

Universidad de Alcalá

Escuela Politécnica Superior

Grado en Ingeniería en Tecnologías de la Telecomunicación

Trabajo Fin de Carrera

Análisis de características para la detección del dolor
mediante el estudio del ECG

Autor: Gonzalo Corral García

Tutor: Manuel Blanco Velasco

2021

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

Grado en Ingeniería en Tecnologías de la
Telecomunicación

Trabajo Fin de Grado
Análisis de características para la detección del dolor
mediante el estudio del ECG

Autor: D. Gonzalo Corral García

Tutor: Dr. Manuel Blanco Velasco

Tribunal:

Presidente: Dr. Roberto Gil Pita

Vocal 1º: Dr. Francisco Javier Escribano Aparicio

Vocal 2º: Dr. Manuel Blanco Velasco

Fecha: 2021

Análisis de características para la detección del dolor mediante el estudio del ECG

Gonzalo Corral García

2 de julio de 2021

*Dedicado a mi familia y a mis amigos. Las personas más importantes de mi vida. Solo ellos saben lo difícil que ha sido llegar hasta aquí ya que sin su ayuda y comprensión nada de esto hubiera sido posible.
Mamá, aunque ya no estés, esto va por ti.*

Agradecimientos

Gracias a Manuel Blanco Velasco por su dedicación tan intensa con este trabajo en su rol como tutor y a Carolina Roza por su apoyo.

Asimismo, doy las gracias a las universidades de Otto-von-Guericke-Universität Magdeburg y de Ulm, en especial, a Philipp Werner por la base de datos BioVid, la cual ha sido una herramienta fundamental para la realización de este trabajo.

Este TFG ha sido financiado parcialmente por los proyectos CCGP19-CCS-013 y COVID-19 2021 2020/00003/016/001/019 de la Universidad de Alcalá, y por el proyecto EPU-INV/2020/002 de la Comunidad de Madrid y de la Universidad de Alcalá.

Índice general

Agradecimientos	III
Resumen	XVII
Abstract	XIX
Resumen extendido	XXI
Lista de acrónimos	XXIII
Lista de símbolos	XXVII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Metodología	3
1.4. Estructura de la memoria	4
2. Antecedentes biomédicos	5
2.1. El sistema nervioso	5
2.2. La sensación de dolor	6
2.3. El electrocardiograma	8
2.3.1. Actividad electromecánica del corazón	9
2.3.2. Medición de la actividad eléctrica del corazón	11
2.3.3. La señal de ECG y sus intervalos temporales	13
2.4. Señal de variabilidad cardiaca	15

3. Base de datos	19
3.1. Señales de biopotenciales de la base de datos	20
3.2. Protocolo de adquisición de los datos	21
3.3. Estructura de la base de datos	23
4. Métodos matemáticos	27
4.1. Métodos temporales	27
4.1.1. SDNN	27
4.1.2. SDRR	28
4.1.3. NN50 y pNN50	28
4.1.4. RMSSD	28
4.2. Métodos espectrales	29
4.3. Métodos de entropía	30
4.3.1. Entropía de Shannon	31
4.3.2. Entropía aproximada	32
4.3.3. Entropía muestral	36
4.3.4. Entropía espectral	37
4.3.5. Entropía de permutación	38
4.4. Máquinas de vectores soporte	41
4.4.1. SVM para ejemplos linealmente separables	42
4.4.2. SVM para datos cuasi-linealmente separables	44
4.4.3. SVM para datos no linealmente o cuasi-linealmente separables	45
5. Tratamiento de los datos	49
5.1. Preprocesado de los datos	49
5.2. Tratamiento del ECG	50
5.3. Experimento de clasificación por SVM	56
6. Conclusiones y trabajo futuro	65
6.1. Contribuciones y conclusiones	65
6.2. Trabajo futuro	67
Bibliografía	68

7. Anexo: Código fuente

73

Lista de figuras

2.1.	Sistema nervioso humano. Imagen de [36]	7
2.2.	Escala visual de intensidad de dolor.	8
2.3.	Estructura física del corazón. Imágenes de [6] y [5]	9
2.4.	Componentes de onda del ECG producidas por cada parte del corazón. Imagen de [27]	11
2.5.	Sistema de 12 derivaciones. Imágenes de [16] y [2]	12
2.6.	Estructura de la señal de ECG, donde int representa intervalo, amp es amplitud y dur duración. Imagen de [31]	13
2.7.	Descripción funcional del algoritmo de Pan-Tomkins.	15
2.8.	Procesado de la señal de ECG en el algoritmo de Pan-Tomkins. Imagen de [32]	15
2.9.	ECG y señal de intervalo tras su muestreo.	16
3.1.	Ejemplo de configuración física un experimento simular, que muestra a un paciente con un sistema de producción de dolor en el brazo. Imagen tomada de [15]	21
3.2.	Niveles de dolor inducidos en los pacientes mediante la aplicación de calor para la adquisición de los datos.	22
3.3.	Temporización de aplicación de dolor a los pacientes para la adquisición de los datos.	23
4.1.	Componentes espectrales del ECG Referencia: [27]	30
4.2.	Ejemplo de los patrones para el cálculo de la entropía aproximada sobre la serie U .	35
4.3.	Ejemplo del cálculo de la entropía de permutación sobre la señal $X[n]$.	40
4.4.	Fronteras y posibles márgenes de separación entre datos linealmente separables, usando distintos hiperplanos.	42

4.5.	Frontera óptima y margen de separación de datos linealmente separables. En rojo se pueden apreciar los vectores soporte de los datos de color negro y, en azul, los vectores soportes de los datos de color naranja.	43
4.6.	Frontera óptima y margen de separación de datos cuasilinealmente separables. En rojo se pueden ver los vectores soporte de los datos de color naranja y, en azul, los vectores soportes de los datos de color negro. En este caso se puede observar cómo algunos de los datos usados como vectores soporte están dentro del margen, incluso se aprecia que uno de los vectores en azul está clasificado de forma errónea en el lado de la frontera perteneciente a los datos de color naranja.	44
4.7.	Clases en el espacio original, separadas en el espacio transformado, y frontera de separación en el espacio original. Los puntos rojos y azules representan los vectores soporte de la clases de color negro y naranja respectivamente.	46
5.1.	Ejemplo de ECG y sus niveles de dolor correspondientes, tras el paso por un detector de Pan-Tomkins.	51
5.2.	Ejemplo de una señal de HRV y sus niveles de dolor correspondientes.	52
5.3.	Ejemplo de una fila de la matriz de HRV.	55
5.4.	Ejemplo de una fila de la matriz de ECG.	56
5.5.	Matriz de confusión resultante de la clasificación de las características con un kernel lineal ($BoxConstraint = 1$) sin downsampling, permutación de los datos o optimización de hiperparámetros.	57
5.6.	Matriz de confusión resultante de la clasificación de las características con un kernel gaussiano ($BoxConstraint = 1$ y $KernelScale = 1,1637$) sin downsampling, permutación de los datos o optimización de hiperparámetros.	58

5.7.	Matriz de confusión resultante de la clasificación de las características con un kernel gaussiano ($BoxConstraint = 1$ y $KernelScale = 1,171$) con downsampling y permutación de los datos, pero sin optimización de hiperparámetros.	59
5.8.	Matriz de confusión resultante de la clasificación de las características con un kernel polinómico de $PolynomialOrder = 2$ y $BoxConstraint = 0,25578$ con downsampling, permutación de los datos y optimización de hiperparámetros.	60
5.9.	Matriz de confusión resultante de la clasificación de las características y la señal de HRV con un kernel gaussiano ($BoxConstraint = 152,12$ y $KernelScale = 10,586$) con downsampling, permutación de los datos y optimización de hiperparámetros.	60
5.10.	Matriz de confusión resultante de la clasificación de las características y la señal de ECG con un kernel gaussiano ($BoxConstraint = 186,9604$ y $KernelScale = 4,8396$) downsampling, permutación de los datos y optimización de hiperparámetros.	62
5.11.	Matriz de confusión resultante de la clasificación de las características junto a las señales de ECG y HRV usando un kernel gaussiano con $BoxConstraint = 138,5564$ y $KernelScale = 87,8377$, con downsampling, permutación de los datos y optimización de hiperparámetros.	62
5.12.	Matriz de confusión resultante de la clasificación de la señal de HRV con un kernel polinómico con $PolynomialOrder = 3$ $BoxConstraint = 0,0032379$, con downsampling, permutación de los datos y optimización de hiperparámetros	63
5.13.	Matriz de confusión resultante de la clasificación de la señal de ECG con un kernel gaussiano ($BoxConstraint = 469,33$ y $KernelScale = 7,7286$) con downsampling, permutación de los datos y optimización de hiperparámetros.	63

-
- 5.14. Matriz de confusión resultante de la clasificación de las señales de ECG y HRV con un kernel gaussiano con $BoxConstraint = 77,0683$ y $KernelScale = 59,6831$ con downsampling, permutación de los datos y optimización de hiperparámetros. 64

Lista de tablas

3.1.	Ejemplo de filas del fichero samples.csv que nos da información básica sobre los pacientes	24
3.2.	Ejemplo de filas de un fichero de biopotenciales, donde para cada muestra recogida en el experimento, se da su instante temporal y el valor en mV de las señales.	24
3.3.	Ejemplo de filas de un fichero de la señal de nieles de dolor, que indica en cada momento temporal el nivel de dolor al que está sometido el paciente, siendo 0 el nivel mínimo y 4 el nivel máximo.	25
4.1.	Tabla de bloques posibles y coincidentes para el cálculo de la entropía aproximada sobre la serie $U'[n]$	36
5.1.	Ejemplo de filas del fichero de datos que se va a emplear tras el preprocesado. Contiene en cada fila una muestra del ECG de un paciente determinado y su nivel de dolor correspondiente.	50
5.5.	Ejemplo de filas pertenecientes a la matriz de características para el sistema de clasificación, que se usan como los distintos vectores de datos x_i para el entrenamiento y la inferencia.	54
5.7.	Ejemplo de filas pertenecientes a la tabla de características de los segmento, donde Nivel de dolor es la clase objetivo en el sistema de clasificación automática y	54
5.8.	Ejemplo de muestras de señales de HRV correspondientes a una ventana	55
5.9.	Ejemplo de muestras de señales de ECG correspondientes a una ventana	56
5.10.	Resultados de los distintos experimentos de clasificación. .	61

Listado de código fuente

1.	Implementación de SDNN	73
2.	Implementación de SDRR	73
3.	Implementación de RMSSD	73
4.	Implementación de pnn50	74
5.	Implementación del análisis de la densidad espectral de potencia del ECG	75
6.	Implementación de la entropía de Shannon	75
7.	Implementación de la entropía espectral	77
8.	Implementación de la entropía aproximada	80
9.	Implementación de la entropía muestral	83
10.	Implementación de la entropía de permutación	84
11.	Implementación de clasificación de segmentos en función de la presencia de dolor mediante SVM	91

Resumen

El objetivo principal de este trabajo es el estudio y extracción de características para la clasificación de dolor mediante la caracterización del electrocardiograma. Esto permitirá, en un futuro, establecer un procedimiento objetivo para el diagnóstico del dolor, de forma que podamos mejorar su tratamiento clínico, ya que en la actualidad, el diagnóstico del dolor se basa en las sensaciones de los pacientes. Por último, se realizan las pruebas de los algoritmos de extracción de características sobre la base de datos BioVid para, posteriormente, emplear estas en el entrenamiento de un clasificador mediante máquinas de vectores soporte y aportar una idea acerca de la efectividad de los parámetros extraídos.

Palabras clave: ECG, HRV, dolor, entropía, SVM.

Abstract

The main objective of this work is the study and extraction of parameters from the electrocardiogram signal for the classification of pain. This will allow, to establish an objective system for the diagnosis of pain, with the goal of improving its treatment, since at the moment it's based on the feelings of the patients. Finally, the algorithms will be tested using the BioVid database, and then, the extracted parameters will be used to design a classifier using support vector machines to test the effectiveness of the extracted parameters.

Keywords: ECG, HRV, pain, entropy, SVM.

Resumen extendido

Este trabajo intenta detectar la presencia de dolor a partir de la señal del electrocardiograma. Esta detección objetiva a partir de indicadores biológicos es deseable, ya que ahora mismo el diagnóstico del dolor es subjetivo, basado en las sensaciones del paciente.

Aunque los mecanismos biológicos que gobiernan el dolor no se entienden en su totalidad, hay amplia evidencia científica de que el dolor se capta gracias a unos sensores que forman parte del sistema nervioso autónomo, cuya actividad también afecta al ritmo con el late el corazón, que se puede observar de manera no invasiva con el electrocardiograma.

Por tanto, la hipótesis principal sobre la que trabajamos es que es posible detectar la presencia de dolor mediante el estudio de la señal de variabilidad cardiaca, que obtenemos a partir del electrocardiograma.

Para estudiar esta hipótesis hemos realizado una caracterización de la señal de electrocardiograma y de variabilidad cardiaca mediante la extracción de características temporales, espectrales y de entropía usadas para su análisis tal y como se describe en la literatura [25].

También describimos y probamos los algoritmos de extracción de características sobre la base de datos BioVid, que contiene mas de 1500 minutos de electrocardiogramas de pacientes cuando están sufriendo dolor.

Por último, entrenamos un clasificador basado en máquinas de vectores soporte sobre esta base de datos, con el cual intentamos clasificar segmentos de electrocardiograma en función de la presencia de dolor para comprobar la capacidad de discriminación de las características extraídas.

Lista de acrónimos

ApEnt Entropía aproximada.

BioVid Base de datos Biopotencial and Video Heat Pain Database).

CO₂ Dióxido de carbono

ECG Electrocardiograma.

EEG Electroencefalograma.

EMG Electromiografía (actividad eléctrica muscular).

FN Falso negativo.

FP Falso positivo.

GSR Actividad eléctrica de la piel.

HF Componentes espectrales de alta frecuencia.

HRV Heart Rate Variability o Señal de variabilidad cardiaca.

LF Componentes espectrales de baja frecuencia.

NN Distancia entre dos complejos QRS normales.

NN50 Número de intervalos NN mayores que 50 ms.

P Onda del ECG que refleja la despolarización auricular, que se manifiesta al inicio del complejo QRS.

PermEnt Entropía de permutación.

pNN50 Porcentajes de intervalos NN mayores que 50 ms.

Pot Potencia.

PQ Intervalo entre el comienzo de la onda P y el final de la onda T en la señal de ECG.

PR Distancia entre el inicio de la onda P y el inicio del complejo QRS en la señal de ECG.

Q Onda intermedia del complejo QRS.

QRS Conjuntos de ondas del ECG que reflejan la despolarización ventricular.

QT Intervalo entre el comienzo de la onda Q y el final de la onda T en la señal de ECG.

ROC Área bajo la curva.

RMSSD Valor cuadrático medio de los intervalos temporales de la serie NN:

RR Distancia entre dos complejos QRS consecutivos en la señal de ECG.

S Onda final del complejo QRS.

SamEnt Entropía muestral.

SDNN Media de los instantes temporales de la serie NN.

SDRR Media de los instantes temporales de la serie NN:

ShEnt Entropía de Shannon.

SNA Sistema Nervioso Autónomo

SNP Sistema Nervioso Periférico

SNC Sistema Nervioso Central

SNS Sistema Nervioso Simpático

SpectalEnt Entropía espectral.

ST Distancia entre complejo QRS y la onda T en la señal de ECG.

SVM Maquina de vector soporte.

T Onda del ECG que refleja la repolarización ventricular.

TN Elemento de la clase negativa (True negative).

TP Elemento de la clase positiva (True positive).

U Onda del ECG que representa las derivaciones precordiales, que aparecen después de la onda T.

VHF Componentes espectrales de muy alta frecuencia.

VLF Componentes espectrales de muy baja frecuencia.

Lista de símbolos

X Serie temporal.

x_n Señal.

x_i i -ésimo elemento de la señal x_n .

f_i i -ésima componente frecuencial de una señal x_n .

mV milivoltios.

ms milisegundo.

V Voltios.

Hz Hertzios.

$H(z)$ Función de transferencia de un filtro.

$RR[n]$ Intervalos RR.

$NN[n]$ Intervalos NN.

$d_{IF}^u(t)$ Señal muestreada no uniformemente.

$\delta(t - k)$ Delta de Dirac en el instante k .

\mathbb{E} Evento de una distribución discreta de probabilidad

$\mathbb{I}(\mathbb{E})$ Cantidad de información de un evento.

$\sigma(x)$ Desviación típica

$P(x_i)$ Probabilidad de un evento x_i

$E[\mathbb{X}]$ Esperanza matemática de \mathbb{X}

$H(\mathcal{X})$ Entropía de la señal \mathcal{X}

p_i Frecuencia relativa de un evento i .

\forall Para todo.

\in Contenido en.

C_i^m Frecuencia relativa del patrón i con longitud m en las entropías aproximada y muestral.

$\phi^m(r)$ Función ϕ de los patrones con longitud m en las entropías aproximada y muestral.

d' Número de patrones que cumplen la condición de distancia en las entropías aproximada y muestral.

$d[u_i, u_j]$ Distancia entre x_i y x_j

R Vector de rangos, en entropía de permutación.

$|\cdot|$ Valor absoluto.

$\max_{k=1, \dots, n}$ máximo de los números entre 1 y n .

π_i Permutación i -ésima.

$D(x)$ Hiperplano en el espacio.

$\|\cdot\|$ Valor absoluto.

$\langle x_i, x_j \rangle$ Producto escalar

ϵ_i Variables de holgura de las máquinas de vectores soporte.

\mathcal{X} Espacio base de las máquinas de vectores soporte.

\mathbb{F} Espacio de características de las máquinas de vectores soporte.

R^d Espacio de d dimensiones, que necesita d componentes para definir un punto o un vector.

$\Phi(x)$ Función de transformación de espacio base a espacio de características.

$K(x_i, x_j)$ Función kernel.

γ Parámetro de anchura del kernel gaussiano en las máquinas de vectores soporte.

Capítulo 1

Introducción

1.1. Motivación

La RAE define el dolor, uno de los grandes problemas médicos del primer mundo, como una *sensación molesta y aflictiva de una parte del cuerpo por causa interior o exterior*,

Clínicamente, diagnosticar el dolor es complicado, ya que no solo depende de sensaciones físicas, sino también de circunstancias emocionales, por lo que la forma más empleada para el diagnóstico es mediante el uso de las sensaciones del paciente de forma subjetiva, sin ningún criterio estandarizado

Por tanto, es deseable encontrar una forma objetiva para diagnosticar la presencia de dolor en los pacientes (pensemos por ejemplo en pacientes incapaces de comunicarse con sus facultativos) a partir de señales biomédicas.

El dolor se capta y transmite al cerebro mediante una parte del sistema nervioso llamada sistema nervioso autónomo (SNA), que a su vez, modifica el ritmo con el que late el corazón, efecto que se puede estudiar de forma no invasiva mediante un electrocardiograma (ECG). Por tanto, teorizamos que se puede detectar el efecto de la presencia de dolor mediante el análisis de la variabilidad cardiaca. Basamos también esta teoría en [1], donde se ajustan modelos de regresión a los niveles de dolor de los pacientes tras una operación con anestesia local frente a distintos parámetros obtenidos del ECG.

Por tanto, en este trabajo vamos a extraer distintas características del

electrocardiograma e intentaremos, a partir de estas, diseñar un sistema de clasificación que permita, determinar si están sintiendo dolor.

El principal campo de aplicación de este trabajo es el futuro diseño e implementación de sistemas para la clasificación objetiva del dolor en seres humanos, ya que en la actualidad, su estudio se basa en una escala subjetiva y personal del paciente, lo que permitirá mejorar la atención que reciben los pacientes por parte de los facultativos.

1.2. Objetivos

El objetivo principal de este trabajo es el estudio de características de la señal de variabilidad cardiaca para su aplicación en sistemas de aprendizaje automático orientados a la caracterización de dolor y la validación de estos métodos mediante la aplicación a un test sintético.

Para alcanzar este objetivo general, se proponen los siguientes objetivos específicos:

OE1. Estudio de las señales de electrocardiograma y variabilidad cardiaca.

OE2. Estudio del estado de la técnica relacionado con el análisis del electrocardiograma.

OE3. Desarrollo de algoritmos basados en entropía y otras técnicas de análisis no lineales.

OE4. Simulaciones sobre series temporales y aplicación a la señal de variabilidad cardiaca.

Para cumplir con los objetivos **OE1** y **OE2**, se ha realizado una familiarización con la bibliografía relevante dedicada al ECG y a su estudio como, por ejemplo, [31], [27], [1], [7], [14] y [25]

Para cumplir con los objetivos **OE3** y **OE4**, se han programado en MATLAB distintos algoritmos para el análisis de las señales del ECG, que luego se han empleado para la extracción de características de la base de datos BioVid, con el objetivo de validar su funcionamiento.

- Se han estudiado las características temporales de la señal de variabilidad cardiaca mediante el cálculo de parámetros como el SDNN, del SDRR y el RMSSD.
- Se ha realizado un análisis de las características espectrales del ECG, mediante el estudio de la distribución de la densidad espectral de potencia.
- También se han usado múltiples métodos no lineales para estudiar la regularidad del ECG y de la señal de variabilidad cardiaca (HRV), como la entropía muestral o la entropía de permutación

Sobre el desarrollo de estos objetivos es importante, recalcar que en este trabajo nos hemos limitado a estudiar e implementar aquellos parámetros y algoritmos que pueden dar resultados significativos sobre la base de datos de la que disponemos, por lo que no hemos estudiado técnicas, que requieran señales de ECG de media o larga duración (con una duración mayor a 5 minutos).

Por último, se ha desarrollado un clasificador a partir de las características extraídas con el objetivo de clasificar segmentos del ECG de la base de datos BioVid en función de la presencia de dolor.

1.3. Metodología

Para la realización de este trabajo se han estudiado e implementado distintos algoritmos en MATLAB para el análisis de características del ECG y del HRV, y posteriormente, estos se han probado sobre las señales de biopotenciales disponibles en la base de datos BioVid [28], que contiene el ECG de 87 pacientes, a los que se aplica dolor en cortos periodos de tiempo, con intensidad aleatoria durante 25 minutos.

Por último, para comprobar la hipótesis fundamental se han empleado estas características para el desarrollo de un clasificador basado en máquinas de vectores soportes (SVM), con el fin de intentar detectar la presencia de dolor en los segmentos de la señales de ECG.

1.4. Estructura de la memoria

Este trabajo sigue la siguiente estructura:

- En el capítulo 2, antecedentes biomédicos, introducimos los conocimientos previos sobre la biología humana y las señales bajo estudio, necesarios para el desarrollo del resto del trabajo.
- En el capítulo 3, base de datos, introducimos la base de datos BioVid, que usamos para probar los métodos que se han desarrollado durante la realización del trabajo.
- En el capítulo 4, métodos matemáticos, explicamos todos los algoritmos que se han empleado para realizar la extracción de características y la clasificación de los segmentos de las señales.
- En el capítulo 5, tratamiento de los datos, describimos como se ha realizado el acondicionamiento de la base de datos, el procedimiento y resultados de la extracción de características de las señales y los resultados de los experimentos de clasificación que hemos realizado.
- Por último, en el capítulo 6, conclusiones, analizamos los resultados obtenidos durante la realización de este trabajo y proponemos posibles líneas de trabajo futuras.

Capítulo 2

Antecedentes biomédicos

En este capítulo vamos a introducir las distintas señales y procesos biológicos sobre las que se basan nuestro estudio posterior.

El sistema nervioso está altamente relacionado con el corazón y con la sensación de dolor, ya que el dolor se capta y se transmite por el sistema nervioso, y la actividad de este influye en el ritmo cardiaco, que podemos estudiar mediante el uso del ECG. De esta forma es posible ver la actividad del sistema nervioso, y por tanto del dolor, usando una prueba barata y no invasiva.

2.1. El sistema nervioso

En esta sección se introduce una pequeña descripción del sistema nervioso basada en [29]

El sistema nervioso es el principal mecanismo de control, regulación y comunicación del cuerpo humano. Es el sistema responsable de toda actividad mental, incluyendo funciones tan dispares como el aprendizaje y la transmisión de sensaciones corporales.

El sistema nervioso está compuesto por el cerebro, la columna espinal, los ganglios y los nervios (que a su vez están compuestos de neuronas y células gliales). Todos estos elementos trabajan de forma conjunta para cumplir las funciones sensitivas, integradoras y motoras del sistema nervioso.

En el sistema nervioso, millones de sensores detectan cambios en el entorno, o el interior del cuerpo, produciendo estímulos, que son transfor-

mados en señales eléctricas. Todas estas señales se transportan al cerebro, donde se emplean para tomar decisiones, que se envían, a su vez, en forma de estímulos, de nuevo a los músculos o a las glándulas, para generar una respuesta.

El sistema nervioso se divide entre el sistema nervioso central (SNC) y el sistema nervioso periférico (SNP):

- El SNC está formado por el cerebro y la médula espinal. El SNC controla la mayor parte de las funciones del cuerpo transmitiendo los mensajes entre el cerebro y el resto del cuerpo de forma bidireccional.
- El SNP consiste en un sistema complejo de neuronas motoras (que estimulan a los músculos para realizar funciones voluntarias o reflejas), neuronas sensoriales (que perciben sensaciones como el dolor o el calor), ganglios (grupos de neuronas) y nervios. Este sistema está conectado al SNC y regula sus funciones.

El SNP a su vez está dividido en:

- Sistema nervioso somático o simpático (SNS) que controla los movimientos de la musculatura voluntaria y se encarga de procesar la información sensorial.
- Sistema nervioso autónomo (SNA) que controla las funciones corporales involuntarias como el ritmo cardíaco y la digestión.

2.2. La sensación de dolor

En esta sección procedemos a describir brevemente los mecanismos biológicos asociados a la sensación de dolor y al estudio de su intensidad basándonos en [20] y [13].

El dolor es una función vital del sistema nervioso al proporcionar al cuerpo una advertencia ante lesiones reales o potenciales. Es una sensación tanto sensorial como emocional, y está influido por factores psicológicos como pueden ser la ansiedad, el miedo, experiencias pasadas, o creencias sobre el dolor. Aunque la ciencia moderna no entiende al completo

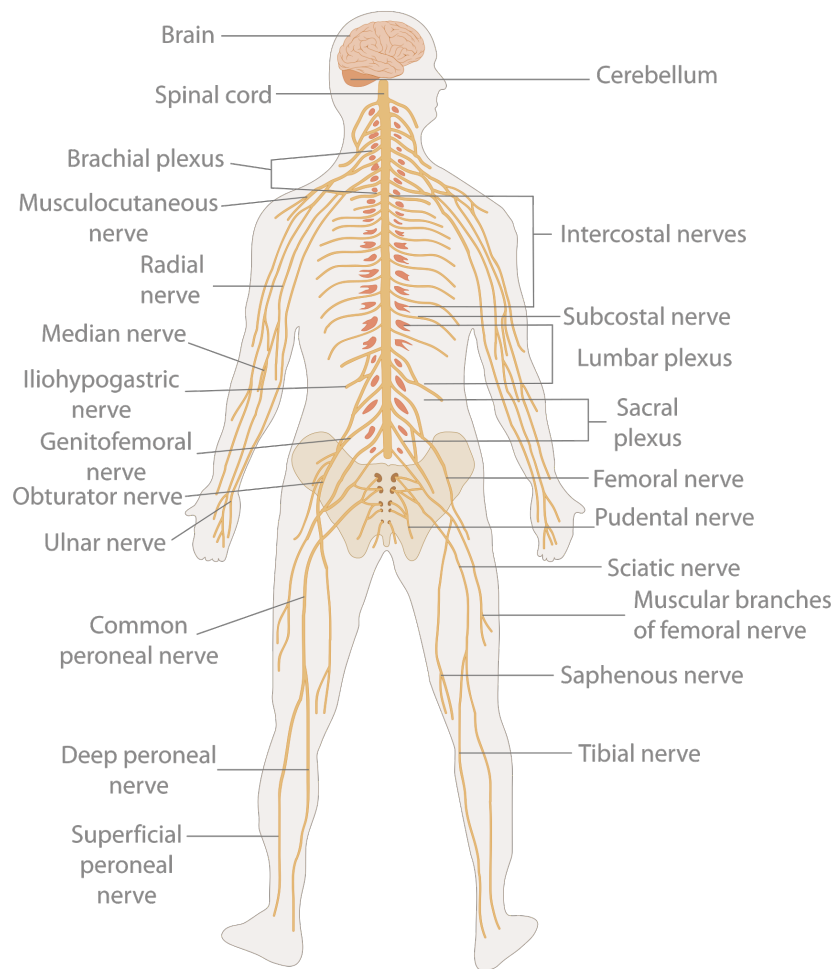


Figura 2.1: Sistema nervioso humano. Imagen de [36]

los mecanismos asociados con la producción y la presencia de dolor, sin embargo se sabe que los tres tipos de dolor (neuropático, nociceptivo y psicológico) se regulan mediante el sistema nervioso [3].

El dolor se detecta mediante los nociceptores que son los receptores sensoriales especializados situados en el SNA, responsables de la detección de estímulos nocivos transformando estos estímulos sensoriales en señales eléctricas, que luego, se transmiten al sistema nervioso central y al cerebro. Estos receptores están distribuidos por todo el cuerpo (piel, órganos, músculos y articulaciones) y pueden ser activados por estímulos de tipo mecánico, térmico o químico.

Al tener el dolor una componente psicológica muy importante, la presencia e intensidad del dolor se mide de una manera subjetiva, depen-

diendo de las sensaciones y las indicaciones del paciente, mediante una escala de intensidad de dolor. La escala más usada es una escala visual y numérica en la que se le solicita al paciente que describa la intensidad del dolor y le asigne una puntuación de 0 a 10, donde 0 indica la ausencia de dolor y 10 refleja el máximo dolor que el paciente se pueda imaginar.

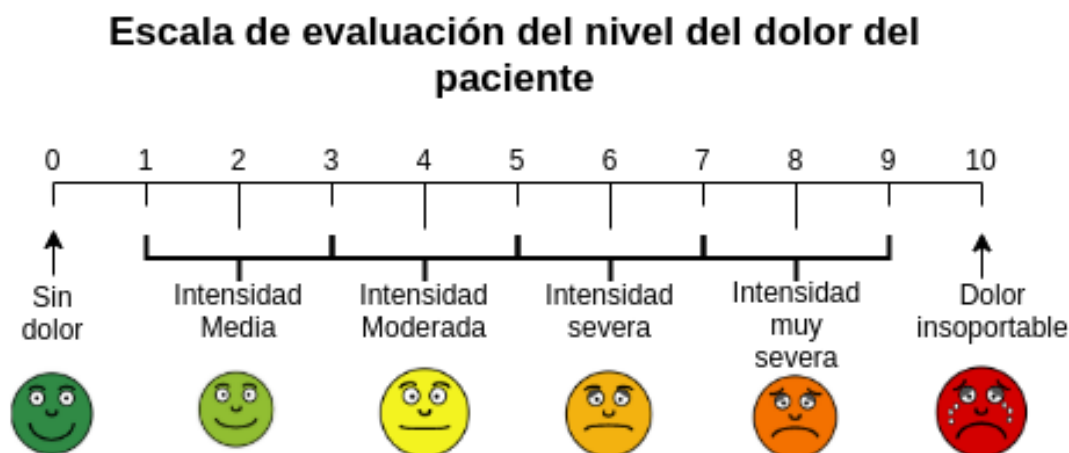


Figura 2.2: Escala visual de intensidad de dolor.

2.3. El electrocardiograma

En este apartado vamos a describir el funcionamiento mecánico y eléctrico del corazón y cómo se refleja este en la señal ECG basándonos en [27]

El electrocardiograma o ECG describe la actividad eléctrica del corazón grabada por electrodos situados en la superficie del cuerpo. Las variaciones del voltaje medidas por los electrodos son causadas por las variaciones de potenciales de acción de las células del corazón cuando se producen las diferentes fases del ciclo cardíaco. La información resultante de esta excitación se manifiesta en el ECG como una serie de ondas, cuya morfología y temporización dan información que puede ser usada para diagnosticar enfermedades. El número de electrodos que se colocan en el paciente dependen del tipo de información clínica deseada, aunque, normalmente es suficiente usar pocos electrodos cuando solo se estudia el ritmo del corazón, mientras que se necesitan hasta 10 electrodos cuando

se necesita estudiar la forma de onda del ECG.

2.3.1. Actividad electromecánica del corazón

El corazón es un órgano musculoso cuya función principal es el bombeo de sangre oxigenada por el cuerpo. Su anatomía está dividida en 2 lados, que se encargan de diferentes sistemas circulatorios, pero que funcionan de una forma rítmica y sincronizada. Los dos lados del corazón consisten en dos cámaras separadas por el tabique, la aurícula, donde entra la sangre al corazón y el ventrículo desde donde la sangre se expulsa al resto del cuerpo. El flujo de sangre se controla mediante cuatro válvulas, dos localizadas entre las aurículas y los ventrículos y otras dos entre los ventrículos y las arterias. El miocardio es la pared del corazón y está compuesto por células musculares que producen fuerza mecánica durante la contracción del corazón y por células conductoras eléctricas que forman una red que permite que los impulsos eléctricos se transmitan rápidamente por el corazón.

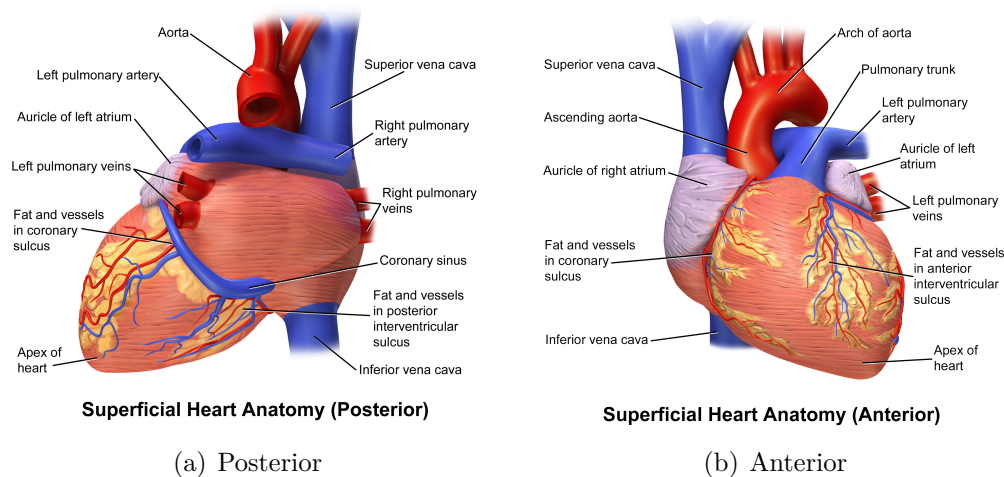


Figura 2.3: Estructura física del corazón. Imágenes de [6] y [5]

Un ciclo cardiaco se inicia cuando un impulso eléctrico se propaga por el sistema de conducción, y da comienzo a la contracción mecánica, que empieza en la aurícula derecha donde la sangre de todo el cuerpo se recolecta. Cuando la aurícula derecha recibe un impulso eléctrico se contrae, tras lo cual la sangre pasa al ventrículo derecho, que se contrae

cuando está lleno para mandar la sangre a los pulmones

En los pulmones se produce el intercambio de CO₂ por oxígeno, y las venas pulmonares devuelven la sangre oxigenada a la aurícula izquierda, desde donde la sangre pasa al ventrículo izquierdo, el cual la bombea a todo el cuerpo.

Cada ciclo cardiaco, si hablamos en términos de actividad eléctrica, está compuesto de 2 fases: despolarización y repolarización, que son equivalentes a la contracción y relajación en términos mecánicos. La despolarización es un cambio del potencial de la membrana de la célula (-90 a 20 mV en 1 ms). Este rápido cambio hace que las células adyacentes se despolaricen. con lo que se forma una corriente eléctrica por el miocardio. Posteriormente se produce la repolarización, fase en la que las células recuperan su potencial original.

El nodo sinoatrial es el marcapasos automático del corazón que determina la frecuencia de los latidos, dado que genera la corriente eléctrica del corazón y, por tanto, controla la contracción y relajación de la aurículas y los ventrículos.

La frecuencia de generación de pulsos por parte del nodo sinoatrial no solo depende de su velocidad de descarga inherente (su "frecuencia de reloj"), sino que, también se ve influida por información transmitida por el sistema nervioso autónomo. El equilibrio entre el sistema simpático y parasimpático del sistema nervioso autónomo afectan al ritmo cardiaco, de forma que un incremento en actividad parasimpática disminuye el ritmo cardiaco, mientras que un incremento en la actividad simpática lo aumenta.

El ECG caracteriza la actividad eléctrica del corazón mediante medidas adquiridas en la superficie del cuerpo y describe las diferentes fases eléctricas del ciclo cardiaco representando una suma en tiempo y espacio de los potenciales de acción generados por millones de células cardiacas. El ECG, en vez de representar directamente cambios en el potencial de las células, da una medida de la corriente eléctrica generada en el fluido extracelular por estos cambios de potencial. Es importante notar que las fases de polarización y despolarización se generan como un desvío de un nivel de base, correspondiente al estado de reposo de las células.

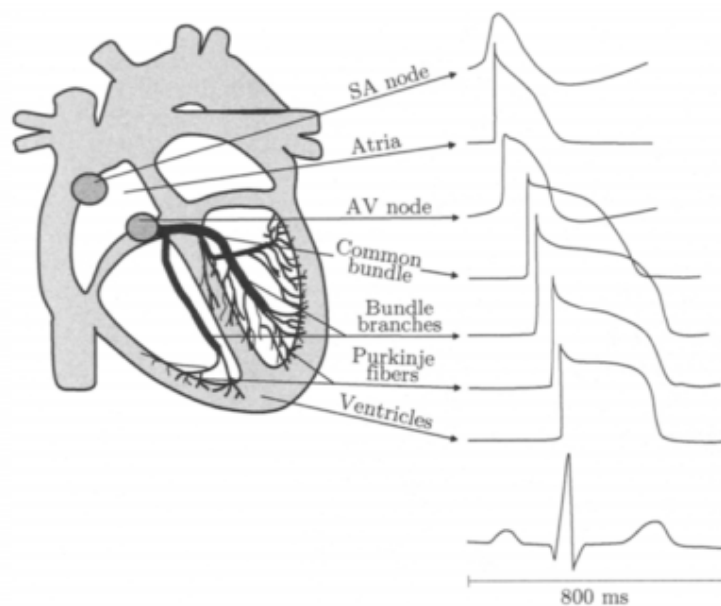


Figura 2.4: Componentes de onda del ECG producidas por cada parte del corazón. Imagen de [27]

De los millones de células individuales en el corazón, solo algunos grupos en el miocardio conducen electricidad en un instante determinado y su despolarización puede representarse como un dipolo equivalente con un vector asociado. El conjunto de estos vectores pueden ser resumidos en un vector dominante, el cual describe la dirección principal del impulso eléctrico, que se ve reflejado en la potencia de la señal captada por los distintos electrodos (dependiendo de su posición en el torso).

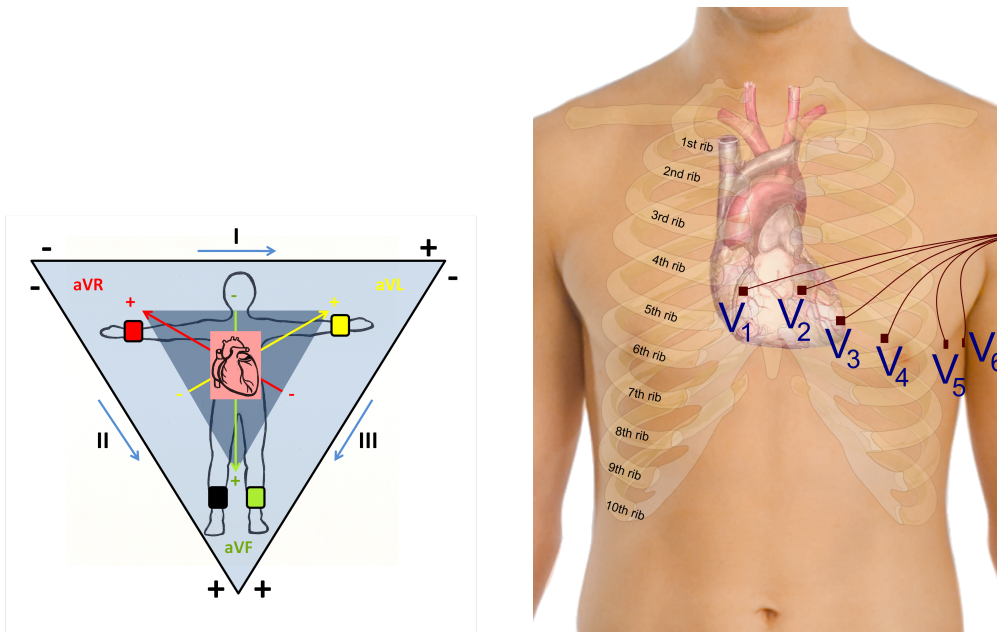
2.3.2. Medición de la actividad eléctrica del corazón

La actividad eléctrica del corazón se capta mediante electrodos colocados en la superficie del cuerpo, entre los que se mide la diferencia de potencial, formando derivaciones. El ECG se graba usando una configuración de múltiples derivaciones que incluye derivaciones de ambos tipos: unipolar y bipolar.

- Una derivación unipolar refleja la variación de voltaje de un solo electrodo y se mide respecto al electrodo de referencia, situado normalmente de manera que el voltaje permanezca casi constante durante

todo el ciclo cardiaco.

- Una derivación bipolar mide la diferencia de voltaje entre dos electrodos, por ejemplo, entre los brazos izquierdo y derecho.



(a) Triángulo de Einthoven, que muestra la posición de los electrodos en las extremidades y las derivaciones obtenidas a partir de estos.

(b) Localización de los electrodos en el torso humano

Figura 2.5: Sistema de 12 derivaciones. Imágenes de [16] y [2]

Los electrodos están conectados a un amplificador diferencial, con gran ganancia y alto rango dinámico, ya que la señal tiene niveles en el rango de pocos microvoltios a 1V, y una componente continua de hasta 1V. El ancho de banda del amplificador suele estar entre 0.05 y 100-500 Hz, dependiendo de la aplicación.

Actualmente existen varios sistemas de derivaciones, donde la posición de las electrodos está estandarizada, aunque en nuestro caso solo discutiremos el sistema de 12 derivaciones, que es el más usado en aplicaciones clínicas. Este sistema se basa en la combinación de tres derivaciones: la derivación bipolar de las extremidades, la derivación mejorada de las extremidades, y la derivación unipolar de la región precordial, que se graban mediante la colocación de 10 electrodos en posiciones predefinidas.

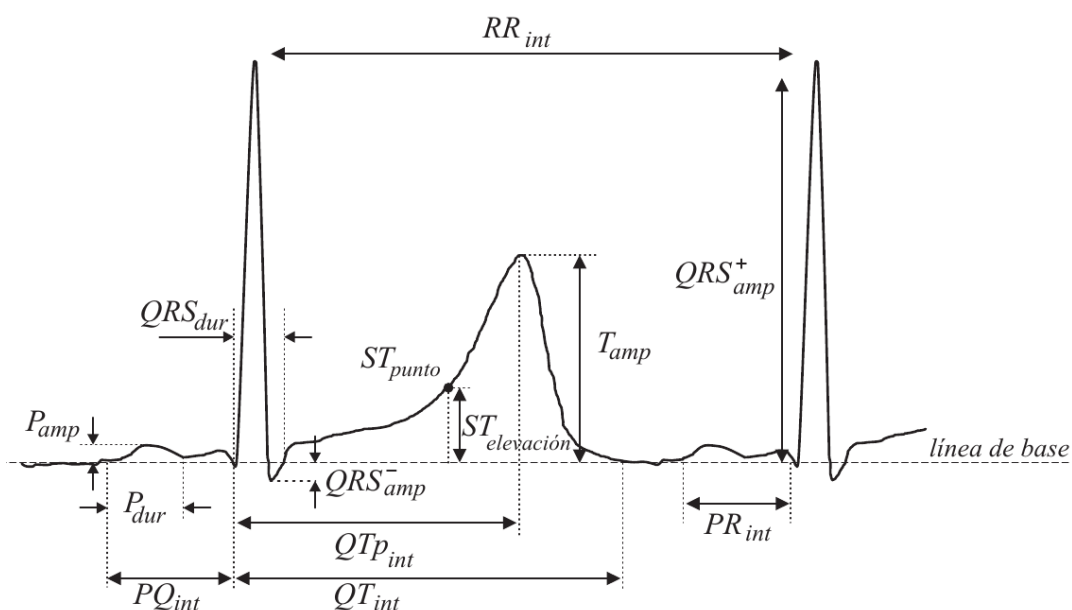


Figura 2.6: Estructura de la señal de ECG, donde int representa intervalo, amp es amplitud y dur duración. Imagen de [31]

2.3.3. La señal de ECG y sus intervalos temporales

Ahora pasamos a describir las características fundamentales del ECG, sus principales componentes y la relación de estas con la actividad mecánica del corazón.

La amplitud del ECG se define en referencia a la componente continua de la señal, que normalmente es la amplitud antes del complejo QRS, mientras que, su periodo se mide entre dos instantes de tiempo definidos por la intersección de la señal con la línea de base (la componente continua).

Las principales partes en las que se divide la señal de ECG son las siguientes:

- La onda P refleja la despolarización en secuencia de las aurículas derecha e izquierda, y normalmente se manifiesta, con una polaridad positiva, baja amplitud y una forma de onda relativamente plana. En el espectro se manifiesta en las bajas frecuencias de 10 a 15 Hz.
- El complejo QRS refleja la despolarización de los ventrículos, y suele durar entre 70 y 110 ms. La primera parte del complejo QRS, en la que se produce una pequeña bajada se llama onda Q, la primera

pendiente positiva se llama onda R, mientras que la pendiente negativa detrás de esta se llama onda S. El complejo QRS presenta la amplitud más grande del ECG (2-3 mV) y es la primera parte a identificar en los análisis. Sus componentes frecuenciales son elevadas, ya que la forma de onda presenta una alta pendiente, por lo que suele encontrar en la franja de 10 a 50 Hz.

- El segmento ST no es una onda, pero representa el intervalo de tiempo en el que los ventrículos permanecen en estado activo, pero despolarizados. El segmento ST comienza al final de la onda S (el punto J) y permanece con un voltaje más o menos constante, hasta que la señal se curva para dar paso a la onda T.
- La onda T refleja la repolarización ventricular y dura unos 300 ms después del complejo QRS. La posición de la onda T depende de la frecuencia cardíaca y suele tener una forma de pico con pendiente positiva. A veces se observa otra onda de baja frecuencia, llamada onda U, que indica la post-repolarización ventricular. La onda T a veces se mezcla con la onda P, sobre todo en situaciones con alta frecuencia cardíaca, lo que puede dificultar la discriminación, y por tanto el análisis de ambas.

Aparte de las ondas individuales de las que se compone el ECG se definen una serie de intervalos:

- El intervalo RR, representa la longitud de un ciclo cardíaco ventricular, medido entre dos ondas RR sucesivas, y sirve como indicador de la frecuencia cardíaca ventricular.
- El intervalo PQ es el intervalo definido desde el comienzo de la despolarización atrial, hasta el comienzo de la despolarización ventricular. El intervalo PQ refleja el tiempo requerido para que el impulso generado en el nodo sinoatrial se propague a los ventrículos, y depende de la frecuencia cardíaca.
- El intervalo QT representa el tiempo desde el inicio de la despolarización ventricular, hasta la finalización de la repolarización ventri-

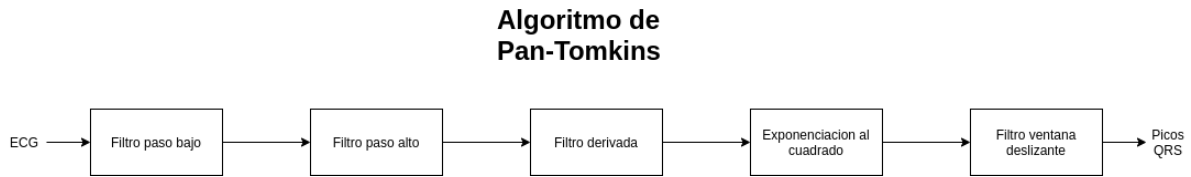


Figura 2.7: Descripción funcional del algoritmo de Pan-Tomkins.

cular. Varía con la frecuencia cardiaca, volviéndose mas corto con frecuencias más altas.

2.4. Señal de variabilidad cardiaca

Ya que el SNA influye en la frecuencia de los latidos, mediante la modificación del ritmo de los impulsos producidos por el nodo sinuatrial, para estudiar su actividad es interesante fijarnos en como cambia el ritmo cardiaco a lo largo del tiempo, y para ello usamos la señal de variabilidad cardiaca (HRV por sus siglas en ingles).

El tacograma se define como la fluctuación de la duración de los intervalos RR y para su obtención se parte de la señal de ECG digitalizada y se realiza una detección de picos del complejo QRS usando un algoritmo como el de Pan-Tomkins, cuyo funcionamiento se puede ver en las figuras 2.7 y 2.8.

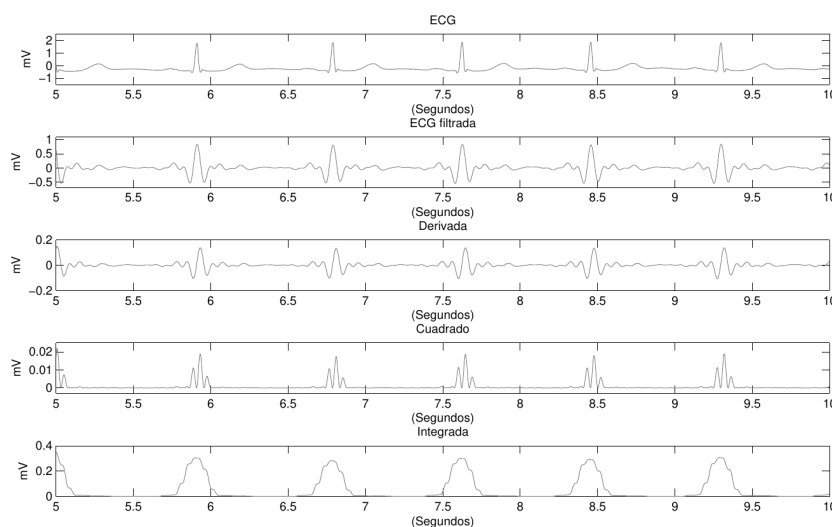


Figura 2.8: Procesado de la señal de ECG en el algoritmo de Pam-Tomkins. Imagen de [32]

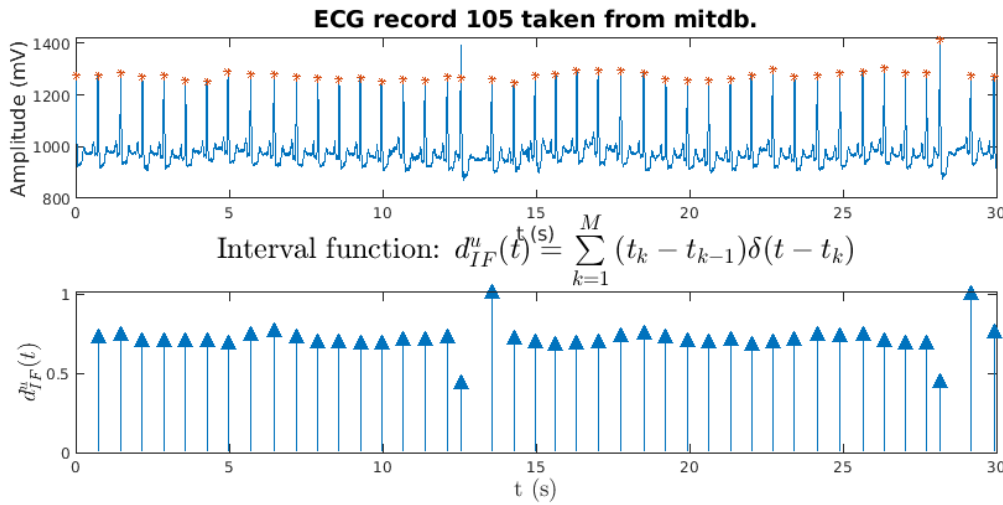


Figura 2.9: ECG y señal de intervalo tras su muestreo.

El algoritmo de Pan y Tomkins [18], comienza el proceso de la detección de los picos mediante un filtro paso banda, compuesto de dos fases un filtro paso bajo con función de transferencia $H(z) = \frac{(1-z^{-6})^2}{(1-z^{-1})^2}$ y otro paso alto con función de transferencia $H(z) = \frac{(-1/32+z^{-16}-z^{-17}+z^{-32}/32)}{(1-z^{-1})}$.

Luego se deriva la señal resultante del paso por los filtros para obtener información sobre la pendiente usando para ello un filtro con función de transferencia $H(z) = 0,1(-2z^{-2} - z^{-1} + z^1 + 2z^2)$, y después se eleva la señal resultante al cuadrado para realizar el complejo QRS, ya que este tiene una amplitud elevada con signo positivo.

Por último, se integra mediante un filtro de ventana deslizante, con una función de transferencia que tiene un número de coeficientes variable dependiente de la frecuencia de muestreo del ECG, y cuya función es: $y(nT) = (1/N)[x(nT - (N - 1)T) + x(nT - (N - 2)T) + \dots + x(nT)]$, (en nuestro caso al tener señales de ECG muestreadas a 512 Hz, es de $N = 30$ coeficientes).

Todo este proceso da como resultado una señal de instantes temporales, indicando la posición de los picos en la señal de ECG en el dominio del tiempo, llamada tacograma, y definida como la señal de variaciones de los sucesivos intervalos RR de forma discreta $Tacograma[n] = RR[n] - RR[n - 1]$.

Para dotar a la señal del dominio temporal, se puede representar en

términos de funciones impulso, con lo que se define la función de intervalo en la cual se representa el pico del complejo QRS que ocurre en un instante t_k como una delta de Dirac con una amplitud escalada por la longitud del intervalo RR anterior $d_{IF}^u(t) = \sum_{k=0}^{M-1} (t_k - t_{k-1})\delta(t - t_k)$. Esta expresión puede interpretarse como el resultado de una señal muestreada no uniformemente, en donde las muestras representan a la señal de variabilidad cardiaca.

El problema con la señal de intervalo obtenida, tal y como se ve en la figura 2.9 donde se muestra un ejemplo de la transformación desde una señal de ECG a una señal de intervalo, es que esta señal no tiene un muestreo uniforme, por lo que es necesario remuestrearla de forma uniforme [8] para poder aplicar técnicas de procesamiento digital, como por ejemplo, técnicas de análisis espectral y de entropía. Con esta transformación obtenemos finalmente, la señal de variabilidad cardiaca, sobre la que vamos a realizar gran parte de nuestro estudio.

Capítulo 3

Base de datos

En este capítulo vamos a describir la base de datos sobre la que vamos a probar los distintos algoritmos de extracción de características del ECG para una futura clasificación dependiente de la presencia de dolor en el paciente.

La base de datos sobre la que se ha trabajado es BioVid (acrónimo de Biopotencial and Video Heat Pain Database), que se describe con detalle en [33], [34] y [35]

En esta se recogen señales de vídeo de la cara de los pacientes junto con diversos biopotenciales. Estos últimos presentan una duración de 25 minutos durante los cuales se somete a los pacientes a 4 niveles de dolor durante cortos periodos de tiempo. Las señales de biopotenciales fueron adquiridos con un dispositivo Nexus-32, que permite grabar simultáneamente y con alta resolución (2 kHz con 24 bits) varias señales de biopotenciales, mientras que las señales de vídeo del rostro se grabaron con 3 cámaras F145C y una cámara Kinect de Microsoft.

En el diseño de nuestro experimento, se han recopilado los datos de 90 personas repartidas homogéneamente entre ambos sexos y con 3 grupos de edad, cada uno con 30 sujetos: de 18 a 35 años, de 36 a 50 años, y de 51 a 65 años.

La base de datos BioVid se compone de las siguientes partes:

- Parte A: estimulación de dolor sin gorro de electroencefalograma (EEG), con señales de biopotenciales filtradas y divididas en periodos cortos.

- Parte B: estimulación de dolor con gorro de EEG, con señales de biopotenciales filtradas y divididas en periodos cortos.
- Parte C: estimulación de dolor sin gorro de EEG, con señales de biopotenciales en periodos largos y sin filtrar.
- Partes D: registro de las emociones básicas en periodos de un minuto.
- Parte E: elicitación de emociones mediante vídeos, mientras se graban distintas señales de biopotenciales asociadas a esas emociones.

En este trabajo nos centramos en las señales de biopotenciales de la parte C de la base de datos, que contiene 87 secuencias de distintos pacientes, con una duración de 25 minutos, y en particular, estamos interesados en el análisis de la señal ECG, ya que inspirándonos en el análisis en [1], consideramos esta señal la principal en nuestro estudio.

3.1. Señales de biopotenciales de la base de datos

Las señales de la base de datos están muestreadas a una frecuencia de 512 Hz y, son las siguientes:

- Conductividad eléctrica de la piel (GSR): la conductividad de la piel se ve modificada por el sudor, en cuya presencia influye la actividad del sistema nervioso simpático, por lo que este parámetro es un buen indicador del estrés de un paciente. Para medirlo se han usado 2 electrodos en los dedos índice y corazón.
- ECG: el electrocardiograma refleja la actividad emocional y es un medidor de esfuerzo mental y de estrés del sujeto bajo estudio. Para medirlo se usan 2 electrodos, uno en la parte superior derecha y otro en la inferior izquierda del torso.
- Electromiografía (EMG): la actividad eléctrica muscular o electromiografía es un indicador de la actividad y el equilibrio de los sistemas simpático y parasimpático, y en particular, la conductividad del trapecio es un indicador del nivel de estrés del individuo. Para

medirlo se han colocado 2 electrodos en cada uno de los siguientes músculos: trapecio, cigomático y corrugador superciliar.

- EEG: la potencia de las distintas ondas del EEG está relacionada con la actividad del sistema nervioso y con el aumento o la disminución del estrés. En este caso para medir el EEG se ha utilizado el sistema 10/20 mediante un gorro de electroencefalograma (Easycap) y electrodos de Ag/AgCl localizados en FP1, FP2, AFz, F3, F5, F7, F6, Fz, C3, C5, Cz, FCz, P3, P5, Pz, O1, O2, T7 y T8. Se ha empleado como referencia los puntos A1 y A2 en los mastoides.

Además de estos biopotenciales, se han recogido una serie de señales, muestreadas a 32 Hz, que indican los niveles de dolor que en cada instante de las señales de biopotenciales se ha aplicado.

3.2. Protocolo de adquisición de los datos

Para la aplicación del dolor se utiliza una célula productora de calor (PATHWAY de Medoc), situada sobre una mesa, y atada alrededor de los brazos de los participantes en el experimento.

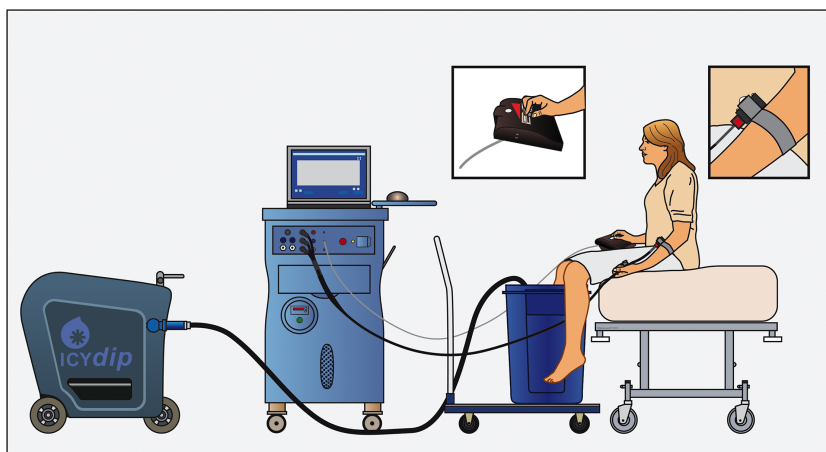


Figura 3.1: Ejemplo de configuración física un experimento simulado, que muestra a un paciente con un sistema de producción de dolor en el brazo. Imagen tomada de [15]

El protocolo con el que se obtiene la base de datos es el siguiente

1. Calibración de los límites de dolor: se realiza una prueba para medir la temperatura que cada paciente asocia con el umbral y con el límite soportable del dolor.
2. Obtención de los niveles de dolor: con esta calibración se obtienen 4 niveles de dolor dependiendo de la temperatura aplicada, mediante la división del intervalo entre las temperaturas que inducen los niveles de dolor mínimo y máximo.

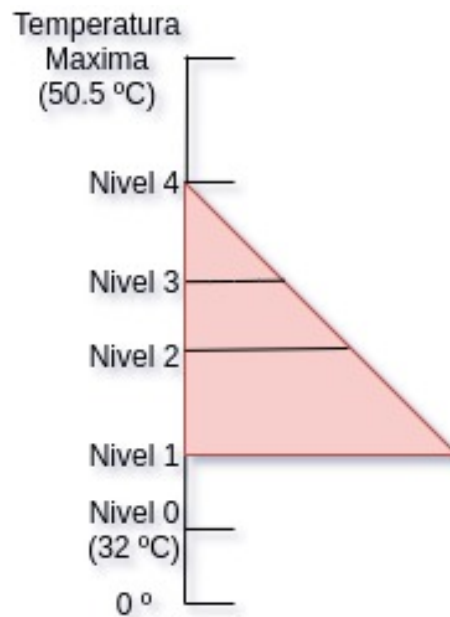


Figura 3.2: Niveles de dolor inducidos en los pacientes mediante la aplicación de calor para la adquisición de los datos.

3. Estimulación del dolor: durante 25 minutos se estimula de forma aleatoria con uno de los niveles de dolor durante 4 segundos, seguidos de pausas de duración variable, de entre 8 y 12 segundos, en las que no se aplica dolor, de forma que se aplican un total de 20 eventos de dolor, de distinta intensidad durante estos 25 minutos. Este periodo de adquisición se realiza sin el gorro que contiene los electrodos del electroencefalograma, para facilitar el estudio de las tomas de vídeo.
4. Registro de las emociones básicas: Los participantes demuestran ante la cámara las emociones básicas, como pueden ser la felicidad o el miedo, durante un minuto.

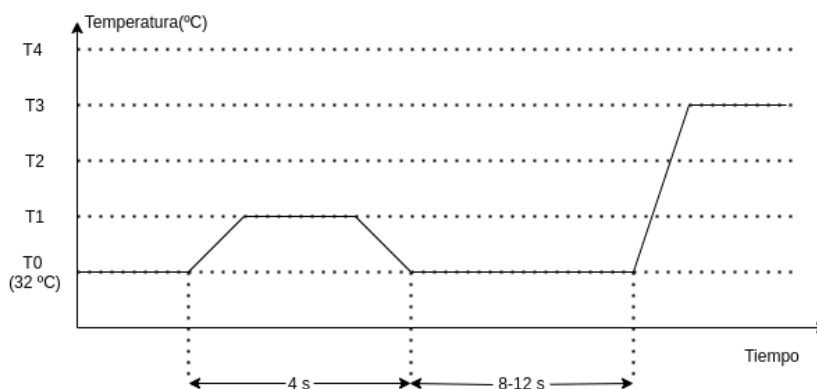


Figura 3.3: Temporización de aplicación de dolor a los pacientes para la adquisición de los datos.

5. Se realiza elicitación de emociones con imágenes y vídeo para cada una de las emociones básicas, que se han mostrado en el paso anterior, y los participantes les asignan una puntuación, en base a la fuerza de la emoción que le han provocado estas fuentes. Este paso y el anterior son necesarios para clasificar la asociación de emociones con el dolor en los registros de vídeo.
6. Se repite la excitación de dolor, siguiendo el mismo protocolo que en el paso 3, pero esta vez con la cubierta facial para la medición del electroencefalograma.

3.3. Estructura de la base de datos

La base de datos BioVid está compuesta por varios ficheros codificados en formato .csv (valores separados por comas); a continuación, describimos los más relevantes para nuestro análisis:

- En `samples.csv` de la carpeta `starting_point`, podemos ver los nombres de las señales de la base de datos y algunos datos sobre el paciente, como su sexo o su edad.

subject_id	subject_name	age	gender
1	071309_w_21	21	1
2	071313_m_41	41	0
3	071614_m_20	20	0
4	071709_w_23	23	1
5	071814_w_23	23	1
6	071911_w_24	24	1
7	072414_m_23	23	0
8	072514_m_27	27	0
9	072609_w_23	23	1

Tabla 3.1: Ejemplo de filas del fichero samples.csv que nos da información básica sobre los pacientes

- En la carpeta biosignal_raw tenemos los valores de las señales biomédicas sin procesar y la temporización de cada muestra tomada.

time	gsr	ecg	emg_trapezius	emg_corrugator	emg_zygomaticus
156	9.08266	-2031.32	2181.68	0	0
2109	9.083	-2033.79	2178.6	0	0
4063	9.08432	-2024.62	2184.44	0	0
6016	9.08301	-2019.41	2188	0	0
7969	9.082	-2024.41	2190.12	0	0
9922	9.08134	-2022.4	2197.61	0	0
11875	9.081	-2014.22	2192.48	0	0
13828	9.081	-2010.9	2188.75	0	0
15781	9.08166	-2002.45	2191.52	0	0

Tabla 3.2: Ejemplo de filas de un fichero de biopotenciales, donde para cada muestra recogida en el experimento, se da su instante temporal y el valor en mV de las señales.

- En la carpeta label los ficheros csv contienen las muestras de los niveles de dolor de cada paciente en un instante temporal determinado.

time	label
27500	0
58750	0
90000	0
121250	0
152500	0
183750	0
215000	0
246250	0
277500	0

Tabla 3.3: Ejemplo de filas de un fichero de la señal de niveles de dolor, que indica en cada momento temporal el nivel de dolor al que está sometido el paciente, siendo 0 el nivel mínimo y 4 el nivel máximo.

Capítulo 4

Métodos matemáticos

En este capítulo vamos a describir las diferentes técnicas que se han empleado para la extracción de características a partir de las señales de ECG y de HRV, que se utilizarían para la posterior clasificación de las señales en función de la presencia o ausencia de dolor en los distintos pacientes. También se hace una breve introducción a las técnicas de clasificación elegidas para comprobar la validez de los métodos de extracción de características aquí expuestos.

4.1. Métodos temporales

Los parámetros temporales son aquellos que se obtienen mediante la realización de cálculos en el dominio del tiempo sobre la señal del ECG o de HRV

En general, cuando se usan este tipo de métodos, no se trabaja directamente con la señal de HRV, sino que se hace una limpieza previa de forma que se eliminan los ectópicos, latidos en los que las aurículas o los ventrículos se contraen de forma prematura, para trabajar con intervalos interlatido asociados a los latidos normales, también llamados intervalos NN. [27] [14]

4.1.1. SDNN

Este parámetro mide la media de la distancia interlatido de los latidos normales en ms (media del tiempo entre intervalos NN).

La fórmula usada para calcular este parámetro es

$$SDNN = \frac{1}{n} \sum_{i=1}^n NN_i - NN_{i-1} \quad (4.1)$$

Este parámetro está muy relacionado con la densidad espectral de potencia del HRV, en concreto con las bandas de baja frecuencia (ULF, VLF y LF).

Es muy utilizado para medir el riesgo cardíaco de un paciente sobre todo en periodos más largos, de 24 o 48 horas, tras la realización de un Holter.

4.1.2. SDRR

Este parámetro se corresponde a la desviación típica de la distancia interlatido de los latidos normales en ms (desviación típica de los intervalo NN).

La fórmula empleada para calcular este parámetro es

$$SDRR = \sqrt{\frac{1}{n-1} \sum_{i=1}^n ((NN_i - NN_{i-1}) - SDNN)^2} \quad (4.2)$$

4.1.3. NN50 y pNN50

El parámetro $NN50$ representa el número de intervalos NN adyacentes que difieren 50 ms, mientras que el parámetro $pNN50$ representa el porcentaje de los intervalos NN adyacentes que difieren más de 50 ms. Está muy asociado con la actividad de las bandas de alta frecuencia (HF y VHF) de la señal de HRV.

4.1.4. RMSSD

Este parámetro es la raíz de la distancia al cuadrado de las sucesivas diferencias entre latidos normales. Se obtiene calculando la diferencia entre latidos en ms, elevando estas diferencias al cuadrado, calculando la media de este cuadrado y por último obteniendo la raíz cuadrada de estas medias.

La fórmula para calcular el RMSSD es

$$RMSSD = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (NN_{i+1} - NN_i)^2} \quad (4.3)$$

El RMSSD está fuertemente correlacionado con el pNN50 y con la potencia en la banda de alta frecuencia (HF) del espectro de la señal de HRV.

4.2. Métodos espectrales

De los métodos usados en el análisis de ECG destacan los relacionados con el análisis del espectro del HRV, que estudian la distribución con la que ocurren las distintas componentes frecuenciales y su potencia absoluta y relativa a otras bandas, o sea, la modulación de los diferentes ritmos cardiacos.

Entre otros indicadores, la cantidad de latidos de baja frecuencia se usa muy comúnmente como una medida de la actividad del sistema nervioso simpático. Otra medida que se usa frecuentemente es la relación $\frac{HF}{LF}$ del espectro del HRV, donde LF se refiere a la potencia de las bandas de baja frecuencia y HF se refiere a la potencia de las componentes de alta frecuencia, lo que proporciona información acerca de la actividad del sistema nervioso autónomo.

Inspirándonos en estos métodos de análisis y siguiendo las pautas dictadas en la literatura relevante como [24], se ha realizado un análisis de la densidad espectral de potencia del propio ECG. No realizamos el análisis directamente sobre el HRV, ya que no tenemos suficiente muestras por segmento como para poder aplicar de manera fiable la transformada de Fourier o el periodograma a estas señales para calcular la densidad espectral de potencia, dado que las señales de HRV solo tienen una longitud de 12 a 30 muestras (señales de 3 a 7.5 segundos muestreadas a 4 Hz)

Para realizar el análisis de la señal de ECG usando métodos espectrales nos basamos en la figura 4.1, donde se puede apreciar que las distintas componentes espectrales de las ondas del ECG se representan mayoritariamente en bandas disjuntas del espectro por lo que calculamos la densi-

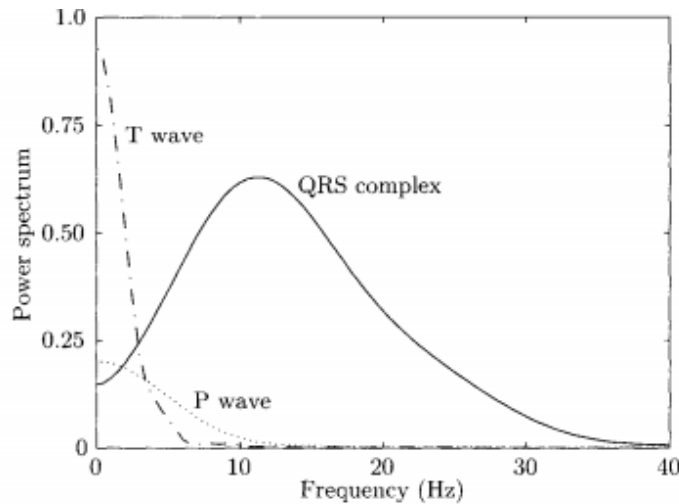


Figura 4.1: Componentes espectrales del ECG Referencia: [27]

dad espectral de potencia de las distintas partes como marcador síntesis para su uso en los sistemas de aprendizaje automático.

El método aquí presentado nos ofrece otra ventaja, al permitirnos el estudio de la actividad y de la influencia de las componentes de onda la T y de la onda P del ECG, dado que estas son muy difíciles de extraer en el dominio del tiempo, ya que el proceso para poder estudiarlas, normalmente, involucra un proceso de delineado manual y el empleo de distintas heurísticas.

Para calcular la potencia en cada banda calculamos un periodograma de Welch, de la señal de ECG, e integramos el área bajo la curva, mediante una suma de las componentes frecuenciales, por lo que calculamos el área comprendida entre 0 y 10 Hz para el cálculo de la potencia las ondas P y T, asimismo, integramos las componentes entre 10 y 40 Hz para obtener la potencia del complejo QRS.

4.3. Métodos de entropía

En los últimos años, la medidas de entropía han jugado un papel predominante para caracterizar la complejidad de sistemas biológicos, que se suelen caracterizar con los mismos métodos de análisis que se usan en el estudio de sistemas dinámicos complejos [22]. Los resultados extraídos gracias al usos de estas medidas permiten a los investigadores extraer

información útil sobre diferentes partes del sujeto bajo estudio sin la necesidad de observación directa y biopsias.

La entropía mide el desorden o la incertidumbre de un sistema y fue introducida inicialmente en el campo de la termodinámica por parte de Clausius [10], como forma de estudiar la propagación de gases en un volumen dependiendo de cambios de temperatura. También introdujo una interpretación estadística entre la propiedad de entropía y el estado del sistema.

4.3.1. Entropía de Shannon

La entropía de Shannon de una variable aleatoria es un parámetro que mide la incertidumbre de una fuente de información, o, desde otro punto de vista, mide la cantidad de información que contienen los símbolos (las muestras) de esta variable aleatoria.

El contenido de información de un evento \mathbb{E} es una función que decrece cuando la probabilidad de ese evento aumenta y formalmente se define como:

$$I(\mathbb{E}) = -\log_2(P(\mathbb{E})) \quad (4.4)$$

donde $P(\mathbb{E})$ se refiere a la probabilidad de un símbolo. La entropía mide la media de la cantidad de información del espacio muestral de una variable aleatoria y se define como:

$$H(\mathbb{X}) = E[I(\mathbb{E})] = E[-\log_2(p(\mathbb{E}))] \quad (4.5)$$

donde $E[\mathbb{X}]$ es la esperanza matemática de la variable aleatoria \mathbb{X} . La entropía también se puede calcular de forma explícita como:

$$H[\mathbb{X}] = \sum_{i=0}^n P(x_i) \log_2\left(\frac{1}{P(x_i)}\right) \quad (4.6)$$

Es importante señalar que la entropía se basa en la probabilidad de los distintos eventos y no en su magnitud, y que, además ésta se maximiza cuando la probabilidad de todos los símbolos es igual.

El ejemplo clásico de entropía de Shannon es considerar 3 símbolos $\{x_1, x_2, x_3\}$ con probabilidades $\{P(x_1), P(x_2), P(x_3)\} = \{0,5, 0,25, 0,25\}$ cuya entropía sería:

$$H(\mathbb{X}) = -(\log_2(0,5) * 0,5 + \log_2(0,25) * 0,25 + \log_2(0,25) * 0,25) = 1,5 \quad (4.7)$$

4.3.2. Entropía aproximada

Esta técnica fue introducida por Pincus [19] para cuantificar la regularidad de una serie temporal sin conocer los detalles sobre el proceso generador del sistema; lo que permite conocer las componentes estocásticas de la señal. Valores bajos de este estadístico significa que el sistema es muy persistente, repetitivo y fácil de predecir, con patrones en la señal que se repiten, mientras que valores altos de este estadístico indican independencia entre los distintos bloques y la ausencia de patrones presentes en los datos. Heurísticamente el algoritmo mide el logaritmo de la probabilidad de que dos patrones consecutivos sean similares, lo que indicaría que la señal es regular. Una nota importante sobre la entropía aproximada es que permite analizar los modelos generadores de los sistemas bajo estudio sin necesidad de conocer detalles de éste, solo son necesarios los datos producidos por el sistema, ya que se basa directamente en los valores de la serie temporal y no en las probabilidades de los distintos símbolos de ésta.

Para implementarla nos hemos basado en [12], que procedemos a resumir.

El algoritmo depende de 2 parámetros m (longitud de la ventana de los patrones a considerar) y r (barrera del filtro de ruido). En el análisis de ECG se suelen tomar los parámetros $m = 2$ y $r = 0,25 * \sigma(U)$ [38], [17].

1. Primero se crean los vectores x_i y x_j de un pseudo-espacio de fase de dimensión m desde una serie $U[n] = \{u_1, u_2, \dots, u_N\}$ de forma que definimos $x_i = \{u_i, \dots, u_{i+m-1}\}$ y $x_j = \{u_j, \dots, u_{j+m-1}\}$, para $\forall i, j \in [1, N]$.

2. Calculamos la distancia entre x_i y x_j como $d[x_i, x_j] = \max_{k=1, \dots, m} (|u_{i+k-1} - u_{j+k-1}|)$
3. Calculamos el número de segmentos $d' \leq N - m + 1$ tal que $d[x_i, x_j] \leq r$ y con esto obtenemos la frecuencia relativa $C_i^m(r) = \frac{d'}{N-m+1}$.
4. Calculamos $\phi^m(r) = \frac{1}{N-m+1} \cdot \sum_{i=1}^{i=N-m+1} \ln(C_i^m(r))$
5. Repetimos el proceso con vectores de longitud $m + 1$ para calcular $C_i^{m+1}(r)$ y $\phi^{m+1}(r)$.
6. Por último, calculamos la entropía aproximada como $ApEnt(m, r) = \phi^m(r) - \phi^{m+1}(r)$

Para intentar aclarar el algoritmo vamos a proceder a desarrollar un ejemplo, el cual se puede en la figura 4.2, en el que fijamos los parámetros $m = 2$ y $r = 0,15$:

Consideremos la serie $U[n] = \{-0,5, -0,4, -0,3, -0,2, -0,1, 0, 0,1, 0,2, 0,3, 0,4, 0,5, 0,4, 0,3, 0,2, 0,1, 0 - 0,1, -0,2, -0,3, -0,4, -0,5\}$

La idea principal es escoger todos los bloque de longitud m y compararlos con el resto de bloques de la misma longitud (incluido él mismo), para ver si son similares en una proporción marcada por el parámetro r ($d[x_i, x_j] \leq r$), para $\forall i, j \in [1, N]$. Si esto se cumple, marcamos el bloque como posible.

Ahora comprobamos los bloque de longitud $m + 1$, donde tenemos que mirar si se cumple, otra vez, la condición de distancia entre los elementos. Si se cumple, entonces marcamos el bloque como coincidente, en caso contrario, los bloques anteriores se quedan solo como posibles. Como optimización podemos simplemente comprobar los últimos elementos (los elementos $m + 1$) de los bloques, ya que, los elementos $1, \dots, m$ los hemos comparado en el paso anterior.

Una vez hemos realizado este proceso con todos los bloque calculamos, la probabilidad condicionada C_i^m como la ratio entre los bloques coincidentes y los bloques posibles

Volviendo con el ejemplo anterior empezamos con $x_i = x_1 = [u_1, u_2] = [-0,5, -0,4]$ como patrón y lo comparamos con todos los bloques en la secuencia de longitud $m = 2$.

Primero lo comparamos consigo mismo, donde obviamente conseguimos una coincidencia (hay que notar que esta auto comparación implica que el estadístico tenga sesgo).

Posteriormente comparamos $x_j = x_2 = [u_2, u_3] = [-0,4, 0,3]$ con x_1 componente a componente, por lo que, calculamos la distancia $|d[u_1, u_2]| = |-0,5 - (-0,4)| = |-0,1| \leq 0,15$, y como cumple la condición entonces evaluamos la distancia entre las segundas componentes de los vectores $|d[u_2, u_3]| = |-0,4 - (-0,3)| = |-0,1| \leq 0,15$, y así repetidamente hasta la m -ésima componente. Como la condición se cumple para todos los componentes $1, \dots, m$ entonces los bloques se marcan como posibles.

Entonces evaluamos la componente siguiente de los bloques que estamos comparando, $m + 1$, localizada inmediatamente después de esos bloques, para ver si las componentes siguen cumpliendo la condición, en este caso, habría que comparar u_3 y u_4 . Con esto vemos que la condición de distancia $d[u_3, u_4] = |-0,3 - (-0,2)| = 0,1 \leq 0,15$ se cumple para $m + 1$, por lo que podemos marcar los segmentos como coincidentes, lo que significa que el patrón continúa después de los 2 segmentos.

Ahora vamos a comparar el bloque x_1 con $x_3 = [u_3, u_4]$, donde es fácil probar que no es un bloque posible, ya que, no cumple la condición en la primera componente $|d[u_1, u_3]| = |-0,5 - (-0,3)| = |0,2| \geq 0,15$, por lo que no hace falta que sigamos comparando este par. Como se puede ver el parámetro r determina el proceso, de forma que valores más altos de r , permitirían que éste fuera un bloque posible.

De la misma forma comparamos el resto de los bloques en la secuencia. Una vez hemos comparado todos los bloques con x_1 escojamos x_2 como patrón, y posteriormente, procedemos a repetir el proceso con los sucesivos bloques hasta que nos quedamos sin bloques de $m = 2$ elementos.

Vamos a centrarnos por un momento en el caso de usar el bloque x_9 como patrón, y compararlo con x_{12} $|d[u_9, u_{12}]| = |0,3 - 0,4| = |0,1| \leq 0,15$ y $|d[u_{10}, u_{13}]| = |0,4 - 0,3| = |0,1| \leq 0,15$, por lo que lo consideramos un bloque posible. Sin embargo, el elemento $m + 1$ no cumple la condición ya que $d[u_{11}, u_{14}] = |0,5 - 0,2| = |0,3| \geq 0,15$, por lo que no es un bloque coincidente. Esto se puede ver en la gráfica de la señal, la figura 4.2, donde

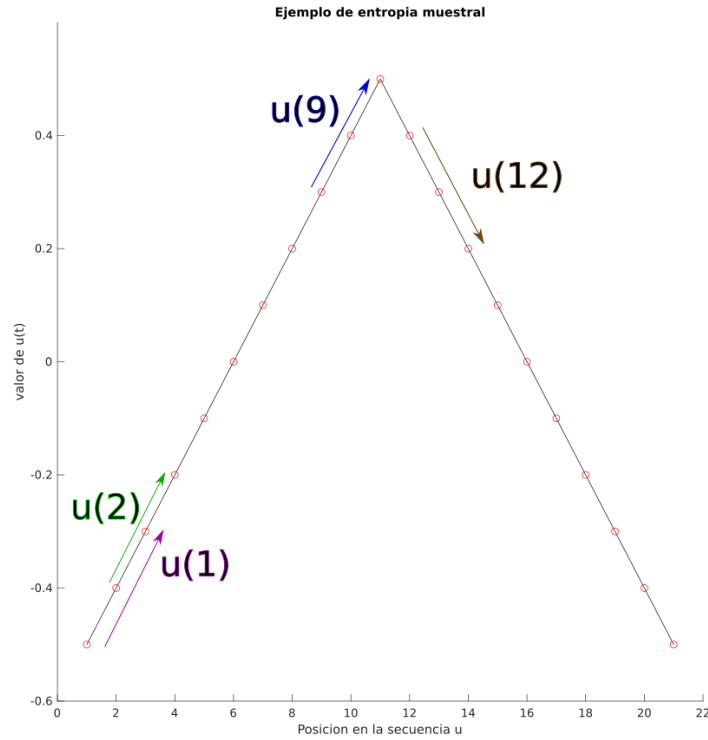


Figura 4.2: Ejemplo de los patrones para el cálculo de la entropía aproximada sobre la serie U .

vemos que x_9 sigue un patrón ascendente el cual no es el mismo patrón que existe en x_{12} donde hay un patrón descendente.

Una vez hemos realizado el proceso para todos los patrones podemos calcular las probabilidades condicionadas C_i^2 dividiendo los bloques coincidentes entre los bloques posibles. Si restringimos el ejemplo anterior a la primera mitad de la serie $U[n]$ de tal forma que $U'[n] = \{-0,5, -0,4, -0,3, -0,2, -0,1, 0, 0,1, 0,2, 0,3, 0,4, 0,5\}$, y mantenemos los valores de los parámetros en $m = 2$ y $r = 0,15$, podemos ver las probabilidades condicionadas de los bloques representados en la figura 4.2 en la tabla 4.1:

Bloque	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
Posibles	2	3	3	3	3	3	3	3	3	3	2
Coincidentes	2	3	3	3	3	3	3	3	3	3	2

Tabla 4.1: Tabla de bloques posibles y coincidentes para el cálculo de la entropía aproximada sobre la serie $U'[n]$.

Por último, calculamos la media del logaritmo de estas probabilidades condicionadas $\frac{\text{posibles}}{\text{coincidentes}}$ y definimos la entropía aproximada como el valor de esa media cambiado de signo.

Las principales ventajas de la entropía aproximada consisten en su estabilidad frente a valores extremos, su inmunidad frente a traslaciones y escalado y su capacidad de obtener medidas significativas con un bajo número de datos en la serie temporal a analizar.

Las principales desventajas de este estadístico radican en la presencia de sesgo (ya que los patrones se emparejan consigo mismos) y en el cambio de forma impredecible del resultado en función de la longitud de la señal y del valor del parámetro r . Además, el cálculo de la entropía aproximada requiere un muestreo uniforme de las señales.

4.3.3. Entropía muestral

Esta técnica fue introducida por Richman y Moorman [21] para evitar los problemas de sesgo y la dependencia de la longitud en la entropía aproximada, mediante la eliminación de las autocomparaciones y la normalización del parámetro de entropía en función de la longitud de la señal por defecto. Para implementarla nos hemos basado en [12] y [37], que procedemos a resumir.

El algoritmo depende de 2 parámetros m (longitud de la ventana de los patrones a considerar) y r (barrera del filtro de ruido). En el análisis de ECG se suelen tomar los parámetros $m = 2$ y $r = 0,25 \cdot \sigma(U)$ [38], [17].

1. Primero se crean los vectores x_i y x_j de un pseudo-espacio de fase de dimensión m desde una serie $U = \{u_1, u_2, \dots, u_N\}$ co-

mo $x_i = \{u_i, \dots, u_{i+m-1}\}$ y $x_j = \{u_j, \dots, u_{j+m-1}\}$. con $i \neq j$ y $i, j = 1, \dots, N - m + 1$.

2. Calculamos la distancia entre x_i y x_j como $d[x_i, x_j] = \max_{k=1, \dots, m} (|u_{i+k-1} - u_{j+k-1}|) \forall i, j$
3. Calculamos el número de segmentos $d' \leq N - m + 1$ tal que $d[x_i, x_j] \leq r$ y obtenemos $C_i^m(r) = \frac{d'}{N-m}$.
4. Calculamos $\phi^m(r) = \frac{1}{N-m+1} * \sum_{i=1}^{N-m+1} (C_i^m(r))$
5. Repetimos el proceso con vectores de longitud $m + 1$ para calcular $C_i^{m+1}(r)$ y $\phi^{m+1}(r)$.
6. Por último calculamos la entropía aproximada como $SamEnt(m, r) = \ln\left(\frac{\phi^m(r)}{\phi^{m+1}(r)}\right)$

Para la implementación de este estadístico referimos al lector a la implementación de la entropía aproximada, ya que ambos métodos son muy similares, siendo sus principales diferencias las siguientes:

- La entropía muestral no admite que se considere como similares un segmento consigo mismo.
- La suma de los vectores patrón está dentro del logaritmo en la entropía muestral, mientras que en el caso de la entropía aproximada la suma está fuera del logaritmo, lo que implica que la entropía muestral considera la serie entera a la hora de encontrar una coincidencia (y si la encuentra ya esta definida), mientras que la entropía aproximada necesita una coincidencia para cada posibilidad.

4.3.4. Entropía espectral

La entropía espectral se define como el valor de la entropía de Shannon sobre la densidad espectral de potencia de una banda frecuencial, y mide la regularidad espectral de la señal, o también, puede ser interpretada como la incertidumbre sobre un evento que ocurra a una frecuencia f_i . Se suele normalizar y se calcula en la banda de frecuencias $[f_i, f_h]$ como

$$SpEnt = \frac{1}{N_f} \sum_{i=f_l}^{f_h} P(f_i) * \log_2\left(\frac{1}{P(f_i)}\right) \quad (4.8)$$

donde N_f es el número de componentes espectrales presentes en el rango $[f_l, f_h]$

Para su cálculo hay que transformar la señal donde se quiera calcular al dominio de la frecuencia (empleando para ello un algoritmo como un periodograma), y en este dominio calcular la entropía de Shannon entre las bandas de frecuencias de interés de la señal.

4.3.5. Entropía de permutación

El método de entropía de permutación fue propuesto por Bandt y Pompe [4] para incorporar el orden de la serie temporal a los análisis tradicionales de entropía para estudiar la complejidad de las señales. El método se basa en el cálculo del orden del rango de las series temporales (orden de mayor a menor de los elementos) en secuencias de longitud dependientes del parámetro de dimensión de las particiones temporales n , con la posibilidad de obviar algunos elementos dependiendo del valor del parámetro de retraso temporal de las particiones τ , que indica el tiempo entre valores considerados para el cálculo de esta medida de entropía.

La formula de la entropía de permutación es:

$$H_n = \sum_{j=1}^{j=n} p'_j \log_2(p'_j) \quad (4.9)$$

donde el símbolo p'_j representa la frecuencia relativa de los posibles patrones de secuencias de símbolos.

Nos basamos en el texto de [23] para la implementación y procedemos a describir brevemente el algoritmo:

1. Se inicializan los contador i a 1, como el índice actual de la ventana sobre la serie temporal considerada.
2. Se define el orden de la permutación n , lo que genera los patrones de permutación π_j con $j = 1, \dots, n!$ (o $j = 1, \dots, n^n$ si es con repetición), mediante el cálculo de todas las permutaciones posibles de los posibles rangos.
3. Se inicializa un vector $Z = \{z_1, z_2, \dots, z_p\}$ a 0 con igual número de elementos como permutaciones, y se asigna cada elemento de éste a

una permutación.

4. Se calculan los rangos de cada valor de la secuencia $X_i = \{x_1, x_2, \dots, x_n\}$, lo que da lugar al vector de rangos $R = \{r_1, r_2, \dots, r_n\}$
5. Se comparan los vectores de rangos con las permutaciones generadas y se incrementa el valor del elemento z_j en 1 cuando coincidan con un vector de rangos.
6. Si $i \leq N - n$, se incrementa el valor de i en τ , se calculan los rangos de las secuencias y se comparan con las permutaciones otra vez. En caso contrario, si $i > N - n$, se prosigue al siguiente paso.
7. Se calcula la frecuencia relativa de las permutaciones como una estimación de la probabilidad de estas mediante la fórmula $p'_j = \frac{z_j}{\sum z_k}$
8. Se seleccionan los valores de p'_j que sean mayores que 0 y se calcula la entropía de permutación con la formula $H_n = \sum_{j=1}^{j=n} p'_j * \log_2(p'_j)$.

Vamos a ilustrar el algoritmo con un ejemplo, considerando la serie $X = \{6, 9, 11, 12, 8, 13, 5\}$, que tiene longitud $N = 7$, con los parámetros $n = 3$ y $\tau = 1$, apoyándonos en la figura 4.3 para explicarlo. (en este ejemplo no consideramos elementos de la serie con el mismo rango)

1. Como $n = 3$ hay $n! = 6$ permutaciones posibles que son $\pi_1 = \{1, 2, 3\}$, $\pi_2 = \{1, 3, 2\}$, $\pi_3 = \{2, 1, 3\}$, $\pi_4 = \{2, 3, 1\}$, $\pi_5 = \{3, 1, 2\}$ y $\pi_6 = \{3, 2, 1\}$.
2. Iniciamos $i = 1$ y $z_{j=1, \dots, 6} = 0$.
3. El rango de la primera secuencia $X_1 = \{6, 9, 11\}$ es $R_1 = \{1, 2, 3\}$.
4. El valor de R_1 es igual a π_1 por lo que incrementamos z_1 en una unidad e incrementamos $i = i + \tau$.
5. Como $i = 2 < 7 - 3$ seleccionamos $X_2 = \{9, 11, 12\}$, que tiene rangos $R_2 = \{1, 2, 3\}$, que otra vez es igual a π_1 , por lo que volvemos a incrementar z_1 y sumamos τ a i .
6. Como $i = 3 < 4$ repetimos el proceso hasta que $i = 5$.
7. Cuando i alcance el valor 5 hemos terminado el proceso de cálculo de rangos, por lo que podemos pasar a calcular las frecuencias relativas.
8. Tenemos los valores de frecuencias $z_1 = 2$, $z_2 = 0$, $z_3 = 1$, $z_4 = 2$, $z_5 = 0$ y $z_6 = 0$, las probabilidades aproximadas son iguales a $p'_1 = \frac{2}{5}$,

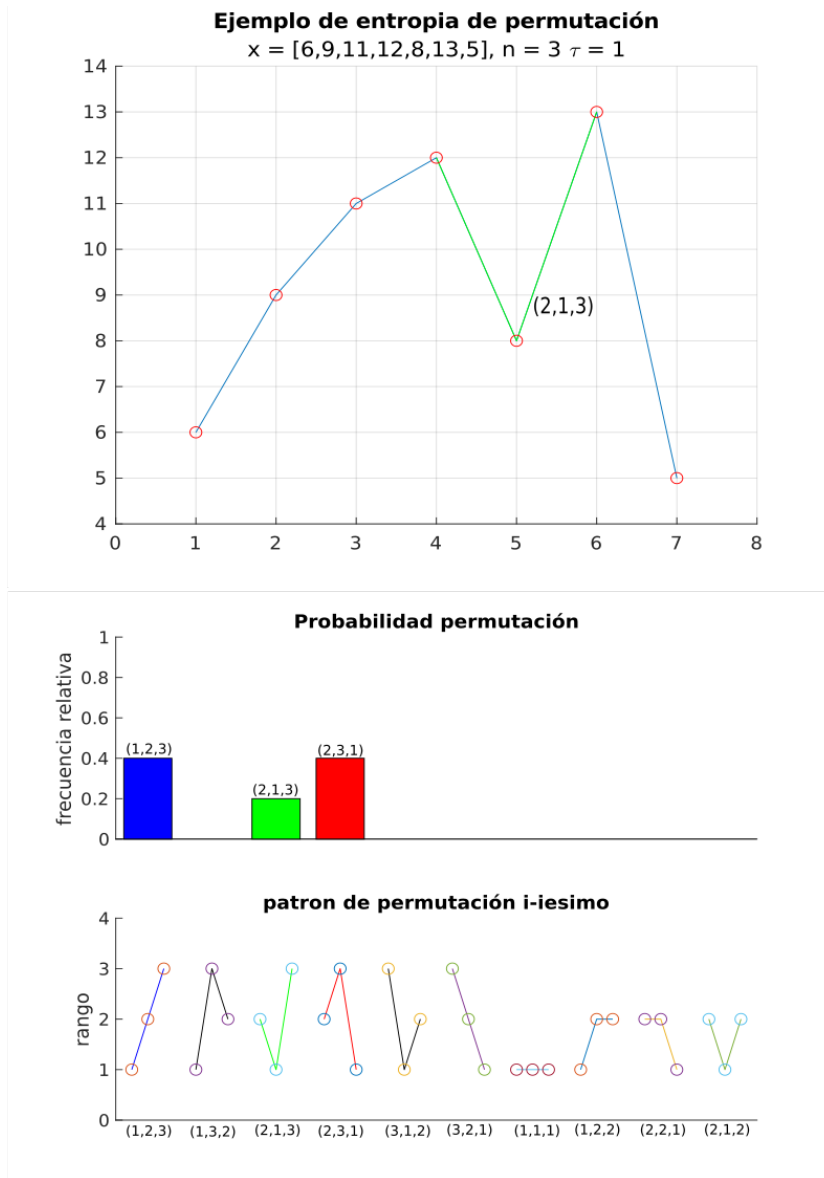


Figura 4.3: Ejemplo del cálculo de la entropía de permutación sobre la señal $X[n]$.

$$p'_2 = 0, p'_3 = \frac{1}{5}, p'_4 = \frac{2}{5}, p'_5 = 0 \text{ y } p'_6 = 0.$$

9. Usando los valores de p'_n distintos de 0, podemos calcular la entropía de permutación de orden 3 como

$$H_3 = -\left(\frac{2}{5} \cdot \log_2\left(\frac{2}{5}\right) + \frac{1}{5} \cdot \log_2\left(\frac{1}{5}\right) + \frac{2}{5} \cdot \log_2\left(\frac{2}{5}\right)\right) \approx 1,5 \quad (4.10)$$

Un posible problema que se puede presentar al realizar el cálculo de entropía de permutación son aquellos valores que tienen el mismo rango en la serie temporal. En este caso la solución que hemos adoptado es considerar que los valores iguales en la secuencia tienen el mismo rango, lo que trae como desventaja el aumento del número de posibles permutaciones hasta un máximo teórico de n^n . Otra posible solución es el incremento del parámetro τ lo que permite saltarse elementos de la serie temporal, lo cual, puede resultar útil en el caso de tener valores similares de forma consecutiva, por ejemplo, si se toma una frecuencia de muestreo demasiado alta.

4.4. Máquinas de vectores soporte

En esta sección vamos a describir el algoritmo de clasificación que hemos usado para clasificar los segmentos extraídos de la base de datos basándonos en [9] y [30]

Las máquinas de vectores soporte (SVM) [11] son una clase de algoritmos de aprendizaje automático que pertenecen a la categoría de los clasificadores lineales supervisados, ya que inducen hiperplanos entre los tipos de los datos a clasificar.

Las SVM minimizan el riesgo estructural del problema de clasificación de forma que intentan separar los ejemplos más cercanos de cada clase para conseguir un margen máximo a cada lado del hiperplano, empleando para realizar la separación de las clases los ejemplos más cercanos al margen.

El entrenamiento de las SVM, es un problema de optimización cuadrática con restricciones y se resuelve mediante multiplicadores de Lagrange.

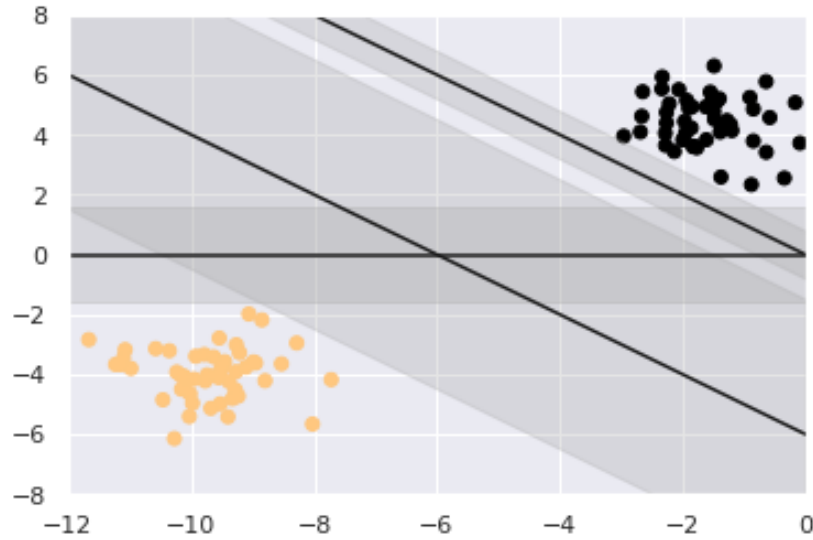


Figura 4.4: Fronteras y posibles márgenes de separación entre datos linealmente separables, usando distintos hiperplanos.

4.4.1. SVM para ejemplos linealmente separables

Dado un conjunto linealmente separable de ejemplos $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ donde $x_i \in \mathbb{R}^d$ e $y_i \in \{+1, -1\}$ se puede definir un hiperplano de separación como una función lineal, que separa dicho conjunto sin error

$$D(x) = (w_1x_1 + \dots + w_dx_d) + b = \langle w, x \rangle + b \quad (4.11)$$

sujeto a la restricciones

$$y_i(\langle w, x_i \rangle + b) \geq 0, \quad i = 1, \dots, n \quad (4.12)$$

donde $w \in \mathbb{R}^d$ y $b \in \mathbb{R}$ son los vectores y escalares que definen un hiperplano de separación de forma que w es el vector que conecta un punto con el hiperplano $D(x)$ y el operador $\langle x_1, x_2 \rangle$ representa el producto escalar de vectores.

Como se puede ver existen múltiples hiperplanos, pero estamos interesados en el hiperplano óptimo, que es el hiperplano donde la distancia, entre $D(x)$ y el ejemplo más cercano de cualquiera de las dos clases (el margen del hiperplano) es máxima.

Para plantear el problema de optimización definamos por geometría la distancia entre el hiperplano de separación $D(x)$ y un ejemplo x' como

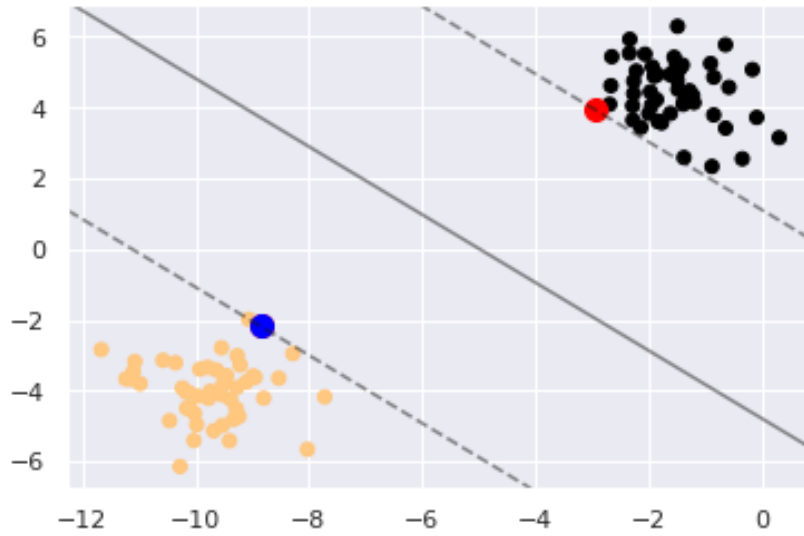


Figura 4.5: Frontera óptima y margen de separación de datos linealmente separables. En rojo se pueden apreciar los vectores soporte de los datos de color negro y, en azul, los vectores soportes de los datos de color naranja.

$$\text{Distancia}(D(x), x') = \frac{D(x')}{\|w\|} \quad (4.13)$$

donde $|\cdot|$ es el operador valor absoluto y $\|\cdot\|$ es el operador norma de un vector.

Entonces todos los ejemplos, bajo la separación con un hiperplano óptimo cumplen la condición

$$\frac{y_i D(x_i)}{\|w\|} \geq \tau \in \mathbb{R} \quad (4.14)$$

siendo la distancia igual al margen para los puntos en la frontera (los llamados vectores soporte), por lo que, si queremos maximizar el margen τ entonces podemos minimizar $\|w\|$. En este proceso fijamos de forma arbitraria $\|w\|\tau = 1$, para solo encontrar una solución.

Con todo esto podemos definir el problema de optimización como la minimización de la función

$$f(x) = \frac{1}{2}\|w\|^2 \quad (4.15)$$

sujeto a

$$y_i(\langle w, x_i \rangle + b) \geq 1 \quad i = 1, \dots, n \quad (4.16)$$

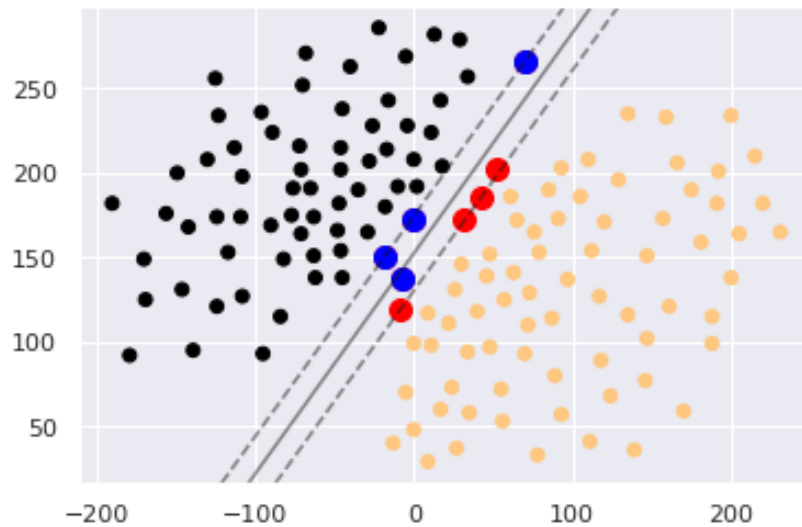


Figura 4.6: Frontera óptima y margen de separación de datos cuasi-linealmente separables. En rojo se pueden ver los vectores soporte de los datos de color naranja y, en azul, los vectores soportes de los datos de color negro. En este caso se puede observar cómo algunos de los datos usados como vectores soporte están dentro del margen, incluso se aprecia que uno de los vectores en azul está clasificado de forma errónea en el lado de la frontera perteneciente a los datos de color naranja.

En este problema se introducen el producto por $\frac{1}{2}$ y el cuadrado de la norma para facilitar el planteamiento del problema de optimización, pero estos cambios no afectan a la solución final.

4.4.2. SVM para datos cuasi-linealmente separables

Si no tenemos ejemplos perfectamente separables mediante un hiperplano, por ejemplo, si existen algunos datos mezclados en la frontera de separación, podemos relajar el grado de separabilidad de forma que permitamos errores de clasificación en algunos de los ejemplos del conjunto de entrenamiento.

La idea es introducir en la condición que define el hiperplano un conjunto de variables de holgura, que representan el error correspondiente a un vector clasificado incorrectamente desde el borde del margen de su clase.

De esta forma la ecuación del hiperplano se transforma en

$$y_i(\langle w, x_i \rangle + b) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, i = 1, \dots, n \quad (4.17)$$

por lo que ya no basta plantear como objetivo la maximización del margen, sino que tenemos que incluir en la función a minimizar un término que cuantifique los errores cometidos por el hiperplano de separación. Para hacerlo transformamos la función a optimizar en

$$f(w, \epsilon) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \epsilon_i \quad (4.18)$$

donde el valor del parámetro de regularización C es elegido por el usuario y permite controlar en que grado influye el término de coste de los ejemplos no separables, de forma si usamos valores de C grandes la anchura del margen es estrecha, lo que produce modelos más simples y poco flexibles, mientras que, si escogemos valores de C pequeños las anchuras de los márgenes resultantes aumentan, lo que produce modelos más complejos, que cometen más errores en el entrenamiento, pero luego son más generalizables.

En este caso podemos definir el problema de optimización a resolver como la minimización de la función

$$f(w, \epsilon) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \epsilon_i \quad (4.19)$$

sujeto a

$$y_i(\langle w, x_i \rangle + b) + \epsilon_i \geq 1 \quad \epsilon_i \geq 0, i = 1 \dots, n \quad (4.20)$$

4.4.3. SVM para datos no linealmente o cuasi-linealmente separables

Cuando los datos no son linealmente o cuasi-linealmente separables, se emplean un conjunto de funciones que transforman los puntos (x_i, y_i) del espacio de entrada a un espacio de características \mathbb{F} , donde los puntos $(\Phi(x_i), y_i)$ son linealmente o cuasi-linealmente separables. Entonces podemos encontrar un hiperplano de separación en \mathbb{F} y transformarlo en una frontera no lineal en el espacio original.

Definamos $\Phi : \mathbb{X} \rightarrow \mathbb{F}$ como una función no lineal de transformación que hace corresponder a cada vector de entrada $x = (x_1, \dots, x_n)$ t.q. $x \in \mathbb{X}$ con un punto en el espacio de características $\phi(x)$ t.q. $\Phi(x) \in \mathbb{F}$. En este

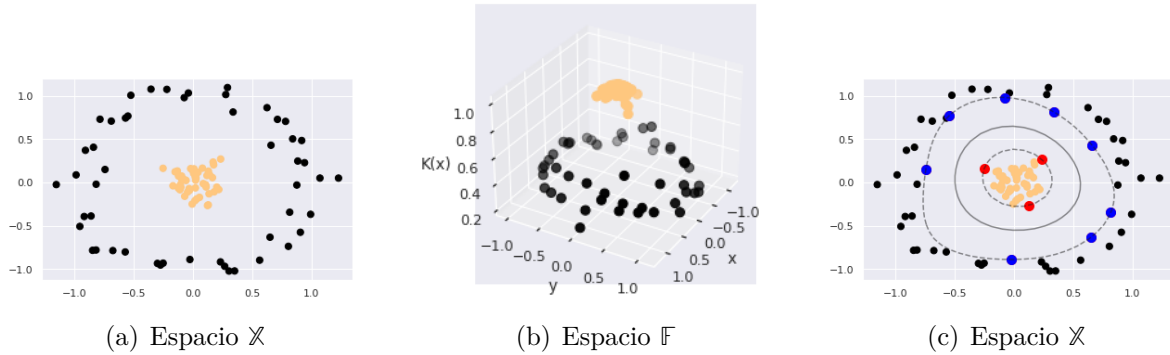


Figura 4.7: Clases en el espacio original, separadas en el espacio transformado, y frontera de separación en el espacio original. Los puntos rojos y azules representan los vectores soporte de la clases de color negro y naranja respectivamente.

contexto la función del hiperplano de separación (donde implícitamente se incluye b como $b = w_1 \phi_1(x)$) se puede definir como

$$D(x) = (w_1 \phi_1(x) + \dots + w_m \phi_m(x)) = \langle w, \Phi(x) \rangle \quad (4.21)$$

que podemos transformar una función equivalente para el proceso de optimización:

$$D(x) = \sum_{i=1}^n \alpha_i y_i \langle \Phi(x_j) \Phi(x_i) \rangle \quad (4.22)$$

Una función kernel $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ es una función semi-definida positiva que asigna a cada par de elementos del espacio de entrada \mathcal{X} un valor correspondiente al producto escalar de las imágenes de dichos elementos en el espacio de características \mathbb{F} .

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle = (\phi_1(x)\phi_1(x') + \dots + \phi_m(x)\phi_m(x')) \quad (4.23)$$

Algunos ejemplos de funciones kernel son funciones lineales, polinómicas o gaussianas (estas dependen de un parámetro extra γ , que influye en la noción de distancia del espacio transformado, mediante el control de la varianza de las funciones gaussianas usadas en el kernel).

Con todo esto la función del hiperplano se convierte en

$$D(x) = \sum_{i=1}^n \alpha_i y_i K(x_j, x_i) \quad (4.24)$$

y entonces podemos definir el problema de optimización como la maximización de

$$\sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (4.25)$$

sujeto a

$$\sum_{i=1}^n \alpha_i y_i = 0 \text{ con } 0 \leq \alpha_i \leq C, i = 1, \dots, n \quad (4.26)$$

Es relevante notar que al resolver este problema de optimización usando funciones kernel conseguimos una frontera no lineal directamente en el espacio de entrada, sin necesidad de realizar el cálculo de funciones de transformación: $K : \mathcal{X} \rightarrow \mathbb{F}$, o de sus inversas: $K^{-1} : \mathbb{F} \rightarrow \mathcal{X}$, lo que simplifica el procedimiento computacional de clasificación no lineal.

Capítulo 5

Tratamiento de los datos

En este capítulo vamos a describir la extracción de características de las señales de la base de datos BioVID para su posterior clasificación en función de la presencia de dolor. También exponemos los resultados de distintos experimentos de clasificación mediante SVM de las características de las señales de ECG obtenidas con los métodos aquí descritos.

5.1. Preprocesado de los datos

Previo a la extracción de características de las señales realizamos un preprocesado de los datos usando un script de Python, para adecuar la base de datos al análisis que vamos a realizar usando MATLAB.

Comenzamos leyendo el fichero `samples.csv` y eliminando los registros correspondientes a los sujetos `082315_w_60`, `082414_m_64`, `082909_m_47`, `083009_w_42`, `083013_w_47`, `083109_m_60`, `083114_w_55`, `091914_m_46`, `092009_m_54`, `092014_m_56`, `092509_w_51`, `092714_m_64`, `100514_w_51`, `100914_m_39`, `101114_w_37`, `101209_w_61`, `101809_m_59`, `101916_m_40`, `111313_m_64`, y `120614_w_61`, ya que estos pacientes no han presentado ninguna reacción visible al dolor estimulado tal y como se sugiere en [28].

Después leemos los ficheros de señales de ECG y de niveles de dolor correspondientes a cada paciente.

Una vez hemos leído los datos, realizamos un upsampling de la señales de niveles de dolor para igualar la frecuencia de muestreo a los 512 Hz

de las señales de biopotenciales. Seguidamente, se sincronizan los ECG con las señales indicativas del nivel de dolor; para ello se emplean las anotaciones temporales que indican en qué instantes se aplica dolor y de qué intensidad, de forma que cada muestra de las señales biomédicas corresponda a una muestra de las señales de niveles de dolor.

El resultado final de este preprocesado se puede ver en la tabla 5.1, donde `ident` da la identidad del paciente, `time` es el instante temporal donde se ha tomado la muestra en segundos, `ecg` es el nivel de la señal de ECG en milivoltios y `label` es el nivel actual de dolor.

ident	ecg	time	label
071309_w_21	-2031.32	0	0
071309_w_21	-2033.79	0.001953	0
071309_w_21	-2024.62	0.003907	0
071309_w_21	-2019.41	0.005860	0
071309_w_21	-2024.41	0.007813	0
071309_w_21	-2022.40	0.009766	0
071309_w_21	-2014.22	0.011719	0
071309_w_21	-2010.90	0.013672	0
071309_w_21	-2002.45	0.015625	0
071309_w_21	-2000.70	0.017578	0

Tabla 5.1: Ejemplo de filas del fichero de datos que se va a emplear tras el preprocesado. Contiene en cada fila una muestra del ECG de un paciente determinado y su nivel de dolor correspondiente.

Por último, esta tabla se guarda en un fichero `.mat`, para leerlo en MATLAB.

5.2. Tratamiento del ECG

Tras cargar el fichero `.mat` anteriormente generado, iniciamos el análisis de la señal de ECG usando un detector de complejo QRS de Pan y Tompkins, sobre la señal.

Como se puede ver en la figura 5.1 el detector de complejo QRS iden-

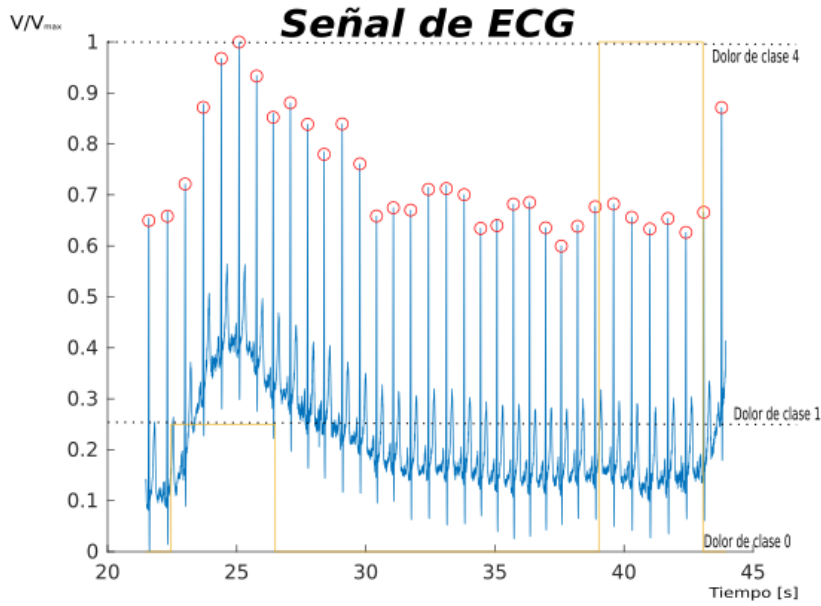


Figura 5.1: Ejemplo de ECG y sus niveles de dolor correspondientes, tras el paso por un detector de Pan-Tomkins.

tifica los picos de la onda R de cada latido del ECG los cuales permiten calcular la señal de variabilidad como la diferencia $RR_i - RR_{i-1}$, donde RR_i representa la onda R del latido i -ésimo. Esta señal de variabilidad se conoce con el nombre de tacómetro y es indicativa exclusivamente del ritmo cardiaco. Puesto que este es irregular, cada valor del tacómetro tiene asociado un instante temporal, pudiendo así conformar una señal que represente esta circunstancia:

$$d_{IF}^u = \sum_{k=1}^M t_k - t_{k-1} \delta(t - t_k) \quad (5.1)$$

que puede considerarse como las muestras resultantes del muestreo no uniforme de la señal de variabilidad cardiaca. Es necesario realizar un remuestreo uniforme de esta señal para obtener la señal de variabilidad cardiaca.

La obtención de la señal de HRV a partir de las diferencias de los complejos QRS resulta en una señal muestreada de manera no uniforme, por lo que, tenemos que volver a muestrear la señal con una frecuencia uniforme cuyo valor recomendado, según la literatura, es de 4 Hz [26], usando interpolación cúbica, También hacemos un remuestreo de las señales de

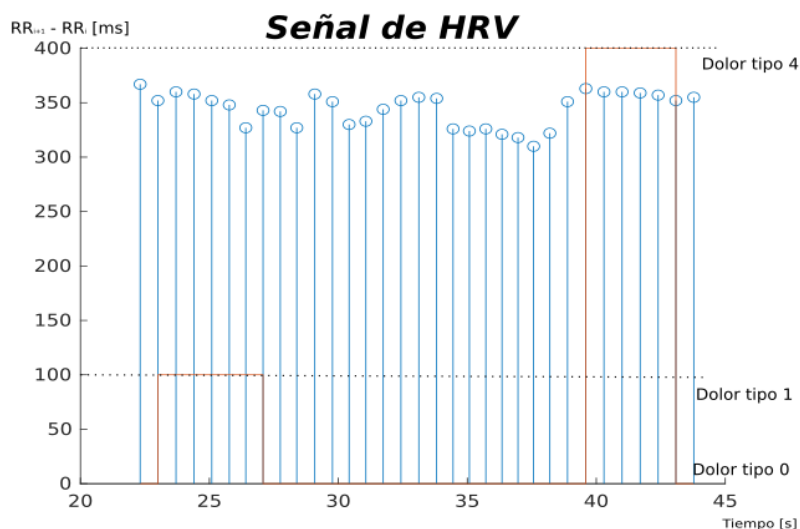


Figura 5.2: Ejemplo de una señal de HRV y sus niveles de dolor correspondientes.

tiempo y de niveles de dolor con la misma frecuencia de forma que exista una correspondencia biyectiva entre las muestras de las señales, para facilitar análisis posteriores.

Seguidamente, procedemos a crear ventanas sobre la señal de HRV de una duración variable de 3 a 7.5 segundos con una superposición del 50 %, sobre las que se calculan todos los parámetros necesarios para las fases siguientes: calculamos la entropías de Shannon, aproximada, muestral y de permutación sobre la señal de HRV. Además, obtenemos las señales del ECG asociadas a cada ventana, para calcular parámetros temporales, espectrales y de entropía sobre estas señales en el contexto de la ventana actual.

En concreto sobre la señal de ECG asociada a la ventana obtenemos los parámetros temporales SDNN, SDRR, PNN50, RRMSD, las densidades espectrales de potencia del complejo QRS y de las ondas P y T y el ratio entre estas potencias; por último, calculamos las entropías de Shannon, aproximada, muestral, espectral y de permutación sobre la señal de ECG

A cada ventana le asignamos el nivel de dolor que le corresponde, calculado como el máximo nivel de dolor de las muestras presentes en la ventana. Como la duración de las ventanas es menor de 8 segundos, que es el mínimo tiempo entre eventos de dolor consecutivos, la ventana solo puede contener muestras con un nivel de dolor distinto de cero, o muestras

sin dolor, nunca muestras con dos niveles de dolor distintos. Además, para ventanas de menos de 4 segundos, almacenamos, los cambios de niveles de dolor para descubrir si estos cambios son entrantes (la señal de HRV empieza con nivel de dolor 0 y se estimula dolor en esa ventana) o salientes (la señal HRV empieza con un nivel de dolor distinto de 0 y se deja de estimular dolor en esa ventana).

Por último, guardamos los parámetros, y las señales asociadas a cada ventana para una clasificación posterior. En este sentido guardamos 4 matrices distintas:

- Una matriz con los datos de las características que hemos calculado en las ventanas de HRV.

ID ejemplo	PermEnt HRV	SamEnt HRV	ApEnt HRV	ShEnt HRV
1	1.84	0.477	0.855	-127490166.4
2	1.57	0.477	0.858	-123266122.9
3	1.21	0.191	0.595	-127572878.8
4	1.14	0.124	0.625	-122160299.6
5	1.49	0.196	0.737	-128408473.8

ID ejemplo	PermEnt ECG	SamEnt ECG	ApEnt ECG	ShEnt ECG
1	2.219	0.337	0.031	-219822375465.3
2	2.174	0.349	0.034	-223831436332.5
3	2.159	0.349	0.033	-231196562869.0
4	2.198	0.344	0.034	-232631971879.8
5	2.212	0.359	0.036	-233594821043.3

ID ejemplo	SpectralEnt ECG	SDNN	SDRR	PNN50	RMSSD
1	-1.42e+15	658.4	30.40	1	659.0
2	-1.47e+15	657.3	32.07	1	658.0
3	-1.57e+15	664.4	25.14	1	664.9
4	-1.59e+15	654.2	25.31	1	654.7

ID ejemplo	SpectralEnt ECG	SDNN	SDRR	PNN50	RMSSD
5	-1.67e+15	670.3	39.64	1	671.3

ID ejemplo	Potencia QRS	Potencia ondas P y T	$\frac{Pot\ QRS}{Pot\ P\ y\ T}$
1	63383.5	66133.8	0.9584
2	64985.9	67040.6	0.9693
3	67070.0	63913.2	1.0493
4	67557.3	211378.9	0.3196
5	74420.2	208167.2	0.3575

Tabla 5.5: Ejemplo de filas pertenecientes a la matriz de características para el sistema de clasificación, que se usan como los distintos vectores de datos x_i para el entrenamiento y la inferencia.

- Una matriz con metadatos de las ventanas que hemos calculado.

ID ejemplo	Nivel de dolor	Porcentaje dolor	Segmento entrante
1	0	0	0
2	0	0	0
3	0	0	0
4	1	0.43	1
5	1	0.53	1

ID ejemplo	Segmento saliente	Tiempo inicio	Tiempo fin
1	0	7.02	14.27
2	0	10.77	18.02
3	0	14.52	21.77
4	0	18.27	25.52
5	0	22.02	29.27

Tabla 5.7: Ejemplo de filas pertenecientes a la tabla de características de los segmento, donde Nivel de dolor es la clase objetivo en el sistema de clasificación automática y

- Una matriz con los valores de HRV muestreados a 4 Hz en cada ventana. Se puede ver una imagen de una fila de la tabla en la figura 5.3.

ID ejemplo	HRV 1	HRV 2	HRV 3	HRV 4	HRV 5	HRV 6	HRV 7
1	321.21	374.18	367.79	339.43	326.10	337.88	358.12
2	340.80	335.17	328.03	320.16	315.00	314.38	317.28
3	355.22	356.18	355.78	354.25	350.81	347.55	345.35
4	322.99	316.61	314.04	315.60	320.37	325.19	328.70
5	367.29	365.14	360.60	356.46	354.23	355.17	358.21

Tabla 5.8: Ejemplo de muestras de señales de HRV correspondientes a una ventana

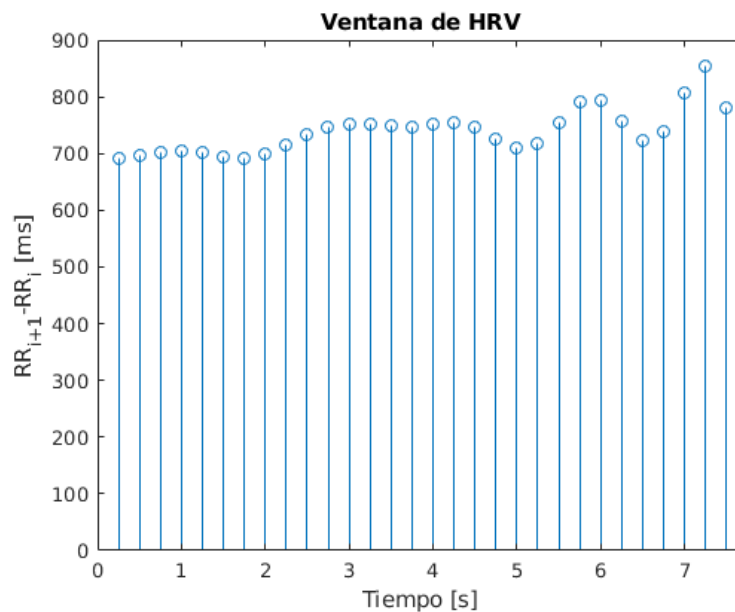


Figura 5.3: Ejemplo de una fila de la matriz de HRV.

- Una matriz con los segmentos originales de ECG asociados a cada ventana. Se puede ver representada una fila de la tabla en la figura 5.4.

ID ejemplo	ECG 1	ECG 2	ECG 3	ECG 4	ECG 5	ECG 6
1	-1087.77	-1176.50	-1250.32	-1404.36	-1647.80	-1860.51
2	-2406.70	-2408.88	-2414.97	-2422.81	-2420.33	-2412.40
3	-2491.48	-2495.22	-2448.06	-2358.72	-2275.56	-2180.46
4	-2498.58	-2506.53	-2507.41	-2514.32	-2508.96	-2512.54
5	-1121.58	-1149.11	-1121.77	-1137.06	-1133.11	-1131.57

Tabla 5.9: Ejemplo de muestras de señales de ECG correspondientes a una ventana

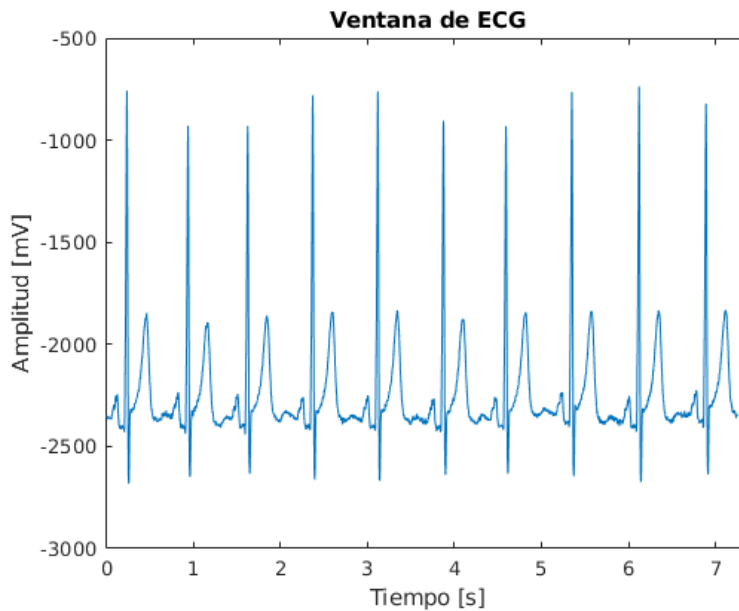


Figura 5.4: Ejemplo de una fila de la matriz de ECG.

5.3. Experimento de clasificación por SVM

En esta sección vamos a discutir los resultados de una clasificación mediante SVM de los segmentos que hemos extraído en señales de ECG con dolor (denotadas como +1) y sin dolor (denotadas como -1). Hemos elegido para la clasificación el conjunto de datos obtenido a partir de una división en segmentos de 7.5 segundos, ya que al tener más muestras en las señales de ECG y HRV los parámetros calculados son más significativos, y además, no se mezclan dos niveles de dolor en un segmento.

En todas las clasificaciones expuestas a continuación empleamos una partición del 60 % de los datos para entrenamiento y 40 % para validación, con distintos pacientes en ambas particiones. Además, siempre se normalizan los datos restándoles la media y dividiendo entre la desviación típica, para mejorar el rendimiento del algoritmo de clasificación.

En un primer experimento lanzamos la SVM con un kernel de tipo lineal sin optimizar hiperparámetros (con $BoxConstraint = 1$) ni realizar ningún tipo de alteración de la base de datos.

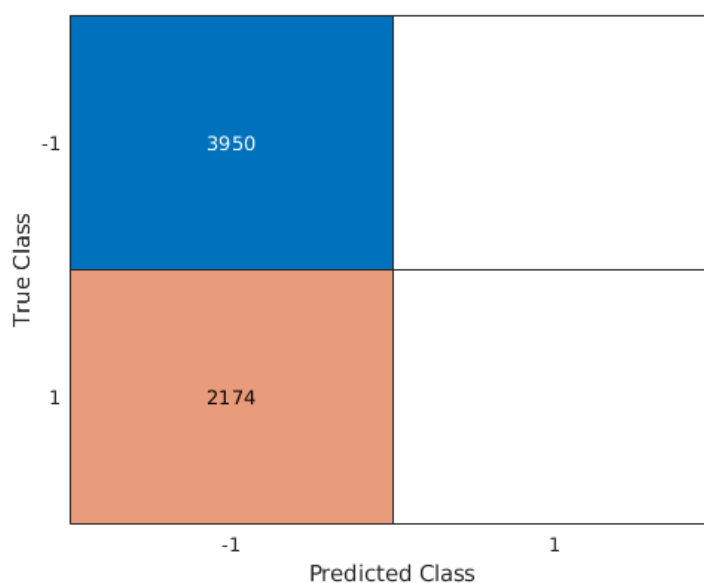


Figura 5.5: Matriz de confusión resultante de la clasificación de las características con un kernel lineal ($BoxConstraint = 1$) sin downsampling, permutación de los datos o optimización de hiperparámetros.

También lanzamos la SVM con kernel de tipo gaussiano sin optimizar hiperparámetros ($BoxConstraint = 1$ y $KernelScale = 1,1637$) ni realizar ningún tipo de alteración de la base de datos.

En las figuras 5.5 y 5.6, se pueden observar las matrices de confusión de estos experimentos, donde nos podemos dar cuenta que todos, o casi todos, los datos se clasifican como no dolor, por lo que debemos realizar un random downsampling de la clase mayoritaria (una eliminación aleatoria de elementos de la clase mayoritaria) en los datos de entrenamiento, de forma que balanceemos el número de ejemplos entre las clases cuando entrenamos el clasificador. Además, empleamos una permutación aleato-

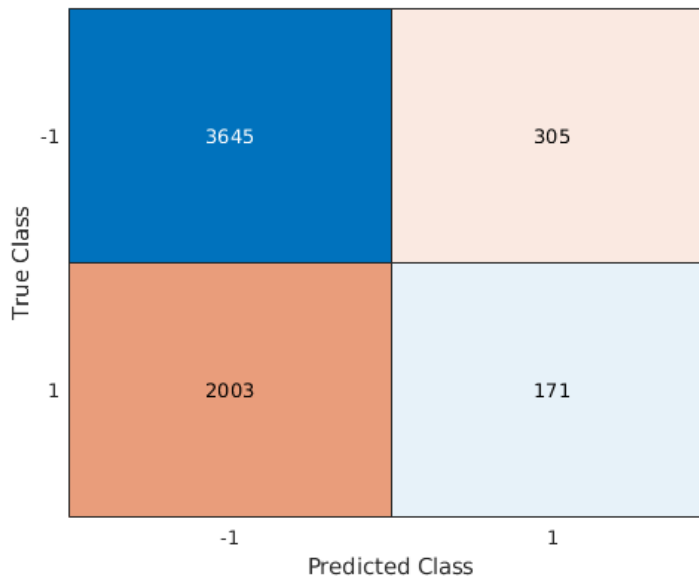


Figura 5.6: Matriz de confusión resultante de la clasificación de las características con un kernel gaussiano ($BoxConstraint = 1$ y $KernelScale = 1,1637$) sin downsampling, permutación de los datos o optimización de hiperparámetros.

ria en los datos de entrenamiento para evitar overfitting, ya que si no se realizara, el clasificador podría aprender patrones dependientes del orden de aparición de los datos.

Como segundo experimento ahora usamos un clasificador con kernel gaussiano con hiperparámetros $BoxConstraint = 1$ y $KernesScale = 1,171$, una vez realizado el downsampling y la permutación, con lo que, conseguimos que el rendimiento del clasificador mejore, como se ve en la matriz de confusión de la figura 5.7:

Ahora, para intentar mejorar aun más el rendimiento del clasificador, usamos optimización de hiperparámetros, en la que optimizamos el margen de error (en MATLAB $BoxConstraint$), el tipo de kernel (entre lineal, polinómico y gaussiano), el parámetro γ cuando usamos un kernel gaussiano (en MATLAB $KernelScale$) y el grado del polinomio (en MATLAB $PolynomialOrder$) en el caso de usar un kernel polinómico. Para ello MATLAB utiliza una k-fold validation, donde tras permutar aleatoriamente los datos de entrenamiento divide estos en 10 partes, de las cuales usa 9 para entrenamiento y la restante para el cálculo de una métrica de error de clasificación que intenta minimizar.

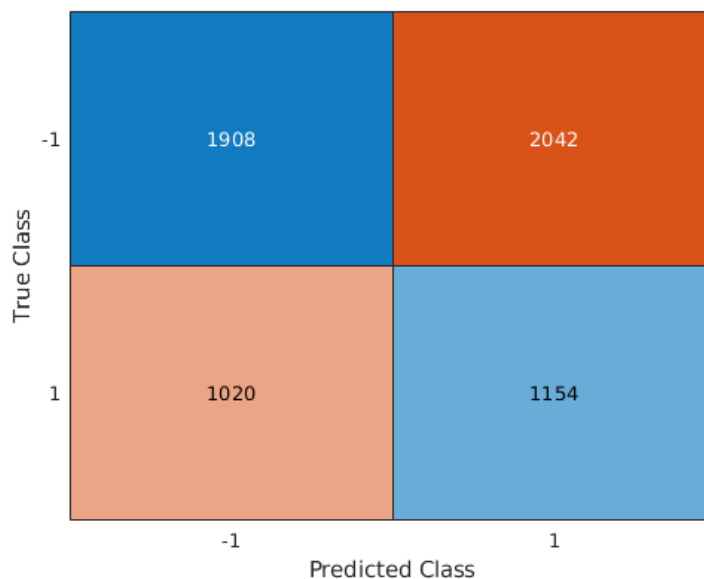


Figura 5.7: Matriz de confusión resultante de la clasificación de las características con un kernel gaussiano ($BoxConstraint = 1$ y $KernelScale = 1,171$) con downsampling y permutación de los datos, pero sin optimización de hiperparámetros.

Con todo esto obtenemos que en este caso el mejor clasificador de las características extraídas es un kernel de tipo polinómico de $PolynomialOrder = 2$ y $BoxConstraint = 0,25578$, con la matriz de confusión representada en la figura 5.8:

También probamos a incorporar los segmentos de HRV y de ECG que se han extraído junto a las características para mejorar los resultados de la clasificación, Las matrices de confusión correspondientes se pueden ver en las figuras 5.9, 5.10 y 5.11

Para terminar los distintos experimentos de clasificación, ahora probamos a solo usar las señales de HRV, ECG o una combinación de ambas (sin usar las características que hemos extraído). Las matrices de confusión de estos experimentos se pueden ver en las figuras 5.12, 5.13 y 5.14.

Por último, resumimos en una tabla los resultados de las distintos experimentos mediante los siguientes parámetros: Accuracy, Precision, Recall y F1 score.

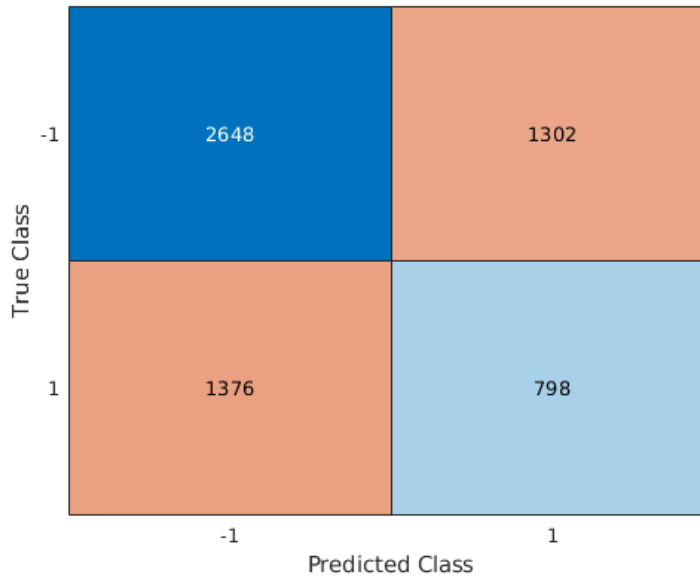


Figura 5.8: Matriz de confusión resultante de la clasificación de las características con un kernel polinómico de $PolynomialOrder = 2$ y $BoxConstraint = 0,25578$ con downsampling, permutación de los datos y optimización de hiperparámetros.

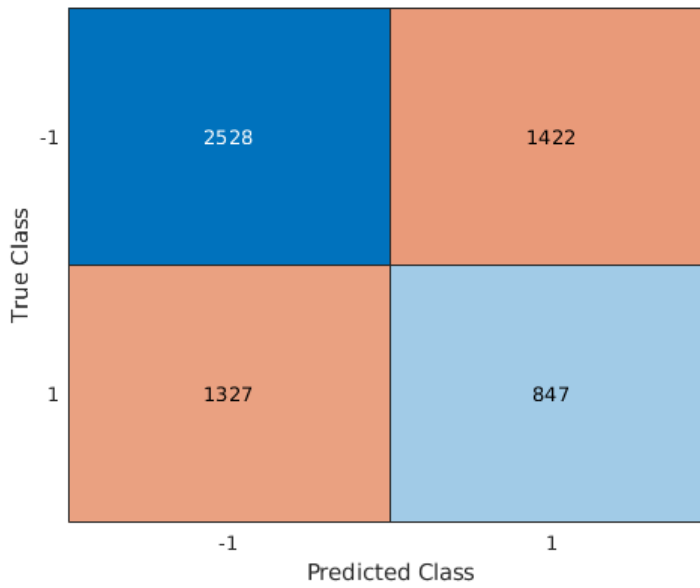


Figura 5.9: Matriz de confusión resultante de la clasificación de las características y la señal de HRV con un kernel gaussiano ($BoxConstraint = 152,12$ y $KernelScale = 10,586$) con downsampling, permutación de los datos y optimización de hiperparámetros.

Experimento	Accuracy	Precision	Recall	F1 score
Características sin downsample ni optimización kernel lineal	0.65	NaN	0	NaN
Características sin downsample ni optimización kernel rbf	0.6231	0.3592	0.0787	0.1291
Características con downsample sin optimización kernel rbf	0.5	0.3611	0.5308	0.4298
Características con downsample y optimización kernel polinómico	0.5627	0.38	0.3671	0.3734
Características y HRV con downsample y optimización kernel rbf	0.5511	0.3733	0.3896	0.3813
Características y ECG con downsample y optimización kernel rbf	0.4863	0.3548	0.5470	0.4863
Características, HRV y ECG con downsample y optimización kernel rbf	0.5304	0.3809	0.5166	0.4385
HRV con downsample y optimización kernel polinómico	0.5629	0.3903	0.4117	0.4007
ECG con downsample y optimización kernel rbf	0.5255	0.371	0.1844	0.4202
HRV y ECG con downsample y optimización kernel rbf	0.5178	0.3769	0.5483	0.4467

Tabla 5.10: Resultados de los distintos experimentos de clasificación.

Para el cálculo de estos parámetros, denotamos como TP a los ejemplos de la clase 1 clasificados correctamente, TN a los ejemplos de la clase -1 clasificados correctamente, FP a los elementos de la clase -1 clasificados como elementos de la clase 1 y FN a los elementos de la clase 1 clasificados como elementos de la clase -1 con lo que definimos:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (5.2)$$

$$Precision = \frac{TP}{TP + FP} \quad (5.3)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.4)$$

$$F1 \text{ score} = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \quad (5.5)$$

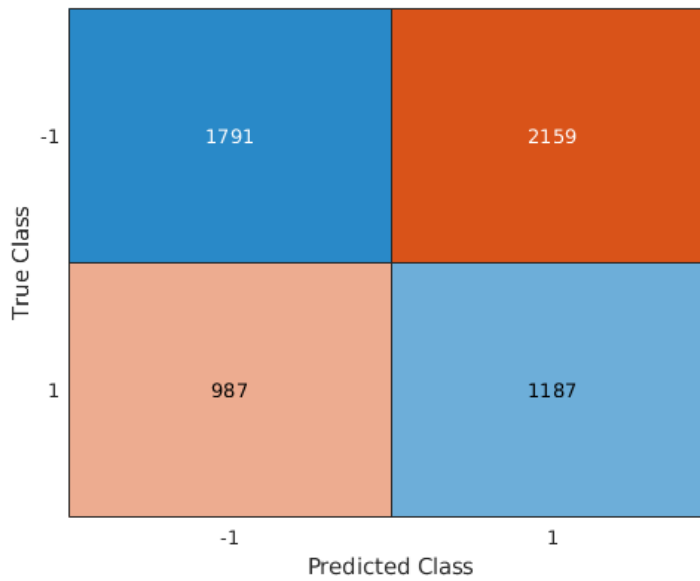


Figura 5.10: Matriz de confusión resultante de la clasificación de las características y la señal de ECG con un kernel gaussiano ($BoxConstraint = 186,9604$ y $KernelScale = 4,8396$) downsampling, permutación de los datos y optimización de hiperparámetros.

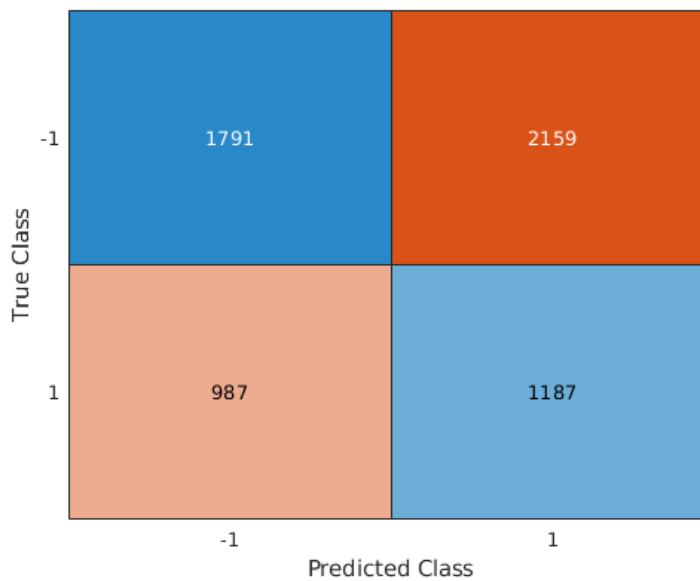


Figura 5.11: Matriz de confusión resultante de la clasificación de las características junto a las señales de ECG y HRV usando un kernel gaussiano con $BoxConstraint = 138,5564$ y $KernelScale = 87,8377$, con downsampling, permutación de los datos y optimización de hiperparámetros.

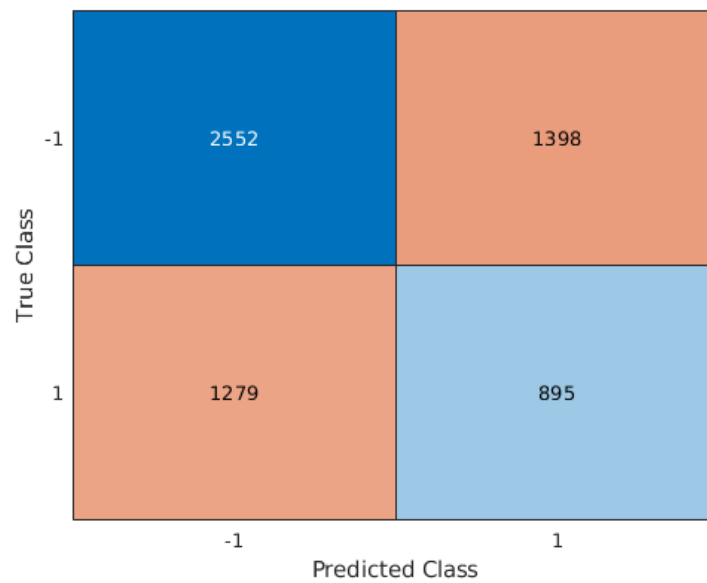


Figura 5.12: Matriz de confusión resultante de la clasificación de la señal de HRV con un kernel polinómico con $PolynomialOrder = 3$ $BoxConstraint = 0,0032379$, con downsampling, permutación de los datos y optimización de hiperparámetros

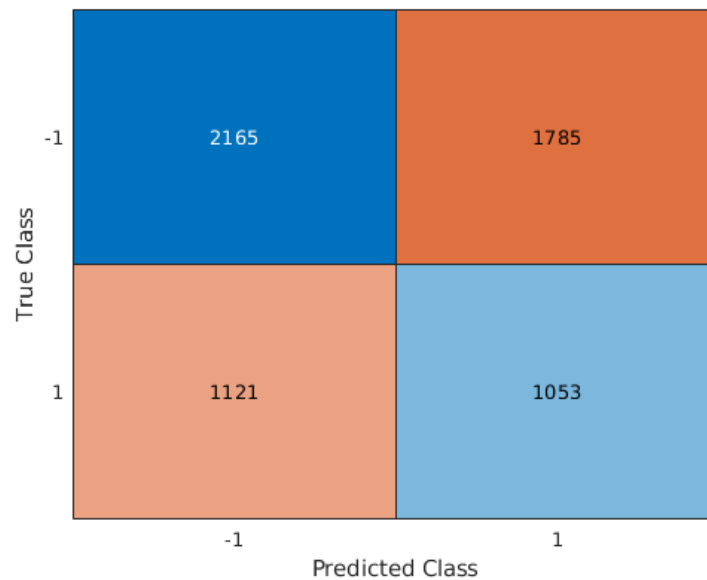


Figura 5.13: Matriz de confusión resultante de la clasificación de la señal de ECG con un kernel gaussiano ($BoxConstraint = 469,33$ y $KernelScale = 7,7286$) con downsampling, permutación de los datos y optimización de hiperparámetros.

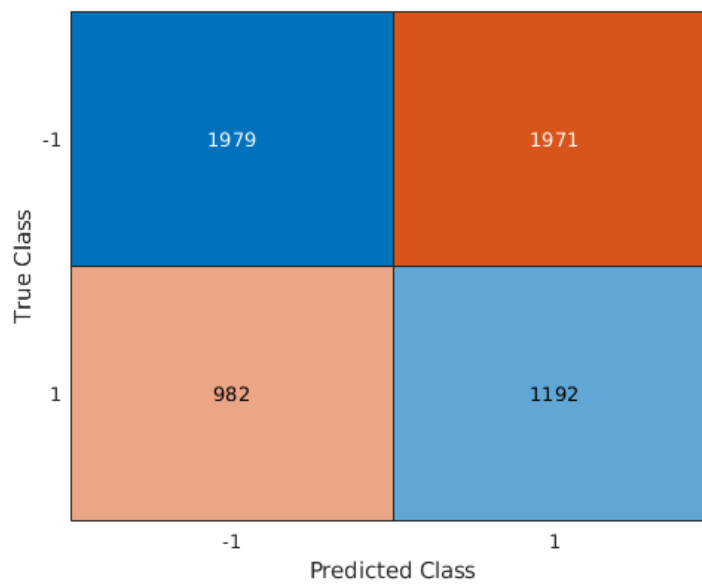


Figura 5.14: Matriz de confusión resultante de la clasificación de las señales de ECG y HRV con un kernel gaussiano con $BoxConstraint = 77,0683$ y $KernelScale = 59,6831$ con down-sampling, permutación de los datos y optimización de hiperparámetros.

Capítulo 6

Conclusiones y trabajo futuro

En este capítulo vamos a realizar una reflexión sobre el contenido del trabajo y las posibles líneas en la que la investigación podría continuar. Para ello nos vamos a centrar en el capítulo 5 cuando reflexionemos sobre las conclusiones extraídas y en los capítulos 4 y 5 cuando propongamos las posibles líneas de trabajo futuras.

6.1. Contribuciones y conclusiones

Los principales hitos que hemos alcanzado durante el desarrollo del trabajo son los siguientes:

- Se ha realizado un estudio de las señales de ECG y HRV y de los algoritmos usados para su análisis en la literatura.
- Se han implementado diversos algoritmos estándar para el análisis de las señales de ECG y HRV en MATLAB.
- Se ha localizado y solicitado acceso a la base de datos BioVid, que contiene señales de ECG en presencia de dolor para la validación de los algoritmos desarrollados.
- Se han probado los algoritmos desarrollados de extracción de características sobre las señales de la base de datos BioVid.
- Se ha desarrollado un sistema de clasificación para la detección de dolor a partir de las características extraídas de la base de datos.

Consideramos que se han cumplido de forma satisfactoria los objetivos relacionados con el estudio de las señales y con el desarrollo y validación de distintos algoritmos de análisis de las señales de ECG y HRV sobre una base de datos, ya que hemos estudiado, implementado y probado de manera exitosa múltiples métodos de análisis para las señales bajo estudio, usando para ello las señales contenidas en la base de datos BioVid.

También se ha desarrollado una etapa de clasificación alimentada por los predictores que se han estudiado en el proyecto, aunque su rendimiento no es satisfactorio, ya que la F1 score del mejor clasificador es menor a 0.5. Creemos que los motivos que pueden dar lugar a esta alta tasa de error se pueden resumir en las tres siguientes causas:

1. Existen errores en la implementación de los programas realizados, en la preparación de los datos o en la clasificación.
2. Hay un error conceptual en el planteamiento teórico del trabajo.
3. La base de datos que estamos usando para este análisis no es adecuada.

La causa 1 se ha intentado evitar mediante la inspección visual de las señales antes de calcular los parámetros, el uso de funciones ya implementadas en MATLAB siempre que ha sido posible (por ejemplo para el cálculo del SDNN, el SDRR o el RMSSD) y la comprobación de resultados de nuestra implementación de las funciones frente a otras de referencia.

Nos hemos planteado la razón 2, dado que los mecanismos biológicos del dolor se desconocen en gran medida, y aunque hay una amplia evidencia sobre la existencia de una relación entre el ritmo cardiaco y el dolor, esta no es la única manifestación de la presencia de dolor en el cuerpo, por lo que es posible que la influencia del dolor en el HRV no sea lo suficientemente relevante como para lograr una detección fiable de dolor, aunque los resultados que se han obtenido en [1], donde se han usado modelos de regresión para relacionar el dolor sufrido por pacientes durante operaciones y distintos parámetros de la señal de ECG, obteniendo modelos resultantes con una ROC superior a 0.75, parece indicar lo contrario.

La razón 3 es probable, ya que de acuerdo con [25] las únicas medidas temporales o espectrales de ritmo cardiaco aplicables en un periodo de

tiempo tan corto, como el disponible en la base de datos BioVid (menos de 10 segundos), son las relacionadas con la media (como pueden ser el SDDN y el RMSDD), mientras que todas las demás métricas necesitan una duración de las señales de al menos 30 segundos. Además, en las medidas calculadas sobre el HRV estamos usando, a lo sumo, 25 muestras por segmento. Las medidas de entropía puede que tampoco tengan una fiabilidad elevada, ya que miden la regularidad de las señales, y debido a la corta duración, sus resultados pueden no ser altamente significativos, dado que cualquier pequeña variación en las señales puede influir en gran medida en los resultados.

6.2. Trabajo futuro

Las principales líneas futuras de trabajo podrían ser las siguientes:

- Investigación de otros algoritmos para caracterizar las señales de ECG. En este trabajo hemos decidido no implementar varias técnicas utilizadas en la literatura, ya que su uso requiere de segmentos de ECG de larga duración con presencia de dolor, como por ejemplo, la comparación de densidades espectrales de potencia del HRV o el uso de Detrended Fluctuation Analysis, que se podrían investigar para un futuro trabajo.
- Aplicación de los principios desarrollados en este trabajo a otra base de datos, con periodos de dolor más largos. Ya que sospechamos que una de las posibles fuentes de error es la corta duración de los segmentos con y sin dolor, se podría buscar otra base de datos con aplicación de dolor más prolongada. Esto puede ser no viable por razones bioéticas, ya que para lograr una cierta fiabilidad se tendría que aplicar dolor a individuos durante largo periodos de tiempo.
- Uso de otros métodos de clasificación usando la misma base de datos. En este trabajo por restricciones temporales solo hemos considerado un método de clasificación. Una posible vía de investigación sería estudiar el uso de otros algoritmos como árboles extremos de regresión

o redes neuronales profundas, para intentar mejorar el rendimiento del clasificador.

Bibliografía

- [1] Tricia Adjei y col. «Pain Prediction from ECG in Vascular Surgery». En: *IEEE Journal of Translational Engineering in Health and Medicine* PP (sep. de 2017), págs. 1-1. DOI: 10.1109/JTEHM.2017.2734647.
- [2] Adnav (Vanda Machová). *Einthovenův trojúhelník*. [Imagen Einthoven.png sin modificar con licencia, CC BY 3.0 CZ]. 2011. URL: <https://www.wikiskripta.eu/w/Soubor:Einthoven.png> (visitado 17-06-2021).
- [3] Carmen Mabel Arroyo-Novoa y col. «Acute Wound Pain: Gaining a Better Understanding». En: *Advances in Skin & Wound Care* 22.8 (2009). ISSN: 1527-7941. URL: https://journals.lww.com/aswcjournal/Fulltext/2009/08000/Acute_Wound_Pain__Gaining_a_Better_Understanding.12.aspx.
- [4] C. Bandt y B. Pompe. «Permutation entropy: a natural complexity measure for time series.» En: *Physical review letters* 88 17 (2002), pág. 174102.
- [5] Blausen Medical Communications, Inc. *Superficial Heart Anatomy (Anterior View)*. [Imagen Blausen 0451 Heart Anterior.png sin modificar con licencia, CC BY 3.0]. 2013. URL: https://commons.wikimedia.org/wiki/File:Blausen_0451_Heart_Anterior.png (visitado 17-06-2021).
- [6] Blausen.com staff. *Superficial Heart Anatomy (Posterior)*. [Imagen Blausen 0456 Heart Posterior.png sin modificar con licencia, CC BY 3.0]. 2014. DOI: 10.15347/wjm/2014.010. URL: https://commons.wikimedia.org/wiki/File:Blausen_0456_Heart_Posterior.png (visitado 17-06-2021).
- [7] Marta Borowska. «Entropy-Based Algorithms in the Analysis of Biomedical Signals». En: *Studies in Logic, Grammar and Rhetoric* 43 (dic. de 2015). DOI: 10.1515/slgr-2015-0039.
- [8] Ping Cao y col. «Preprocessing Unevenly Sampled RR Interval Signals to Enhance Estimation of Heart Rate Deceleration and Acceleration Capacities in Discriminating Chronic Heart Failure Patients from Healthy Controls». En: *Computational and Mathematical Methods in Medicine* 2020 (2020). Ed. por Guy Carrault, pág. 9763826. ISSN: 1748-670X. DOI: 10.1155/2020/9763826.
- [9] Enrique Carmona. «tutorial sobre maquinas de vectores soporte (SVM)». En: *UNED, Consultada en http://www.ia.uned.es/~ejcarmona/publicaciones/[2013-Carmona] % 20SVM. pdf (fecha de consulta 01-07-2021)* (2014).

- [10] Rudolf Clausius. *Mechanical Theory of Heat*. 1850-1865.
- [11] Corinna Cortes y Vladimir Vapnik. «Support-vector networks». En: *Machine Learning* 20.3 (1995), págs. 273-297. ISSN: 1573-0565. DOI: 10.1007/BF00994018.
- [12] Alfonso Delgado-Bonal y Alexander Marshak. «Approximate Entropy and Sample Entropy: A Comprehensive Tutorial». En: *Entropy* 21.6 (2019). ISSN: 1099-4300. DOI: 10.3390/e21060541. URL: <https://www.mdpi.com/1099-4300/21/6/541>.
- [13] Mathias Haefeli y Achim Elfering. «Pain assessment». En: *European Spine Journal* 15.1 (2006), S17-S24. ISSN: 1432-0932. DOI: 10.1007/s00586-005-1044-x.
- [14] Marek Malik. «Heart rate variability: Standards of measurement, physiological interpretation, and clinical use». En: *Circulation* 93 (mar. de 1996), págs. 1043-1065. DOI: 10.1161/01.CIR.93.5.1043.
- [15] Chris Martini y col. «A Randomized Controlled Trial on the Effect of Tapentadol and Morphine on Conditioned Pain Modulation in Healthy Volunteers». En: *PLoS one* 10 (jun. de 2015), e0128997. DOI: 10.1371/journal.pone.0128997.
- [16] Mikael Häggström. *Placement of the precordial leads in electrocardiography*. [Imagen Precordial leads in ECG.png sin modificar con licencia, CC0 1.0 Universal (CC0 1.0) Dedicación de Dominio Público]. 2012. URL: https://commons.wikimedia.org/wiki/File:Precordial_leads_in_ECG.png (visitado 17-06-2021).
- [17] Luis Montesinos, Rossana Castaldo y Leandro Pecchia. «On the use of approximate entropy and sample entropy with centre of pressure time-series». En: *Journal of NeuroEngineering and Rehabilitation* 15.1 (2018), pág. 116. ISSN: 1743-0003. DOI: 10.1186/s12984-018-0465-9.
- [18] Jiapu Pan y Willis J. Tompkins. «A Real-Time QRS Detection Algorithm». En: *IEEE Transactions on Biomedical Engineering* BME-32.3 (1985), págs. 230-236. DOI: 10.1109/TBME.1985.325532.
- [19] S M Pincus. «Approximate entropy as a measure of system complexity.» En: *Proceedings of the National Academy of Sciences* 88.6 (1991), págs. 2297-2301. ISSN: 0027-8424. DOI: 10.1073/pnas.88.6.2297. eprint: <https://www.pnas.org/content/88/6/2297.full.pdf>. URL: <https://www.pnas.org/content/88/6/2297>.
- [20] D. Reddi, N. Curran y R. Stephens. «An introduction to pain pathways and mechanisms». En: *Br J Hosp Med (Lond)* 74 Suppl 12 (2013), págs. C188-191.
- [21] Joshua Richman y Joseph Moorman. «Physiological Time-Series Analysis Using Approximate Entropy and Sample Entropy». En: *American journal of physiology. Heart and circulatory physiology* 278 (jul. de 2000), H2039-49. DOI: 10.1152/ajpheart.2000.278.6.H2039.

- [22] Joshua S. Richman y J. Randall Moorman. «Physiological time-series analysis using approximate entropy and sample entropy». En: *American Journal of Physiology-Heart and Circulatory Physiology* 278.6 (2000). PMID: 10843903, H2039-H2049. DOI: 10.1152/ajpheart.2000.278.6.H2039. eprint: <https://doi.org/10.1152/ajpheart.2000.278.6.H2039>. URL: <https://doi.org/10.1152/ajpheart.2000.278.6.H2039>.
- [23] Maik Riedl, Andreas Müller y Niels Wessel. «Practical considerations of permutation entropy: A tutorial review». En: *The European Physical Journal Special Topics* 222 (jun. de 2013). DOI: 10.1140/epjst/e2013-01862-7.
- [24] Roberto Sassi y col. «Advances in heart rate variability signal analysis: joint position statement by the e-Cardiology ESC Working Group and the European Heart Rhythm Association co-endorsed by the Asia Pacific Heart Rhythm Society». En: *EP Europace* 17.9 (jul. de 2015), págs. 1341-1353. ISSN: 1099-5129. DOI: 10.1093/europace/euv015. eprint: <https://academic.oup.com/europace/article-pdf/17/9/1341/1747851/euv015.pdf>. URL: <https://doi.org/10.1093/europace/euv015>.
- [25] Fred Shaffer y J P Ginsberg. «An Overview of Heart Rate Variability Metrics and Norms». En: *Frontiers in public health* 5 (sep. de 2017), págs. 258-258. ISSN: 2296-2565 2296-2565. DOI: 10.3389/fpubh.2017.00258. URL: <https://pubmed.ncbi.nlm.nih.gov/29034226;%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5624990>.
- [26] D Singh, Vinod Kumar y S Saxena. «Sampling frequency of the RR interval time series for spectral analysis of heart rate variability». En: *Journal of medical engineering & technology* 28 (nov. de 2004), págs. 263-72. DOI: 10.1080/03091900410001662350.
- [27] Leif Sörnmo y Pablo Laguna. «Bioelectrical Signal Processing in Cardiac and Neurological Applications». En: ed. por Leif Sörnmo y Pablo Laguna. *Biomedical Engineering*. Burlington: Academic Press, 2005. ISBN: 978-0-12-437552-9. DOI: <https://doi.org/10.1016/B978-0-12-437552-9.X5000-4>.
- [28] *The BioVid Heat Pain Database*. Mayo de 2020. URL: <http://www.iikt.ovgu.de/BioVid.html> (visitado 30-09-2020).
- [29] National Cancer Institute. U. S. National Institutes of Health. *SEER Training Modules, Introduction to the Nervous System*. URL: <https://training.seer.cancer.gov/anatomy/nervous> (visitado 17-06-2021).
- [30] Jake VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data*. 1st. O'Reilly Media, Inc., 2016. ISBN: 1491912057.
- [31] Manuel Blanco Velasco. *Nociones acerca del electrocardiograma*. Publicado en XI Semana de la Ciencia, UAH. 2018.
- [32] Manuel Blanco Velasco. *Nociones básicas de electrocardiografía*. Apuntes de la asignatura Tratamiento de señales biomédicas UAH. 2018.

- [33] S. Walter y col. «The biovid heat pain database data for the advancement and systematic validation of an automated pain recognition system». En: *2013 IEEE International Conference on Cybernetics (CYBCO)*. 2013, págs. 128-131. DOI: 10.1109/CYBCConf.2013.6617456.
- [34] Philipp Werner, Ayoub Al-Hamadi y Steffen Walter. «Analysis of Facial Expressiveness During Experimentally Induced Heat Pain». En: oct. de 2017. DOI: 10.1109/ACIIW.2017.8272610.
- [35] Philipp Werner y col. «Towards Pain Monitoring: Facial Expression, Head Pose, a new Database, an Automatic System and Remaining Challenges». En: sep. de 2013. DOI: 10.5244/C.27.119.
- [36] Wikipedia user "The Emirr". *Location Of Nerves*. [Imagen TE-Nervous system diagram.svg sin modificar con licencia, CC BY 3.0]. 2010. URL: https://commons.wikimedia.org/wiki/File:TE-Nervous_system_diagram.svg (visitado 17-06-2021).
- [37] Ping Yao y col. «Sample Entropy-Based Approach to Evaluate the Stability of Double-Wire Pulsed MIG Welding». En: *Mathematical Problems in Engineering* 2014 (2014). Ed. por Hamid Reza Karimi, pág. 869631. ISSN: 1024-123X. DOI: 10.1155/2014/869631.
- [38] Jennifer Yentes y col. «The Appropriate Use of Approximate Entropy and Sample Entropy with Short Data Sets». En: *Annals of biomedical engineering* 41 (oct. de 2012). DOI: 10.1007/s10439-012-0668-3.

Capítulo 7

Anexo: Código fuente

```
function [m] = sdnrr(rr_int)
    % Calculate the SDNN of an HRV signal.
    m = mean(rr_int);
end
```

Listado 1: Implementación de SDNN

```
function [sd] = sdrdrr(rr_int)
    % Calculate the SDRR of an HRV signal
    sd = std(rr_int);
end
```

Listado 2: Implementación de SDRR

```
function [rr_mean_square] = rr_rms(rr_int)
    % Calculate the RMSSD of an HRV signal
    rr_mean_square = rms(rr_int);
end
```

Listado 3: Implementación de RMSSD

```
function [nn50, pnn50] = rr50(rr_int)
    % Calculate the pnn50 of an HRV signal
    nn50 = sum(diff(rr_int) > 50);
    pnn50 = sum(diff(rr_int) > 50) / length(diff(rr_int));
end
```

Listado 4: Implementación de pnn50

```

function [power_ptwave, power_QRS, ratioHFLF] =
    ↪ power_bands_ecg(x_n, freq_sample)
    % Calculate the PSD of the bands on the signals
    % Arguments:
    % x_n: signal
    % freq_sample: sample freq. Default to 512 Hz

    if nargin < 2
        freq_sample = 512;
    end
    fs = freq_sample;

    % Eliminate the DC component, as it has very high power
    x_n_detrended = x_n - mean(x_n);

    % From the documentation: fs must be the fifth input to
    ↪ pwelch. To input a
    % sample rate and still use the default values of the
    ↪ preceding optional
    % arguments, specify these arguments as empty, [].
    [psd, f] = pwelch(x_n_detrended, [], [], [], fs);

    % Find the first index that oks the condition
    index_f1 = find(f >= 0, 1, 'first');
    index_f2 = find(f >= 10, 1, 'first');
    index_f3 = find(f >= 10, 1, 'first');
    index_f4 = find(f >= 40, 1, 'first');

    psd_f1_f2 = psd(index_f1:index_f2);
    psd_f3_f4 = psd(index_f3:index_f4);

    power_ptwave = sum(psd_f1_f2);

```



```

power_QRS = sum(psd_f3_f4);
ratioHFLF = power_QRS/power_ptwave;
end

```

Listado 5: Implementación del análisis de la densidad espectral de potencia del ECG

```

function [ent] = sh_ent(x)
    % Implementation of Shannon entropy
    % The entropy is calculated as the shannon_ent =
    % ↪ sum(log2(1/p)*p)
    % where p is the probability of each symbol in the
    % ↪ alphabet of the
    % time series.
    if nargin ~= 1
        error('expected 1 argument');
    end

    unique_elems = unique(x);
    probs = zeros(1,length(unique_elems));
    for elem = 1:length(unique_elems)
        probs(elem) = sum(unique_elems(elem) == x);
    end

    p = probs ./ length(x);

    log_p = log2(p);
    aux = p .* log_p;
    ent = -sum(aux);
end

```

Listado 6: Implementación de la entropía de Shannon

```

function [ent] = sp_ent(x,fs,method,normalize)
    % Calculate the spectral entropy of a signal
    % Arguments:

```

```

% x: signal
% fs: sampling freq. Default -1, use the default value
    ↪ to calculate periodogram.
% method: Method to calculate the PSD. Values accepted:
    ↪ 'welch' or 'periodogram'.
%           If 'welch' we use the pwelch function
%           If 'periodogram' we use the periodogram
    ↪ funtion
%           Default: 'welch'
% normalize: If true divide the entropy by length of
    ↪ PSD to compare different segments. Default: false

% Note: default to a fs of -1 to indicate the default
    ↪ value of the
% parameters of the MATLAB functions.
if nargin < 2
    fs = -1;
    method = 'welch';
    normalize = 0;
end

if nargin < 3
    method = 'welch';
    normalize = 0;
end

if nargin < 4
    normalize = 0;
end

is_welch = strcmp(method, 'welch');
is_fft = strcmp(method, 'periodogram');
if ~(is_welch || is_fft)

```

```

        error("Method invalid, returning 0 for the entropy
        ↪ ");
    end

    if is_fft
        if fs == -1
            x_omega = periodogram(x);
        else
            x_omega = periodogram(x,fs);
        end
    end

    if is_welch
        if fs == -1
            x_omega = pwelch(x);
        else
            x_omega = pwelch(x,fs);
        end
    end

    ent_psd = sh_ent(x_omega);

    if normalize
        ent = ent_psd ./ log2(length(x_omega));
    else
        ent = ent_psd;
    end

end

```

Listado 7: Implementación de la entropía espectral

```

function [appent, ratio] = appent(u, m, lag, tol)
% Calculate the approximate entropy of the signal.
% Parameters:

```

```
% - u: signal
% - m: state space dimension (how long the vectors should
↪ be).
% - lag: state space lag (how many samples should we skip
↪ on each iteration to form the vectors)
% - r: filter (where is the limit of similiraty iff diff <=
↪ r then consider a match)
if nargin < 2
    m = 3;
    tol = 0.2;
    lag = 1;
    r = tol * std(u);
end

if nargin < 3
    tol = 0.2;
    lag = 1;
    r = tol * std(u);
end

if nargin < 4
    tol = 0.2;
    r = tol * std(u);
end

if nargin == 4
    r = tol;
end

N = length(u) - m;

% initiate variables to zero
pos = zeros(1,N);
mat = zeros(1,N);
```

```
for i = 1:N
    posibles_loop = 0;
    matches_loop = 0;
    % Look for matches_loop in the whole series
    for j = 1:N
        % This simulate an state space reconstruction
        ↪ comparing all the
        % differences between the coordinates, so the lag is
        ↪ to skip n elements
        % on each iterations of the vectors
        for k = 1:lag:(m+1)
            % Calculate the difference between the
            ↪ coordinates
            dif = abs(max(u(i+k-1) - u(j+k-1)));

            if dif > r
                % If the difference is greater than the
                ↪ maximum is it not possible to have a
                ↪ match.
                break
            end

            if k < m
                % first possibility: it does not fit the
                ↪ condition for k < m
                continue;
            elseif k == m
                % second possibility: check if it is a
                ↪ possible block or not, k = m
                posibles_loop = posibles_loop + 1;
            elseif k > m
                % third possibility: check if it is a match
                ↪ block or not, k = m+1
```

```

        matches_loop = matches_loop + 1;
    end
end
end
pos(i) = posibles_loop; % save the probabilities for
    ↪ each template vector i
mat(i) = matches_loop; % save the probabilities for
    ↪ each template vector i
end

posibles = sum(pos);
matches = sum(mat);

ratio = matches ./ posibles;
appent = 0;
for i = 1:N
    appent = appent + log(ratio);
end

appent = -appent/N;

if isinf(appent)
    appent = log(N-m)+log(N-m-1)-log(2);
end
end

```

Listado 8: Implementación de la entropía aproximada

```

function [sampen, ratio] = sament(u, m, lag, tol)
% Calculate the sample entropy of the signal.
% Parameters:
% - u: signal
% - m: state space dimension (how long the vectors should
    ↪ be).

```

```
% - lag: state space lag (how many samples should we skip  
↳ on each iteration to form the vectors)  
% - r: filter (where is the limit of similarity iff diff <=  
↳ r then consider a match)  
  
% Note: SampEn has the following limits:  
% - Lower bound: 0  
% - Upper bound: log(N-m)+log(N-m-1)-log(2)  
  
if nargin < 2  
    m = 3;  
    tol = 0.2;  
    lag = 1;  
    r = tol * std(u);  
end  
  
if nargin < 3  
    tol = 0.2;  
    lag = 1;  
    r = tol * std(u);  
end  
  
if nargin < 4  
    tol = 0.2;  
    r = tol * std(u);  
end  
  
if nargin == 4  
    % If we have an explicit tolerance we do not multiply  
    r = tol;  
end  
  
N = length(u) - m;
```

```
% initiate variables to zero
pos = zeros(1,N);
mat = zeros(1,N);

for i = 1:N
    posibles_loop = 0;
    matches_loop = 0;
    % Look for matches_loop in the whole series
    for j = 1:N
        if i == j; break; end % Avoid self matches_loop

        % This simulate an state space reconstruction
        ↪ comparing all the
        % differences between the coordinates, so the lag is
        ↪ to skip n elements
        % on each iterations of the vectors
        for k = 1:lag:(m+1)
            % Calculate the difference between the
            ↪ coordinates
            dif = abs(max(u(i+k-1) - u(j+k-1)));

            if dif > r
                % If the difference is greater than the
                ↪ maximum is it not possible to have a
                ↪ match.
                break
            end

            if k < m
                % first possibility: it does not fit the
                ↪ condition for k < m
                continue;
            elseif k == m
```

```

        % second possibility: check if it is a
        ↪ possible block or not, k = m
        posibles_loop = posibles_loop + 1;
    elseif k > m
        % third possibility: check if it is a match
        ↪ block or not, k = m+1
        matches_loop = matches_loop + 1;
    end
end
end
pos(i) = posibles_loop; % save the probabilities for
↪ each template vector i
mat(i) = matches_loop; % save the probabilities for
↪ each template vector i
end

posibles = sum(pos);
matches = sum(mat);

ratio = matches./posibles;
sampen = -log(ratio);

if isinf(sampen)
    sampen = log(N-m)+log(N-m-1)-log(2);
end

end

```

Listado 9: Implementación de la entropía muestral

```

function [ent] = sh_ent(x)
    % Implementation of Shannon entropy
    % The entropy is calculated as the shannon_ent =
    ↪ sum(log2(1/p)*p)

```

```

% where p is the probability of each symbol in the
↪ alphabet of the
% time series.
if nargin ~= 1
    error('expected 1 argument');
end

unique_elems = unique(x);
probs = zeros(1,length(unique_elems));
for elem = 1:length(unique_elems)
    probs(elem) = sum(unique_elems(elem) == x);
end

p = probs ./ length(x);

log_p = log2(p);
aux = p .* log_p;
ent = -sum(aux);
end

```

Listado 10: Implementación de la entropía de permutación

```

% Perform the clasification
% load the data
dataset = load("ml_matrix_hrv_75_sec_full.mat");

% Use parameters
use_chars = false;
% Use HRV
use_hrv = false;
% Use ECG
use_ecg = true;
% Do a downsample
downsample = false;

```

```

% How much the ratio of pain-nopain should be. With one
↳ len(pain)==len(npain)
scale_factor = 1;
% Perform hyperparameter optimization
optimize = true;
% Optimize the kernel, not only the parameters
optimize_model = true;

% Note: props matrix fields are
% pain_level, percent_pain, starts, ends, time_begin,
↳ time_end
%      1           2           3           4           5
↳ 6

% Here we filter and replace the pain level indicator by a
↳ one and the not pain indicator by a -1
pain_filter = dataset.props(:,1) == 1 ...
    | dataset.props(:,1) == 2 ...
    | dataset.props(:,1) == 3 ...
    | dataset.props(:,1) == 4;

dataset.props(pain_filter, 1) = 1;

nopain_filter = dataset.props(:,1) == 0;
dataset.props(nopain_filter, 1) = -1;

% Make filters to select only the signals we are interested
↳ in:
% Here we filter based in characteristics of the segments
% If you want to select all make an array of 1 column and
% length(dataset.props) array.
% If not you can make a filter as based in the previous
↳ example or-ing the
% various characteristics you want to select

```

```
% Pain filters: Select signals with all the characteristics  
↳ that are pain  
% segments we are interested in.  
% Only select the signals with pain class 1, (For example  
↳ to exclude  
% certain pain level)  
isp_filter = dataset.props(:,1) == 1;  
  
% Select only signals with a certain id  
idp_filter = ones(length(dataset.identifiers_string),1);  
  
% Select signal that begins or ending the pain segment on  
↳ the window.  
beg_filter = ones(length(dataset.props),1);  
end_filter = ones(length(dataset.props),1);  
  
% Select percent pain.  
percentpain_filter = dataset.props(:,2) >=.50; % Pain  
↳ segments with more than 50 % pain  
  
p_filter = isp_filter & idp_filter & beg_filter & end_filter  
↳ & percentpain_filter;  
  
% Non pain filters: Select the non pain segmentes we are  
↳ interested in.  
% Only select the signals with pain class -1, (For example  
↳ to exclude  
% selecting too many non pain segments)  
isnp_filter = dataset.props(:,1) == -1;  
  
% Select only signals with a certain id  
idnp_filter = ones(length(dataset.identifiers_string), 1);
```

```
np_filter = isnp_filter & idnp_filter;

filter = p_filter | np_filter;

ids = dataset.identifiers_string(filter,:);
hrv = dataset.HRV_signal(filter,:);
ecg = dataset.ecg_matrix(filter,:);
sprops = dataset.predictors(filter,:);
schar = dataset.props(filter,:);
classes = schar(:,1);

patients = unique(ids);
num_patients = length(patients);
num_train_patients = round(length(patients) * 60/100);
train_patients = patients(1:num_train_patients);
num_test_patients = num_patients - num_train_patients;
test_patients = patients(num_train_patients+1:end);

filter_train_patients = zeros(length(ids),1);

for i= 1:length(train_patients)
    filter_train_patients = filter_train_patients | (ids ==
        ↪ train_patients(i));
end

filter_test_patients = zeros(length(ids),1);

for i= 1:length(test_patients)
    filter_test_patients = filter_test_patients | (ids ==
        ↪ test_patients(i));
end

train_data = sprops(filter_train_patients,:);
test_data = sprops(filter_test_patients,:);
```

```
if use_chars == false
    train_data = [];
    test_data = [];
end

if use_hrv == true
    hrv_train = hrv(filter_train_patients,:);
    hrv_test = hrv(filter_test_patients,:);

    train_data = [train_data, hrv_train];
    test_data = [test_data, hrv_test];
end

if use_ecg == true
    ecg_train = ecg(filter_train_patients,:);
    ecg_test = ecg(filter_test_patients,:);

    train_data = [train_data, ecg_train];
    test_data = [test_data, ecg_test];
end

train_classes = classes(filter_train_patients,:);
test_classes = classes(filter_test_patients,:);

train_data_downsampled = train_data;
train_classes_downsampled = train_classes;

num_train_pain = sum(classes(filter_train_patients,:) == 1);
desired_num_train_no_pain = round(scale_factor *
    ↪ num_train_pain);
if downsample == true
    while sum(train_classes_downsampled == -1) >
        ↪ desired_num_train_no_pain
```

```

    % generate a random index.
    idx = randi([1 length(train_classes_downsampled)],
        ↪ 1);
    if train_classes_downsampled(idx) == -1
        train_data_downsampled(idx,:) = [];
        train_classes_downsampled(idx) = [];
    end
end
end

perm = randperm(length(train_data_downsampled));
train_data_perm = train_data_downsampled(perm,:);
train_classes_perm = train_classes_downsampled(perm,:);

kernel = 'rbf';
% poly_order = 5; % Only used if kernel == polynomial

if optimize == true && optimize_model == false
    % Do hyperparameter tuning only on BoxConstraint and
    ↪ KernelScale if
    % auto, on BoxConstraint and KernelScale,
    ↪ KernelFunction, Standardize,
    % PolynomialOrder if 'all'.
    % If CrossVal is 'on' do 10 k-fold cross validation
    ↪ only iff
    % 'OptimizeHyperparameters' if 'None' as it does
    ↪ internally 5 k-fold
    % cross validation
    SVMModel = fitcsvm(train_data_perm, train_classes_perm,
        ↪ 'KernelFunction', kernel, ...
        'Standardize',
        ↪ true, 'OptimizeHyperparameters', 'auto', ...
        'ClassNames', [1,-1]);
elseif optimize == true && optimize_model == true

```

```

SVMModel = fitcsvm(train_data_perm, train_classes_perm,
↳ ...
    'Standardize', true, 'ClassNames', [1, -1],
↳ 'OptimizeHyperparameters', ...
    {'BoxConstraint', 'KernelFunction', 'KernelScale',
↳ 'PolynomialOrder'} ...
);
[ScoreCVSVMModel,ScoreParameters] =
↳ fitSVMPosterior(SVMModel);
else
SVMModel = fitcsvm(train_data_perm, train_classes_perm,
↳ 'KernelFunction', kernel, ...
    'Standardize',
↳ true, 'OptimizeHyperparameters', 'none', ...
    'ClassNames', [1,-1], 'KernelScale', 'auto');
[ScoreCVSVMModel,ScoreParameters] =
↳ fitSVMPosterior(SVMModel);
end

[ytest_predicted,score] = predict(SVMModel,test_data);
accuaracy = ytest_predicted == test_classes;
ac_percent = sum(accuaracy)/length(accuaracy) * 100;

% If we consider positive the pain class, and it is the
↳ label 1. So under
% this system the negative class is the non pain one with
↳ label -1.
confusion_mat_svm =
↳ confusionmat(test_classes,ytest_predicted);

% Accuracy = (TP+TN)/(TP+FP+FN+TN)
Accuracy_svm = (confusion_mat_svm(1,1) +
↳ confusion_mat_svm(2,2))/(confusion_mat_svm(1,1) + ...

```

```
confusion_mat_svm(1,2) + confusion_mat_svm(2,1) +
    ↪ confusion_mat_svm(2,2));

% Precision = TP/(TP + FP)
Precision_svm =
    ↪ confusion_mat_svm(2,2)/(confusion_mat_svm(1,2) +
    ↪ confusion_mat_svm(2,2));

% Recall = TP/(TP + FN)
Recall_svm = confusion_mat_svm(2,2)/(confusion_mat_svm(2,2)
    ↪ + confusion_mat_svm(2,1));

% F1 Score = 2*(Recall * Precision) / (Recall + Precision)
F1_score_svm = 2*(Recall_svm * Precision_svm)/ (Recall_svm +
    ↪ Precision_svm);

confusion_chart_svm = confusionchart(test_classes,
    ↪ ytest_predicted);
```

Listado 11: Implementación de clasificación de segmentos en función de la presencia de dolor mediante SVM

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá