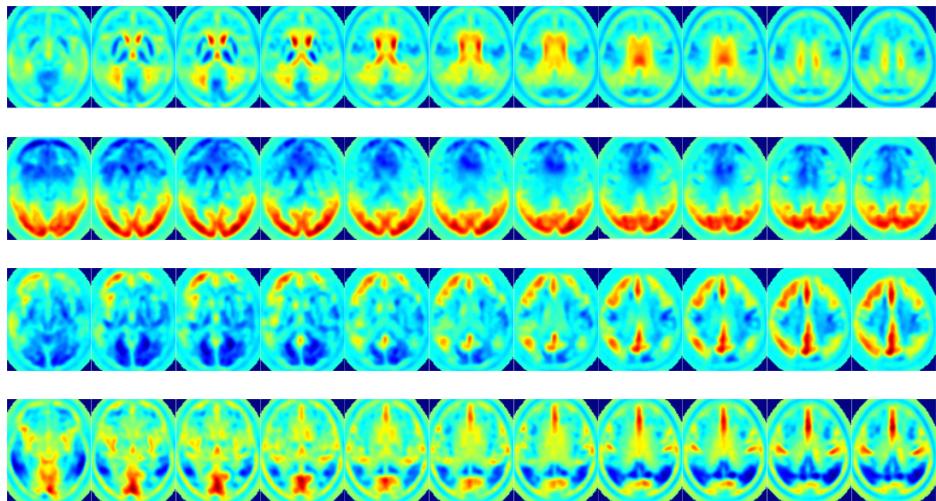


Figura 6.4: Representación de los 4 primeros eigenbrains en orden descendente tras aplicar PCA al Grupo 1 (AD+NC).

## 6.4. Modelado y simulación de Componentes Principales

Con PCA conseguimos trasladar el conjunto de variables originales al espacio de eigenbrains, de modo que cada imagen se puede considerar como un conjunto de coordenadas en dicho espacio, las CPs.

El objetivo ahora, es estimar la distribución de densidad que siguen estas coordenadas, de modo que una vez conocida, podremos generar nuevas coordenadas en el espacio de eigenbrains y así simular las nuevas imágenes funcionales cerebrales (ver Sección 6.5).

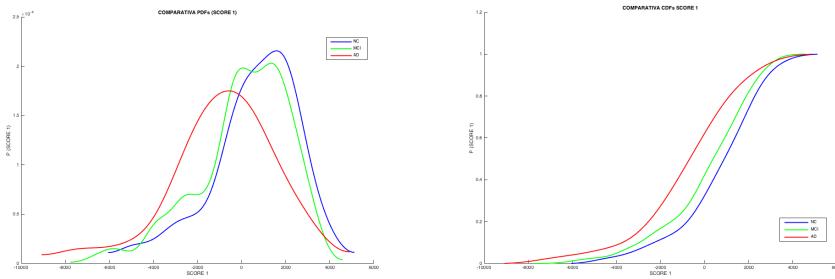
### 6.4.1. KDE vía difusión

Existen numerosos métodos paramétricos para la estimación de la función de densidad que sigue una población muestral. Sin embargo, estamos interesados en simular imágenes funcionales lo más cercanas posibles a la realidad, e intuitivamente se deduce que hacer suposiciones previas sobre la distribución que siguen, estableciendo un modelo paramétrico, nos aleja de este objetivo.

Es por ello, que para la estimación de la función de densidad de las Componentes Principales hemos hecho uso de un modelo no paramétrico, evitando así cualquier suposición a priori sobre la distribución de las CPs. Concretamente, hemos aplicado un estimador tipo núcleo, KDE vía difusión<sup>2</sup>

---

<sup>2</sup>Función *KDE* de Matlab extraída de *Annals of Statistics, Volume 38, Number 5,*



(a) PDF de la primera componente principal.  
(b) CDF de la primera componente principal

Figura 6.5: Distribuciones de la primera componente principal para las 3 clases, estimadas mediante KDE sobre la base de datos completa: NC+AD+MCI.

(ver Sección 4.4).

Este estimador hace uso de un kernel Gaussiano y realiza una estimación automática del ancho de banda,  $h$ . Este parámetro es de vital importancia pues define el compromiso entre el sesgo del estimador y su varianza. La ventaja principal de KDE vía difusión frente a otros métodos, es que no hace ninguna suposición previa sobre la distribución de las muestras y estima automáticamente el ancho de banda,  $h$ , a diferencia de otros métodos de estimación tipo núcleo.

Puesto que hemos etiquetado y agrupado convenientemente cada una de las clases, aplicaremos KDE a cada una de ellas, obteniendo así la distribución de cada componente para cada clase (NC,MCI y AD). A modo de ejemplo, en la figura 6.5 mostramos la estimación de la primera Componente Principal para cada uno de los 3 clases, tras haber aplicado PCA a la base de datos original y no a cada uno de los grupos por separado. Podemos observar que las tres clases siguen una distribución aproximadamente Gaussiana, en especial la distribución de la clase AD.

En la Figura 6.6 mostramos las PDFs estimadas sobre cada uno de los grupos. Puesto que el objetivo es simular imágenes funcionales desde una base de datos dada, para generar imágenes AD y NC por ejemplo, usamos el Grupo 1 (NC+AD), pues nos permitirá una mayor caracterización en el espacio de eigenbrains que el espacio formado desde una base de datos NC+AD+MCI.

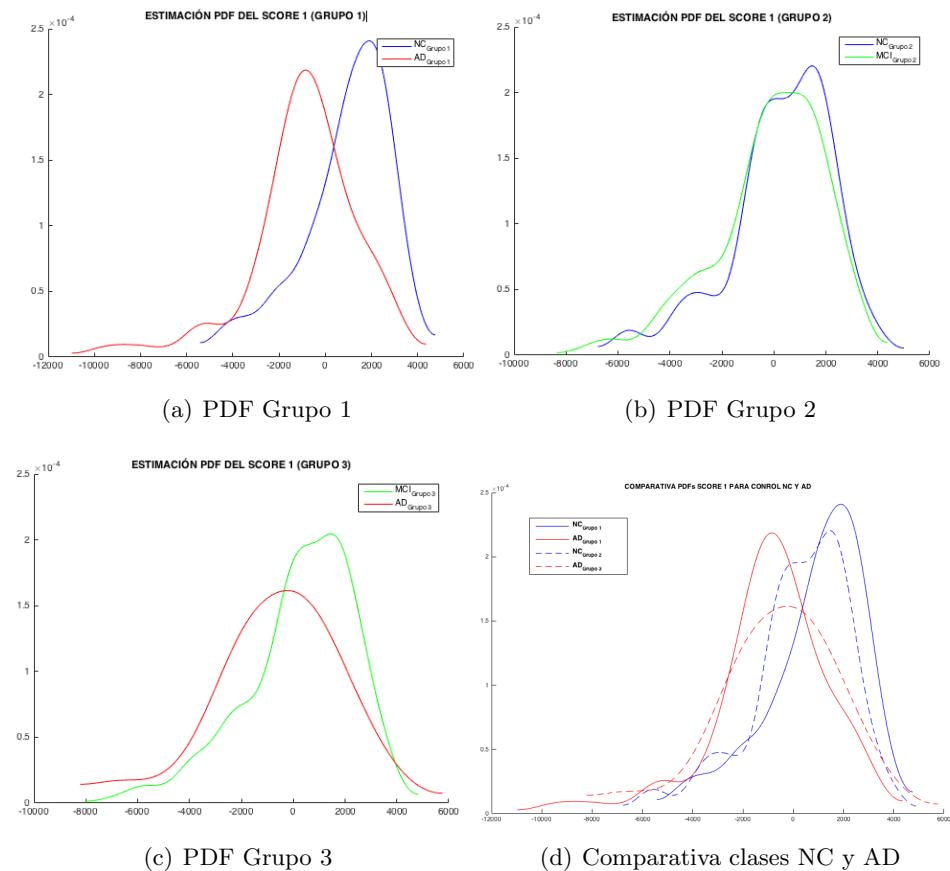


Figura 6.6: Distribuciones estimadas sobre la base de datos original de la primera CP para cada grupo a) Grupo 1, b) Grupo 2, c) Grupo 3 y comparativa de las distribuciones NC y AD.

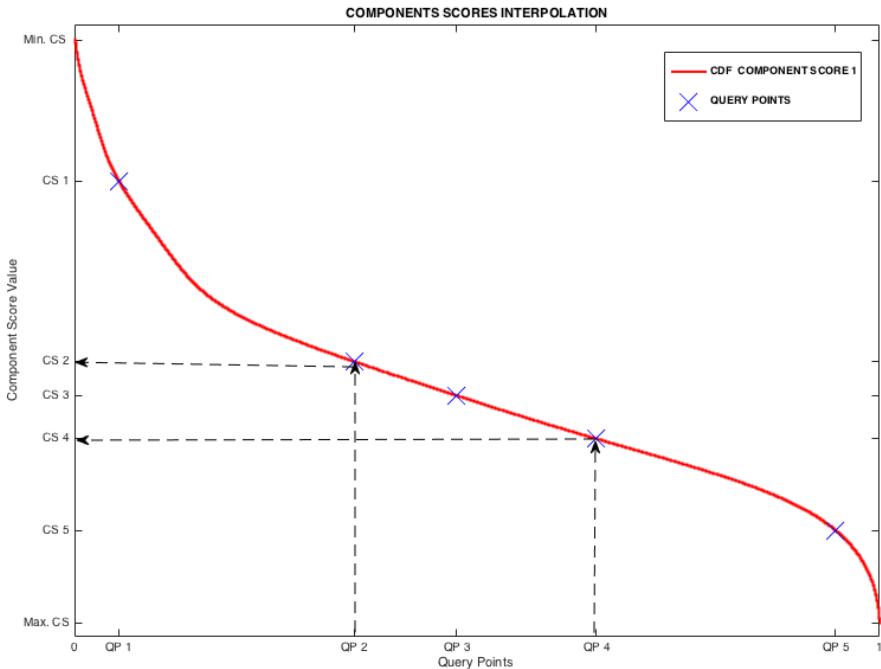


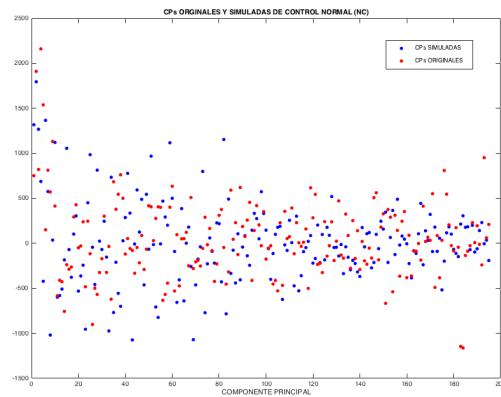
Figura 6.7: Generación de coordenadas en el espacio de eigenbrains.

#### 6.4.2. Interpolación de coordenadas

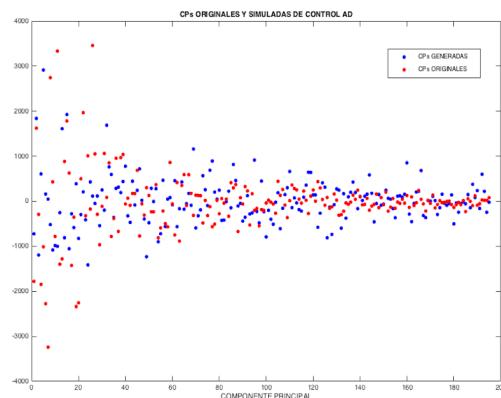
Una vez estimada la función de densidad que siguen cada una de las  $n - 1$  Componentes Principales, podemos generar un nuevo conjunto de coordenadas en el espacio de eigenbrains con la misma distribución que las coordenadas originales.

Para ello usaremos un generador de números aleatorios, que proporciona valores aleatorios uniformemente distribuidos en el intervalo  $[0, 1]$ . De esta manera, usando estos puntos de estimación como probabilidades de la función de densidad perteneciente a cada uno de los scores, podemos interpolar las nuevas coordenadas desde la función empírica de distribución acumulada al espacio de eigenbrains.

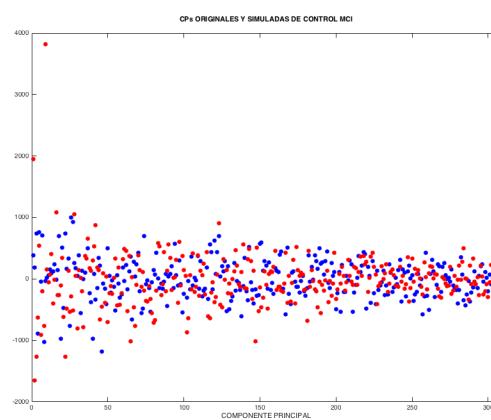
Podemos ver una representación gráfica de este procedimiento en la figura 6.7. En ella aparece representada la CDF de la primera Componente Principal (notar que en el eje X representamos la probabilidad, mientras que en el eje Y representamos el rango de valores que puede tomar la CP). Así, generando aleatoriamente 5 puntos a estimar (QP1, QP2, QP3, QP4 Y QP5), podemos interpolarlos mediante la cdf, obteniendo así 5 valores para dicha componente. Si repetimos este proceso sobre las  $n_i - 1$  componentes principales, obtendríamos las coordenadas de 5 imágenes en el espacio de eigenbrains.



(a) Clase NC



(b) Clase AD



(c) Clase MCI

Figura 6.8: Componentes Principales originales y generadas mediante interpolación para clase a) NC, b) AD y c) MCI. En rojo las CPs originales y en azul las CPs generadas.

Al igual que en caso anterior, generamos uno o varios conjuntos de coordenadas (tantos como imágenes queramos simular) para cada clase (NC, MCI, AD). Por ejemplo, en la figura 6.8 podemos observar en rojo el valor de las  $n_1 - 1$  Componentes Principales originales, mientras que en azul se muestran los  $n_i - 1$  valores generados, que darán lugar a una nueva imagen funcional simulada, como veremos en el siguiente apartado.

## 6.5. Reconstrucción de imágenes funcionales

En el apartado anterior generábamos nuevas coordenadas en el espacio de eigenbrains, que seguían la misma distribución que las originales. De este modo, para visualizar las nuevas imágenes funcionales simuladas, solo tendremos que proyectar el nuevo conjunto de coordenadas en espacio original.

En 3.28 veíamos que las coordenadas de las imágenes en el espacio de eigenbrains eran una combinación lineal de los vectores de la nueva base ortogonal y las imágenes originales. Así, para reconstruir el conjunto de imágenes originales solo es necesario multiplicar la matriz de  $p$  eigenbrains  $\tilde{\mathbf{E}}$  de dimensión  $m \times p$ , por la matriz de componentes principales traspuesta  $\mathbf{X}^T$  de dimensión  $p \times n$ , donde  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$  y  $\mathbf{x}_n \in \mathbb{R}^p$ :

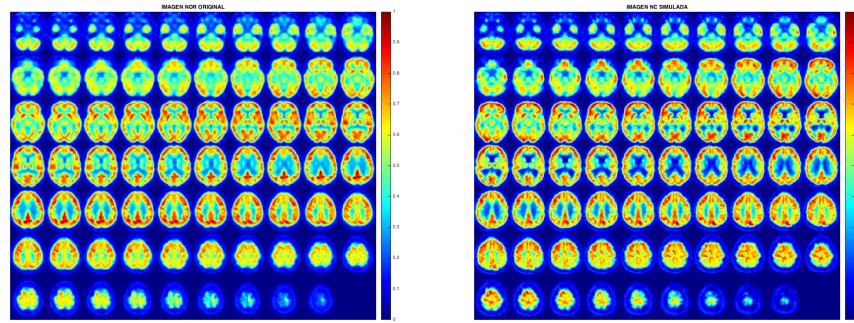
$$\hat{\mathbf{A}} = \tilde{\mathbf{E}} \mathbf{X}^T \quad (6.6)$$

Así, sin más que sustituir  $\mathbf{X}$  por el nuevo conjunto de coordenadas generadas en el espacio de eigenbrains,  $\mathbf{X}'$ , obtendremos las imágenes funcionales simuladas:

$$\mathbf{R} = \tilde{\mathbf{E}} (\mathbf{X}')^T \quad (6.7)$$

donde  $\mathbf{R}$  es ahora el conjunto de imágenes reconstruidas.

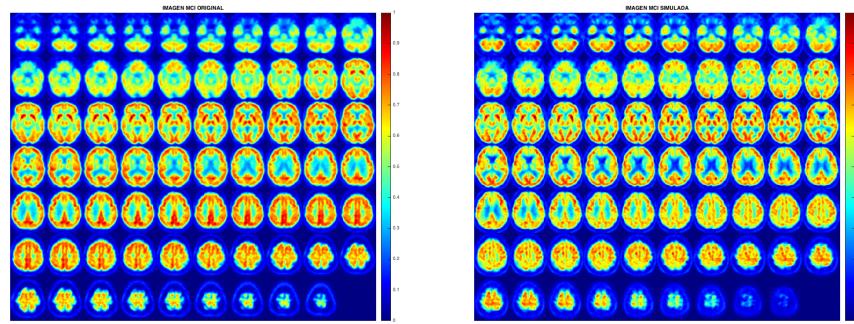
En las figuras 6.9, 6.10 y 6.11, podemos ver la comparación entre la imagen original y la imagen simulada mediante el algoritmo propuesto.



(a) Imagen clase NC original.

(b) Imagen clase NC simulada.

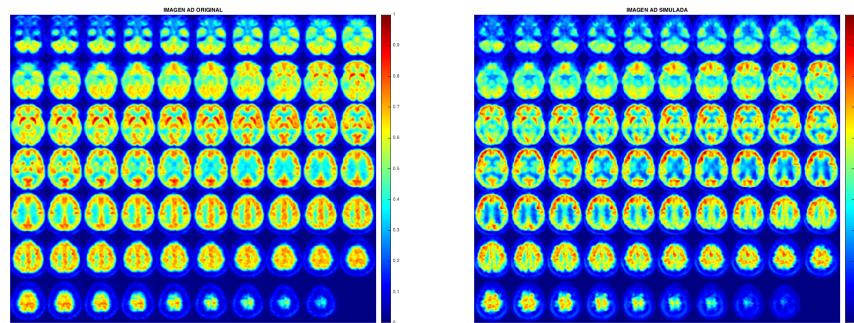
Figura 6.9: Imagen de Control Normal a) original y b) simulada.



(a) Imagen clase MCI original.

(b) Imagen clase MCI simulada.

Figura 6.10: Imagen de clase MCI a) original y b) simulada.



(a) Imagen clase AD original.

(b) Imagen clase AD simulada.

Figura 6.11: Imagen de clase AD a) original y b) simulada.

## Capítulo 7

# Validación y Resultados

En el Capítulo 6 detallábamos el algoritmo para la simulación de imágenes cerebrales funcionales, a partir de la base de datos existente. Llegados a este punto, necesitamos validar la base de datos generada para comprobar si las nuevas imágenes funcionales, aunque simuladas, respeten las características y patrones anatómicos de las originales, y sean a su vez independientes de estas. Usaremos la clasificación para este propósito, mediante el uso de Máquinas de Soporte Vectorial (SVM) con kernel lineal.

Como indicábamos, para validar la base de datos simulada realizamos tres experimentos diferentes. El análisis de clasificación se ha efectuado usando los véxeles para construir los vectores de características, ( $v = I_1, \dots, I_m$ ). Se ha demostrado en [48] que esta aproximación es igual de exacta que el diagnóstico visual usando como apoyo SPM<sup>1</sup>.

### 7.1. Experimento 1

En este primer experimento realizamos un análisis mediante clasificación para comprobar que el conjunto de imágenes simuladas coincide con el original. El objetivo final es obtener los valores de precisión, sensibilidad y especificidad, para comprobar que el conjunto de imágenes originales y simuladas tienen la misma capacidad discriminatoria, a la vez que se reduce la varianza de los estimadores cuando se incrementa el tamaño muestral.

Tras haber diseñado un esquema de validación cruzada de 10 iteraciones, realizamos el experimento para los 3 grupos previamente definidos:

---

<sup>1</sup>SPM (Statistical Parametric Mapping) es una herramienta para el análisis de imágenes funcionales, ampliamente usada en investigación médica, así como ayuda al diagnóstico de enfermedades neurodegenerativas, que permite realizar entre otras, una gran variedad de estudios estadísticos.

- Grupo 1: AD vs NC
- Grupo 2: MCI vs NC
- Grupo 3: AD vs MCI

Tabla 7.1: Resultados Experimento 1.

Escenario	Precisión ( $\pm SD$ )	Sensibilidad ( $\pm SD$ )	Especificidad ( $\pm SD$ ) (%)	
Original	$87.8 \pm 7.1$	$87.2 \pm 13.6$	$88.1 \pm 9.2$	Grupo 1
Simulado (N=200)	$77.5 \pm 8.2$	$78.0 \pm 11.4$	$77.0 \pm 10.6$	
Simulado (N=500)	$81.4 \pm 5.5$	$81.2 \pm 9.2$	$83.5 \pm 2.8$	
Simulado (N=1000)	$81.5 \pm 3.4$	$80.0 \pm 5.8$	$83.0 \pm 6.1$	
Simulado (N=1500)	$81.9 \pm 1.4$	$80.4 \pm 3.6$	$81.6 \pm 10.2$	
Simulado (N=2000)	$81.4 \pm 2.3$	$80.7 \pm 3.5$	$82.0 \pm 3.1$	
Original	$70.1 \pm 6.9$	$88.0 \pm 7.3$	$33.9 \pm 17.3$	Grupo 2
Simulado (N=200)	$69.9 \pm 8.4$	$68.0 \pm 1.69$	$70.0 \pm 11.5$	
Simulado (N=500)	$72.0 \pm 6.0$	$74.0 \pm 7.4$	$70.0 \pm 8.7$	
Simulado (N=1000)	$80.3 \pm 2.9$	$81.2 \pm 4.2$	$79.4 \pm 3.9$	
Simulado (N=1500)	$80.7 \pm 2.5$	$80.9 \pm 5.6$	$80.5 \pm 3.8$	
Simulado (N=2000)	$80.6 \pm 2.8$	$80.5 \pm 4.4$	$80.7 \pm 3.2$	
Original	$70.9 \pm 7.4$	$31.9 \pm 16.3$	$88.9 \pm 8.0$	Grupo 3
Simulado (N=302)	$75.7 \pm 5.1$	$36.6 \pm 16.1$	$93.2 \pm 7.3$	
Simulado (N=500)	$77.4 \pm 4.0$	$76.0 \pm 9.6$	$78.8 \pm 5.0$	
Simulado (N=1000)	$78.2 \pm 2.6$	$75.8 \pm 4.8$	$80.6 \pm 6.3$	
Simulado (N=1500)	$80.02 \pm 4.0$	$79.3 \pm 5.1$	$81.1 \pm 6.5$	
Simulado (N=2000)	$81.2 \pm 3.0$	$79.8 \pm 3.6$	$82.6 \pm 4.2$	

En la tabla A.1 se muestran los resultados correspondientes al primer experimento, en el que se han obtenido la estimación de los distintos parámetros (precisión, sensibilidad y especificidad) tras realizar un análisis de clasificación siguiendo la aproximación VAF. En el escenario *Original*, se ha realizado el análisis sobre el conjunto de imágenes original para cada grupo (ver 6.2) y en el escenario *simulado* sobre un conjunto simulado para distintos tamaños.

El esquema de los resultados para este primer experimento, se repite en los 3 grupos sobre los que se ha realizado el análisis. Observando el primer grupo NCvsAD, vemos que los valores de precisión, sensibilidad y especificidad correspondientes a la clasificación sobre el conjunto simulado se mantienen por debajo de los valores correspondientes a la clasificación del conjunto de imágenes originales, sin embargo se muestra que la tendencia al aumentar en número de imágenes simuladas es obtener valores cada vez más próximos a los originales.

En la gráfica 7.1 se observa esta evolución de los valores de precisión correspondiente a la clasificación AD vs NC. La línea discontinua roja indica el valor de precisión obtenido a partir de la bases de datos original, mientras

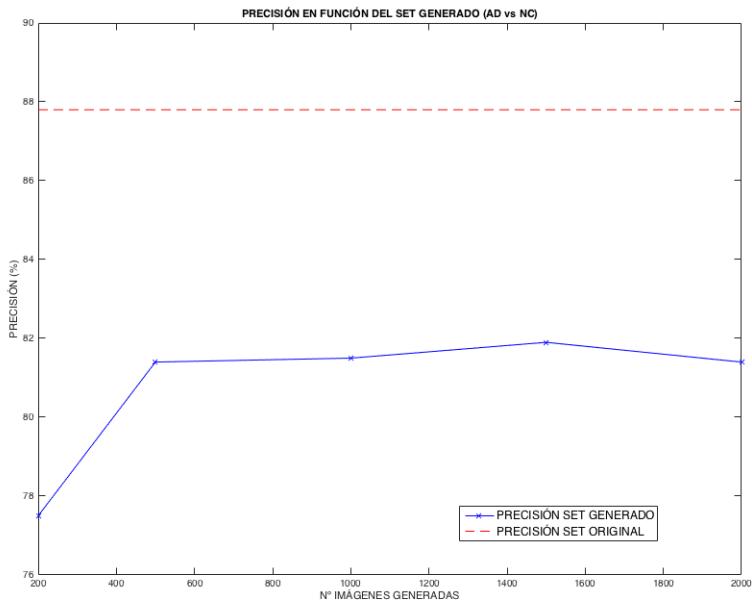


Figura 7.1: Evolución de la precisión en función del tamaño del conjunto generado correspondiente a la clasificación del Grupo 1 (AD vs NC). Experimento 1.

que en azul se muestran los valores correspondientes a los sets de imágenes generados para distintos tamaños. Como hemos indicado estos valores están próximos al 87.8 % y se deduce una aproximación mayor para sets de generación mayores.

Por otro lado, considerando los resultados obtenidos para el Grupo 2 NC vs MCI, se observa que mientras los valores de precisión y sensibilidad tienen las mismas características que el Grupo 1, la especificidad obtenida sobre la clasificación del conjunto original es del 33.9 %, un valor muy bajo, que mejora considerablemente hasta un 79.4 % cuando se aumenta el tamaño de las muestras hasta  $N = 1000$ . Se trata de un resultado coherente, pues la clase MCI al ser un clase intermedia entre las clases NC y AD, que con el tiempo se convertirá en una clase AD o no, presentará muchas características comunes a la clase NC, sobre todo para aquellas imágenes etiquetadas como MCI que presentan un deterioro cerebral leve.

Algo similar ocurre con la evaluación del Grupo 3 MCI vs AD, que presenta una sensibilidad muy baja al evaluar el conjunto original, cuyo valor aumenta llegándose incluso a duplicar cuando se genera un número de imágenes funcionales elevado. Implícitamente se comprueba que efectivamente, obtendremos una clasificación con mayor capacidad predictiva y

generalizable, al trabajar en espacios con mayor número de muestras.

Por lo tanto, con un elevado número de imágenes se mantienen valores aproximados de precisión, sensibilidad y especificidad entre el conjunto original y simulado, mientras que la varianza se reduce sustancialmente. Por ello podemos decir que el conjunto generado mantiene la misma capacidad discriminatoria frente al conjunto original. Por otro lado, notar que el hecho de que disminuya la varianza puede ser debido al elevado número de imágenes generadas sobre el que se realiza en análisis.

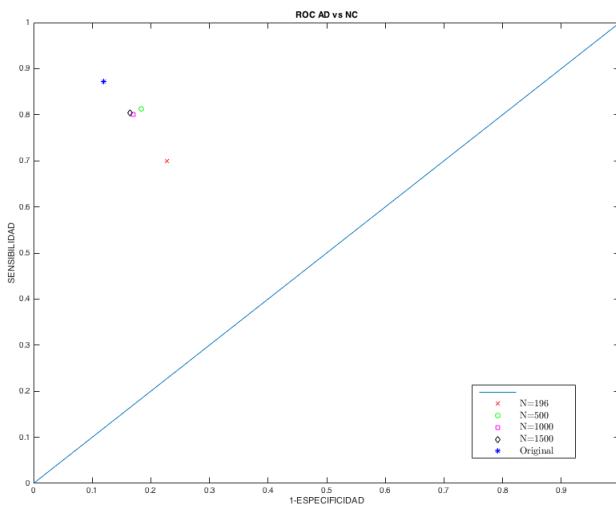


Figura 7.2: Espacio ROC experimento 1: Grupo 1 (AD vs NC).

En las figuras 7.2, 7.3, 7.4, mostramos los resultados del análisis de clasificación en el espacio ROC. A primera vista se observan mejores resultados de clasificación respecto al Grupo 1 (AD vs NC), cuyos puntos están más próximos a la esquina superior izquierda y cuyos valores son más próximo a una clasificación perfecta (ningún NF y ningún PF). Además, se comprueba que al aumentar el tamaño de generación, los puntos están más próximos al punto correspondiente a los resultados de clasificación del conjunto original.

Con respecto al Grupo 2 (MCI vs AD) es importante notar la mejoría que se produce con los sets simulados con respecto del conjunto original, cuando se aumenta considerablemente el tamaño del conjunto de imágenes. Aunque los resultados correspondientes al conjunto original presenta una sensibilidad muy alta, la especificidad es muy baja; en cambio, al aumentar el tamaño del conjunto de generación la sensibilidad se mantiene, mientras que aumenta el valor de la especificidad.

Por otro lado, la figura 7.4 muestra un comportamiento similar, pero

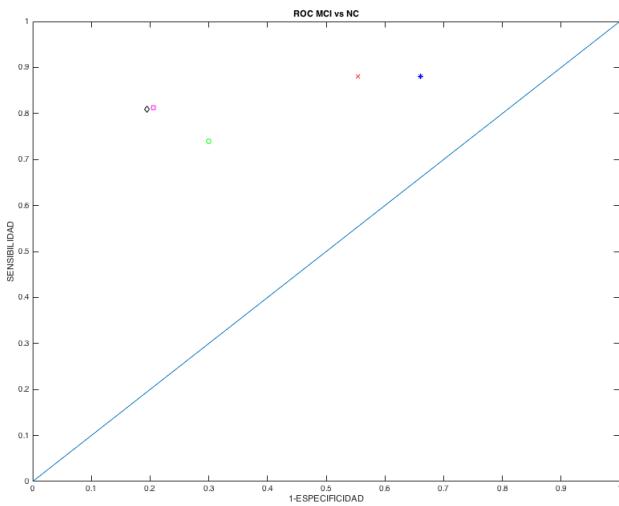


Figura 7.3: Espacio ROC experimento 1: Grupo 2 (MCI vs NC).

ahora es la sensibilidad del conjunto original la que presenta un valor bajo que aumenta considerablemente al aumentar el tamaño del set simulado, mientras que los valores de especificidad se mantienen en comparación.

## 7.2. Experimento 2

Con el objetivo de analizar la capacidad predictiva de las imágenes simuladas sobre imágenes originales, diseñamos un segundo esquema de validación cruzada de 10 iteraciones, de modo que en cada una de las iteraciones usamos una fracción del conjunto de imágenes originales, para generar un conjunto de  $N$  imágenes (pertenecientes a cada grupo) que usamos para entrenar el clasificador lineal. Con él clasificamos el conjunto de imágenes de test originales (distinto del conjunto de generación). Así, podemos comprobar la precisión de los modelos entrenados con los datos simulados, prediciendo los datos originales.

Del mismo modo que en el caso anterior, realizamos el experimento para los 3 grupos mencionados.

Los resultados correspondientes al segundo experimento quedan detallados en la Tabla A.2. Con un objetivo similar al experimento anterior, en este se pretende comparar la capacidad predictiva de las imágenes simuladas, entrenando para ello un clasificador con el conjunto generado y prediciendo nuevas imágenes originales. Así, hemos realizado el experimento para los 3 grupos, generando 4 sets por grupo de diferentes tamaños ( $N=200$ ,  $N=500$ ,

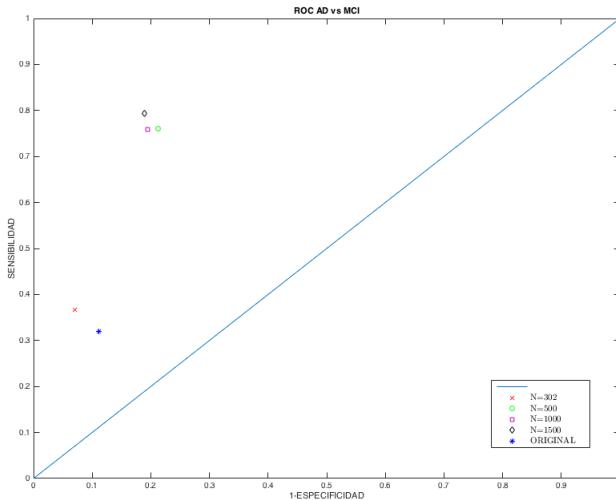


Figura 7.4: Espacio ROC experimento 1: Grupo 3 (AD vs MCI).

$N=1000$  y  $N=2000$ ). Para complementar este análisis, repetimos el experimento creando dos escenarios adicionales en los que incluimos la clase MCI como NC y también MCI como AD, dando lugar a MCI as NC y MCI as AD.

Analizando los resultados de la clasificación observamos por un lado valores altos de precisión, sensibilidad y especificidad del clasificador entrenado con imágenes simuladas. Además, en los tres primeros escenarios, correspondientes a cada uno de los grupos, se observa una tendencia general: tanto los valores de precisión, como especificidad y sensibilidad, aumentan a medida que aumentamos la base de datos de imágenes simuladas con las que entrenamos el clasificador y se aproximan a los valores obtenidos en el primer experimento sobre la base de datos original. Por ejemplo, para el escenario NCvsAD, se obtiene una precisión del 83.2% cuando generamos un set de 2000 imágenes, muy próximo al 87.8% obtenido en el primer experimento al entrenar y clasificar con el conjunto original. Del modo similar ocurre con los escenarios del Grupo 2 y 3. Esto se puede ver reflejado en la Figura 7.5, donde se representa la evolución del valor de precisión en función del tamaño del conjunto generado. La tendencia es claramente ascendente y cada vez más próxima a valor de precisión correspondiente a la clasificación sobre el conjunto original.

Como hemos visto, los dos últimos escenarios se realizan incluyendo la clase MCI como clase NC y otro incluyendo la clase MCI como AD, llevando a cabo la clasificación NC vs AD en ambos casos. Es de esperar peores resultados, pues teniendo en cuenta que estamos usando una clasificación

Tabla 7.2: Resultados Experimento 2.

Escenario	Precisión ( $\pm SD$ )	Sensibilidad ( $\pm SD$ )	Especificidad ( $\pm SD$ ) (%)
AD vs NC (N=200)	79.6 $\pm$ 10.7	71.3 $\pm$ 18.5	87.3 $\pm$ 12.1
AD vs NC (N=500)	80.1 $\pm$ 9.2	74.9 $\pm$ 17.5	85.4 $\pm$ 12.8
AD vs NC (N=1000)	80.6 $\pm$ 11.6	75.9 $\pm$ 15.6	85.1 $\pm$ 12.7
AD vs NC (N=2000)	83.2 $\pm$ 9.2	77.8 $\pm$ 19.8	88.1 $\pm$ 9.2
MCI vs NC (N=200)	59.9 $\pm$ 4.0	65.0 $\pm$ 9.8	49.4 $\pm$ 18.6
MCI vs NC (N=500)	71.8 $\pm$ 8.9	81.0 $\pm$ 9.3	52.4 $\pm$ 11.7
MCI vs NC (N=1000)	69.6 $\pm$ 5.4	75.9 $\pm$ 6.2	56.9 $\pm$ 14.3
MCI vs NC (N=2000)	69.2 $\pm$ 8.3	77.8 $\pm$ 6.1	52.2 $\pm$ 17.2
AD vs MCI (N=200)	66.2 $\pm$ 8.6	52.7 $\pm$ 25.6	72.6 $\pm$ 12.1
AD vs MCI (N=500)	68.5 $\pm$ 4.5	47.1 $\pm$ 14.8	78.3 $\pm$ 9.1
AD vs MCI (N=1000)	70.2 $\pm$ 7.6	51.9 $\pm$ 12.0	78.8 $\pm$ 13.5
AD vs MCI (N=2000)	73.2 $\pm$ 6.5	48.6 $\pm$ 10.7	84.6 $\pm$ 10.0
MCI as NC (Original)	79.7 $\pm$ 4.0	25.7 $\pm$ 18.8	96.4 $\pm$ 4.2
MCI as NC (N=200)	74.9 $\pm$ 7.5	52.6 $\pm$ 14.8	81.8 $\pm$ 9.1
MCI as NC (N=500)	77.2 $\pm$ 4.4	58.1 $\pm$ 18.3	83.1 $\pm$ 6.1
MCI as NC (N=1000)	76.9 $\pm$ 5.8	50.4 $\pm$ 13.0	85.0 $\pm$ 7.5
MCI as NC (N=1500)	78.7 $\pm$ 5.3	52.6 $\pm$ 15.2	86.7 $\pm$ 5.9
MCI as AD (Original)	78.7 $\pm$ 5.2	92.7 $\pm$ 6.3	36.6 $\pm$ 2.0
MCI as AD (N=200)	69.2 $\pm$ 7.3	71.2 $\pm$ 8.5	63.7 $\pm$ 16.4
MCI as AD (N=500)	72.2 $\pm$ 8.8	76.6 $\pm$ 9.6	59.5 $\pm$ 16.0
MCI as AD (N=1000)	75.2 $\pm$ 4.0	81.1 $\pm$ 6.8	57.5 $\pm$ 11.8
MCI as AD (N=1500)	73.9 $\pm$ 6.8	80.8 $\pm$ 6.7	53.5 $\pm$ 22.3

lineal, la clase MCI tendrá muchas características comunes a la clase NC y a la clase AD. Así en el primer caso, MCI as NC, la precisión desciende desde un 70.6 % a un 76.9 %, mientras que en el caso MCI as AD desciende hasta un 75.2 %.

En las Figuras 7.6, 7.7, 7.8 representamos ahora el espacio ROC con los resultados de la clasificación correspondiente al segundo experimento. Puesto que en este experimento a diferencia del primero, los clasificadores son entrenados con las imágenes simuladas, hemos representado también en el espacio ROC el resultado correspondiente a la clasificación sobre el conjunto original cuando el clasificador se entrena también con las imágenes originales.

Estas gráficas muestran resultados muy importantes acerca la capacidad predictiva de las imágenes generadas. Analizando así la Figura 7.6, observamos que para los distintos tamaños del conjunto de entrenamiento, se obtienen buenos resultados con un sensibilidad entorno al 80 % y una especificidad en torno al 90 %, muy próximos al 87.2 % y 88.1 % de sensibilidad y especificidad respectivamente, correspondientes al valor de la clasificación sobre el conjunto original del primer experimento.

En el caso de la clasificación correspondiente al Grupo 2 (MCI vs NC)

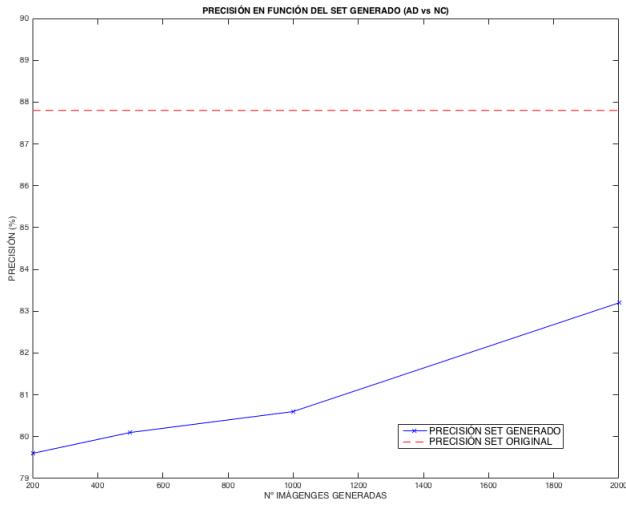


Figura 7.5: Evolución de la precisión en función del tamaño del conjunto generado correspondiente a la clasificación del Grupo 1 (Experimento 2).

en la Figura 7.7, se observa que no siendo resultados muy buenos usando un clasificador lineal, estos mejoran notablemente con respecto al resultado sobre la clasificación original del experimento 1. De manera similar ocurre en el Grupo 3.

Por tanto, los resultados presentados en la Tabla A.2 y las figuras correspondientes al espacio ROC, reflejan que las imágenes simuladas pueden predecir de manera efectiva la clase de las imágenes originales en los 4 escenarios. Además, a medida que generamos un conjunto de imágenes simuladas mayor, aumentan la precisión del clasificador, mientras se reduce de nuevo el valor de la varianza para cada uno de ellos.

### 7.3. Experimento 3

En este tercer experimento, el objetivo es probar la independencia de las imágenes simuladas con respecto de las originales, pero preservando las mismas características. Para ello “trucaremos” el clasificador de la siguiente manera:

Extraemos un pequeño subconjunto de la base de datos original. Entrenamos el clasificador con este subconjunto (en lugar del conjunto de entrenamiento) y lo testamos con el conjunto de test, extraído del mismo subconjunto. De esta manera, hemos “trucado” el clasificador, pues predice muestras que estaban dentro del conjunto de entrenamiento. Dado que se produce un sobre-

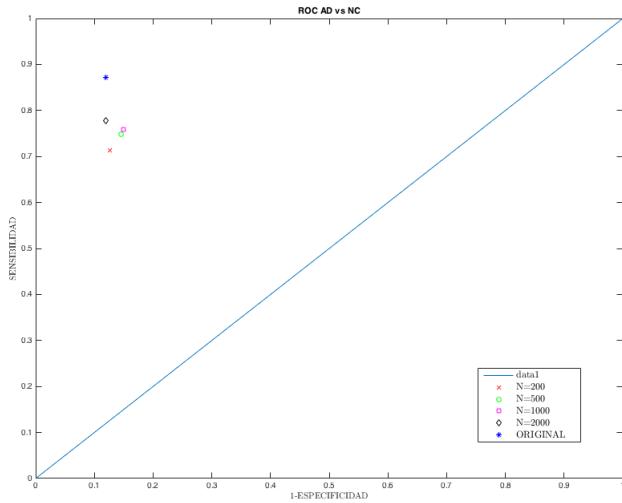


Figura 7.6: Espacio ROC experimento 2: Grupo 1 (AD vs NC).

ajuste, el rendimiento del clasificador debería ser próximo a 1.

A continuación, generamos a partir del subconjunto original un nuevo conjunto de imágenes que testamos usando el mismo clasificador inicial “trucado”. De manera que, si las imágenes que simulamos son independientes de las originales, el rendimiento del sistema debería disminuir notablemente. En caso contrario, estaríamos generando imágenes sin capacidad de generalización, y el sobreajuste del modelo daría un rendimiento similar.

Tabla 7.3: Resultados Experimento 3.

Escenario	Precisión ( $\pm$ SD)	Sensibilidad ( $\pm$ SD)	Especificidad ( $\pm$ SD) (%)	
Original	100 $\pm$ 0.0	100 $\pm$ 0.0	100 $\pm$ 0.0	Grupo 1
Simulado (N=N <sub>test</sub> )	80.0 $\pm$ 25.8	60.0 $\pm$ 51.6	100 $\pm$ 0.0	
Simulado (N=1000)	85.5 $\pm$ 2.9	81.6 $\pm$ 4.2	89.4 $\pm$ 5.5	
Original	90.0 $\pm$ 2.11	100 $\pm$ 0.0	80.0 $\pm$ 42.2	Grupo 2
Simulado (N=N <sub>test</sub> )	90.0 $\pm$ 21.1	90.0 $\pm$ 31.6	90.0 $\pm$ 31.6	
Simulado (N=1000)	77.8 $\pm$ 10.9	83.8 $\pm$ 12.1	71.8 $\pm$ 28.2	
Original	85.0 $\pm$ 24.2	70.0 $\pm$ 48.3	100 $\pm$ 0	Grupo 3
Simulado (N=N <sub>test</sub> )	80.0 $\pm$ 25.8	70.0 $\pm$ 48.3	90.0 $\pm$ 31.6	
Simulado (N=1000)	75.7 $\pm$ 14.5	60.8 $\pm$ 35.3	90.7 $\pm$ 7.2	

Decíamos que pretendemos demostrar la independencia de las imágenes simuladas con respecto de las originales trucando para ello el clasificador y obtener un sobreajuste intencionado con un conjunto pequeño de las imágenes originales, usándolo después para predecir la clase de nuevas imágenes simuladas. Así en la Tabla A.3 podemos observar que, mientras el rendimien-

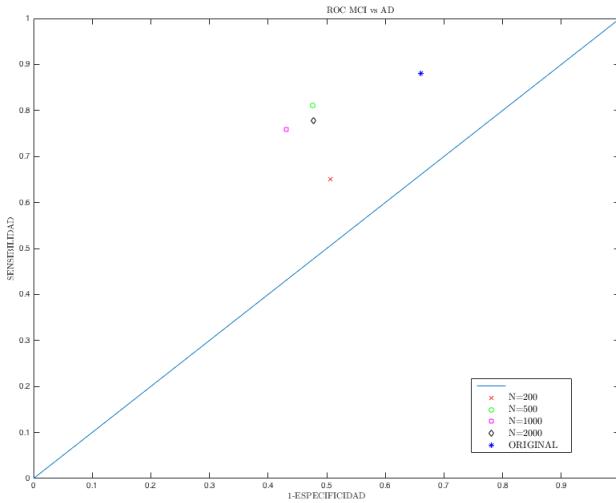


Figura 7.7: Espacio ROC experimento 2: Grupo 2 (MCI vs NC).

to del clasificador es aproximadamente perfecto en el escenario original, este disminuye notablemente cuando trata de predecir etiquetas pertenecientes a imágenes simuladas. Con sets de generación de 1000 imágenes se obtiene una reducción de aproximadamente el 15 % en los 3 casos.

Es natural pensar que en el escenario original de este experimento, los resultados fuesen del 100 % en los 3 casos, sin embargo solo observamos esto en el primer de ellos (AD vs NC), mientras que en el resto aún siendo muy buenos no alcanzan el 100 %. Como ya hemos explicado anteriormente, el motivo reside en que la clase MCI (ver 2.1.3), junto con la clase NC o AD, comparten características a nivel de voxel y son clases linealmente no separables. Así, aunque tengamos un sobre ajuste, usando un clasificador lineal, no conseguiremos una clasificación perfecta a no ser que las clases sean linealmente separables.

Por ello, podemos decir que el conjunto de imágenes simuladas es independiente del conjunto original, pues como habíamos planteado, el uso de un clasificador sobreajustado a los datos originales, tendría un rendimiento mucho menor sobre imágenes independientes de las originales.

## 7.4. Evolución de las PDFs

Hemos visto que para poder generar nuevas imágenes funcionales, generábamos nuevas coordenadas en el espacio de eigenbrains, habiendo estimado la distribución de cada uno de los scores previamente. Así en la Figura

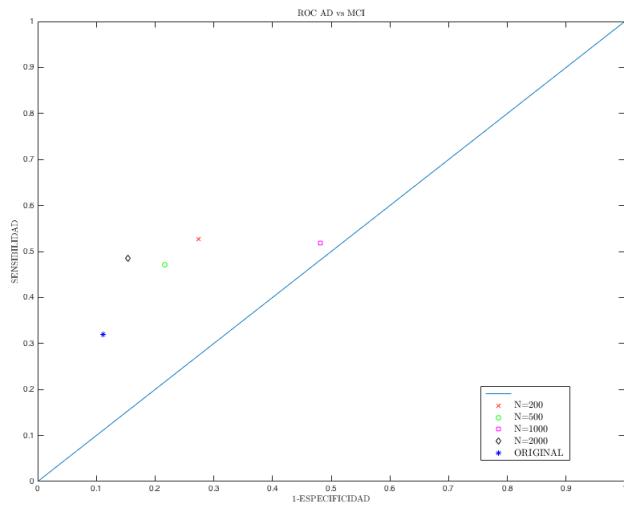
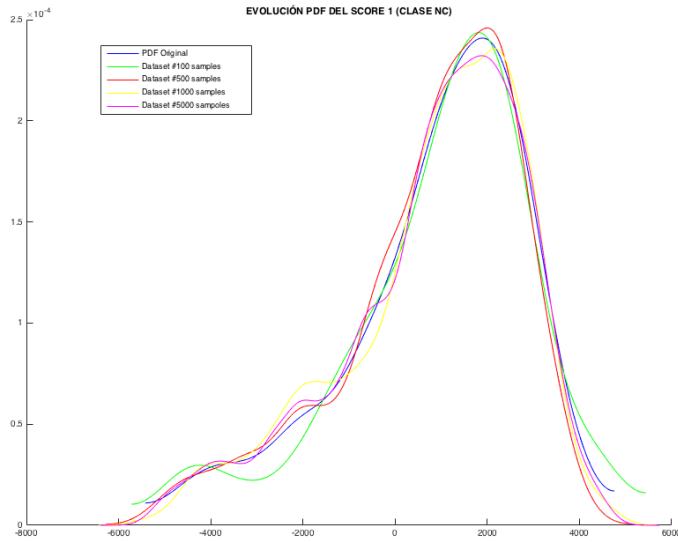


Figura 7.8: Espacio ROC experimento 2: Grupo 3 (AD vs MCI).

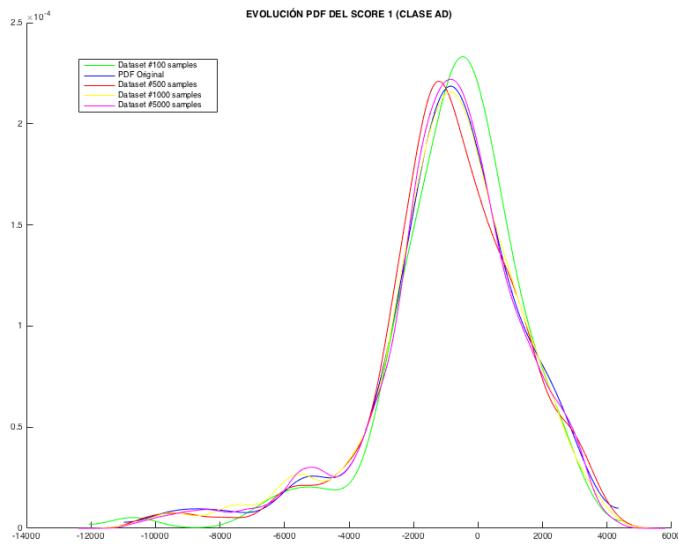
6.6 veíamos la estimación de la primera Componente Principal para las 3 clases y su origen desde cada uno de los 3 grupos definidos.

En la Figura 7.9 mostramos ahora no solo la distribución de la primera CP para la base de datos original, sino la estimación de la PDF para distintos sets de diferentes tamaños muestrales ( $N=100$ ,  $N=500$ ,  $N=1000$  y  $N=5000$ ). Para ello, una vez generado el conjunto de  $N$  imágenes AD y NC, hemos estimado de nuevo la distribución de las componentes principales mediante KDE.

Se puede comprobar que, efectivamente, la distribución de la primera componente principal para los distintos conjuntos simulados, presentan la misma distribución que el conjunto original.



(a) Clase NC



(b) Clase AD

Figura 7.9: Evolución de la PDF correspondiente a la primera Componente Principal para distintos tamaños de sets generados, para la clases NC y AD del Grupo 1.

## Capítulo 8

# Planificación y estimación de costes

Este capítulo está dedicado a la planificación del proyecto en su globalidad, desde su puesta en marcha con la documentación del problema hasta su resolución. Además, en él se detalla la organización temporal dedicada a cada una de las tareas que se llevan a cabo, así como una estimación de costes del mismo.

### 8.1. Fases del desarrollo

En esta sección enumeramos y describimos brevemente las distintas etapas desarrolladas durante el proyecto.

#### 8.1.1. Revisión del estado del arte

Como indicábamos en el primer capítulo, el objetivo de este proyecto era proponer una solución al problema del pequeño tamaño muestral en estudios funcionales. Previamente a la propuesta de la solución se ha realizado una exhaustiva revisión de la literatura relacionada con dicho problema, realizando diferentes lecturas de publicaciones que lo abordan y teniendo una idea general del trabajo realizado hasta la fecha en ese contexto.

La búsqueda y lectura de las publicaciones en este campo ha llevado un tiempo aproximado de 20 horas.

### 8.1.2. Solución propuesta y desarrollo

El proyecto elegido consiste en sí en una solución orientada al problema planteado, cuya implementación, estudio y discusión forman el bloque general del presente proyecto.

Como solución al problema del pequeño tamaño muestral, se propone un simulador de imágenes funcionales cerebrales para ampliar la base de datos en estudios funcionales. Se ha realizado una implementación de la solución en Matlab, desarrollando un algoritmo para tal fin.

Este bloque constituye uno de los dos bloques más importantes que forman el proyecto. La programación de la solución se ha llevado a cabo durante 4 meses, con un tiempo estimado de 80 horas.

### 8.1.3. Resultados experimentales

Una vez obtenida la solución, se han llevado a cabo diferentes experimentos con el fin de validar la bases de datos generada y comprobar la validez del algoritmo propuesto.

En total se han llevado a cabo tres esquemas experimentales, formados a su vez por diversas pruebas y configuraciones. Debido al gran tamaño de la base de datos original y simulada, el coste computacional implicado en la validación de los resultados ha sido muy elevado, estimando un tiempo empleado de 30 horas.

### 8.1.4. Documentación

La parte final del proyecto consiste en la documentación de todo el proceso realizado, incluyendo presentación y origen del problema, introducción teórica y matemática a los métodos usados, documentación del algoritmo implementado, así como los experimentos realizados y su posterior discusión e interpretación.

Este bloque, desarrollado a la par con el bloque anterior, forman el segundo gran bloque del proyecto. Así la búsqueda de información y la redacción del mismo ha supuesto un total de 70 horas.

### 8.1.5. Diagrama de Gantt

Con el objetivo de plasmar gráficamente las distintas etapas en la realización de este proyecto, mostramos en la Figura 8.1 el diagrama de Gantt realizado con Gantt Project.

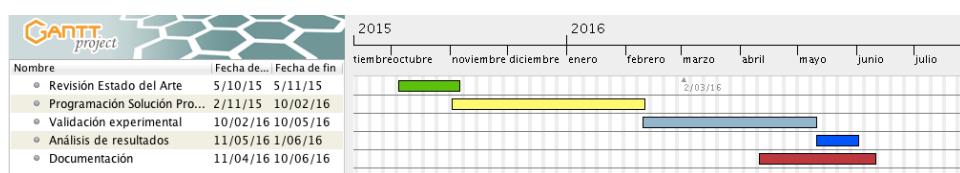


Figura 8.1: Diagrama de Gantt asociado.

## 8.2. Recursos

### Recursos Humanos

Los recursos humanos que han formado parte de este proyecto son:

- Francisco Jesús Martínez Murcia, becario del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, en calidad de tutor del proyecto.
- D. Juan Manuel Górriz Sáez, profesor del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, en calidad de tutor del proyecto.
- D. Javier Ramírez Pérez de Inestrosa, profesor del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada, en calidad de tutor del proyecto.
- Ulises Vidal Sanz, alumno de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación, de la universidad de Granada, como autor del proyecto.

### Recursos Hardware

Para la realización de este proyecto únicamente he hecho uso del ordenador personal, en el que he realizado toda la implementación del algoritmo en Matlab y redacción de la memoria con Latex.

### Recursos Software

Para el desarrollo del proyecto he hecho uso de los siguientes recursos software:

- Sistema operativo OS X Yosemite Versión 10.10.5
- Programación: Matlab 2015\_Rb y Matlab 2009
- Ofimática: TexMaker (LaTex)

### 8.3. Coste asociado

En esta última sección estimamos el coste derivado del desarrollo del algoritmo, como solución al problema del pequeño tamaño muestral. Te-

Nombre	Precio/hora (€/h)	Nº Horas	Total (€)
Tutores (Ingenieros senior)	50	15	750
Ulises Vidal Sanz (Ingeniero Junior)	20	200	4.000

Tabla 8.1: Coste asociado a los recursos humanos.

niendo en cuenta el tiempo empleado por los Ingenieros Senior (tutores del proyecto), así como las horas realizadas por el Ingeniero Juniors (autor del proyecto), la estimación de costes debido a los recursos humanos están reflejados en la Tabla 8.1.

Recurso Software	Coste (€)
S.O I OS	Precio PC
Matlab 2015Rb	2000
TexMaker	Software libre

Tabla 8.2: Coste asociado a los recursos software.

Por otro lado, los costes asociados a los recursos software quedan detallados en la Tabla 8.2.

Y por último los costes asociados a los recursos hardware se detallan en la Tabla 8.3.

Recurso Hardware	Coste inicial(€)	Vida media (ciclos)	Tiempo de uso (ciclos)	Coste (€)
Portátil McBook Air	1200	1000	100	120

Tabla 8.3: Coste asociado a los recursos hardware.

Teniendo en cuenta todo lo anterior, el coste total asociado a la realización del proyecto en su totalidad queda especificado en la Tabla 8.3.

Recursos	Coste (€)
Humanos	4750
Software	2000
Hardware	120
Total	6870

Tabla 8.4: Coste total asociado.



## **Capítulo 9**

# **Conclusiones y Trabajo futuro**

Este capítulo tiene como objetivo mostrar las conclusiones fruto de la realización del proyecto y mostrar las vías de trabajo futuras relacionadas con el mismo, que puedan ampliar y mejorar el trabajo hasta ahora existente.

### **9.1. Conclusiones**

Al comienzo de este proyecto veíamos que el incremento del uso de imágenes funcionales cerebrales para el diagnóstico y seguimiento de enfermedades neurodegenerativas, ha supuesto un avance sin precedentes en el campo de la medicina permitiendo así que radiólogos y médicos tengan a su alcance numerosas herramientas para el estudio del cerebro. Esto a su vez, implica que investigadores de todo el mundo estén llevando a cabo diferentes proyectos en diferentes áreas con un objetivo común: permitir que los avances en las imágenes funcionales y anatómicas sirvan para mejorar la práctica clínica y por tanto, facilitar el tratamiento a los enfermos.

La última tendencia en esta línea consiste en el uso de algoritmos de aprendizaje automático para obtener sistemas de diagnóstico con una alta precisión que permitan efectuar un diagnóstico automático. Sin embargo, existe un problema fundamental a la hora de evaluar el rendimiento de todos estos sistemas basados en técnicas estadísticas, el conocido como problema del pequeño tamaño muestral. Este ocurre cuando el número de muestras (100-500 pacientes) es muy pequeño en comparación con la dimensión del espacio de características, que se correspondería con los voxels de la imagen ( $\simeq 500.000$  voxels) [46]. En la sección 5.4 veíamos los fundamentos de este problema y sus implicaciones. Es por tanto necesario, obtener una solución que permita desarrollar y evaluar algoritmos de diagnóstico de manera real

y precisa.

Este proyecto se presenta como una solución novel al problema del pequeño tamaño muestral. Aunque numerosas soluciones parciales han sido estudiadas en la literatura [49, 50, 51], es posible que muchas de las publicaciones existentes hasta ahora muestren resultado subestimados o erróneos [52]. En contraste con las soluciones hasta ahora existentes, basadas en la generalización de los clasificadores o métodos para la extracción de características, en este proyecto presentamos un algoritmo novel capaz de simular imágenes cerebrales funcionales con el objetivo de aumentar el conjunto de entrenamiento y test, abordando así el problema desde un nuevo punto de vista.

De manera simplificada el algoritmo se divide en:

- Caracterización de la base de datos original
- Estimación de la distribución de las características
- Simulación de nuevas imágenes funcionales

La base de datos con la que hemos trabajado durante la realización del proyecto pertenece a la iniciativa ADNI (véase 2) y está formada por 403 imágenes FDG-PET que incluye imágenes de pacientes sanos (NC), pacientes que sufren la Enfermedad del Alzheimer (AD) y pacientes con deterioro cognitivo leve (MCI).

Puesto que se trata de un sistema multivariado, hemos hecho uso de la transformación PCA para caracterizar la base de datos. Esta transformación aprovecha la relación existente entre las variables originales (voxels) que dominan el comportamiento del sistema, para agruparlas en unas nuevas variables denominadas Componentes Principales, resultado de proyectar las imágenes originales en el espacio de eigenbrains cuya base ortogonal agrupa las variables en orden decreciente de varianza. Por su efectividad, este método ha sido usado en diversos trabajos campos como la detección de AD [64], reconocimiento de emociones [65], reconocimiento facial [66], clasificación de electrocardiogramas [67] o escoliosis [68].

Una vez caracterizada la bases de datos original formadas por las 403 imágenes, se ha estimado la distribución que siguen cada nueva variable en el espacio de eigenbrains. Aunque existen numerosos métodos paramétricos y no paramétricos para la estimación de la distribución que sigue una determinada población muestral, hemos hecho uso KDE vía difusión 4.4.1 para estimar la distribución de las CPs. La motivación principal para usar este método se basa en que no hace ninguna suposición previa sobre la distribución que sigue una población y esto es fundamental a la hora de caracterizar una base de datos cuya distribución se desconoce. Además a diferencia de

otros métodos, el parámetro de suavizado  $h$  se elige automáticamente sobre los datos sin hacer ninguna suposición previa y sin usar un modelo de referencia.

Conociendo la distribución que siguen las CPs en el nuevo espacio, podemos usar la CDF para interpolar nuevos valores y obtener así nuevas CPs en el espacio de eigenbrains. La figura 6.7 explica gráficamente este procedimiento. Invirtiendo la transformación PCA, podemos trasladar estas nuevas variables simuladas al espacio original (voxels), obteniendo así nuevas imágenes simuladas con las mismas características y patrones que las imágenes originales.

De hecho, comprobar la validez de la nueva base de datos generada es parte fundamental del proyecto. Idealmente queremos comprobar que las imágenes simuladas presentan las mismas características que las originales y representan patrones de difusión realistas en función de la etiqueta asociada (AD, MCI o NC). Un primer síntoma de ello se observa al representar los eigenbrains o vectores ortogonales de la nueva base ortogonal, obtenidos al aplicar PCA al conjunto de imágenes originales. Algunos cortes representativos de los 4 primeros eigenbrains se muestran en 6.4, mostrando patrones característicos de la enfermedad del Alzheimer, como diferencias en el metabolismo de glucosa entre el lóbulo principal y occipital o entre el giro cingulado y el precúneo.

Con el fin de validar el conjunto de imágenes generadas, hemos llevado a cabo 3 experimentos diferentes usando un análisis de clasificación mediante Máquinas de Soporte Vectorial, que por su simplicidad y atractivo computacional ha sido ampliamente usado en la literatura, desde la detección de actividad vocal [112, 113, 114, 115, 116, 117], clasificación de texturas [108] y diagnóstico de imágenes médicas [53, 109, 110, 111]. Además los vectores de características se han generado siguiendo una aproximación VAF, siendo los voxels las características que lo forman. El objetivo de esto es obtener los valores de rendimiento del clasificador (sensibilidad, sensibilidad y precisión) con el fin de compararlos para distintos esquemas de validación.

En el primero de ellos hemos usado la base de datos simulada para entrenar y testar un clasificador lineal, obteniendo los parámetros de rendimiento del clasificador que hemos comparado con la clasificación de la base de datos original usada como conjunto de entrenamiento y test. Los resultados obtenidos en el experimento A.1, muestra que la base de datos simulada presenta características muy similares a la base de datos original, pues el rendimiento de ambos clasificadores es próximo, y aumentando el tamaño del conjunto generado la diferencia entre ellos es aún menor.

En el segundo experimento entrenábamos el clasificador con el conjunto de imágenes simuladas para clasificar así imágenes originales, con el objetivo de comprobar la capacidad de predicción de las imágenes generadas. Los

resultados de este segundo experimento quedan recogidos en la tabla A.2 y a partir de su representación en el espacio ROC 7.6,7.7,7.8, se demuestra que en la clasificación AD vs NC es muy similar a la clasificación AD vs NC usando solo imágenes originales. Del mismo modo ocurre para los casos MCI vs NC y AD vs MCI. Además se observa que el rendimiento aumenta a medida que aumentamos el tamaño de la base de datos simulada, de la cual hemos llegado a generar hasta diez veces más (4000) sujetos que los 403 que formaban la base original.

Con el objetivo de comprobar la independencia de la base de datos simuladas con respecto de la base de datos original, describimos el experimento 3, en el que forzamos un sobreajuste del clasificador usando un pequeño conjunto de imágenes originales. Con esas mismas imágenes generamos un set mayor de imágenes y las clasificamos usando el mismo clasificador inicial. Mientras el rendimiento al clasificar las imágenes originales es próximo al 100 %, observamos una disminución sustancial en el rendimiento cuando clasificamos las imágenes generadas, lo que indica que las imágenes simuladas son independientes de las originales. De no ser así el rendimiento se mantendría próximo al 100 % al clasificar el conjunto generado.

Así el algoritmo novel presentado es este proyecto para la simulación de imágenes funcionales cerebrales, es capaz de generar bases de datos simuladas cuyas imágenes mantienen la misma capacidad predictiva y además se ha comprobado que son independientes de las originales. Por tanto, podemos afirmar que el algoritmo presentado puede usarse para incrementar el tamaño muestral en estudios funcionales con el objetivo de validar sistemas de clasificación obtenido resultados más fiables y consistentes, que se traducirán en sistemas de diagnóstico reales.

También mencionar que fruto de la realización de este proyecto, hemos elaborado un artículo que ha sido aceptado en el congreso NSS-MIC organizado por el IEEE:

- U. Vidal-Sanz, F.J. Martínez-Murcia, J.M. Górriz, J. Ramírez, I.A. Illán, F. Segovia, D. Salas-González. *Simulating Functional Brain Images in Alzheimer's Disease*.

## **9.2. Trabajo futuro**

Aunque el objetivo principal para el desarrollo de un algoritmo capaz de generar una base de datos de imágenes funcionales es paliar uno de los problemas más frecuentes en neuroimagen, este puede tener un uso puramente educativo. Así, en términos de funcionalidad, se propone como trabajo futuro el desarrollo de un aplicación que implemente este algoritmo y evalúe los conocimientos del estudiante o médico el cual intentará predecir la clase

de imagen simulada por la aplicación.

En lo que se refiere a metodología se proponen varias líneas de trabajo futuro. Por un lado, en lo relativo al reconocimiento de patrones en sistemas multivariados, además de PCA se propone el uso de:

- Análisis de Componentes Independientes (ICA por sus siglas en inglés).
- Análisis Discriminante Lineal (LDA por sus siglas en inglés)

ICA pretende descomponer una señal observada en una combinación lineal de fuentes indendientes, mientras que LDA es una técnica de aprendizaje supervisado para analizar datos. Caracterizar por tanto la base de datos original con estos métodos y analizar los resultados en comparación con PCA forman parte del trabajo futuro.

Por otro lado, como bien es sabido, existen numerosos métodos de estimación de la distribución de una determinada población. Se propone el uso de otros métodos no paramétricos con el fin de comparar las distribuciones estimadas y los resultados que se infieren sobre las imágenes finales.

En este trabajo las imágenes FDG-PET que estiman la actividad metabólica del cerebro, han sido usadas para generar y validar las imágenes simuladas. Se propone así su extensión a otras modalidades de imágenes y trazadores como PiB-PET o SPECT-HMPAO, usados tanto en la Enfermedad del Alzheimer como en la Enfermedad de Parkinson.



# Bibliografía

- [1] Karen Ritchie and Simon Lovestone. The dementias. *The Lancet*, 360(9347):1759–1766, 2002.
- [2] K Maurer, S Volk, and H Gerbaldo. Auguste d. première patiente du docteur alzheimer. *La recherche*, 303, 1997.
- [3] M J de Leon, S H Ferris, A E George, B Reisberg, D R Christman, I I Kricheff, and A P Wolf. Computed tomography and positron emission transaxial tomography evaluations of normal aging and Alzheimer's disease. *Journal of Cerebral Blood Flow and Metabolism: Official Journal of the International Society of Cerebral Blood Flow and Metabolism*, 3(3):391–4, September 1983. PMID: 6603463.
- [4] Norman L. Foster, Thomas N. Chase, Paul Fedio, Nicholas J. Patronas, Rodney A. Brooks, and Giovanni Di Chiro. Alzheimer's disease: Focal cortical changes shown by positron emission tomography. *Neurology*, 33(8):961, August 1983.
- [5] T N Chase, N L Foster, P Fedio, R Brooks, L Mansi, and G Di Chiro. Regional cortical dysfunction in Alzheimer's disease as determined by positron emission tomography. *Annals of Neurology*, 15 Suppl:S170–4, 1984. PMID: 6611118.
- [6] R. Duara, C. Grady, J. Haxby, M. Sundaram, N. R. Cutler, L. Heston, A. Moore, N. Schlageter, S. Larson, and S. I. Rapoport. Positron emission tomography in Alzheimer's disease. *Neurology*, 36(7):879–887, July 1986.
- [7] E G McGeer, R P Peppard, P L McGeer, H Tuokko, D Crockett, R Parks, H Akiyama, D B Calne, B L Beattie, and R Harrop. 18Fluoro-deoxyglucose positron emission tomography studies in presumed Alzheimer cases, including 13 serial scans. *The Canadian Journal of Neurological Sciences. Le Journal Canadien Des Sciences Neurologiques*, 17(1):1–11, February 1990. PMID: 2311010.

- [8] Satoshi Minoshima, Norman L. Foster, and David E. Kuhl. Posterior cingulate cortex in Alzheimer's disease. *The Lancet*, 344(8926):895, September 1994.
- [9] V Ibañez, P Pietrini, G E Alexander, M L Furey, D Teichberg, J C Rajapakse, S I Rapoport, M B Schapiro, and B Horwitz. Regional glucose metabolic abnormalities are not the result of atrophy in Alzheimer's disease. *Neurology*, 50(6):1585–1593, June 1998. PMID: 9633698.
- [10] J. M. Hoffman, K. A. Welsh-Bohmer, and M. Hanson. FDG PET imaging in patients with pathologically verified dementia. *Journal of Nuclear Medicine*, 41(11):1920–1928, 2000.
- [11] D. Kogure, H. Matsuda, T. Ohnishi, T. Asada, M. Uno, T. Kunihiro, S. Nakano, and M. Takasaki. Longitudinal Evaluation of Early Alzheimer Disease Using Brain Perfusion SPECT. *The Journal of Nuclear Medicine*, 41(7):1155–1162, 2000.
- [12] G. E. Alexander, P. Pietrini, S. I. Rapoport, and E. M. Reiman. Longitudinal PET evaluation of cerebral metabolic decline in dementia: A potential outcome measure in Alzheimer's disease treatment studies. *The American Journal of Psychiatry*, 159(5):783–745, 2002.
- [13] L. Mosconi, W. H. Tsui, K. Herholz, A. Pupi, A. Drzezga, G. Lucignani, E. M. Reiman, V. Holthoff, E. Kalbe, S. Sorbi, J. Diehl-Schmid, R. Perneczky, F. Clerici, R. Caselli, B. Beuthien-Baumann, A. Kurz, S. Minoshima, and M. J. de Leon. Multicenter standardized 18F-FDG PET diagnosis of mild cognitive impairment, Alzheimer's disease and other dementias. *Journal of Nuclear Medicine*, 49(3):390–398, 2008.
- [14] J. B. Langbaum, K. Vhen, W. Lee, C. Reschke, D. Bandy, A. S. Fleisher, G. E. Alexander, N. L. Foster, M. W. Weiner, R. A. Koeppe, W. J. Jagust, and E. M. Reiman. Categorical and correlation analyses of baseline fluorodeoxyglucose positron emission tomography images from the Alzheimer's disease neuroimaging initiative (ADNI). *Neuroimage*, 45(4):1107–1116, 2009.
- [15] K. Lange and R. Carson. EM reconstruction for emission and transmission tomography. *Journal of Computer Assisted Tomography*, 8:306–312, 1984.
- [16] Y. Vardi, L. A. Shepp, and L. Kaufman. A Statistical Model for Positron Emission Tomography. *Journal of the American Statistical Association*, 80(389):8–20, 1985.
- [17] H. Malcolm Hudson and Richard S. Larkin. Accelerated Image Reconstruction Using Ordered Subsets of Projection Data. *IEEE Transactions on Medical Imaging*, 13(4):601–609, 1994.

- [18] P. P. Bruyant. Analytic and iterative reconstruction algorithms in SPECT. *The Journal of Nuclear Medicine*, 43(10):1343–1358, 2002.
- [19] D. H. Silverman, G. W. Small, and C. Y. Chang. Positron emission tomography in evaluation of dementia: regionalbrain metabolism and long-term outcome. *Journal of the American Medical Association*, 286(17):2120–2127, 2001.
- [20] R. Higdon, N. L. Foster, R. A. Koeppe, C. S. DeCarli, W. J. Jagust, C. M. Clark, N. R. Barbas, S. E. Arnold, R. S. Turner, J. L. Heidebrink, and S. Minoshima. A comparison of classification methods for differentiating fronto-temporal dementia from Alzheimer’s disease using FDG-PET imaging. *Statistics in Medicine*, 23:315–326, 2004.
- [21] Cristina Messa, Daniela Perani, Giovanni Lucignani, Arturo Zenorini, Felicia Zito, Giovanna Rizzo, Franco Grassi, Angelo Del Sole, Massimo Franceschi, Maria Carla Gilardi, and Ferruccio Fazio. High-Resolution Technetium-99m-HMPAO SPECT in Patients with Probable Alzheimer’s Disease: Comparison with Fluorine-18-FDG PET. *Journal of Nuclear Medicine*, 35(2):210–216, 1994.
- [22] Karl Herholz, Helge Schopphoff, Mathias Schmidt, Rudiger Mielke, Wolfgang Eschner, Klemens Scheidhauer, Harald Schicha, Wolf-Dieter Heiss, and Klaus Ebmeier. Direct Comparison of Spatially Normalized PET and SPECT Scans in Alzheimer’s Disease. *Journal of Nuclear Medicine*, 43(1):21–26, 2002.
- [23] Daniel H.S. Silverman. Brain 18F-FDG PET in the Diagnosis of Neurodegenerative Dementias: Comparison with Perfusion SPECT and with Clinical Evaluations Lacking Nuclear Imaging. *Journal of Nuclear Medicine*, 45(4):594–607, April 2004.
- [24] K.J. Friston, J. Ashburner, S.J. Kiebel, T.E. Nichols, and W.D. Penny. *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, 2007.
- [25] S Minoshima, K A Frey, R A Koeppe, N L Foster, and D E Kuhl. A diagnostic approach in Alzheimer’s disease using three-dimensional stereotactic surface projections of fluorine-18-FDG PET. *Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine*, 36(7):1238–48, July 1995. PMID: 7790950.
- [26] A. Drzezga, N. Lautenschlager, H. Siebner, M. Riemschneider, F. Willoch, S. Minoshima, M. Schwaiger, and A. Kurz. Cerebral metabolic changes accompanying conversion of Mild Cognitive Impairment into Alzheimer’s disease: a PET follow-up study. *European Journal of Nuclear Medicine and Molecular Imaging*, 30(8):1104–1113, 2003.

- [27] K. Ishii, A. K. Kono, H. Sasaki, N. Miyamoto, T. Fukuda, S. Sakamoto, and E. Mori. Fully automatic diagnostic system for early- and late-onset mild Alzheimer's disease using FDG PET and 3D-SSP. *European Journal of Nuclear Medicine and Molecular Imaging*, 33(5):575–583, 2006.
- [28] Nikolaos Scarmeas, Christian G. Habeck, Eric Zarahn, Karen E. Anderson, Aileen Park, John Hilton, Gregory H. Pelton, Matthias H. Tabert, Lawrence S. Honig, James R. Moeller, Davangere P. Devanand, and Yaakov Stern. Covariance PET patterns in early Alzheimer's disease and subjects with cognitive impairment but no dementia: utility in group discrimination and correlations with functional performance. *Neuroimage*, 23(1):35–45, 2004.
- [29] Stefan J. Teipel, Robert Stahl, Olaf Dietrich, Stefan O. Schoenberg, Robert Perneczky, Arun L.W. Bokde, Maximilian F. Reiser, Hans-Jürgen Möller, and Harald Hampel. Multivariate network analysis of fiber tract integrity in Alzheimer's disease. *Neuroimage*, 34(3):985–995, February 2007.
- [30] Eric Salmon, Nacer Kerrouche, Daniela Perani, Françoise Lekeu, Vjera Holthoff, Bettina Beuthien-Baumann, Sandro Sorbi, Christian Lemaire, Fabienne Collette, and Karl Herholz. On the multivariate nature of brain metabolic impairment in Alzheimer's disease. *Neurobiology of Aging*, 30(2):186–197, 2009.
- [31] P.J. Markiewicz, J.C. Matthews, J. Declerck, and K. Herholz. Robustness of multivariate image analysis assessed by resampling techniques and applied to FDG-PET scans of patients with Alzheimer's disease. *Neuroimage*, 46:472–485, 2009.
- [32] Flavio Nobili, Dario Salmaso, Silvia Morbelli, Nicola Girtler, Arnoldo Piccardo, Andrea Brugnolo, Barbara Dessì, Stig A. Larsson, Guido Rodriguez, and Marco Pagani. Principal component analysis of FDG PET in amnestic MCI. *European Journal of Nuclear Medicine and Molecular Imaging*, 35(12):2191–2202, 2008.
- [33] Kewei Chen, Eric M. Reiman, Zhongdan Huan, Richard J. Caselli, Daniel Bandy, Napatkamon Ayutyanont, and Gene E. Alexander. Linking functional and structural brain images with multivariate network analyses: A novel application of the partial least square method. *NeuroImage*, 47(2):602–610, 2009.
- [34] Stefan Klöppel, Cynthia M Stonnington, Carlton Chu, Bogdan Draganski, Rachael I Scahill, Jonathan D Rohrer, Nick C Fox, Clifford R

- Jack, John Ashburner, and Richard SJ Frackowiak. Automatic classification of mr scans in alzheimer's disease. *Brain*, 131(3):681–689, January 2008.
- [35] Benoît Magnin, Lilia Mesrob, Serge Kinkignéhun, Mélanie Pélégrini-Issac, Olivier Colliot, Marie Sarazin, Bruno Dubois, Stéphane Lehéricy, and Habib Benali. Support vector machine-based classification of alzheimer's disease from whole-brain anatomical mri. *Neuroradiology*, 51(2):73–83, 2009.
- [36] Daoqiang Zhang, Yaping Wang, Luting Zhou, Hong Yuan, Dinggang Shen, Alzheimer's Disease Neuroimaging Initiative, et al. Multimodal classification of alzheimer's disease and mild cognitive impairment. *Neuroimage*, 55(3):856–867, 2011.
- [37] Benson Mwangi, Klaus P Ebmeier, Keith Matthews, and J Douglas Steele. Multi-centre diagnostic classification of individual structural neuroimaging scans from patients with major depressive disorder. *Brain*, 135(5):1508–1521, 2012.
- [38] Ling-Li Zeng, Hui Shen, Li Liu, Lubin Wang, Baojuan Li, Peng Fang, Zongtan Zhou, Yaming Li, and Dewen Hu. Identifying major depression using whole-brain functional connectivity: a multivariate pattern analysis. *Brain*, 135(5):1498–1507, 2012.
- [39] Christine Ecker, Andre Marquand, Janaina Mourão-Miranda, Patrick Johnston, Eileen M. Daly, Michael J. Brammer, Stefanos Maltezos, Clodagh M. Murphy, Dene Robertson, Steven C. Williams, and Declan G M. Murphy. Describing the brain in autism in five dimensions—magnetic resonance imaging-assisted diagnosis of autism spectrum disorder using a multiparameter classification approach. *The Journal of Neuroscience*, 30(32):10612–10623, August 2010.
- [40] Madhura Ingalkar, Drew Parker, Luke Bloy, Timothy PL Roberts, and Ragini Verma. Diffusion based abnormality markers of pathology: toward learned diagnostic prediction of asd. *Neuroimage*, 57(3):918–927, 2011.
- [41] Nikolaos Koutsouleris, Eva M Meisenzahl, Christos Davatzikos, Ronald Bottlender, Thomas Frodl, Johanna Scheuerecker, Gisela Schmitt, Thomas Zetzsche, Petra Decker, Maximilian Reiser, et al. Use of neuroanatomical pattern classification to identify subjects in at-risk mental states of psychosis and predict disease transition. *Archives of general psychiatry*, 66(7):700–712, 2009.
- [42] Sven Haller, Karl O Lovblad, and Pantelimon Giannakopoulos. Principles of classification analyses in mild cognitive impairment (mci) and

- alzheimer disease. *Journal of Alzheimer's Disease*, 26(s3):389–394, 2011.
- [43] Lena Lim, Andre Marquand, Ana A Cubillo, Anna B Smith, Kaylita Chantiluke, Andrew Simmons, Mitul Mehta, and Katya Rubia. Disorder-specific predictive classification of adolescents with attention deficit hyperactivity disorder (adhd) relative to autism using structural magnetic resonance imaging. *PloS one*, 8(5):e63660, 2013.
  - [44] João Ricardo Sato, Marcelo Queiroz Hoexter, André Fujita, and Luis Augusto Rohde. Evaluation of pattern recognition and feature extraction methods in adhd prediction. *Frontiers in systems neuroscience*, 6(68):23015782, 2012.
  - [45] CZ Zhu, YF Zang, M Liang, LX Tian, Y He, XB Li, MQ Sui, YF Wang, and TZ Jiang. Discriminative analysis of brain function at resting-state for attention-deficit/hyperactivity disorder. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005*, pages 468–475. Springer, 2005.
  - [46] S. J. Raudys and A. K. Jain. Small sample size effects in statistical pattern recognition: recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264, March 1991.
  - [47] J. Stoeckel, G. Malandain, O. Migneco, P. M. Koulibaly, P. Robert, N. Ayache, and J. Darcourt. Classification of SPECT Images of Normal Subjects versus Images of Alzheimer's Disease Patients. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, volume 2208 of *Lecture Notes in Computer Science*, pages 666–674. Springer, 2001.
  - [48] J. Stoeckel, N. Ayache, G. Malandain, P. M. Koulibaly, K. P. Ebmeier, and J. Darcourt. Automatic Classification of SPECT Images of Alzheimer's Disease Patients and Control Subjects. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, volume 3217 of *Lecture Notes in Computer Science*, pages 654–662. Springer, 2004.
  - [49] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.
  - [50] Jianping Hua, Waibhav D Tembe, and Edward R Dougherty. Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42(3):409–424, 2009.

- [51] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.
- [52] Katherine S Button, John PA Ioannidis, Claire Mokrysz, Brian A Nosek, Jonathan Flint, Emma SJ Robinson, and Marcus R Munafò. Power failure: why small sample size undermines the reliability of neuroscience. *Nature Reviews Neuroscience*, 14(5):365–376, 2013.
- [53] G. Fung and J. Stoeckel. SVM feature selection for classification of SPECT images of Alzheimer’s disease using spatial information. *Knowledge and Information Systems*, 11(2):243–258, 2007.
- [54] F.J. Martínez-Murcia, J.M. Górriz, J. Ramírez, C.G. Puntonet, and D. Salas-González. Computer Aided Diagnosis tool for Alzheimer’s Disease based on Mann-Whitney-Wilcoxon U-Test. *Expert Systems with Applications*, 39(10):9676–9685, August 2012.
- [55] Andrés Ortiz, Juan M Górriz, Javier Ramírez, and Francisco Jesús Martínez-Murcia. Lvq-SVM based CAD tool applied to structural MRI for the diagnosis of the alzheimer’s disease. *Pattern Recognition Letters*, 34(14):1725–1733, October 2013.
- [56] L. Khedher, J. Ramírez, J.M. Górriz, A. Brahim, and F. Segovia. Early diagnosis of alzheimer’s disease based on partial least squares, principal component analysis and support vector machine using segmented MRI images. *Neurocomputing*, 151:139–150, March 2015.
- [57] I. A. Illán, J. M. Górriz, J. Ramírez, D. Salas-Gonzalez, M. M. López, F. Segovia, R. Chaves, M. Gómez-Rio, and C. G. Puntonet.  $^{18}\text{F}$ -FDG PET imaging analysis for computer aided Alzheimer’s diagnosis. *Information Sciences*, 181:903–916, February 2011.
- [58] Clifford R Jack, Matt A Bernstein, Nick C Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J Britson, Jennifer L Whitwell, Chadwick Ward, et al. The alzheimer’s disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging*, 27(4):685–691, 2008.
- [59] Susan S Spencer, William H Theodore, and Samuel F Berkovic. Clinical applications: Mri, spect, and pet. *Magnetic resonance imaging*, 13(8):1119–1124, 1995.
- [60] R. P. Woods. Spatial Transformation Models. In Isaac N. Bankman, editor, *Handbook of Medical Imaging*, chapter 29, pages 465–490. Academic Press, San Diego, 2000.

- [61] J. Ramírez, J. M. Górriz, M. Gómez-Río, A. Romero, R. Chaves, A. Lassl, A. Rodríguez, C. G. Puntonet, F. Theis, and E. W. Lang. Effective Emission Tomography Image Reconstruction Algorithms for SPECT Data. In *ICCS 2008, Part I, LNCS*, volume 5101/2008, pages 741–748. Springer-Verlag Berlin Heidelberg, 2008.
- [62] J. Ashburner and K. J. Friston. Nonlinear spatial normalization using basis functions. *Human Brain Mapping*, 7(4):254–66, 1999.
- [63] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.
- [64] M. López, J. Ramírez, J.M. Górriz, I. Álvarez, D. Salas-Gonzalez, F. Segovia, R. Chaves, P. Padilla, and M. Gómez-Río. Principal component analysis-based techniques and supervised classification schemes for the early detection of Alzheimer’s disease. *Neurocomputing*, 74(8):1260 – 1271, 2011. Selected Papers from the 3<sup>rd</sup> International Work-Conference on the Interplay between Natural and Artificial Computation (IWINAC 2009).
- [65] A. P. Gosavi and S. R. Khot. Emotion recognition using principal component analysis with singular value decomposition. In *Proc. Int Electronics and Communication Systems (ICECS) Conf*, pages 1–5, February 2014.
- [66] Z. Lihong and G. Zikui. Face recognition method based on adaptively weighted block-two dimensional principal component analysis. In *Proc. Third Int Computational Intelligence, Communication Systems and Networks (CICSyN) Conf*, pages 22–25, July 2011.
- [67] R. Kottaimalai, M. P. Rajasekaran, V. Selvam, and B. Kannapiran. EEG signal classification using principal component analysis with neural network in brain computer interface applications. In *Proc. Int Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN) Conf*, pages 227–231, March 2013.
- [68] Hong Lin. Clustering property of spinal deformity classification with simplified 3-D model and principal component analysis. In *Proc. Int. Conf. Information Technology and Applications in Biomedicine*, pages 358–361, May 2008.
- [69] Abhilash Alexander Miranda, Yann-Aël Le Borgne, and Gianluca Bon-tempi. New routes from minimal approximation error to principal components. *Neural Processing Letters*, 27(3):197–207, 2008.
- [70] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.

- [71] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.
- [72] Antonio Cuevas. *El análisis estadístico de grandes masas de datos: Algunas tendencias actuales*.
- [73] Antonio Miñarro. *Estimación no paramétrica de la función de densidad*. Enero 1998.
- [74] ZI Botev, JF Grotowski, DP Kroese, et al. Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.
- [75] Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis–nonparametric discrimination: consistency properties. Technical report, DTIC Document, 1951.
- [76] Murray Rosenblatt et al. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.
- [77] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.
- [78] Simon J Sheather and Michael C Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690, 1991.
- [79] Jeffrey S Simonoff. *Smoothing methods in statistics*. Springer Science & Business Media, 2012.
- [80] Matt P Wand and M Chris Jones. *Kernel smoothing*. Crc Press, 1994.
- [81] Daniel Cremers, Stanley J Osher, and Stefano Soatto. Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. *International Journal of Computer Vision*, 69(3):335–351, 2006.
- [82] Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, 2002.
- [83] Anurag Mittal and Nikos Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–302. IEEE, 2004.

- [84] V. Akbarzadeh, C. Gagné, and M. Parizeau. Kernel density estimation for target trajectory prediction. In *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, pages 3449–3456, September 2015.
- [85] R. Annoni and C. H. Q. Forster. Analysis of aircraft trajectories using Fourier descriptors and kernel density estimation. In *Proc. 15th Int. IEEE Conf. Intelligent Transportation Systems*, pages 1441–1446, September 2012.
- [86] Y. Cao, L. Zhang, and D. Sun. An air traffic prediction model based on kernel density estimation. In *Proc. American Control Conf*, pages 6333–6338, June 2013.
- [87] A. Banerjee and P. Burlina. Efficient particle filtering via sparse kernel density estimation. *IEEE Transactions on Image Processing*, 19(9):2480–2490, September 2010.
- [88] I. B. P. P. Dinata and B. Hardian. Predicting smart home lighting behavior from sensors and user input using very fast decision tree with kernel density estimation and improved Laplace correction. In *Proc. Int Advanced Computer Science and Information Systems (ICACIS) Conf*, pages 171–175, October 2014.
- [89] MC Jones, James Stephen Marron, and SJ Sheather. Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, 11(3):337–381, 1996.
- [90] Duc Devroye, J Beirlant, R Cao, R Fraiman, P Hall, MC Jones, Gábor Lugosi, Enno Mammen, JS Marron, C Sánchez-Sellero, et al. Universal smoothing factor selection in density estimation: theory and practice. *Test*, 6(2):223–320, 1997.
- [91] M Chris Jones, James S Marron, and Simon J Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996.
- [92] Adrian W Bowman and Adelchi Azzalini. *Applied smoothing techniques for data analysis*. Clarendon Press, 2004.
- [93] J Steve Marron and Matt P Wand. Exact mean integrated squared error. *The Annals of Statistics*, pages 712–736, 1992.
- [94] George R Terrell and David W Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.
- [95] James Stephen Marron and David Ruppert. Transformations to reduce boundary bias in kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 653–671, 1994.

- [96] BU Park, Seok-Oh Jeong, MC Jones, and Kee-Hoon Kang. Adaptive variable location kernel density estimators with good performance at boundaries. *Journal of Nonparametric Statistics*, 15(1):61–75, 2003.
- [97] MC Jones and DF Signorini. A comparison of higher-order bias kernel density estimators. *Journal of the American Statistical Association*, 92(439):1063–1073, 1997.
- [98] Ian S Abramson. On bandwidth variation in kernel estimates-a square root law. *The annals of Statistics*, pages 1217–1223, 1982.
- [99] Peter Hall. On the bias of variable bandwidth curve estimators. *Biometrika*, 77(3):529–535, 1990.
- [100] Peter Hall, Tien Chung Hu, and James Stephen Marron. Improved variable window kernel estimates of probability densities. *The Annals of Statistics*, pages 1–10, 1995.
- [101] MC Jones, IJ McKay, and T-C Hu. Variable location and scale kernel density estimation. *Annals of the Institute of Statistical Mathematics*, 46(3):521–535, 1994.
- [102] Stig Larsson and Vidar Thomée. *Partial differential equations with numerical methods*, volume 45. Springer Science & Business Media, 2008.
- [103] Probal Chaudhuri and James Steven Marron. Scale space view of curve estimation. *Annals of Statistics*, pages 408–428, 2000.
- [104] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.
- [105] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [106] T. Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In *Lecture Notes in Computer Science*, volume 1398, pages 137–142, 1998.
- [107] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006.
- [108] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim. Support Vector Machines for Texture Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1542–1550, 2002.

- [109] I. Kalatzis, D. Pappas, N. Piliouras, and D. Cavouras. Support vector machines based analysis of brain SPECT images for determining cerebral abnormalities in asymptomatic diabetic patients. *Medical Informatics and the Internet in Medicine*, 28(3):221–230, 2003.
- [110] Ignacio A. Illán. *Análisis en Componentes de Imágenes Funcionales para la Ayuda al Diagnóstico de la Enfermedad de Alzheimer*. PhD thesis, Universidad de Granada, June 2009.
- [111] M. López, J. Ramírez, J. M. Górriz, D. Salas-Gonzalez, I. Álvarez, F. Segovia, and C. G. Puntonet. Automatic Tool for the Alzheimer’s Disease Diagnosis Using PCA and Bayesian Classification Rules. *IET Electronics Letters*, 45(8):389–391, 2009.
- [112] D. Enqing, L. Guizhong, Z. Yatong, and Z. Xiaodi. Applying support vector machines to voice activity detection. In *6<sup>th</sup> International Conference on Signal Processing*, volume 2, pages 1124–1127, 2002.
- [113] D. Enqing, Z. Heming, and L. Yongli. Low bit and variable rate speech coding using local cosine transform. In *Proc. of the 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering (TENCON ’02)*, volume 1, pages 423–426, 2002.
- [114] F. Qi, C. Bao, and Y. Liu. A novel two-step SVM classifier for voiced/unvoiced/silence classification of speech. In *International Symposium on Chinese Spoken Language Processing*, pages 77–80, 2004.
- [115] J. Ramírez, P. Yélamos, J. M. Górriz, C. G. Puntonet, and J. C. Segura. SVM-enabled Voice Activity Detection. In *Lecture Notes in Computer Science*, volume 3972, pages 676–681, 2006.
- [116] J. Ramírez, P. Yélamos, J. M. Górriz, and J. C. Segura. SVM-based speech endpoint detection using contextual speech features. *Electronics Letters*, 42(7):877–879, 2006.
- [117] P. Yélamos, J. Ramírez, J. M. Górriz, C. G. Puntonet, and J. C. Segura. Speech Event Detection Using Support Vector Machines. In *Lecture Notes in Computer Science*, volume 3991, pages 356–363, 2006.
- [118] Ron Kohavi and George H. John. Wrappers for Feature Subset Selection, 1995.
- [119] Pierre A. Devijver and Josef Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, first edition edition, 1982.
- [120] T. M. Cover. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, 14(3):326–334, 1965.

- [121] P. Saxena, D. G. Pavel, J. C. Quintana, and B. Horwitz. An Automatic Threshold-Based Scaling Method for Enhancing the Usefulness of Tc-HMPAO SPECT in the Diagnosis of Alzheimer's Disease. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, volume 1496 of *Lecture Notes in Computer Science*, pages 623–630. Springer, 1998.
- [122] D. Salas-Gonzalez, J. M. Górriz, J. Ramírez, I. A. Illán, M. López, F. Segovia, R. Chaves, P. Padilla, and C. G. Puntonet. Feature selection using factor analysis for Alzheimer's diagnosis using F-FDG PET images. *Medical Physics*, 37(11):6084–95, November 2010.
- [123] Míriam López, JM Górriz, Javier Ramírez, Diego Salas-Gonzalez, Rosa Chaves, and Manuel Gómez-Río. SVM with bounds of confidence and pls for quantifying the effects of acupuncture on migraine patients. In *Hybrid Artificial Intelligent Systems*, pages 132–139. Springer, 2011.
- [124] I. Goethals, C. van deWiele, D. Slosman, and R. Dierckx. Brain SPET perfusion in early Alzheimer disease: where to look? *European Journal of Nuclear Medicine*, 29(8):975–978, 2002.



## Apéndice A

# Resultados Detallados

En este apéndice mostramos todos los resultados de clasificación obtenidos en los 3 experimentos.

Tabla A.1: Resultados Experimento 1.

Escenario	Precisión ( $\pm$ SD)	Sensibilidad ( $\pm$ SD)	Especificidad ( $\pm$ SD) (%)	
Original	$87.8 \pm 7.1$	$87.2 \pm 13.6$	$88.1 \pm 9.2$	Grupo 1
Simulado (N=200)	$77.5 \pm 8.2$	$78.0 \pm 11.4$	$77.0 \pm 10.6$	
Simulado (N=500)	$81.4 \pm 5.5$	$81.2 \pm 9.2$	$83.5 \pm 2.8$	
Simulado (N=1000)	$81.5 \pm 3.4$	$80.0 \pm 5.8$	$83.0 \pm 6.1$	
Simulado (N=1500)	$81.9 \pm 1.4$	$80.4 \pm 3.6$	$81.6 \pm 10.2$	
Simulado (N=2000)	$81.4 \pm 2.3$	$80.7 \pm 3.5$	$82.0 \pm 3.1$	
Original	$70.1 \pm 6.9$	$88.0 \pm 7.3$	$33.9 \pm 17.3$	Grupo 2
Simulado (N=200)	$69.9 \pm 8.4$	$68.0 \pm 1.69$	$70.0 \pm 11.5$	
Simulado (N=500)	$72.0 \pm 6.0$	$74.0 \pm 7.4$	$70.0 \pm 8.7$	
Simulado (N=1000)	$80.3 \pm 2.9$	$81.2 \pm 4.2$	$79.4 \pm 3.9$	
Simulado (N=1500)	$80.7 \pm 2.5$	$80.9 \pm 5.6$	$80.5 \pm 3.8$	
Simulado (N=2000)	$80.6 \pm 2.8$	$80.5 \pm 4.4$	$80.7 \pm 3.2$	
Original	$70.9 \pm 7.4$	$31.9 \pm 16.3$	$88.9 \pm 8.0$	Grupo 3
Simulado (N=302)	$75.7 \pm 5.1$	$36.6 \pm 16.1$	$93.2 \pm 7.3$	
Simulado (N=500)	$77.4 \pm 4.0$	$76.0 \pm 9.6$	$78.8 \pm 5.0$	
Simulado (N=1000)	$78.2 \pm 2.6$	$75.8 \pm 4.8$	$80.6 \pm 6.3$	
Simulado (N=1500)	$80.02 \pm 4.0$	$79.3 \pm 5.1$	$81.1 \pm 6.5$	
Simulado (N=2000)	$81.2 \pm 3.0$	$79.8 \pm 3.6$	$82.6 \pm 4.2$	

Tabla A.2: Resultados Experimento 2.

Escenario	Precisión ( $\pm SD$ )	Sensibilidad ( $\pm SD$ )	Especificidad ( $\pm SD$ ) (%)
AD vs NC (N=200)	$79.6 \pm 10.7$	$71.3 \pm 18.5$	$87.3 \pm 12.1$
AD vs NC (N=500)	$80.1 \pm 9.2$	$74.9 \pm 17.5$	$85.4 \pm 12.8$
AD vs NC (N=1000)	$80.6 \pm 11.6$	$75.9 \pm 15.6$	$85.1 \pm 12.7$
AD vs NC (N=2000)	$83.2 \pm 9.2$	$77.8 \pm 19.8$	$88.1 \pm 9.2$
MCI vs NC (N=200)	$59.9 \pm 4.0$	$65.0 \pm 9.8$	$49.4 \pm 18.6$
MCI vs NC (N=500)	$71.8 \pm 8.9$	$81.0 \pm 9.3$	$52.4 \pm 11.7$
MCI vs NC (N=1000)	$69.6 \pm 5.4$	$75.9 \pm 6.2$	$56.9 \pm 14.3$
MCI vs NC (N=2000)	$69.2 \pm 8.3$	$77.8 \pm 6.1$	$52.2 \pm 17.2$
AD vs MCI (N=200)	$66.2 \pm 8.6$	$52.7 \pm 25.6$	$72.6 \pm 12.1$
AD vs MCI (N=500)	$68.5 \pm 4.5$	$47.1 \pm 14.8$	$78.3 \pm 9.1$
AD vs MCI (N=1000)	$70.2 \pm 7.6$	$51.9 \pm 12.0$	$78.8 \pm 13.5$
AD vs MCI (N=2000)	$73.2 \pm 6.5$	$48.6 \pm 10.7$	$84.6 \pm 10.0$
MCI as NC (Original)	$79.7 \pm 4.0$	$25.7 \pm 18.8$	$96.4 \pm 4.2$
MCI as NC (N=200)	$74.9 \pm 7.5$	$52.6 \pm 14.8$	$81.8 \pm 9.1$
MCI as NC (N=500)	$77.2 \pm 4.4$	$58.1 \pm 18.3$	$83.1 \pm 6.1$
MCI as NC (N=1000)	$76.9 \pm 5.8$	$50.4 \pm 13.0$	$85.0 \pm 7.5$
MCI as NC (N=1500)	$78.7 \pm 5.3$	$52.6 \pm 15.2$	$86.7 \pm 5.9$
MCI as AD (Original)	$78.7 \pm 5.2$	$92.7 \pm 6.3$	$36.6 \pm 2.0$
MCI as AD (N=200)	$69.2 \pm 7.3$	$71.2 \pm 8.5$	$63.7 \pm 16.4$
MCI as AD (N=500)	$72.2 \pm 8.8$	$76.6 \pm 9.6$	$59.5 \pm 16.0$
MCI as AD (N=1000)	$75.2 \pm 4.0$	$81.1 \pm 6.8$	$57.5 \pm 11.8$
MCI as AD (N=1500)	$73.9 \pm 6.8$	$80.8 \pm 6.7$	$53.5 \pm 22.3$

Tabla A.3: Resultados Experimento 3.

Escenario	Precisión ( $\pm SD$ )	Sensibilidad ( $\pm SD$ )	Especificidad ( $\pm SD$ ) (%)	(%)
Original	$100 \pm 0.0$	$100 \pm 0.0$	$100 \pm 0.0$	Grupo 1
Simulado (N=N <sub>test</sub> )	$80.0 \pm 25.8$	$60.0 \pm 51.6$	$100 \pm 0.0$	
Simulado (N=1000)	$85.5 \pm 2.9$	$81.6 \pm 4.2$	$89.4 \pm 5.5$	
Original	$90.0 \pm 2.11$	$100 \pm 0.0$	$80.0 \pm 42.2$	Grupo 2
Simulado (N=N <sub>test</sub> )	$90.0 \pm 21.1$	$90.0 \pm 31.6$	$90.0 \pm 31.6$	
Simulado (N=1000)	$77.8 \pm 10.9$	$83.8 \pm 12.1$	$71.8 \pm 28.2$	
Original	$85.0 \pm 24.2$	$70.0 \pm 48.3$	$100 \pm 0$	Grupo 3
Simulado (N=N <sub>test</sub> )	$80.0 \pm 25.8$	$70.0 \pm 48.3$	$90.0 \pm 31.6$	
Simulado (N=1000)	$75.7 \pm 14.5$	$60.8 \pm 35.3$	$90.7 \pm 7.2$	

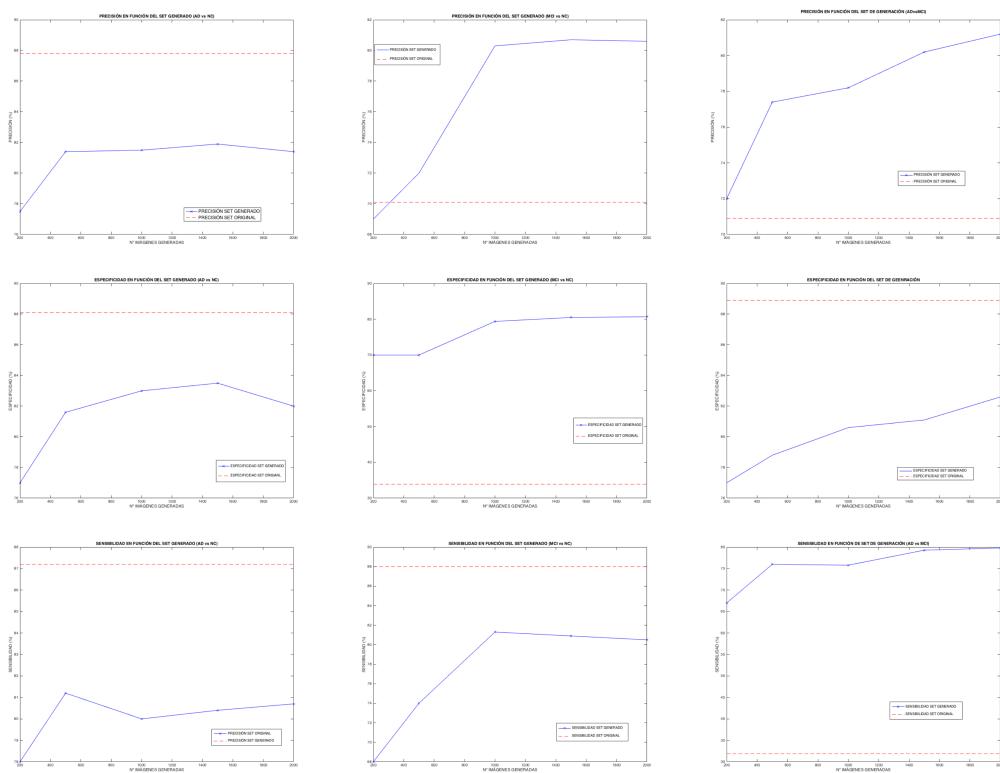


Figura A.1: Evolución de la precisión, especificidad y sensibilidad para los 3 grupos del experimento 1. Columna 1: ADvsNC. Columna 2: MCIvsNC. Columna 3: ADvsMCI.

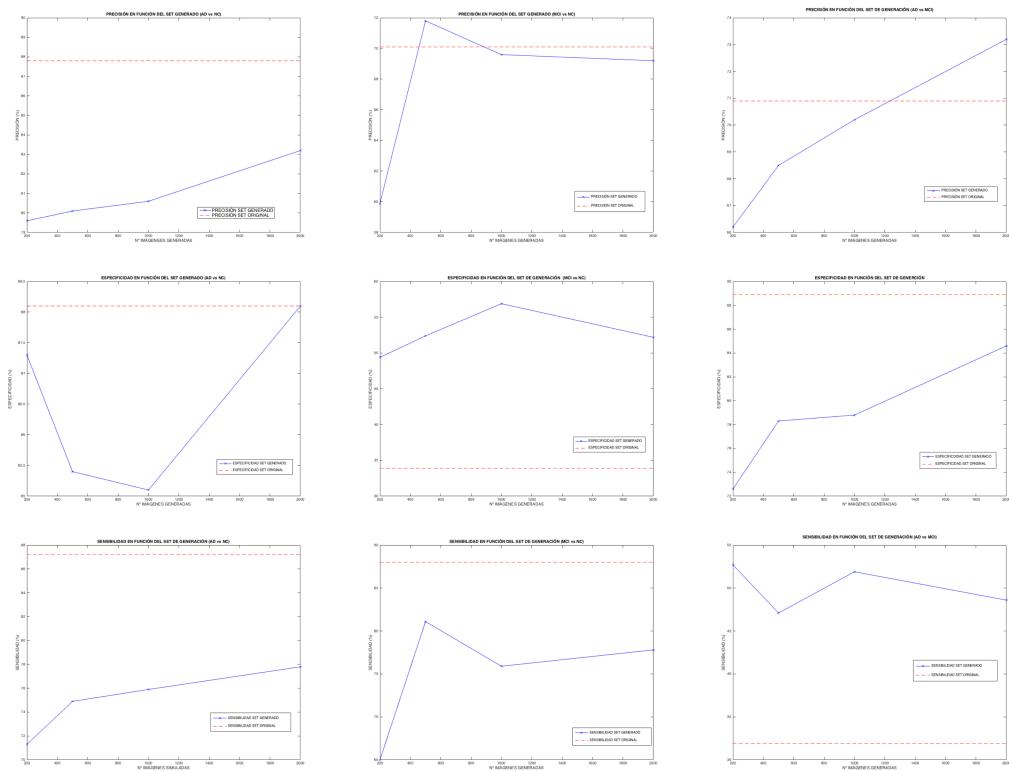


Figura A.2: Evolución de la precisión, especificidad y sensibilidad para los 3 grupos del experimento 2. Columna 1: ADvsNC. Columna 2: MCIvsNC. Columna 3: ADvsMCI.

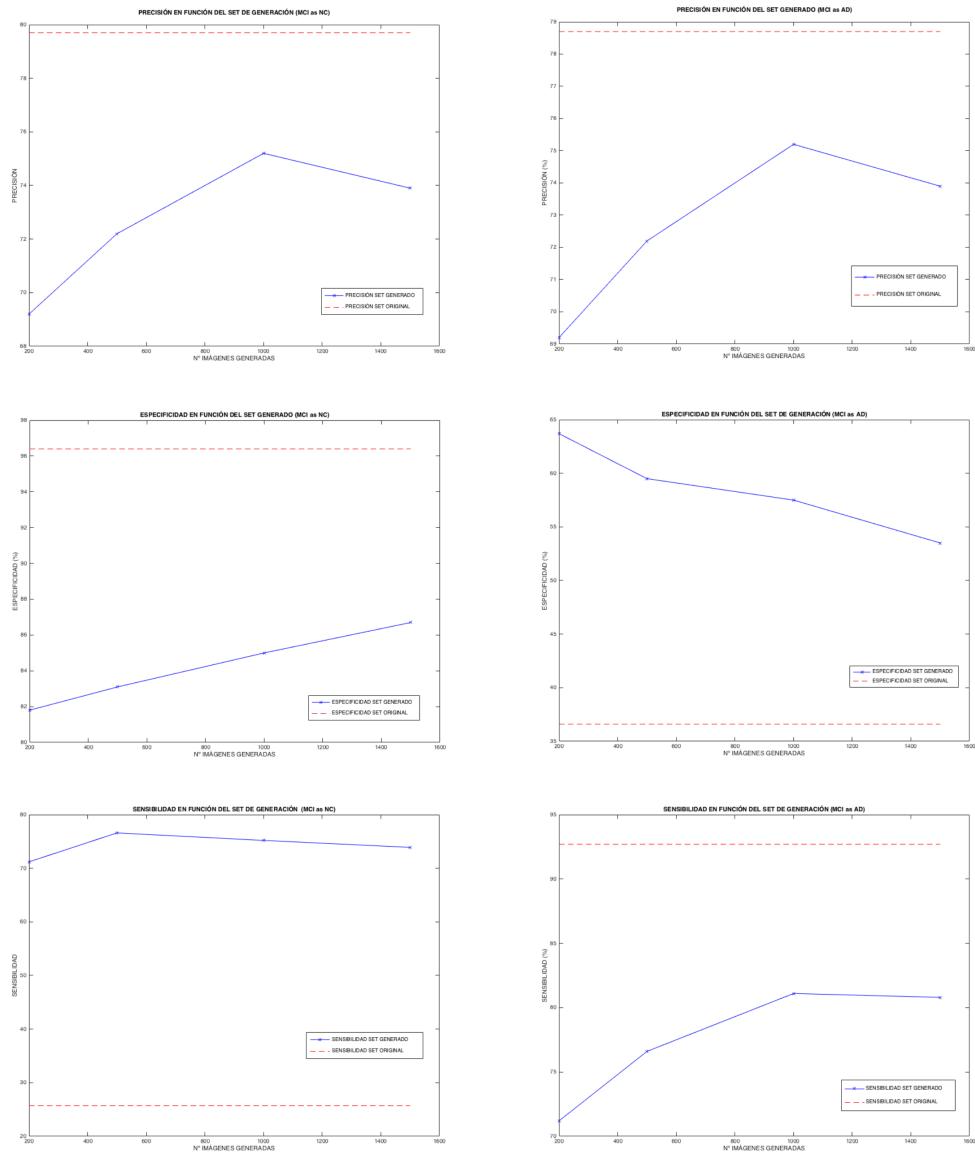


Figura A.3: Evolución de la precisión, especificidad y sensibilidad para los 3 grupos del experimento 2. Columna 1: MCIasNC. Columna 2: MCIasAD.

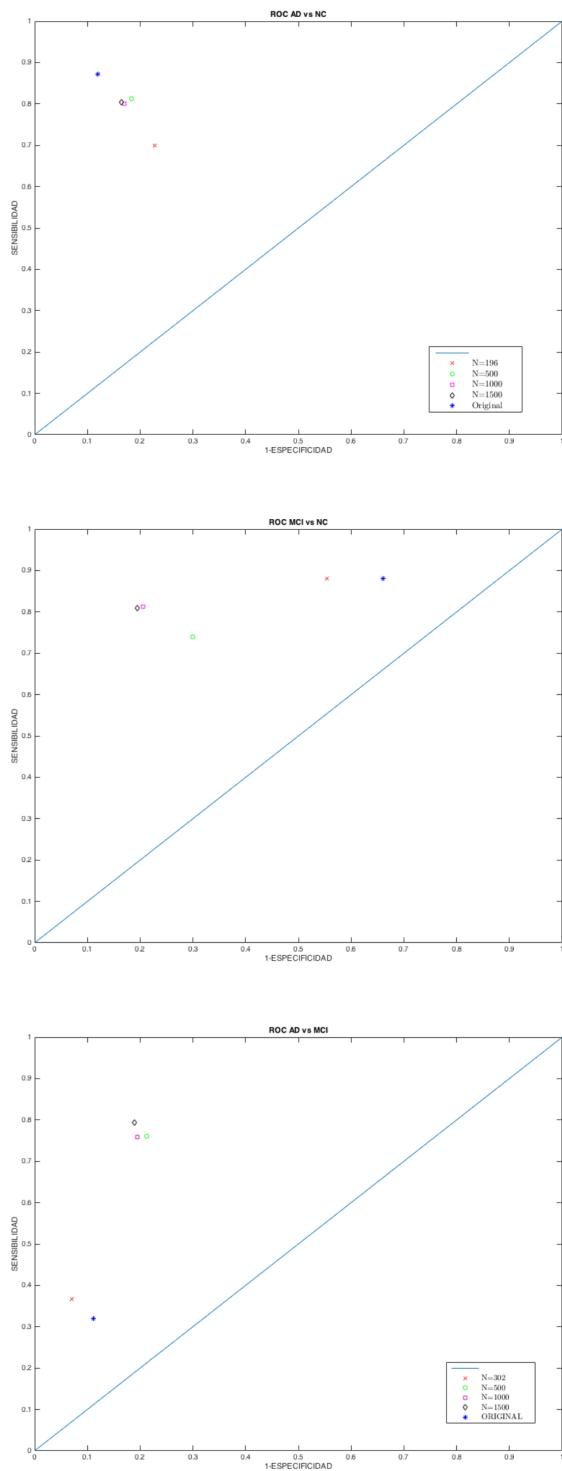


Figura A.4: Resultados de la clasificación del experimento 1 en el espacio ROC.

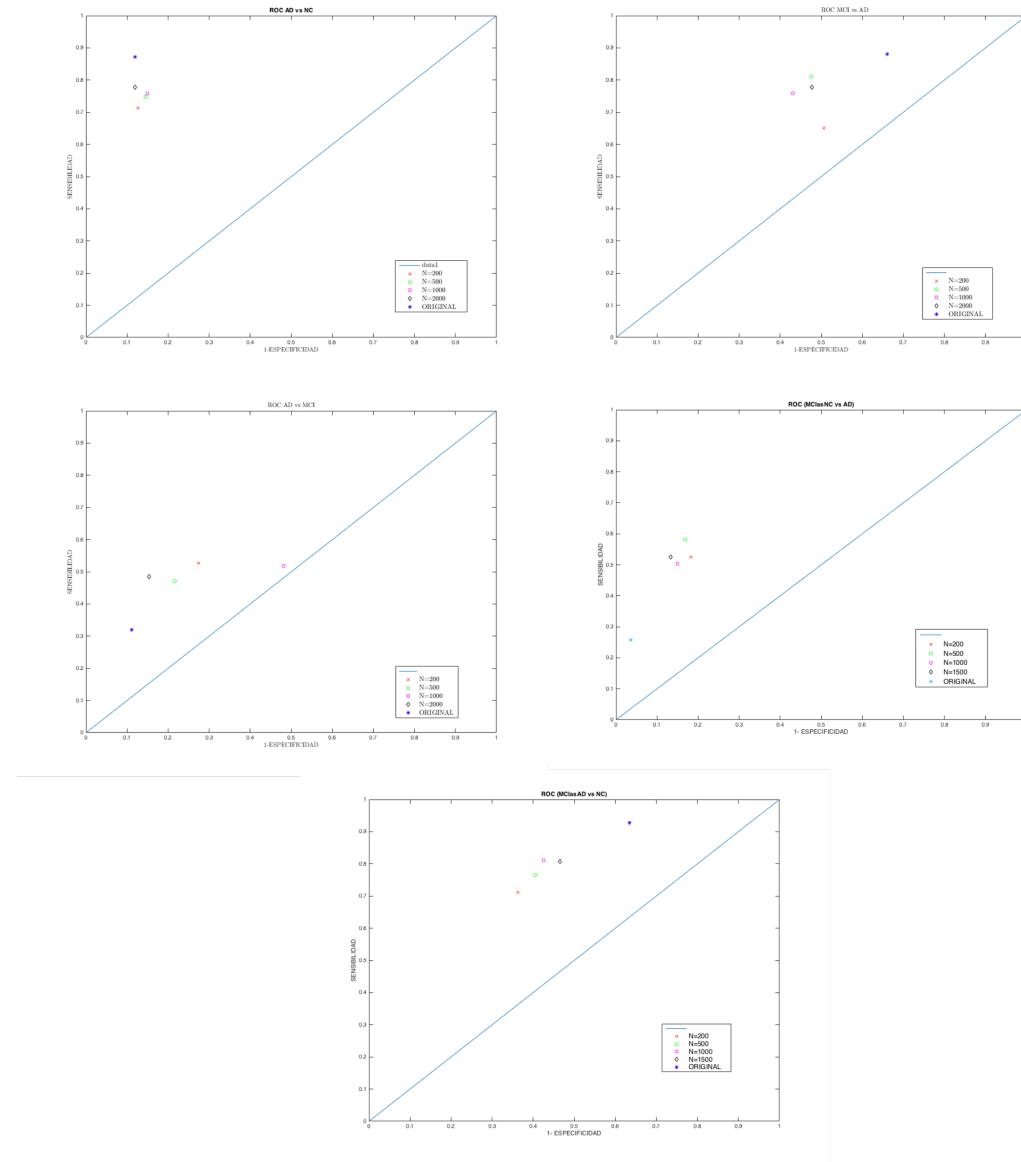


Figura A.5: Resultados de la clasificación del experimento 2 en el espacio ROC.



