

CARDIDENT

Sistema de detección, clasificación e identificación en línea de complejos QRS

Pablo Aguirre

Juan Cardelino

Nicolás Loeff

Tutor:

Prof. Agr. Ing. Franco Simini

Núcleo de Ingeniería Biomédica
de las Facultades Medicina e Ingeniería
Universidad de la República

Abril 2001 - Mayo 2002

Instituto de Ingeniería Eléctrica
Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

A Karen y Virginia.

AGRADECIMIENTOS

Durante el desarrollo de este trabajo contamos con la ayuda de varias personas. Si bien no es posible nombrarlas a todas, quisiéramos mencionar a algunas de ellas. Franco Simini, nuestro tutor, que se aventuró a guiarnos en un proyecto propuesto por nosotros mismos. Eduardo Migliaro que siempre estuvo disponible para aclarar nuestras dudas sobre el aspecto médico de este proyecto. Anders Lansner y la gente del Grupo de Redes Neuronales del KTH de Estocolmo, que amablemente recibieron a Nicolás durante un par de semanas y sugirieron el uso del algoritmo CSL. Alvaro Pardo por sus sugerencias y recomendaciones bibliográficas de gran ayuda.

Quisiéramos agradecer también a Margot y Diego por proporcionarnos un lugar donde instalarnos y soportarnos dentro de su casa por casi un año. A nuestras familias, que ahora nos van a tener viviendo con ellas otra vez. Por último a las panaderías y supers del barrio por suministrarnos con aproximadamente 420 litros de Coca-Cola y 65Kg de bizcochos y muy especialmente al OSO que nos acompañó durante todo el proyecto.

Indice General

Indice de Tablas	IX
Indice de Figuras	XI
1.. Introducción	1
1.1. Descripción del problema	3
1.2. Estado del Arte	4
1.3. Requerimientos Temporales y de Validación	6
1.4. Diseño del Sistema	11
1.4.1. Introducción	11
1.4.2. Modelo de Capas	11
1.4.3. Detección	12
1.4.4. Extracción de Características	12
1.4.5. Clasificador	12
1.4.6. Interfaces	13
1.4.7. Separación de Tareas	13
1.5. Organización de la tesis	14
2.. Detección	15
2.1. Introducción	17
2.2. Procesamiento Previo	17
2.2.1. Filtro Pasabanda	18
2.2.2. Derivador	20
2.2.3. Energía	22
2.3. Algoritmo de Detección	22
2.3.1. Umbrales	23
2.3.2. Detección de Picos R	26
2.3.3. SearchBack	26
2.3.4. Período Refractario	27
2.4. Resultados	28
2.4.1. Performance del Algoritmo Desarrollado	29
2.4.2. Comparación de la Performance con otros Algoritmos	30
2.5. Conclusiones	33
2.6. Posibles Mejoras	33

3.. Extracción de Características	35
3.1. Introducción	37
3.2. Preprocesamiento	37
3.3. Transformada de Karhunen-Loève	39
3.3.1. Base de la transformada	39
3.3.2. Elección de la muestra (para generar la base)	41
3.3.3. Extracción de características	41
3.4. Modelo paramétrico de Hermite	41
3.4.1. Base de Hermite	42
3.4.2. Procesamiento	43
3.4.3. Método iterativo para encontrar el b óptimo	44
3.5. Resultados	49
3.6. Generación de vectores de características	50
4.. Clasificación	53
4.1. Introducción	55
4.2. Algoritmos de Clustering	56
4.2.1. Aprendizaje Competitivo	57
4.2.2. Aprendizaje Competitivo Selectivo	60
4.2.3. Comparación	65
4.3. Algoritmos de Clasificación	66
4.3.1. Learning Vector Quantization	66
4.3.2. Optimized Learning Vector Quantization	66
4.3.3. Algoritmo CSL supervisado	68
4.4. Agrupador	68
4.4.1. Introducción	68
4.4.2. Implementación	69
4.4.3. Entrenamiento	69
4.4.4. Resultados	69
4.4.5. Resultados del agrupador basado en Hermite	74
4.4.6. Comparación con otros algoritmos	76
4.5. Experto Local	78
4.5.1. Introducción	78
4.5.2. Implementación	79
4.5.3. Entrenamiento	79
4.5.4. Resultados	79
4.5.5. Comparación con otros algoritmos	80

4.6.	Experto Global	83
4.6.1.	Introducción	83
4.6.2.	Implementación	84
4.6.3.	Entrenamiento	84
4.6.4.	Resultados	85
4.6.5.	Comparación con otros Algoritmos	85
4.7.	Mezcla de Expertos	87
4.7.1.	Introducción	87
4.7.2.	Fundamento Teórico	88
4.7.3.	Implementación	92
4.7.4.	Entrenamiento	92
4.7.5.	Resultados	93
4.7.6.	Comparación con otros algoritmos	94
4.8.	Conclusiones	97
5.	Implementación de CARDIDENT	99
5.1.	Introducción	101
5.2.	Implementación de CARDIDENT en Matlab	101
5.2.1.	Introducción	101
5.2.2.	Resultados	102
5.3.	Implementación de CARDIDENT en C	103
5.3.1.	Solución Propuesta	103
5.3.2.	Determinación del tamaño de los Buffers	106
5.4.	Requerimientos de Precisión y Memoria	107
5.5.	Requerimientos de Tiempo y Capacidad de Procesamiento	108
5.6.	Líneas de código utilizadas	110
5.7.	Conclusiones	110
6.	Conclusiones	113
6.1.	Introducción	115
6.2.	Detección	115
6.3.	Extracción de características	115
6.4.	Clasificación	116
6.5.	Integración e implementación	117
6.6.	Mejoras	117

A..Manual de usuario CARDIDENT DEMO	121
A.1. Introducción	123
A.2. Instalación	123
A.3. Estructura del Programa	123
A.3.1. Evaluar Detección	123
A.3.2. Evaluar Clasificación	125
A.3.3. Estructura ECG: Visualizador de latidos y neuronas	126
B..Tablas de Resultados	129
C..Desarrollos auxiliares	139
D..Especificación inicial	143
E.. Tiempos y Costos	147
E.1. Tiempos	149
E.2. Costos	151
Bibliografía	155

Indice de Tablas

1.1.	Códigos de anotaciones de la base MIT-BIH y sus explicaciones.	8
1.2.	Clases recomendadas por la AAMI.	11
1.3.	Clases utilizadas para evaluar el clasificador.	12
2.1.	Performance del Detector según AAMI	30
2.2.	Performance estricta del Detector	31
2.3.	Comparación de Performances de algoritmos de Detección	33
3.1.	Máximo b admisible para N dado	44
3.2.	Comparación de Karhunen-Loève y Hermite	51
4.1.	Performance por clase del agrupador sobre la base de datos.	70
4.2.	Agrupamiento de los latidos sobre la base de datos, por clase.	71
4.3.	Performance global por registro del agrupador CSL de CARDIDENT.	72
4.4.	Comparación con algoritmo SOM de Lagerholm <i>et al.</i> [1].	77
4.5.	Comparación con algoritmo MART	78
4.6.	Performance por clase del LE de CARDIDENT sobre la base de datos	80
4.7.	Performances totales del LE de CARDIDENT por registro	81
4.8.	Clasificación del LE de la base	82
4.9.	Comparación de performance del LE con HU <i>et al.</i>	83
4.10.	División de la base para el GE	85
4.11.	Performances totales del GE de CARDIDENT por clase	85
4.12.	Performances totales del GE de CARDIDENT por registro	86
4.13.	Clasificación del GE de la base	86
4.14.	Comparación de performance del GE con HU <i>et al.</i>	87
4.15.	Performance por clase del MOE de CARDIDENT sobre la base de datos	93
4.16.	Comparación de performance del MOE y los expertos (por clase).	94
4.17.	Performances totales del MOE de CARDIDENT por registro	95
4.18.	Clasificación del MOE sobre la base	95

4.19.	Comparación de las performances totales del MOE con la de los expertos (por registro).	96
4.20.	Comparación de performance del MOE desarrollado con el de Hu <i>et al.</i>	97
5.1.	Desempeños por registro y globales de los expertos al integrar las tres etapas.	104
5.2.	Clasificación del LE de la base	105
5.3.	Clasificación del LE de la base	105
5.4.	Clasificación del MOE de la base	106
5.5.	Ritmo cardíaco mínimo que soporta el sistema	107
5.6.	Reglas para contar operaciones de punto flotante[46]	109
5.7.	Resumen de operaciones punto flotante G1	109
5.8.	Resumen de operaciones punto flotante G2	109
5.9.	Frecuencia cardíaca máxima contra velocidad de procesamiento	110
5.10.	Cantidad de líneas de código de MatLab empleadas	111
5.11.	Cantidad de líneas generadas en código C	111
B.1.	Performance del LE por clase y registro	131
B.2.	Performance del GE por clase y registro	135
B.3.	Performance del MOE por clase y registro	137
E.1.	Horas trabajadas por persona y por semana.	149
E.2.	Horas trabajadas discriminadas por área	151
E.3.	Equipo utilizado en el proyecto y su precio estimado a Abril del 2001	154
E.4.	Costos de CARDIDENT	154

Indice de Figuras

1.1.	Latido cardíaco normal y sus formas de onda	5
1.2.	Ejemplos de los diferentes tipos de latidos presentes en la base de Datos utilizada	9
1.3.	Ejemplos de los diferentes tipos de latidos presentes en la base de Datos utilizada	10
1.4.	Ejemplos de los diferentes tipos de latidos presentes en la base de Datos utilizada	11
2.1.	Densidad espectral de potencia relativa de complejos QRS	17
2.2.	Esquema del procesamiento de la señal para la detección	18
2.3.	Ejemplo de procesamiento para Detección	19
2.4.	Filtro Pasa Bajos	20
2.5.	Filtro Pasa Altos	21
2.6.	Filtro Pasa Banda	21
2.7.	Derivador	22
2.8.	Detección de picos en la energía	23
2.9.	Algoritmo de Detección de complejos QRS	24
2.10.	Ejemplo del funcionamiento de los umbrales adaptativos	25
2.11.	Ejemplo de detección de un pico R.	27
3.1.	Filtrado del <i>baseline drift</i> en el registro 203	38
3.2.	Respuesta en frecuencia y fase del filtro utilizado para eliminar el <i>baseline drift</i>	38
3.3.	Primeras seis funciones base de la transformada KL (abscisas en seg).	40
3.4.	Latido 20 (Normal) del Registro 200: Proyección con la transformada K-L	42
3.5.	Primeras seis funciones base para $b=15$ ms.	43
3.6.	Proyección en las primeras 6 bases de Hermite	46
3.7.	Análisis de la transformada de Hermite para el latido 20 del Registro 200	47
3.8.	Latido 137 del Registro 106: (a)Parámetro b vs Iteraciones, (b)Coeficientes	48
3.9.	Comparacion de errores para Hermite y Karhunen-Loève.	50

4.1.	Agrupamiento del registro x_124.	73
4.2.	Agrupamiento del registro 214.	75
4.3.	Sistema MOE	88
4.4.	Detalle de la implementación realizada del MOE	89
5.1.	Primeras seis funciones base de la transformada KL en la integración (abscisas en seg).	102
A.1.	Ventana Principal	124
A.2.	Ventana de Evaluación del Detector	125
A.3.	Ventana Evaluador de la Clasificación	127
A.4.	Estructura para Visualización del espacio de los latidos y neuronas	128
A.5.	Ejemplo de los reportes generados	128
E.1.	Distribución de horas por Semana y por Persona	151
E.2.	Distribución de las tareas realizadas a lo largo del tiempo detallando las horas dedicadas	152
E.3.	Distribución de las tareas realizadas a lo largo del tiempo	152
E.4.	Distribución de horas por área	153

Capítulo 1

INTRODUCCIÓN

“Coronary heart disease is a silent disease and the first manifestation frequently is sudden death.”

Dr. Herman Hellerstein

1.1. Descripción del problema

Varias décadas de investigación han hecho de la electrocardiografía una disciplina básica para el diagnóstico de desórdenes cardíacos. Debido a su simplicidad, bajo costo y al ser una técnica no-invasiva, sigue siendo ampliamente utilizada.

En los últimos años el descenso progresivo en el costo del procesamiento digital de alta performance ha impulsado la introducción de técnicas de análisis automático de Electrocardiogramas (ECG). Las mismas son fundamentales en diversos entornos, como por ejemplo en los centros de cuidados intensivos donde es esencial la detección y clasificación en tiempo real de arritmias cardíacas. En estos casos la clasificación incorrecta o tardía de la señal puede significar una amenaza a la vida del paciente. Otro campo de aplicación es el análisis de señales de larga duración, el cual puede resultar tedioso y consumir mucho tiempo de un especialista. Esto lleva a que el mismo se concentre en algunos intervalos determinados, en desmedro del resto de la señal. La automatización permite seleccionar los segmentos relevantes para que el especialista pueda estudiarlos con detenimiento, y prestar menos atención a los tramos rutinarios.

El incremento en los costos de la salud hace imperiosa la necesidad de desarrollar sistemas ambulatorios que reduzcan la permanencia de los pacientes en los hospitales. Por lo tanto el diseño de un sistema portable, de bajo costo, alta performance y uso sencillo que permita el análisis y diagnóstico automático es una necesidad. Un equipo de estas características debe integrar diversas técnicas de análisis de datos como: procesamiento de señales, reconocimiento de patrones, toma de decisiones e interacción hombre-máquina.

Los equipos portátiles existentes se reducen, por razones tecnológicas, al registro de la señal durante un lapso de tiempo limitado por la capacidad de almacenamiento de los dispositivos. Los equipos de registro Holter graban 24 horas de ECG en cassette o más recientemente en memoria de silicio.

Una señal típica de 24 horas de duración consta de aproximadamente 100.000 latidos los cuales pueden ser agrupados morfológicamente en un número pequeño de clases. En la mayoría de estas clases donde la actividad cardíaca es normal, es suficiente registrar el número de latidos que la componen y un latido representativo. En el lapso de tiempo donde la actividad cardíaca presenta anomalías o síntomas de enfermedades, es necesario el registro completo de la señal. Esto sólo es posible si el equipo portátil además de registrar la señal, también es capaz de analizarla.

Dada una cierta capacidad de almacenamiento del dispositivo, el tiempo de observación de la actividad cardíaca aumenta considerablemente implementando el análisis propuesto anteriormente. La relación de tiempos de observación entre los métodos existentes y el propuesto, depende claramente del tipo de actividad cardíaca desarrollada por el paciente durante el lapso de tiempo considerado. Otra ventaja del análisis en línea es la posibilidad de alertar ante la presencia de anomalías en la señal que signifiquen riesgo para el paciente, en forma local o remota. La acción más simple es la del registro detallado del episodio de riesgo, disponible hoy en los llamados *regis-tradores de eventos cardíacos*.

Las patologías observables mediante la electrocardiografía pueden agruparse en tres grandes categorías:

- Alteraciones del ritmo cardíaco o arritmias.

- Disfunciones de la perfusión de la sangre en el miocardio o isquemia cardíaca.
- Alteraciones crónicas de la estructura mecánica del corazón, como por ejemplo hipertrofia del ventrículo izquierdo.

El presente trabajo se centrará en el estudio de la primera de estas patologías. En particular se trabajará sobre el complejo QRS ¹ que está asociado con la actividad eléctrica ventricular. El mismo contiene importante información clínica, por ejemplo su morfología sufre cambios importantes en los latidos ventriculares. También está presente en la enorme mayoría de los latidos y su relación señal-ruido es la más alta de todas las ondas presentes en la señal.

El principal problema a los que se enfrenta un analizador de ECG es la gran variabilidad de las morfologías, no sólo entre pacientes sino también debido a movimientos del paciente, modificaciones de las características de conducción eléctrica del cuerpo, entre otros. Debido a esto no es posible construir un conjunto de entrenamiento que tome en cuenta todos los casos.

Además, la señal de ECG se ve contaminada por varias fuentes de ruido, tanto externas (interferencia de la red de alimentación, movimiento de los electrodos) como de origen biológico (el movimiento muscular que provoca interferencia en alta frecuencia y la respiración que provoca un desplazamiento de la línea base). Es por eso que debe tomarse un cuidado especial en la elección de técnicas adecuadas de acondicionamiento de la señal (pre-procesamiento), ya que la calidad de la señal a la entrada del clasificador tiene impacto directo sobre su desempeño.

Organización del capítulo

En la sección anterior se presentó una descripción del problema a resolver. En la sección 1.2 se presenta un panorama general de las técnicas de análisis de ECG existentes hasta el momento. La especificación de los requerimientos del sistema CARDIDENT se muestra en la sección 1.3. En la sección 1.4 se describe el diseño del sistema a implementar para satisfacer dichos requerimientos. Por último en la sección 1.5 se presenta un esquema de la organización de este trabajo.

1.2. Estado del Arte

Las técnicas de análisis automático de ECG tradicionalmente se pueden dividir en dos grandes grupos: aquellos que usan las muestras directamente extraídas de la señal y los que están basados en un conjunto de características extraídas.

Entre los primeros se encuentran generalmente los que utilizan técnicas de identificación de patrones (“*template matching*”). Este enfoque presenta el inconveniente de ser muy sensible a los corrimientos temporales y el hecho de comparar cada latido con una plantilla es computacionalmente costoso.

Los métodos basados en la extracción de características usualmente miden magnitudes como el ancho, alto y área del complejo QRS, el intervalo RR, etc. Los algoritmos de

¹La señal de un ECG es la manifestación en el cuerpo de la actividad eléctrica miocárdica, la cual aparece como una señal casi periódica. Cada patología cardíaca afecta permanentemente o temporalmente, completa o parcialmente, la forma de onda básica del ECG. Esta forma de onda está compuesta por varias ondas, llamadas P, Q, R, S y T respectivamente (ver figura 1.1). En este trabajo se utiliza la expresión “complejo QRS” como referencia a la señal formada por las tres ondas centrales de cada latido.

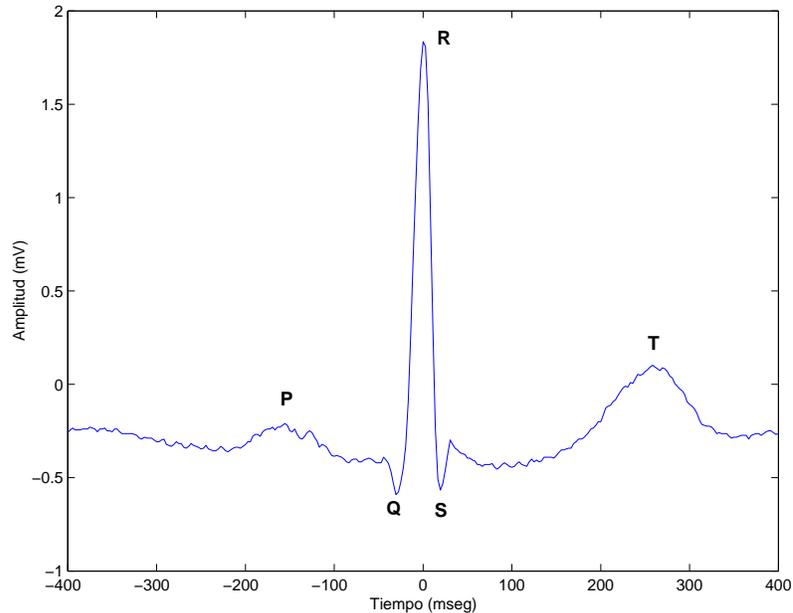


Figura 1.1: Latido cardíaco, extraído del registro 103, donde se pueden apreciar las ondas que lo componen

clasificación consisten en crear un árbol de decisión o un conjunto de reglas heurísticas que determinan el resultado de la clasificación mediante la comparación con umbrales (en general determinados empíricamente).

Existen otros que usan un procedimiento sintáctico de clasificación basados en un conjunto de reglas gramaticales aplicadas sobre vectores de características del patrón de entrada. En general estos clasificadores son más robustos a los corrimientos temporales, pero cabe destacar que la performance de estos clasificadores depende fuertemente de la representatividad del conjunto de características elegido y de los parámetros usados por los mecanismo de decisión (umbrales). Además en algunos casos la complejidad agregada por el pre-procesamiento necesario puede ser muy alta. Existen además en la literatura clasificadores Bayesianos, sistemas expertos y modelos de Markov. La principal dificultad de todos estos enfoques reside en la gran variabilidad de las señales. Por lo cual los desempeños de estos clasificadores son bastantes irregulares.

En los últimos años varias Redes Neuronales Artificiales (ANN) han sido propuestas como herramientas para el diseño de clasificadores para enfrentar problemas donde sea necesaria la discriminación no lineal de las clases o donde haya datos incompletos o ambiguos. En la literatura existen numerosos trabajos, con diversas performances, que aplican ANN para resolver problemas de clasificación de ECG ([1], [2], [3], [4], [5], [6], [7]). Los algoritmos utilizados son variados y aparecen tanto algoritmos supervisados como no-supervisados. Dentro de los primeros se puede destacar:

- Watrous *et al.* [7] utiliza un Perceptron Multi-Capa (*Multi-Layer Perceptron MLP*) sintetizado a partir de un clasificador basado en reglas lógicas. En [5] los mismos autores presentan una variante interesante que tiene la habilidad de adaptarse al paciente. Con este método los autores de [5] reportan mejoras en

las tasas de error que oscilan entre 2 y 10 veces comparados con el clasificador original.

- Silipo *et al.* [6] presenta un repaso de algoritmos de clasificación y se implementan varios, esencialmente MLP y redes recurrentes. Además se presentan en este trabajo criterios de rechazo en los casos en los cuales el clasificador no está seguro de cómo clasificarlo.
- Hu *et al.* [4] presenta un algoritmo que se adapta al paciente por medio de una mezcla de expertos. Con este algoritmo se logra una performance de 94.0%.
- Hu *et al.* [8] presenta un algoritmo de clasificación basado en un MLP que separa los latidos en dos clases con una performance del 90%. Los mismos autores reportan una performance del 65% con el mismo algoritmo pero clasificando los latidos en 13 clases. Finalmente, utilizando una estructura jerárquica del MLP logran una performance de 84.5% sobre las 13 clases.

En cuanto a los algoritmos no supervisados presentes en la literatura se puede destacar:

- El trabajo [1] de Lagerholm *et al.* que utiliza Mapas Auto-Organizativos para agrupar los latidos en 16 clases. En este trabajo se reporta una performance de 98.5%.
- Fernández-Delgado *et al.* [3] utilizan una red ART multi-canal con una cantidad de clases variable, con la cual logran una performance de 98.72%.

Llama la atención el pobre desempeño de los algoritmos supervisados cuando se los compara con los no supervisados. Esto se debe a que los primeros emplean intervalos diferentes para el entrenamiento y la evaluación, mientras que los segundos deben usar el mismo. Además los no supervisados se limitan a agrupar latidos similares mientras que los supervisados intentan clasificarlos.

1.3. Requerimientos Temporales y de Validación

Antes de diseñar CARDIDENT se estableció que debía cumplir con un requerimiento de tiempo de ejecución para tener utilidad. Además se estableció que el resultado de CARDIDENT debía ser validado de manera objetiva y reproducible.

Requerimientos Temporales

Existen tres tipos de algoritmos de detección en cuanto a las condiciones de tiempo para observar la señal, extraer los complejos QRS y clasificarlos.

Algoritmos Off-line Son aquellos en los que se cuenta con toda la señal a analizar de antemano, pudiendo el algoritmo recorrer la señal hacia atrás y hacia adelante sin restricciones. Este es el caso de los clasificadores de ECG grabados por equipos Holter.

Algoritmos On-line Son aquellos que trabajan sobre una señal que está siendo generada en ese momento. Es decir que no pueden conocer la señal más allá de la última muestra que se recibió. Estos algoritmos tienen un margen de entre 2

a 3 segundos para detectar los latidos, lo cual les da la posibilidad de repasar la señal en caso de perderse un latido.

Algoritmos Real-Time Estos son un subconjunto de los On-line, con la diferencia de que deben detectar cada latido antes de que llegue el siguiente.

El algoritmo necesario para CARDIDENT debe cumplir por lo menos con la segunda definición, es decir que debe ser un algoritmo **On-line**, ya que deberá ser implementable en dispositivos de monitoreo. Esta condición debe cumplirse para la salida del sistema, pero no para el entrenamiento que se podrá hacer fuera de línea.

Validación

Para poder objetivamente juzgar el desempeño del sistema desarrollado es necesario que su evaluación sea hecha en base a criterios estándar. Es por eso que se usó la base de datos de arritmias del Instituto Tecnológico de Massachusetts (MIT) y el Hospital Beth-Israel (de aquí en adelante MIT-BIH) [9], y hasta donde fue posible se siguió las guías de la Asociación para el Avance de la Instrumentación Médica (AAMI) [10]. A continuación se describen ambas someramente.

Base de datos MIT-BIH . La misma fue construida a partir de un conjunto de unos 4000 registros Holter. Consta de 23 registros (numerados del 100 al 124 inclusive con algunos faltantes) de aproximadamente 30 minutos de duración elegidos aleatoriamente del conjunto y de 25 registros (numerados del 200 al 234 con algunos faltantes) de idéntica duración elegidos del mismo conjunto debido a que contienen una variedad de fenómenos raros pero de importancia clínica que no estaría bien representada en un conjunto aleatorio tan pequeño de registros electrocardiográficos. El primer grupo sirve como representativo de una variedad de morfologías y artefactos (señales de origen artificial) con los cuales un analizador de ECG se puede encontrar durante el uso clínico normal. Como se mencionó fueron elegidos aleatoriamente del conjunto, desechando solamente los intervalos que contenían ruido en *todos* los canales y por lo tanto no podían ser analizados por los cardiólogos. En el segundo grupo de encuentran arritmias ventriculares, supraventriculares, de conducción y nodales complejas. Muchos de estos registros fueron seleccionados por las características del ritmo, variación en la morfología de los QRS y cambios en la calidad de la señal que se espera presenten una dificultad especial para los analizadores de arritmias. Los registros fueron tomados de hombres y mujeres entre los 23 y 89 años.

Todos los registros cuentan con dos canales. El canal 1 es en casi todos los registros el electrodo de miembro modificado II, mientras que el canal 2 es generalmente el electrodo modificado V1 aunque en algunos es el V2 o el V5 y en un caso es el V4. La excepción es el registro 114 donde los canales están invertidos, pero el analizador debe estar preparado a aceptar esta eventualidad. Los latidos normales son generalmente más visibles en el canal 1, y los ectópicos en el 2.

Los registros fueron registrados analógicamente por el mismo modelo de instrumento en todos los casos, y más tarde digitalizados con una frecuencia de muestreo de 360 Hz y resolución de 11 bits (para ± 5 mV).

Los registros fueron anotados por dos cardiólogos independientemente, y las discrepancias fueron resueltas por consenso. Además de anotaciones de arritmia se

Código	Explicación
!	<i>flutter</i> ventricular
/	Latido de marcapaso
?	No clasificado durante el entrenamiento
A	Latido prematuro atrial
E	Escape ventricular
B	Bloqueo de rama (sin especificar)
F	Fusión entre Normal y Ventricular
J	Latido prematuro nodal
L	Bloqueo de rama izquierda
N	Latido normal
Q	Inclasificable
R	Bloqueo de rama derecha
S	Latido supraventricular prematuro o ectópico (atrial o nodal)
V	Prematuro ventricular
a	Prematuro aberrante atrial
e	Escape atrial
f	Fusión entre Normal y Paced
j	Escape nodal
n	Escape supraventricular (atrial o nodal)
r	Prematuro ventricular R-sobre-T

Tabla 1.1: Códigos de anotaciones de la base MIT-BIH y sus explicaciones.

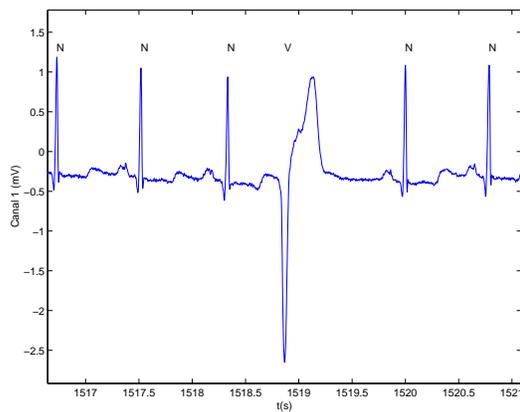
incluyeron cambios en la calidad de la señal, de ritmo y artefactos entre otros. Las anotaciones que fueron utilizadas en este estudio se pueden apreciar en la tabla 1.1 y las figuras 1.2, 1.3 y 1.4 muestran ejemplos de algunos de los tipos de latidos distinguidos en las anotaciones.

Criterios de la AAMI (Association for the Advancement of Medical Instrumentation)

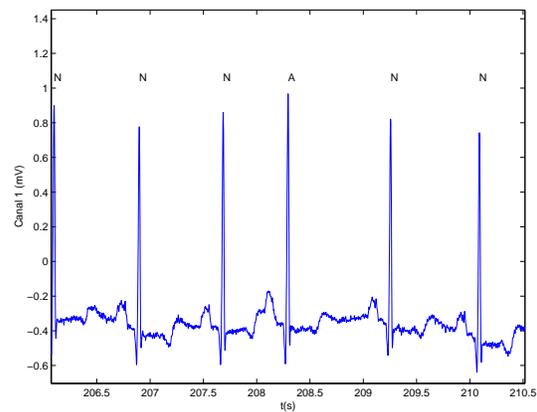
Es un conjunto de protocolos estándar para la evaluación de analizadores de electrocardiogramas². Recomienda utilizar los primeros cinco minutos de cada registro para adaptar los parámetros de detección y clasificación, y el resto para la evaluación. Pone énfasis también en la presentación de los resultados por registro. Según la AAMI también es posible excluir de las necesidades de reporte los registros (102, 104, 107, 217) que contiene latidos estimulados por marcapasos artificiales (*paced*, /). También usa una convención según la cual se pueden agrupar los latidos en cuatro clases, como se puede apreciar en la tabla 1.2.

Para hacer más exigente la evaluación, se decidió dividir la primer clase de la AAMI en dos (según [11]). También se incluyó los latidos *paced* y los *flutter* ventriculares por su importancia clínica (tabla 1.3). Para evaluar la performance del agrupador se emplearon las veinte clases de la base de datos.

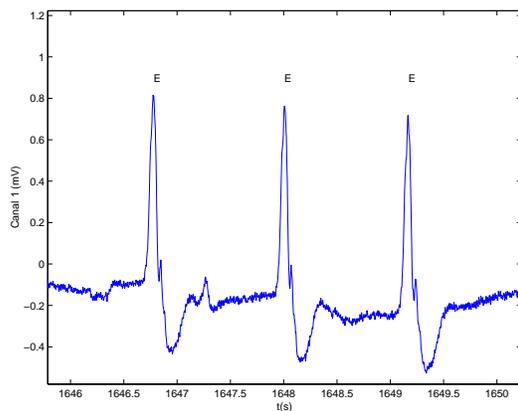
²American National Standard for Ambulatory Electrocardiographs (ANSI/AAMI EC38:1998 y su predecesor, ANSI/AAMI EC38:1994) y el American National Standard for Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms (ANSI/AAMI EC57:1998)



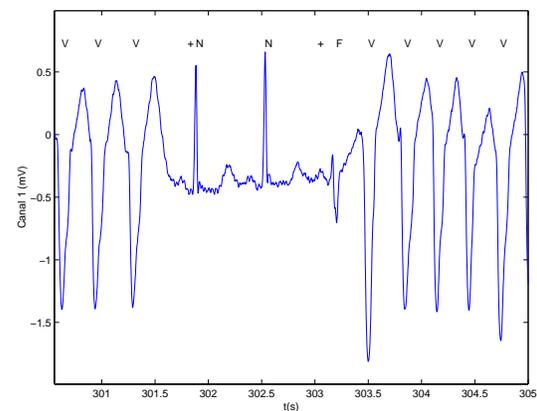
(a) Latido ventricular (V) extraído del registro 100.



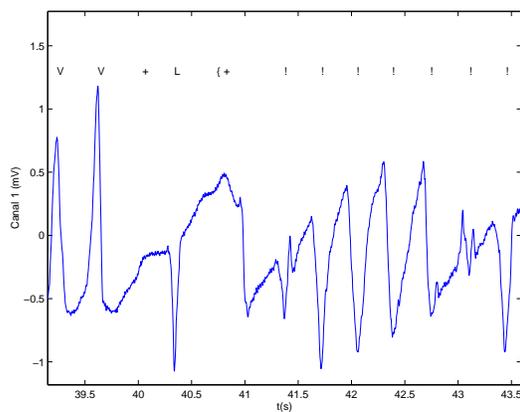
(b) Latido prematuro atrial (A) extraído del registro 100.



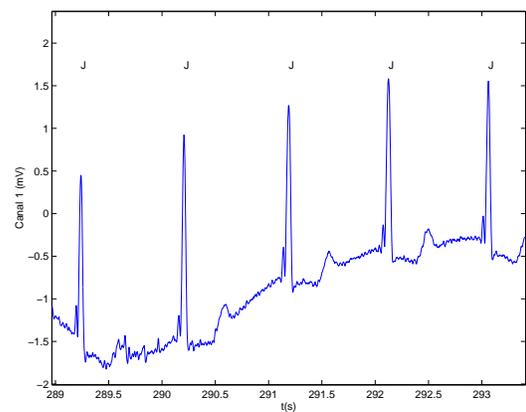
(c) Escape ventricular (E) extraído del registro 207.



(d) Fusión entre Normal y Ventricular (F) extraído del registro 205.

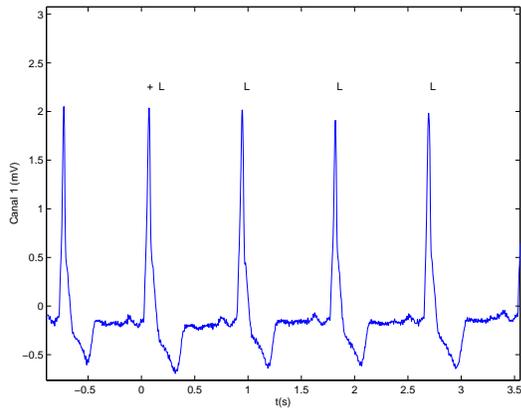


(e) *flutter* ventricular (!) extraído del registro 207.

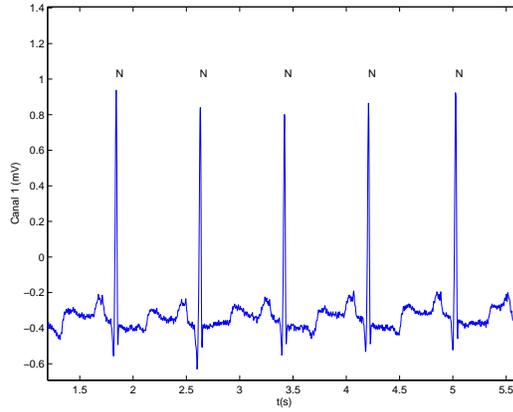


(f) Latido prematuro nodal (J) extraído del registro x_124.

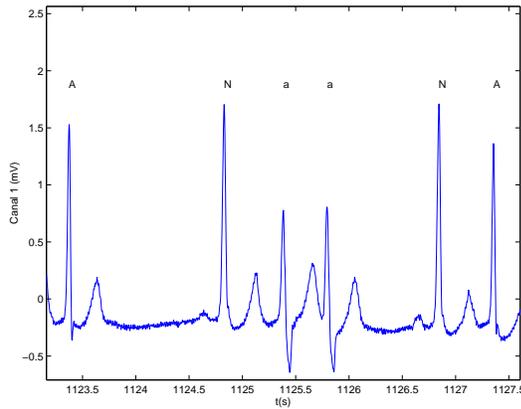
Figura 1.2: Ejemplos de los diferentes tipos de latidos presentes en la base de Datos utilizada



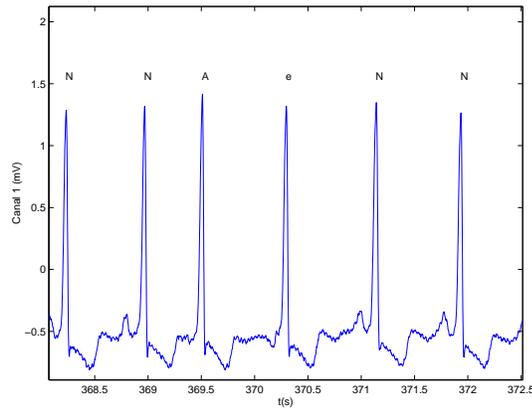
(a) Bloqueo de rama izquierda (L) extraído del registro 214.



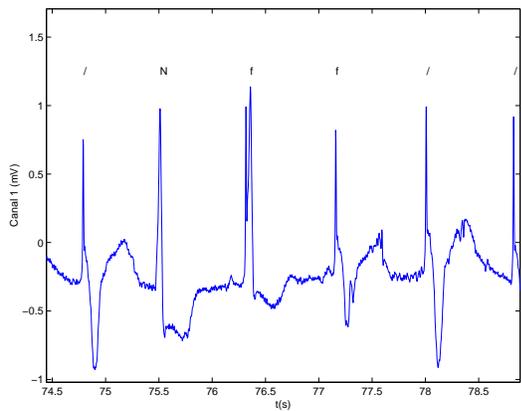
(b) Latido normal (N) extraído del registro 100.



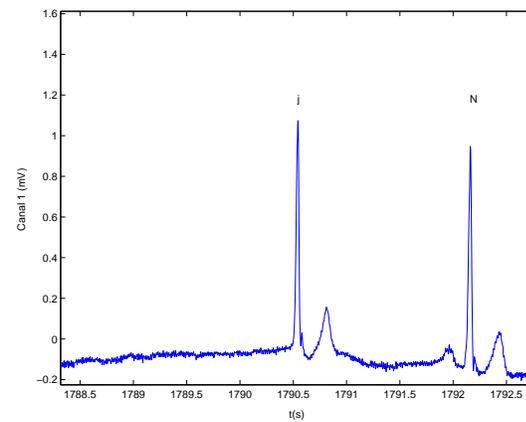
(c) Prematuro aberrante atrial (a) extraído del registro 202.



(d) Escape atrial (e) extraído del registro x_223.

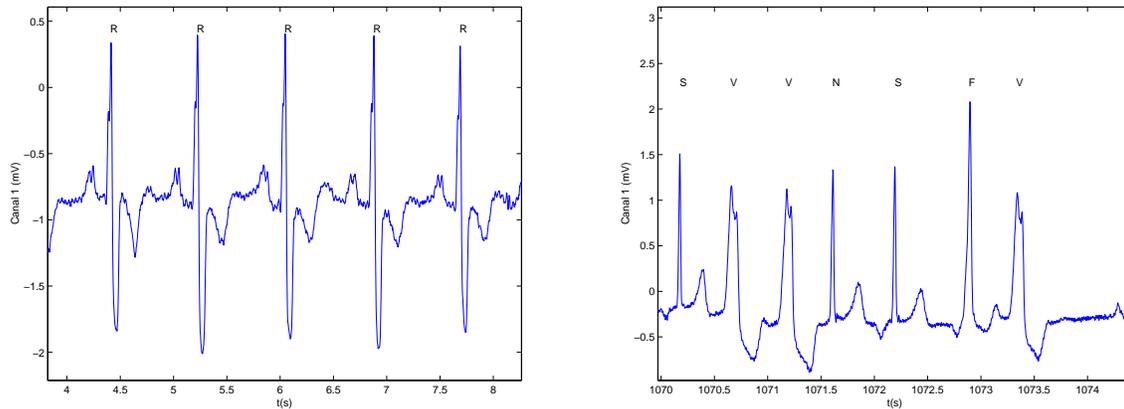


(e) Fusión entre Normal y Paced (f) extraído del registro 104.



(f) Escape nodal (j) extraído del registro 201.

Figura 1.3: Ejemplos de los diferentes tipos de latidos presentes en la base de Datos utilizada



(a) Bloqueo de rama derecha (R) extraído del registro 118.

(b) Latido supraventricular prematuro (S) extraído del registro 208.

Figura 1.4: Ejemplos de los diferentes tipos de latidos presentes en la base de Datos utilizada

Clase	Tipos de latidos
N	Todos los que no caen en las categorías V, F o Q (N, L, R, B, e, j, n, S, A, a, J)
V	Latidos ventriculares ectópicos (V, r, E)
F	Latidos fusión entre N y V (F)
Q	Inclasificables

Tabla 1.2: Clases recomendadas por la AAMI.

1.4. Diseño del Sistema

1.4.1. Introducción

El problema de análisis e identificación de complejos QRS propuesto en este trabajo será abordado utilizando un modelo de capas, que se mostrará en la sección 1.4.2, con las cuales se facilita la modularidad de la solución propuesta. Además se dividirán las tareas según el momento de su ejecución (en línea o no), la cual se mostrará en la sección 1.4.7

1.4.2. Modelo de Capas

La ventaja principal de la utilización de un modelo de capas para describir un problema radica en la posibilidad de obtener una solución general independiente de la solución utilizada en cada capa. Es decir que una vez definida la función de cada capa y su interfaz con las demás, se tiene total libertad sobre la técnica utilizada para llevar a cabo cada tarea. Incluso es posible intercambiar dos soluciones diferentes para la implementación de una capa, afectando únicamente el rendimiento del sistema de acuerdo al rendimiento de la solución utilizada, pero sin variar la función del mismo. En nuestro problema particular existen tres etapas claramente definidas: Detección de complejos QRS, Extracción de Características y Clasificación. Es necesario definir la función de cada una de estas etapas y la interfaz para la comunicación entre ellas

Clase	Tipos de latidos
N	Latidos con QRS <i>normal</i> (N, L, R, B, e, j, n)
S	Latidos con QRS <i>supraventricular</i> (S, A, a, J)
V	Latidos ventriculares (V, r, E, !)
P	Latidos de marcapaso (/)
F	Latidos fusión (F, f)
Q	Inclasificables (Q, ?)

Tabla 1.3: Clases utilizadas para evaluar el clasificador.

y con el exterior.

1.4.3. Detección

El primer paso a la hora de realizar una clasificación de los complejos QRS de un electrocardiograma es ubicarlos dentro de la señal ECG que se quiere analizar. La función de la capa de detección es, entonces, mirar la señal, determinar donde aparecen los complejos y extraer la sección de señal correspondiente al complejo QRS, descartando el resto.

A la hora de definir que información debe recopilar esta etapa, hay que tener en cuenta que la información que recibe una capa y no transmite (procesada o no) no estará disponible al resto de las capas. Por lo tanto es importante que esta etapa transmita toda la información que caracteriza un complejo QRS. Esta información puede estar disponible directamente (como el intervalo de tiempo desde el último latido) o indirectamente (como información sobre la morfología de la señal que debe ser extraída del complejo QRS por otra capa).

1.4.4. Extracción de Características

La idea de esta etapa es aplicar una transformada al complejo QRS provisto por la capa anterior de manera de obtener en un número reducido de coeficientes la información clínica relevante. De esta forma es posible representar los complejos en un espacio cuya métrica minimice la distancia entre latidos de una misma clase y maximice la distancia entre latidos de distinta clase. Además esta capa debe agregar la información temporal que recibe del Detector (intervalo RR, ancho del QRS, etc.). La función de esta capa es, entonces, obtener un vector n-dimensional que incluya los coeficientes de la transformada y los valores temporales. Para que el vector tenga sentido, los elementos deben estar normalizados para que los coeficientes y los valores temporales tengan el mismo peso relativo para todos los latidos.

1.4.5. Clasificador

El Clasificador es la etapa medular de todo el sistema, ya que define a que clase pertenece el latido detectado y procesado por las etapas anteriores. En resumen la función de esta capa es analizar los vectores recibidos y definir a qué clase pertenece.

Para llevar a cabo esta tarea se utilizarán en este trabajo redes neuronales artificiales, aunque cualquier otro algoritmo de clasificación es aceptable.

1.4.6. Interfaces

Interfaz Entrada-Detector El detector debe recibir la señal digitalizada del ECG que se está analizando muestra a muestra. La frecuencia de muestreo es un parámetro configurable: para CARDIDENT se utilizaron señales digitalizadas a 360 muestras por segundo.

Interfaz Detector-Extractor de Características El detector debe entregarle a la siguiente capa la sección correspondiente a cada complejo QRS formado por una ventana de 200 milisegundos de señal centrada en el pico R, el tiempo en que ocurrió el pico R, el intervalo RR entre este complejo y el anterior, el promedio de los últimos 8 intervalos RR y una medida del ancho del complejo QRS.

Interfaz Extractor de Características-Clasificador El Clasificador debe recibir de parte del Extractor de Características un vector de n dimensiones que sitúe en el espacio de la transformada elegida al latido que se está analizando. La dimensión del vector es un parámetro general fijado a priori para todo el sistema.

Interfaz Clasificador-Salida El Clasificador debe registrar en el momento de señal detectado como pico R la clase en el que fue clasificado el latido. Los tipos de latidos que conforman cada clase, así como la cantidad de clases son también parámetros configurables de todo el sistema.

1.4.7. Separación de Tareas

A la hora de diseñar un sistema portátil de análisis *On-Line* de ECG es necesario tomar una decisión sobre que tareas serán implementadas *On-line* y cuales pueden o deben ser implementadas *Off-line*.

Esta decisión influirá sobre la potencia de procesamiento necesaria por parte del sistema portátil, ya que todo aquello que sea implementado *Off-line* puede ser procesado por un PC. Sin embargo es probable que mientras más información pueda analizar el sistema portátil, para así adaptarse mejor a la señal particular que está analizando, mejor resulte la performance del conjunto. Queda en definitiva tomar una decisión de compromiso entre la performance y la potencia de procesamiento que se está dispuesto a utilizar en el sistema, que no es otra cosa que una decisión entre resultados y costo del sistema desarrollado.

Para este trabajo se eligió dejar el entrenamiento de las redes neuronales que realizan la clasificación para ser efectuado *Off-Line*. Esta decisión está basada en que el entrenamiento es muy costoso computacionalmente y debe ser efectuado una sola vez. De todas formas el sistema portátil podría incluir algoritmos de re-entrenamiento que realizan ajustes finos al entrenamiento original, sin embargo este tipo de algoritmos no son incluidos en este trabajo.

En la capa de Extracción de Características decisiones de este tipo también pueden ser tomadas. Existen transformadas como la de Karhunen-Loève cuya base de proyección puede ser generada *Off-Line* y que por lo tanto no utilizan recursos del sistema portátil para ello. Por otro lado existen transformadas como la de Hermite cuya base

de proyección se adapta a la señal que analiza. Esto tiene la ventaja de lograr una mejor representación, sin embargo se utilizan recursos en adaptar la base en forma *On-Line*.

1.5. Organización de la tesis

En este capítulo se presentó una descripción del problema a resolver, se describió el estado del arte en el área y se mostró un esquema de la solución propuesta. En el capítulo 2 se presenta el detector de QRS desarrollado. En el capítulo 3 se explica la etapa de extracción de características. El desarrollo de la teoría y la implementación de la etapa de clasificación se encuentra en el capítulo 4. En el capítulo 6 se presentan las conclusiones elaboradas sobre este trabajo. Finalmente en el capítulo 5 se muestra la implementación propuesta y se estiman los requerimientos para la misma. En los apéndices se explican: el manual de usuario, algunas tablas de resultados, algunos desarrollos auxiliares, la especificación inicial del proyecto y los tiempos y costos del mismo respectivamente.

Capítulo 2

DETECCIÓN

*“Dices que tienes corazón, y sólo
lo dices porque sientes sus latidos;
eso no es corazón... es una máquina
que al compás que se mueve hace ruido.”*

*Gustavo Adolfo Bécquer
RIMAS Y LEYENDAS
LXXVII*

2.1. Introducción

En este capítulo se explica la implementación elegida para el detector de complejos QRS. Previo al desarrollo del mismo es necesario explicar brevemente como se compone el espectro de la energía de la señal de un ECG. En el mismo coexisten diversas componentes como los propios complejos QRS, las ondas P o T, ruido muscular, interferencia de la red de alimentación, etc. Thakor *et al.* [12] muestran los resultados del análisis del espectro de energía de un ECG en la figura 2.1.

En ella se puede apreciar que la energía de los complejos QRS se encuentra centrada

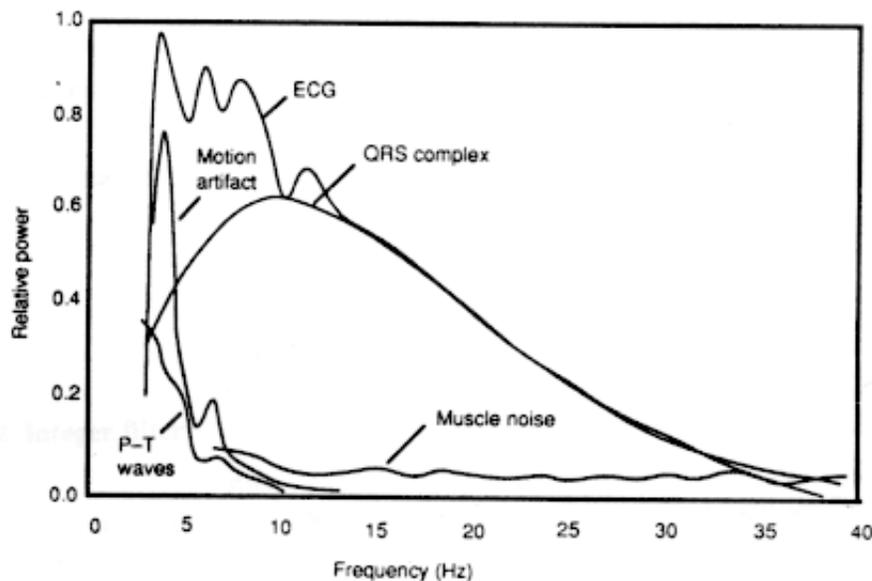


Figura 2.1: Densidad espectral de potencia relativa de complejos QRS y de las fuentes de ruido que aparecen en un ECG, basado en un promedio de 150 latidos. Tomado de Afonso *et al.* [13]

alrededor de los 10Hz. Este resultado permite diseñar un filtro que aisle en frecuencia la componente del ECG correspondiente a los complejos QRS.

Organización del capítulo

El capítulo está dispuesto de la siguiente manera. En las secciones 2.2 y 2.3 se explican los fundamentos del algoritmo implementado y del procesamiento que se le realiza a la señal. En la sección 2.4 se muestran los resultados y se los compara con los resultados publicados en la literatura utilizando distintas estrategias. Finalmente en las secciones 2.5 y 2.6 se dan las conclusiones y se comentan algunas ideas para mejorar la performance del algoritmo que no fueron implementadas en este trabajo.

2.2. Procesamiento Previo

Pan y Tompkins [14] desarrollaron un algoritmo de detección en tiempo real basado en el análisis de la pendiente de la señal en la banda de interés. Es decir, dada la señal de un ECG, se filtra en una banda de frecuencia conveniente y se analiza la

señal extraída del filtro. El detector que se va a desarrollar se basa en este algoritmo y en la implementación del mismo que realiza Afonso [13].

El primer paso en la detección de los complejos QRS es realizar un procesamiento de la señal para obtener una medida de la energía instantánea de la pendiente de la señal. Así se puede diferenciar las oscilaciones en la señal que corresponden a verdaderos complejos QRS de ruido o interferencias que aparecen en la misma banda de frecuencias.

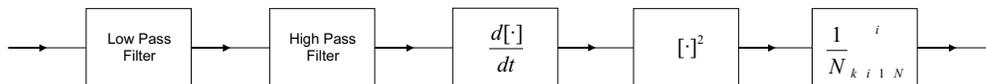


Figura 2.2: Esquema del procesamiento de la señal para la detección

La figura 2.2 muestra los pasos del proceso de análisis y en la figura 2.3 se ven todos los pasos en un latido tipo que aparece en la figura 2.3(a). Primero, se filtra la señal con un filtro pasabanda que permite obtener sólo la banda donde se espera encontrar los complejos QRS y elimina otras señales que interfieren en la detección. Esto se realiza con un filtro pasa bajos (fig.2.3(b)) que elimina todo el ruido de alta frecuencia (lo cual incluye por ejemplo, el ruido de 50Hz de la red de alimentación) y con un filtro pasa altos (fig.2.3(c)) que como se ve, elimina las componentes de continua y las ondas P y T. Luego, se deriva la señal filtrada para detectar las pendientes pronunciadas características de los complejos QRS (fig.2.3(d)). Hasta aquí el procesamiento es lineal, pero a continuación se realiza una medida de la energía instantánea de la derivada de la señal elevando al cuadrado muestra a muestra y promediando mediante una integral móvil ($E[x(t)] = \langle x^2 \rangle_{|t-\tau}^t$). Esta medida de la energía permite: primero, que cuando la señal es elevada al cuadrado todas las muestras sean positivas y se acentúe la diferencia entre las distintas pendientes detectadas en la etapa de diferenciación (fig.2.3(e)); segundo, cuando se promedia se eliminan las oscilaciones de poca duración que no pueden corresponder a un complejo QRS y se obtiene un pulso uniforme en la sección de señal correspondiente a un complejo QRS (fig.2.3(f)).

2.2.1. Filtro Pasabanda

El filtro tiene el cometido de reducir el ruido y la interferencia de señales fuera de la banda de interés definida entre 5Hz y 15Hz. Esta banda se define a partir del análisis de las señales presentes en un ECG como se comentó en la Introducción. La implementación del filtro pasabanda se realiza con un filtro pasa bajos y un filtro pasa altos concatenados. Cada uno es implementado con filtros recursivos que son fácilmente implementables en procesadores digitales. Estos filtros podrían ser implementados con coeficientes enteros para posibilitar la utilización de procesadores de bajo poder, sin embargo, teniendo en cuenta el poder actual de procesamiento de los procesadores digitales de señal (más conocidos como DSP) y el hecho que más adelante en este trabajo es imposible evitar la utilización de lógica de punto flotante, se decidió buscar los mejores filtros sin tener en cuenta que solo utilizarán lógica de números enteros.

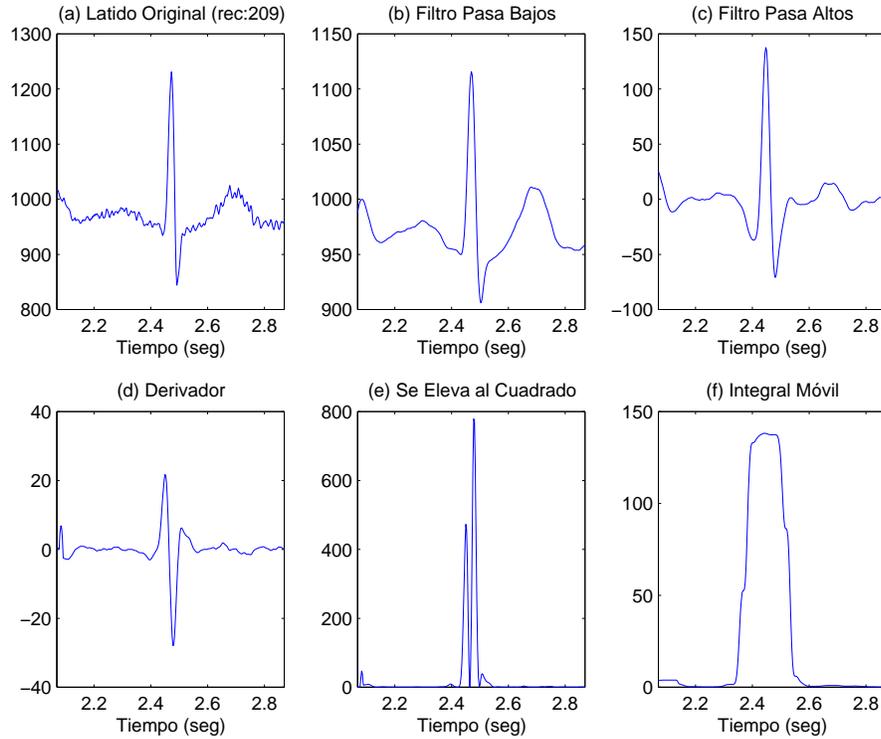


Figura 2.3: Ejemplo de procesamiento sobre un latido del registro 209 en la base de datos del MIT-BIH

Filtro Pasa Bajos

El filtro implementado debe tener una frecuencia de corte en el entorno de los 15Hz, para eso se utilizó un filtro de segundo orden con la siguiente función de transferencia:

$$H(z) = \frac{1}{7^2} \frac{(1 - z^{-7})^2}{(1 - z^{-1})^2} \quad (2.1)$$

o si se prefiere, la siguiente ecuación en diferencias:

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - 7T) + x(nT - 14T) \quad (2.2)$$

La frecuencia de corte de este amplificador se encuentra en el orden de los 16Hz y el retardo es de 6 muestras (16.7 ms para $f_s = 360$). Para darle una ganancia 1 en la banda pasante, se multiplica por el factor $1/7^2$. La figura 2.4 muestra la respuesta en frecuencia para el filtro pasa bajos. En el se puede apreciar que, por ejemplo, hay una atenuación superior a los 40dB para 50Hz, por lo que la interferencia de la red de alimentación es prácticamente eliminada. También se puede apreciar la fase lineal que presenta el filtro para toda frecuencia, lo que asegura que no exista distorsión de fase en la señal.

Filtro Pasa Altos

El filtro pasa altos se diseñó restando la señal de un filtro pasa bajos de primer orden de la señal original convenientemente retardada. El filtro obtenido, cuya transferencia se puede observar en la ecuación 2.3, tiene una frecuencia de corte de 5.6Hz y un retardo de 24 muestras (66.7ms). La respuesta se puede observar en la figura

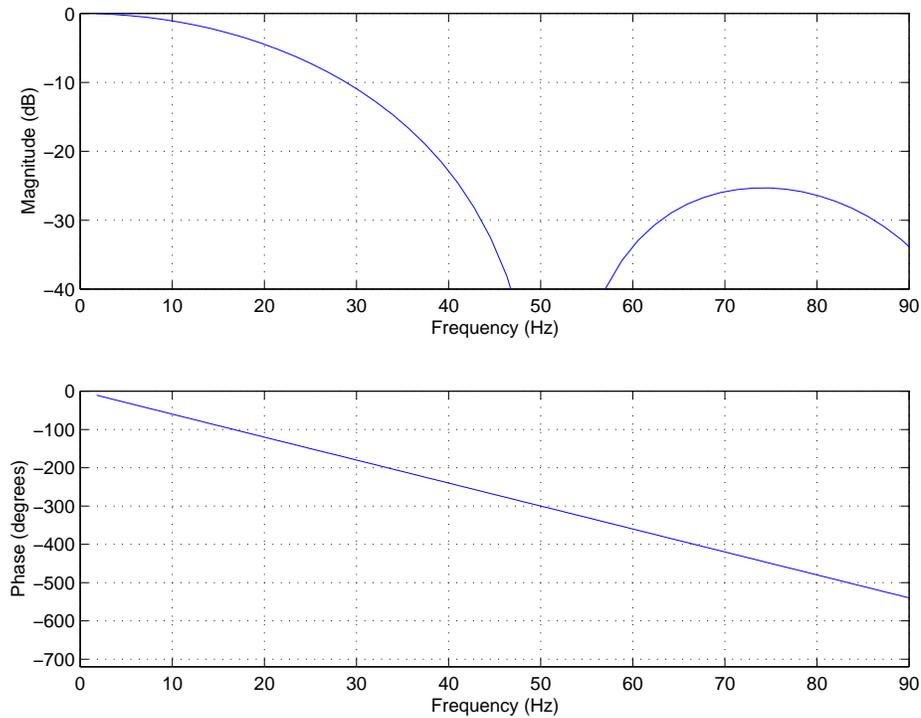


Figura 2.4: Filtro Pasa Bajos

2.5, donde se ve que la fase es lineal.

$$H(z) = z^{24} - \frac{1}{48} \frac{1 - z^{48}}{1 - z^{-1}} = \frac{1}{48} \frac{-1 + 48z^{-24} - 48z^{-25} + z^{48}}{1 - z^{-1}} \quad (2.3)$$

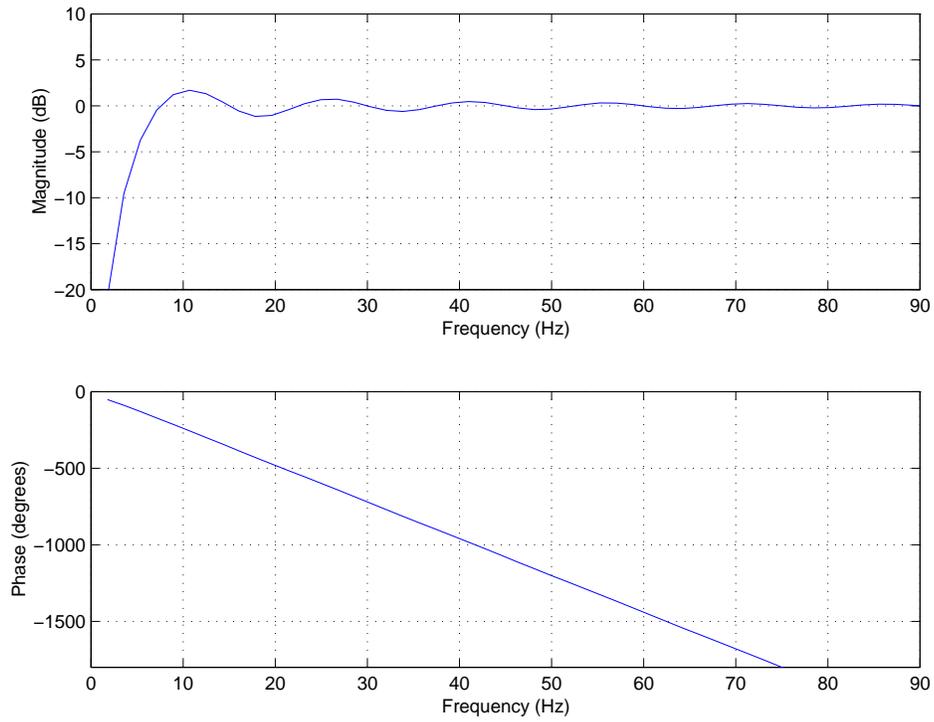
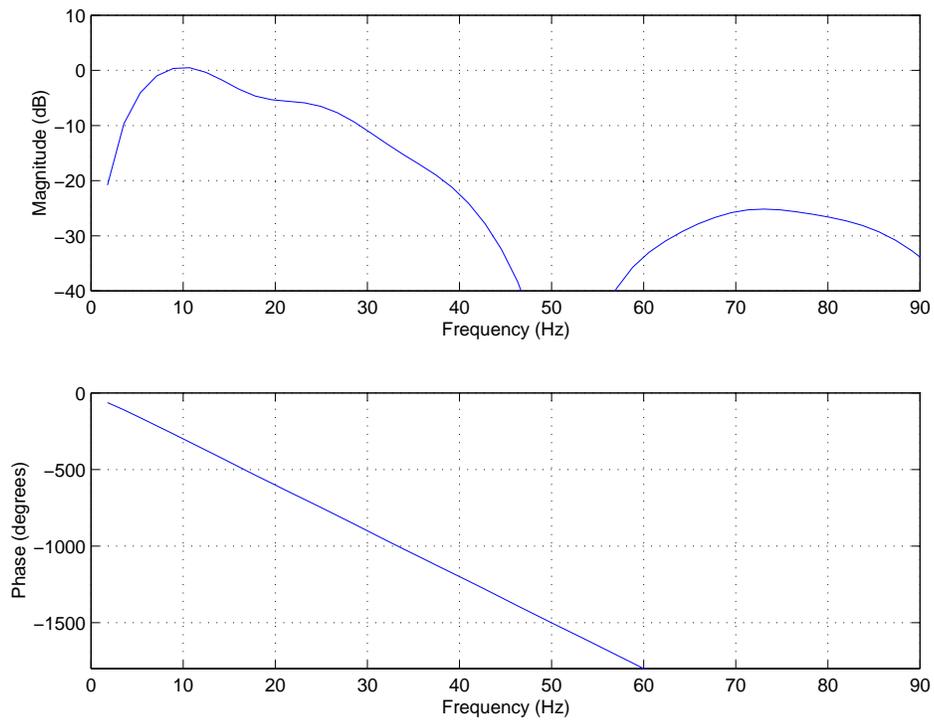
El filtro pasabanda compuesto a partir de estos dos filtros, tiene la respuesta en frecuencia que muestra la figura 2.6. En ella se puede observar que la banda pasante esta centrada en los 10Hz. De hecho, las medidas de la amplitud de la respuesta en frecuencia dan que las caídas de 3dB a cada lado de la banda pasante se dan en 6Hz y 15Hz, lo cual muestra que se obtiene la banda buscada con mucha precisión ya que todas las señales por fuera de la banda de interés se ven significativamente atenuadas.

2.2.2. Derivador

La derivada de la señal se realiza con un filtro que implementa la respuesta de un derivador ($H(s) = s$) en una banda de frecuencias que incluye holgadamente la banda de interés y que está dado por:

$$H(z) = 0,1(2 + z^{-1} - z^{-3} - z^{-4}) \quad (2.4)$$

Este derivador presenta la respuesta en frecuencia que muestra la figura 2.7. En ella se puede apreciar que efectivamente la respuesta aproxima la de un derivador ideal en una banda de frecuencias entre DC y 40Hz. La fase del filtro es lineal y el retardo introducido por el filtro es de dos muestras (5.6ms).

**Figura 2.5: Filtro Pasa Altos****Figura 2.6: Filtro Pasa Banda**

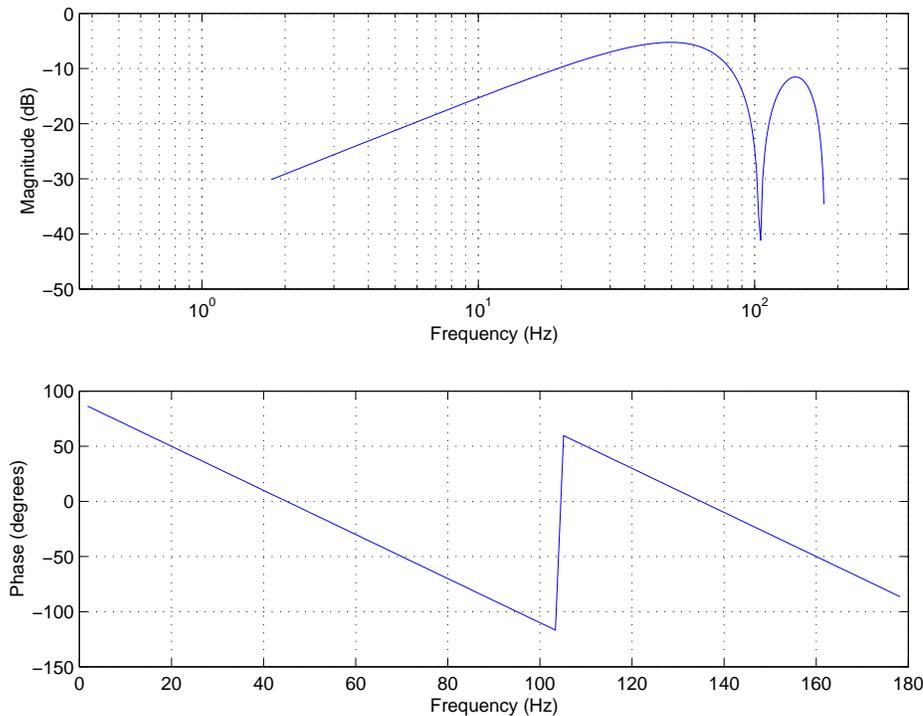


Figura 2.7: Derivador

2.2.3. Energía

La medida de la energía de la derivada que se realizó está dada por la siguiente ecuación:

$$E[x(nT)] = \frac{1}{N} \sum_{i=n-N+1}^n x(iT)^2 \quad (2.5)$$

donde $x(nT)$ es la derivada de la señal.

Esta medida se realiza en dos pasos. Primero se eleva al cuadrado punto a punto y luego se promedia mediante una integral de ventana móvil. Como ya se comentó, al elevar la señal al cuadrado hacemos positivos todos los puntos de la señal y damos mayor importancia a las frecuencias centrales de la banda pasante contra las que están en el borde y fuera de la banda que son atenuadas. Con la integral móvil se obtiene información sobre características adicionales a la pendiente de la onda R. En la misma, el ancho de la ventana (dado por N) debe ser aproximadamente igual al ancho de un complejo QRS, ya que si queda demasiado grande se mezclan con las ondas T, mientras que si queda demasiado chico un solo complejo QRS puede generar varios picos de energía. Este valor se ajusta experimentalmente, aunque se sabe que es del orden de los 150ms.

2.3. Algoritmo de Detección

Una vez en poder de la medida de energía que nos da el procesamiento, se debe determinar donde se ubica exactamente cada complejo QRS para extraer una ventana de 200ms de señal original centrada en el pico R. Para ello se utilizan umbrales adaptivos (sec.2.3.1), según los cuales se determina si un pico de energía corresponde

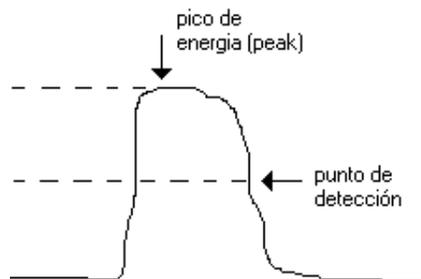


Figura 2.8: Detección de picos en la energía

efectivamente a un complejo QRS o si debe ser considerado ruido. También se incluyen en el algoritmo implementado técnicas de búsqueda hacia atrás (SearchBack, sec.2.3.3) y determinación del pico R (sec.2.3.2), fundamental para las etapas posteriores del sistema.

La figura 2.9 muestra el algoritmo con un diagrama de flujo. En la misma se aprecia el bucle principal que recorre la señal obtenida con el procesamiento (la medida de energía) mediante un puntero 'pos'. Después de cada detección exitosa, se busca el nivel más alto de energía (peak) hasta que la señal cae a la mitad del máximo nivel encontrado hasta el momento (figura 2.8). Cuando sucede, se determina que se ha encontrado un posible complejo QRS y se pasa al siguiente nivel donde se determina si el pico de energía encontrado corresponde efectivamente a un complejo o si se trata solamente de ruido. Para eso se utilizan, como se anticipó, umbrales adaptativos. Cuando el nivel de pico supera el umbral, se determina que se encontró un complejo QRS y se pasa a grabar la ventana de la señal original donde se encuentra. Sin embargo es fundamental ubicar precisamente la posición del pico R para las etapas posteriores, por lo cual se realiza una búsqueda de extremos relativos de la señal y se evalúa cual de ellos es efectivamente el pico buscado.

Cuando se termina de grabar el complejo, se actualiza el umbral y se recorre la energía buscando donde comienza a subir nuevamente para evitar que el algoritmo analice una y otra vez el mismo pico de energía. A este punto se le refiere como "comienzo de loma". Sin embargo antes de volver al principio se evalúa si es necesario realizar un salto hacia adelante en la señal basado en la frecuencia cardíaca que se viene registrando, lo que mejora la velocidad del detector.

Finalmente se mide el tiempo que ha pasado desde la última detección y en caso de ser excesivo se realiza una búsqueda hacia atrás que se explica más adelante.

2.3.1. Umbrales

Los umbrales son parte esencial del algoritmo de detección. En ellos se basa la decisión de determinar cuales picos de energía corresponden a complejos QRS y cuales a ruido en la señal. Un umbral fijo a lo largo de toda la detección en este tipo de señales es obviamente una opción sumamente deficiente, por lo que la utilización de umbrales adaptativos es imprescindible.

En este algoritmo se utiliza un umbral que se actualiza después de cada pico de energía. Para ello se lleva un registro del nivel de picos que corresponden a señal (SPK) y otro que corresponde a los de ruido (NPK). Como se ve en las ecuaciones 2.6

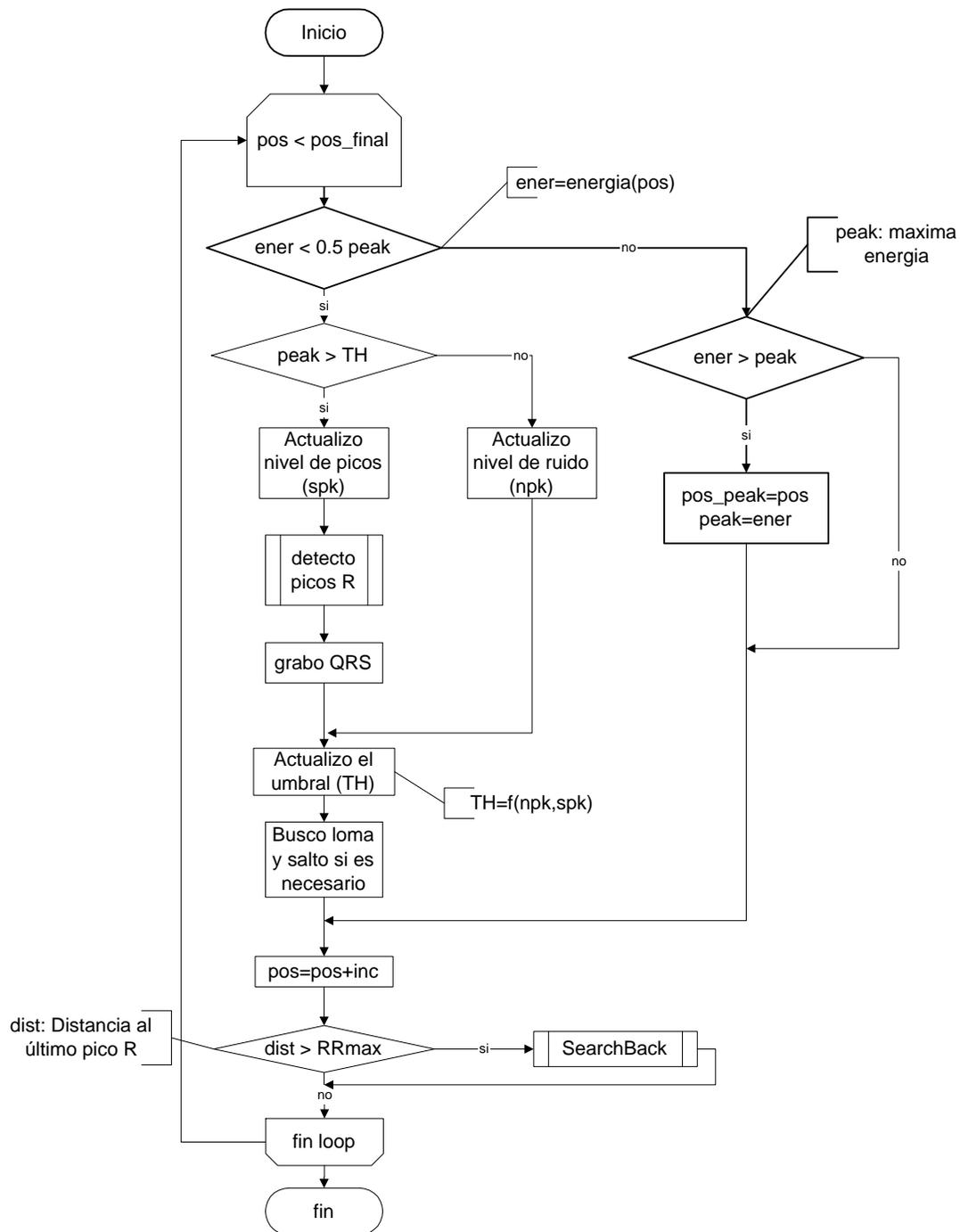


Figura 2.9: Algoritmo de Detección de complejos QRS

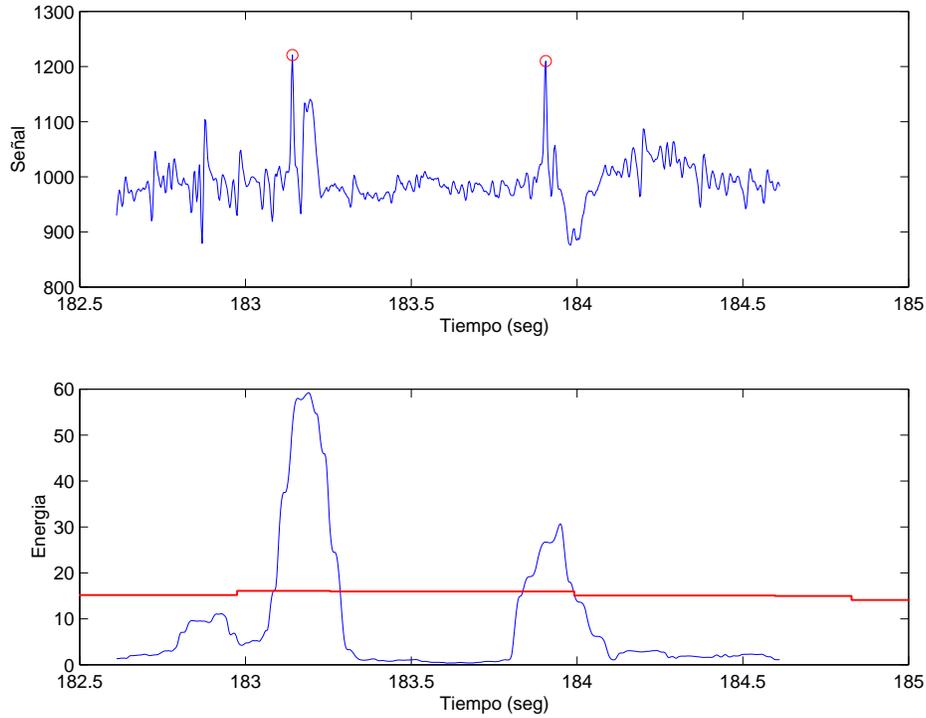


Figura 2.10: Ejemplo del funcionamiento de los umbrales adaptativos

y 2.7 estos registros se realizan con un promedio ponderado del nivel del último pico y el anterior registro. Así, se le va dando cada vez menor importancia en el promedio a los niveles más alejados del pico actual, y son los niveles más cercanos los que dominan el promedio. El parámetro w_{PK} permite ajustar cuanto afecta cada nuevo pico incluido en el promedio. Este parámetro se determinó experimentalmente y su valor final fue $w_{PK} = 0,125$

$$SPK_i = w_{PK} \times peak + (1 - w_{PK}) \times SPK_{i-1} \quad (2.6)$$

$$NPK_i = w_{PK} \times peak + (1 - w_{PK}) \times NPK_{i-1} \quad (2.7)$$

NPK se inicializa con valor cero, y para SPK se tomó un valor típico (60) en la señal de energía. Cada uno de estos dos niveles es actualizado cada vez que se determina que un pico (cuyo valor está dado en la variable 'peak') corresponde a un complejo (SPK) o a ruido (NPK). A partir de ellos se calcula el umbral con la ecuación 2.8.

$$TH_1 = NPK + w_{TH} \times (SPK - NPK) \quad (2.8)$$

Así, w_{TH} determina a que altura entre los dos niveles se fija el umbral. En la literatura [13] se sugiere un valor de aproximadamente 0.25, pero las experiencias mostraron una mejor performance para $w_{TH} = 0,21$. Este umbral es utilizado durante el funcionamiento normal, sin embargo cuando se pasa a modo "Search Back" (sec.2.3.3) se utiliza como umbral de detección la mitad del valor de TH_1 , lo que equivale a un nivel de exigencia menor para detectar un pico:

$$TH_2 = \frac{TH_1}{2} \quad (2.9)$$

La figura 2.10 muestra el funcionamiento de los umbrales adaptativos en una sección del registro 104 de la base de datos del MIT-BIH ([9]) donde existe un nivel considerable de ruido. En ella se puede observar en la gráfica superior, el ECG con los picos R de los complejos QRS marcados y en la gráfica inferior, la energía de la derivada y como varía el nivel del umbral a medida que el detector reconoce picos de ruido y picos de señal. De acuerdo a las ecuaciones 2.7 y 2.8, cuando analiza el pico de energía debido a ruido que aparece al comienzo de la ventana, el nivel del umbral sube. Luego, en el segundo latido que aparece, al tener éste un nivel de energía considerablemente menor que el anterior latido, de acuerdo a las ecuaciones 2.6 y 2.8, el umbral decae.

2.3.2. Detección de Picos R

La determinación del pico R se realiza mediante una evaluación de extremos relativos en una ventana de búsqueda. Una vez determinado el pico de energía, se estima el punto Q 100ms atrás sobre la señal original y se marca a partir de allí una ventana de búsqueda de 200 ms o hasta la última muestra recibida. En esta ventana, se hallan los máximos y mínimos relativos los cuales son evaluados para determinar cual de ellos es el pico R.

Los criterios para elegir el pico son:

Energía: El valor de la energía de la señal en la muestra. Se normaliza el valor de cada pico con el máximo. Este es el criterio con mayor peso a la hora de elegir el pico R.

Distancia al pico de energía: La distancia (en valor absoluto) al máximo de energía del complejo QRS. Se normaliza el valor de cada pico con el máximo. Para elegir el pico se toma uno menos el valor de esta característica en cada pico para darle mayor importancia a los que están cerca del pico de energía.

Distancia horizontal a Q: La separación en el tiempo entre la extremo evaluado y el punto Q estimado. Se normaliza el valor de cada pico con el máximo. Debido a como se eligen los extremos relativos, los valores mas alejados a Q tienen mayor probabilidad de ser el pico R.

Distancia vertical a Q: Es igual a la anterior, pero la separación es en amplitud (en valor absoluto).

A cada uno de estos criterios se le asigna un peso relativo en la toma de la decisión, [0.7, 0.3, 0.3, 0.2] respectivamente. Estos pesos fueron fijados experimentalmente a partir de pruebas realizadas con los registros del MIT-BIH. En la figura 2.11 se muestra un ejemplo representativo donde se ve el pico de energía, los límites de la ventana de 200ms, los extremos relativos detectados y el extremo elegido como pico R.

2.3.3. SearchBack

La técnica de Search-Back consiste en la búsqueda hacia atrás de un pico R, cuando no ha sido detectado durante un lapso de tiempo luego del pico R anterior. Para implementar esta técnica se mantienen dos promedios del intervalo R-R. El primero ($RRAV_1$) es el promedio de los últimos 8 intervalos R-R, es decir

$$RRAV_1 = \frac{RR_n + \dots + RR_{n-7}}{8} \quad (2.10)$$

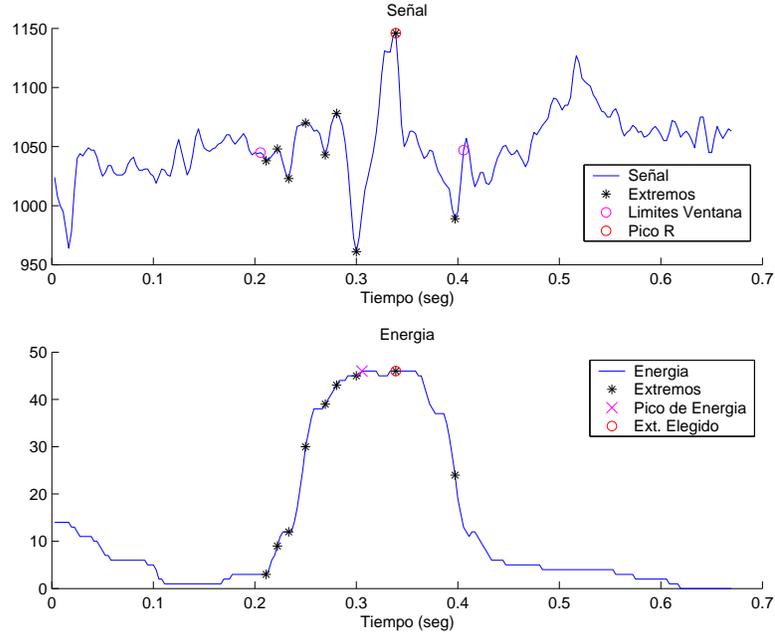


Figura 2.11: Ejemplo de detección de un pico R.

El segundo ($RRAV_2$) es el promedio de los últimos 8 intervalos R-R "normales". Se le llama normales a los intervalos RR que caen dentro de los siguientes límites:

$$\begin{aligned} RR_{bajo} &= 0,92 \cdot RRAV_1 \\ RR_{alto} &= 1,16 \cdot RRAV_1 \end{aligned}$$

Entonces el Search-Back se aplica cuando no se detecta un pico R durante un lapso de tiempo dado por:

$$RR_{max} = 1,66 \cdot RRAV_2 \quad (2.11)$$

Una vez que el algoritmo entra en el estado de Search-Back, retrocede hasta el último comienzo de loma" (sec.2.3) seleccionado. A partir de ese punto comienza a buscar picos R disminuyendo el umbral a la mitad (usando TH_2 , ecuación 2.9), y actualizando los umbrales usando un peso de $w_{PK} = 0,25$. Si se recuerda que el peso del umbral normal era 0.125 (sec.2.3.1), se puede ver que el efecto de este cambio es darle más importancia a los picos nuevos³.

El algoritmo se detiene cuando encuentra un pico R o cuando llega al punto donde entró en Search-Back en cuyo caso decide que no hubo ningún latido. Al finalizar el algoritmo, cualquiera sea el resultado, el umbral y la ponderación de picos nuevos vuelven a sus valores normales.

2.3.4. Período Refractorio

El período refractorio tiene como objeto dejar en suspenso la detección durante un período de tiempo inmediatamente posterior a un complejo QRS donde no se

³Esto es porque si no se detectó el pico en la primera pasada, significa que el umbral es muy alto, y si se pondera mucho el pico nuevo, se logra bajar sensiblemente el umbral, y evitar este tipo de situaciones se repitan.

espera que aparezca otro latido dada la frecuencia cardíaca⁴ que se viene registrando. Para ello el salto se calcula a partir del promedio de los últimos 8 intervalos RR (ecuación 2.10):

$$\text{salto} = \frac{RRAV_1}{4} \quad (2.12)$$

Sin embargo, si la frecuencia cardíaca no se está comportando de forma estable (si existe una desviación muy grande en los últimos 8 intervalos RR con los que se calcula el promedio), se utiliza un salto mínimo dado por

$$\text{salto}_{min} = 0,39 \times \sqrt{RR_{ultimo}} \quad (2.13)$$

donde es fisiológicamente imposible que se de un nuevo latido [15].

2.4. Resultados

Se desarrolló un detector capaz de funcionar 'on-line' gracias a que ninguno de los algoritmos implicados necesita utilizar información en el futuro con respecto a la última muestra adquirida. El retardo entre la adquisición de la muestra correspondiente a un pico R y la detección y extracción del complejo QRS es inferior a medio segundo (del orden de algunas centenas de milisegundo). Este retardo es un poco alto para poder catalogar el detector de real-time, sin embargo es perfectamente aceptable para la clasificación en línea de complejos QRS ya que la memoria necesaria sería inferior a 100 muestras.

La performance de este detector, al igual que el resto del proyecto, fue evaluada utilizando la base de datos del MIT-BIH [9] con los criterios de evaluación recomendados por la AAMI (Association for the Advancement of Medical Instrumentation [10]). La utilización de la metodología de evaluación recomendada en los estándares desarrollados por la AAMI para electrocardiógrafos ambulantes (ANSI/AAMI EC38:1998) y para test y evaluación de la performance de algoritmos de medida de ritmo cardíaco y segmentos ST (ANSI/AAMI EC57:1998) permite realizar evaluaciones reproducibles que dan objetividad a los resultados obtenidos por una investigación. En particular, el paquete de software WFDB [16] incluye programas que realizan la evaluación de los resultados de un detector de acuerdo a los estándares de la AAMI. Para poder comprender los resultados de ésta evaluación se definen las siguientes clases de latidos:

TP: Verdaderos Positivos (True Positive) son los latidos correctamente detectados.⁵

FP: Falsos Positivos (False Positive) son picos de la señal que el detector marcó erróneamente como latidos.

FN: Falsos Negativos (False Negative) son latidos que no fueron detectados.⁶

Estas definiciones permiten calcular dos medidas de la performance del detector:

Valor Predictivo Positivo (+P): Es una medida de cuantos de los latidos detectados son verdaderos.

$$+P = \frac{TP}{TP + FP} \quad (2.14)$$

⁴La frecuencia cardíaca instantánea se define como: $f_{cardiaca} = \frac{1}{RR}$

⁵Esto incluye una correcta determinación del pico R

⁶Esto, a su vez, incluye latidos detectados a los que no se les determinó correctamente el pico R

Sensibilidad (Se): Es una medida de cuantos latidos se detectan de entre todos los latidos analizados por el detector.

$$Se = \frac{TP}{TP + FN} \quad (2.15)$$

La idea de estas medidas es poder ver si un detector tiende a perderse latidos (Se baja) para evitar detectar latidos que no lo son (+P alto) o si, todo lo contrario, tiende a detectar latidos que no lo son (+P bajo) para no perderse ningún latido (Se alta).

Los criterios de la AAMI, establecen que en la evaluación de un detector los primeros 5 minutos de señal no deben ser considerados, dejándose para la adaptación del algoritmo. Así mismo determina que un latido fue bien detectado cuando el pico R determinado por el algoritmo se encuentra dentro de una ventana de 150ms centrada en el verdadero pico R. El resultado de la evaluación según los estándares de la AAMI se pueden ver en la tabla 2.1.

Sin embargo, para las siguientes etapas de este trabajo estas dos condiciones de la evaluación según los estándares de la AAMI no resultaban convenientes. Primero porque los primeros 5 minutos de señal serán usados para el entrenamiento de las redes neuronales encargadas de la clasificación y es necesario saber que el detector mantiene la performance en niveles aceptables durante ese período, en segundo lugar la ventana de 150ms resulta un poco excesiva, ya que para una correcta extracción de características y por consiguiente una correcta clasificación es necesario identificar el pico R con mayor precisión. Por ello también se realizó una evaluación considerando los primeros 5 minutos de señal y tomando una ventana de 100ms para la ubicación del pico R. El resultado de ésta evaluación más restrictiva se puede ver en la tabla 2.2.

2.4.1. Performance del Algoritmo Desarrollado

En las tablas 2.1 y 2.2 se aprecia la performance del algoritmo desarrollado. Tanto la sensibilidad como el valor predictivo positivo permiten afirmar que el algoritmo mantiene alto el nivel de detección (Sensibilidad) y no confunde ruido u otras señales presentes en el ECG con latidos (Valor Predictivo Positivo) para una gran variedad de señales electrocardiográficas como las que contiene la base de datos del MIT-BIH [9].

La merma en la performance apreciada en la tabla 2.2 con respecto a la obtenida utilizando los criterios de la AAMI para la evaluación (tabla 2.1) es debida, principalmente, a la reducción en la ventana de tiempo utilizada por el evaluador para determinar cuando la detección del pico R es correcta. La evaluación del detector en los primeros 5 minutos no agrega casi problemas ya que el algoritmo desarrollado prácticamente no precisa de un período de adaptación.

En términos más claros, considerando que a un ritmo cardíaco normal ocurren aproximadamente entre 4000 y 5000 latidos por hora, este detector perderá 9.2 latidos por hora e identificará erróneamente sectores de señal como latidos 8 veces por hora, lo cual, como se verá a continuación, se encuentra al nivel del estado del arte de los detectores automáticos de latidos en electrocardiogramas.

Registro	Latidos	TP	FN	FP	Se	+P
100	1902	1901	1	0	99,95 %	100,00 %
101	1523	1523	0	2	100,00 %	99,87 %
102	1821	1821	0	0	100,00 %	100,00 %
103	1729	1727	2	0	99,88 %	100,00 %
104	1857	1857	0	6	100,00 %	99,68 %
105	2155	2145	10	50	99,54 %	97,72 %
106	1696	1695	1	0	99,94 %	100,00 %
107	1784	1782	2	0	99,89 %	100,00 %
118	1916	1916	0	0	100,00 %	100,00 %
119	1661	1661	0	0	100,00 %	100,00 %
200	2168	2165	3	4	99,86 %	99,82 %
201	1521	1482	39	0	97,44 %	100,00 %
202	1871	1868	3	0	99,84 %	100,00 %
203	2481	2464	17	20	99,31 %	99,19 %
205	2201	2198	3	0	99,86 %	100,00 %
207	1592	1591	1	4	99,94 %	99,75 %
208	2437	2422	15	7	99,38 %	99,71 %
209	2518	2518	0	3	100,00 %	99,88 %
210	2204	2193	11	3	99,50 %	99,86 %
212	2285	2284	1	0	99,96 %	100,00 %
213	2700	2698	2	0	99,93 %	100,00 %
214	1878	1878	0	1	100,00 %	99,95 %
215	2795	2795	0	0	100,00 %	100,00 %
217	1845	1840	5	2	99,73 %	99,89 %
219	1773	1773	0	0	100,00 %	100,00 %
Total	50313	50197	116	102	99,77 %	99,80 %

Tabla 2.1: Performance del detector en la base de datos del MIT-BIH, evaluada según los criterios de la AAMI.

2.4.2. Comparación de la Performance con otros Algoritmos

En la literatura especializada se encuentran numerosos trabajos sobre detectores que utilizan diversas técnicas y estrategias. En esta sección se va a comparar como es la performance de esos algoritmos para tener una idea de que nivel se alcanzó con el algoritmo desarrollado.

A la hora de comparar performances, siempre es necesario aclarar en que condiciones se evalúa y cómo se evalúa cada detector que se compara, ya que ello puede llevar a confusiones.

A real-time QRS detection algorithm [14]

Gran parte de los detectores desarrollados se han basado en este algoritmo publicado por Pan y Tompkins [14] en 1985. El mismo muestra una performance muy buena. Tiene un sensibilidad de 99.76 % y un valor predictivo positivo de 99.56 %, lo cual es apenas superado por el detector implementado en CARDIDENT. Es necesario notar que en el año 1985 los criterios de la AAMI [10] aún no existían por lo que puede haber alguna discrepancia en el modo de evaluar.

Registro	Latidos	TP	FN	FP	Se	+P
100	2273	2272	1	0	99,96 %	100,00 %
101	1865	1864	1	4	99,95 %	99,79 %
102	2187	2187	0	0	100,00 %	100,00 %
103	2084	2082	2	0	99,90 %	100,00 %
104	2229	2214	15	37	99,33 %	98,36 %
105	2572	2560	12	51	99,53 %	98,05 %
106	2027	2026	1	0	99,95 %	100,00 %
107	2137	2134	3	1	99,86 %	99,95 %
118	2278	2278	0	0	100,00 %	100,00 %
119	1987	1987	0	0	100,00 %	100,00 %
200	2601	2598	3	4	99,88 %	99,85 %
201	1963	1924	39	0	98,01 %	100,00 %
202	2136	2133	3	0	99,86 %	100,00 %
203	2980	2954	26	25	99,13 %	99,16 %
205	2656	2653	3	0	99,89 %	100,00 %
207	1860	1857	3	6	99,84 %	99,68 %
208	2955	2933	22	10	99,26 %	99,66 %
209	3004	3004	0	3	100,00 %	99,90 %
210	2650	2635	15	8	99,43 %	99,70 %
212	2748	2747	1	0	99,96 %	100,00 %
213	3251	3249	2	0	99,94 %	100,00 %
214	2261	2258	3	2	99,87 %	99,91 %
215	3363	3363	0	1	100,00 %	99,97 %
217	2208	2203	5	2	99,77 %	99,91 %
219	2154	2154	0	0	100,00 %	100,00 %
Total	60429	60269	160	154	99,74 %	99,75 %

Tabla 2.2: Performance del detector en la base de datos del MIT-BIH, incluyendo los primeros 5 minutos de señal y tomando una ventana para el pico R de 100ms.

ECG Beat Detection Using Filter Banks [17]

Afonso *et al.* diseñan un algoritmo que incorpora bancos de filtros digitales (filter banks) que descomponen el ECG en sub-bandas de ancho uniforme, lo que permite realizar análisis en tiempo y frecuencia independientemente. El algoritmo tiene una sensibilidad de 99.59 % y un valor predictivo positivo de 99.56 %. Este trabajo está validado contra la base de datos del MIT-BIH [9] y utiliza los criterios de la AA-MI para la evaluación. Queda claro que la performance de este algoritmo es inferior a la alcanzada por el detector de CARDIDENT, aunque vale aclarar que Afonso *et al.* utilizaban la base completa (48 registros) del MIT-BIH [9].

Wavelet Transform-Based QRS Complex Detector [18]

Kadambe *et al.* describen un detector de complejos QRS basado en la transformada wavelet diádica (D_yWT , dyadic wavelet transform) que es robusta a las morfologías variables en el tiempo de los complejos QRS y al ruido. Se diseña una wavelet spline apropiada para la detección de QRS y las escalas de esta wavelet se

eligen en base a las características espectrales de la señal ECG. La performance del detector se realiza contra señales de la American Heart Association (AHA). La forma de evaluar la performance de este algoritmo es ligeramente diferente a como se evalúa en este trabajo, en el mismo, los resultados se registran como el porcentaje de registros (o 'tapes') en los que el error de detección es menor a 1 % o 5 %. Los mejores resultados son de 67.1 % de los registros con menos de 1 % de error y 87.1 % con menos de 5 % de error. En el detector desarrollado en CARDIDENT 20 de 25 registros tienen un error menor al 1 % (80 %)⁷ y ninguno tiene un error superior al 5 % (100 % de los registros). Si bien la base de datos sobre la que evalúan Kadambe *et al.* [18] tiene 70 registros (contra 25 de los 48 de la base del MIT con los se evalúa el detector propuesto) la diferencia es bastante grande como para suponer que la performance superior de este algoritmo se va a mantener.

An Efficient R-R Interval Detection for ECG Monitoring System [19]

Kohama *et al.* proponen un detector de intervalos R-R para señales ECG. El algoritmo propuesto detecta las ondas R eficientemente buscando solamente en el rango de tiempo donde se espera la siguiente onda R. Para realizar esta detección se realiza un preprocesamiento que reduce el ruido y el corrimiento de la señal ECG lo que mejora la performance del algoritmo. Finalmente en las simulaciones hechas en la base de datos de la American Heart Association (AHA) para los registros de las series 1200s y 2200s la performance registrada fue de una sensibilidad de 99.36 % y un valor predictivo positivo de 99.91 %. Es claro que la performance de este algoritmo tiene una pequeña tendencia a no detectar latidos falsos, al precio de perderse algunos latidos.

CLASICAR: Clasificador de complejos QRS [15]

En CLASICAR se realiza un detector muy similar al implementado en este trabajo, ya que ambos están basados en la implementación del algoritmo de Pan-Tompkins ([14]) propuesta por Afonso *et al.* [13]. Sin embargo existen algunas diferencias fundamentales con este trabajo, siendo la principal el hecho de que CLASICAR utiliza información de hasta un minuto adelante en el tiempo para ajustar sus umbrales de detección, limitándolo a la clase de algoritmos **off-line**. Para la evaluación se utilizaron los mismos 25 registros de la base de datos del MIT-BIH que se utilizan en CARDIDENT, sin embargo el número de latidos considerados reales en cada registro no es correcto. Esto es debido a que en este trabajo no se utilizó la información que incluye la base de datos ni las herramientas de evaluación disponibles en el paquete de software WFDB [16]. La información sobre la ubicación de los latidos fue, al parecer, recreada durante el desarrollo de este trabajo. La performance reportada es de una sensibilidad de 99.83 % y un valor predictivo positivo de 99.36 %. Más allá de la evaluación poco objetiva realizada, se puede ver que este detector tiene una pequeña tendencia a no perderse latidos al precio de detectar latidos falsos.

⁷El error se midió como

$$err = \frac{FP + FN}{TP}$$

	Se	+P	err: 1 % / 5 %	Base	AAMI
Pan-Tompkins	99,76 %	99,56 %	–	MIT-BIH	No
Afonso et al.	99,59 %	99,56 %	–	MIT-BIH	Si
Kadambe et al.	–	–	67,1 % / 87,1 %	AHA	No
Kohama et al.	99,36 %	99,91 %	–	AHA	No
CLASICAR	99,83 %	99,36 %	–	MIT-BIH	No
CARDIDENT (AAMI)	99,77 %	99,80 %	–	MIT-BIH	Si
CARDIDENT (estricta)	99,74 %	99,75 %	80 % / 100 %	MIT-BIH	No

Tabla 2.3: Comparación de las performances de algunos algoritmos publicados en la literatura especializada con el algoritmo desarrollado en **CARDIDENT**.

2.5. Conclusiones

Se desarrollo un algoritmo de detección on-line de complejos QRS en señales ECG. El mismo utiliza filtros digitales para aislar la banda del espectro correspondiente a los complejos QRS y evitar así las ondas P y T y todas las fuentes de ruido que aparecen en un electrocardiograma que fueron mencionadas anteriormente. Luego utiliza umbrales adaptativos para la detección de los complejos y se realiza un algoritmo de búsqueda de extremos relativos para detectar correctamente el pico R. El algoritmo funciona eficientemente ya que después de cada detección exitosa inhibe la detección durante un período adaptativo de tiempo para no analizar partes de la señal donde no se esperan nuevos complejos. Finalmente cuenta con un respaldo para el caso de no detectar un latido de baja energía realizando una re-búsqueda con un umbral menor.

La performance del algoritmo de detección es comparable e inclusive ligeramente superior a los detectores que aparecen en la literatura, por lo que se puede decir que el mismo se encuentra al nivel del estado del arte de los detectores actuales.

La elevada tasa de detección permite estar seguros de que el clasificador desarrollado en este trabajo podrá trabajar con un buen nivel de seguridad de que las señales que esta intentando clasificar son efectivamente complejos QRS.

2.6. Posibles Mejoras

A pesar de los buenos resultados que muestra la evaluación de este detector, quedan aún varios aspectos donde es posible mejorar un poco más el rendimiento.

La principal mejora a implementar en el futuro es la incorporación en la detección de un segundo canal (disponible en la base de datos) lo que permitiría que cuando el nivel de ruido en el canal principal haga muy difícil la detección, utilizar el segundo canal para confirmar o descartar la detección. Otra posibilidad es realizar la detección en los dos canales en paralelo (mayor poder de procesamiento necesario). En ese caso se lleva un promedio ponderado del nivel de ruido que se va registrando en cada canal, así cuando la detección de cada uno no coincide, se puede decidir que canal se encuentra en mejores condiciones para realizar la detección.

Un segundo punto a mejorar en este detector es la detección del pico R en los complejos QRS. Si bien el algoritmo desarrollado para ese fin tiene una performance muy superior a soluciones más simples (como elegir el punto más distante en amplitud

al punto Q estimado) en la performance total del detector alcanzada hasta el momento hay un componente de error debido a una incorrecta determinación del pico R, aún cuando se hubo detectado correctamente la existencia de un complejo QRS, tal como se ve en la tabla 2.2 donde la performance cae por esta causa principalmente.

Capítulo 3

EXTRACCIÓN DE CARACTERÍSTICAS

“To measure the man, measure his heart.”

Malcolm Stevenson Forbes

3.1. Introducción

Para la clasificación de las arritmias en los latidos es necesario antes extraer de la señal un conjunto de características que describan la misma. Varios enfoques han sido propuestos, siendo el uso directo de las muestras temporales el más utilizado ([2],[3],[5],[6]). Aunque tiene un buen desempeño en condiciones favorables, su performance baja con la presencia de ruido en la señal y es sensible a la ubicación temporal de la ventana del QRS. Otra propuesta es la medición de características temporales de la señal como son pendientes, ancho del QRS, área del QRS, etc. (por ejemplo en [7]) y su comparación con umbrales establecidos experimentalmente. Desgraciadamente, este enfoque no es robusto frente a la variabilidad de las morfologías de los complejos QRS, y su efectividad depende de la representatividad de las características elegidas. Para enfrentar varias de estas dificultades, especialmente la sensibilidad al ruido, fueron propuestos espacios de transformadas cuyas métricas son más robustas. Además, la reducción de la dimensionalidad dada por el truncamiento de los coeficientes de la transformada ayuda en la velocidad de comparación entre latidos. Como contrapartida la dimensión de la transformada no debe ser muy baja para no perder información clínica relevante. La transformada debe elegirse de forma de minimizar la distancia entre latidos de la misma clase y maximizar la distancia inter-clase. Es común en la literatura para problemas de clasificación de complejos QRS trabajar con la transformada de Karhunen-Loève y la de Hermite.

Organización del capítulo

En la sección 3.3 se describe la transformada de Karhunen-Loève y su aplicación a la extracción de características de la señal de ECG. Lo mismo se hace en 3.4 con la transformada de Hermite. En la misma se presenta el algoritmo desarrollado para el ajuste eficiente del parámetro de la transformada. La comparación de ambas se hace en 3.5. Por último la incorporación de las características temporales se discute en 3.6.

3.2. Preprocesamiento

Como ya fue mencionado en el capítulo 1, la señal del ECG se ve contaminada por una variedad de fuentes de ruido. Se destaca entre ellas la respiración del paciente que causa el *baseline drift*. El mismo se manifiesta como un movimiento de la línea de base del ECG (ver señal original en la figura 3.1), que afecta la forma de los complejos QRS. Debido a su naturaleza, la mayor parte del espectro del *baseline drift* se concentra en las frecuencias bajas, del orden de 1 Hz. Para atenuar su presencia en la señal, así como el componente de continua, se diseñó un filtro pasaalto FIR de orden elevado ($N = 512$), ganancia de banda pasante 1, frecuencia de -3 dB de 1.25 Hz y ganancia a 0.1 Hz de -50 dB. El mismo fue diseñado utilizando una ventana de Kaiser. Su respuesta en frecuencia y fase se puede apreciar en la figura 3.2. El filtro atenúa el *baseline drift*, un ejemplo de esto se puede apreciar en la figura 3.1 donde se muestra su comportamiento en una sección del registro 203. Sin embargo, presenta el inconveniente de demandar 512 operaciones por muestra y un retardo de 256 (aprox. 700 ms). En [20] se utiliza un spline cúbico para eliminar la línea de base, lo que puede ser una alternativa si es imposible implementar el filtro por costo computacional.

Una vez detectado el complejo QRS y estimada la posición de pico R, una ventana de ancho $t_0 = 200$ ms es extraída de la señal

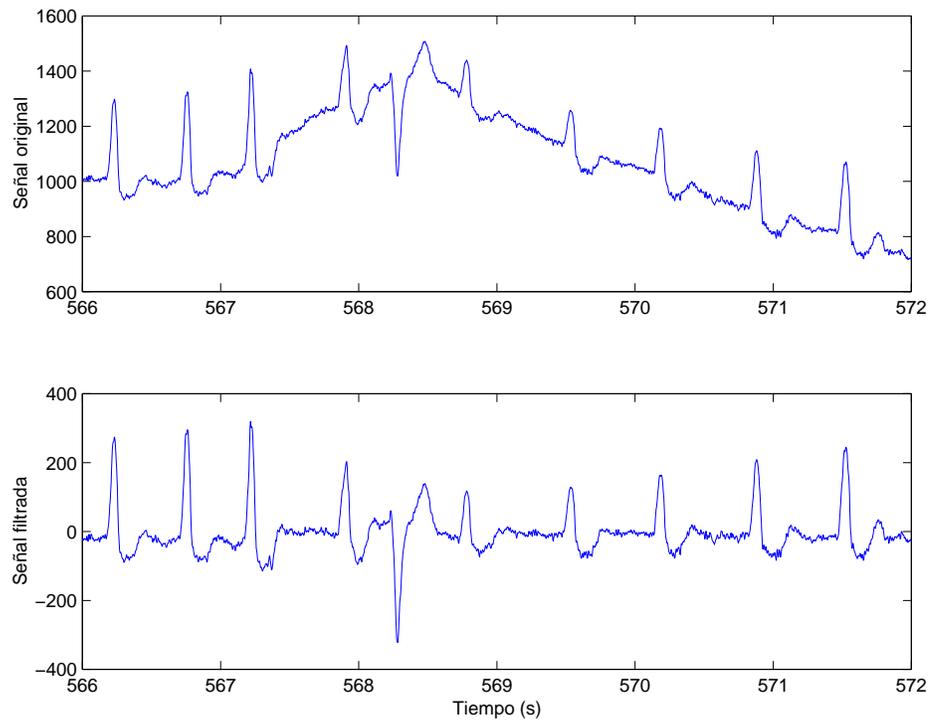


Figura 3.1: Filtrado del *baseline drift* en el registro 203

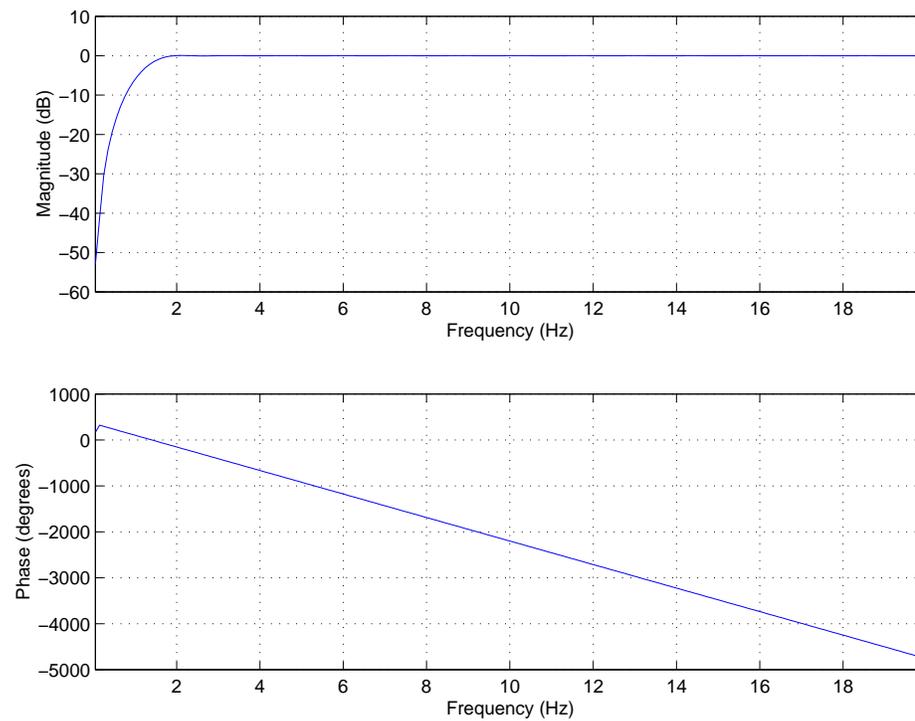


Figura 3.2: Respuesta en frecuencia y fase del filtro utilizado para eliminar el *baseline drift*

3.3. Transformada de Karhunen-Loève

La transformada de Karhunen-Loève (K-L) es una transformación lineal dependiente de la señal, que es óptima en el sentido del error cuadrático medio. Tiene otras propiedades interesantes como son entropía de representación mínima y coeficiente no correlacionados.

La transformada K-L ha sido empleada extensamente en el análisis del segmento ST y la onda T ([21],[22],[23],[24],[25],[26]), compresión de ECG ([27],[28]) y clasificación de latidos ([4]).

3.3.1. Base de la transformada

Para obtener la base de la transformada K-L (también conocida como *Análisis de Componentes Principales*) se minimiza el error medio de proyección de los vectores sujeto a la condición de ortonormalidad de la base.

Cualquier vector \mathbf{s} puede expandirse en una base de vectores ortonormales

$$\mathbf{s} = \sum_{n=1}^D c_n \omega_{\mathbf{n}} \quad (3.1)$$

donde D es la dimensión del espacio, c_n es el coeficiente correspondiente al n -ésimo vector de la base $\omega_{\mathbf{n}}$.

Al reducir la dimensionalidad del espacio se trabaja con una aproximación $\hat{\mathbf{s}}$ resultante de truncar la expansión anterior

$$\hat{\mathbf{s}} = \sum_{n=1}^N c_n \omega_{\mathbf{n}} \quad (3.2)$$

sujeta a un error medio sobre la muestra

$$e = E[(\mathbf{s} - \hat{\mathbf{s}})^T (\mathbf{s} - \hat{\mathbf{s}})] \quad (3.3)$$

donde $E[\mathbf{s}]$ es la esperanza sobre el espacio muestral Usando 3.1 y 3.2 se puede expresar el error como

$$e = E \left[\sum_{n=N+1}^D c_n \omega_{\mathbf{n}}^T \sum_{m=N+1}^D c_m \omega_{\mathbf{m}} \right] \quad (3.4)$$

y debido a la ortonormalidad de las bases ω ,

$$e = E \left[\sum_{n=N+1}^D c_n^2 \right] \quad (3.5)$$

Tomando en cuenta que,

$$\omega_{\mathbf{m}}^T \mathbf{s} = \sum_{n=1}^D c_n \omega_{\mathbf{m}}^T \omega_{\mathbf{n}} = c_m \quad (3.6)$$

se lo puede sustituir en 3.5 y en entonces se obtiene

$$e = \sum_{n=N+1}^D \omega_{\mathbf{n}}^T R \omega_{\mathbf{n}} \quad (3.7)$$

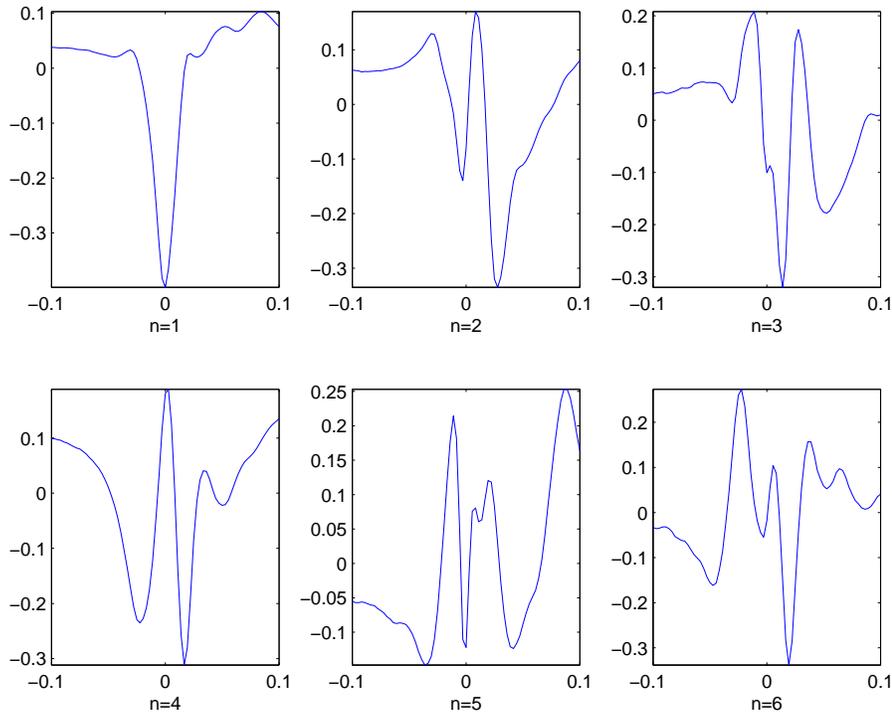


Figura 3.3: Primeras seis funciones base de la transformada KL (abscisas en seg).

con $R = E[\mathbf{ss}^T]$.

Para encontrar las bases ω óptimas en el sentido que minimicen e sujeto a la condición de ortonormalidad se puede emplear el método de multiplicadores de Lagrange. La función a minimizar entonces es

$$E = e - \sum_i \lambda_i (\omega_i^T \omega_i - 1) \quad (3.8)$$

Sujeto a la condición

$$\frac{\partial E}{\partial \omega_{\mathbf{n}}} = 0 \quad (3.9)$$

Entonces, dado que la matriz R es simétrica por construcción,

$$\frac{\partial E}{\partial \omega_{\mathbf{n}}} = 2(R\omega_{\mathbf{n}} - \lambda_n \omega_{\mathbf{n}}) = 0 \quad (3.10)$$

con lo que se obtiene el problema de valores propios

$$R\omega_{\mathbf{n}} = \lambda_n \omega_{\mathbf{n}} \quad (3.11)$$

De esta forma se puede ver que los vectores correspondientes a valores propios más altos representan mayor proporción de energía proyectada⁸. Por lo tanto como extractor de características de orden N se deben usar los primeros N vectores propios $\mathbf{u}_1, \dots, \mathbf{u}_N$, ordenados por valor propio descendente.

⁸Ver [29] para una demostración de esto.

Definiendo la matriz U ,

$$U = [\mathbf{u}_1, \dots, \mathbf{u}_N] \quad (3.12)$$

entonces el vector

$$\hat{\mathbf{x}} = U^T \mathbf{s} \quad (3.13)$$

contiene la *mayor parte de la información* de \mathbf{s} resumida en N coeficientes.

3.3.2. Elección de la muestra (para generar la base)

Los latidos utilizados para construir la base fueron elegidos de manera que se obtenga una buena representación de cada tipo de latido, independientemente de su proporción en la muestra original. Para realizar esto se eligieron aproximadamente la misma cantidad de latidos de cada una de las clases disponibles en la base de datos⁹. Éstos se tomaron de algunos registros solamente, que entonces representan la estadística conocida *a priori* de las señales de ECG. Esto se hace así, en vez de tomar toda la base de datos, ya que en un caso general es imposible en general conocer la estadística de los pacientes antes de estudiarlos. Una vez seleccionada la muestra, a cada latido se le aplicó un pre-procesamiento que consiste en restar su media y dividir por su potencia (o varianza). Es decir, si \mathbf{s} es la señal original, entonces la señal a transformar es $\tilde{\mathbf{s}}$ calculada como

$$\tilde{\mathbf{s}} = \frac{\mathbf{s} - \mu}{\|\mathbf{s} - \mu\|^2} \quad (3.14)$$

En la figura 3.3 se muestran las primeras seis bases de la transformada calculadas de esta forma.

3.3.3. Extracción de características

Después de ser filtrada la señal, una ventana de ancho $t_0 = 200$ ms centrada en el pico R estimado es extraída de la señal de ECG. A la misma se le resta el promedio de la señal de manera que su media sea nula. Los coeficientes de la transformada se obtienen utilizando (3.13) sobre la base elegida como en la sección 3.3.2.

En la figura 3.4 se observa la señal del canal 1 del latido 20 del registro 200, y su proyección en la base de la transformada KL de orden 6.

3.4. Modelo paramétrico de Hermite

Las funciones ortogonales de Hermite fueron propuestas como representación paramétrica de los complejos QRS, por su similitud con las bases de la transformada de Karhunen-Loève aplicada al estudio de los mismos. Al igual que las de la transformada KL, las bases de Hermite son ortonormales y por lo tanto cada coeficiente representa información independiente de las características de la señal, por lo que la misma puede representarse con pocos coeficientes. En 1981, Sörnmo [30] informó que en promedio el 98.6% de la energía de la señal de QRS puede representarse utilizando tres coeficientes de Hermite.

A diferencia de la KL las bases de Hermite no dependen de la estadística de la señal sino que son fijas salvo por un parámetro de ancho. El modelo de Hermite ha sido usado extensamente en el estudio de electrocardiogramas, tanto en el contexto de la

⁹Existen algunas clases de las cuales no existen latidos.

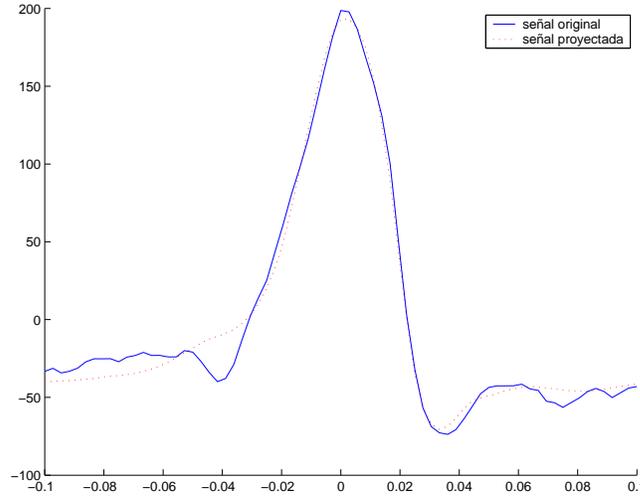


Figura 3.4: Latido 20 (Normal) del Registro 200: Proyección en las primeras 6 bases de la transformada KL (abscisas en seg).

compresión ([31], [20], [32]), así como en la clasificación ([33]).

3.4.1. Base de Hermite

Los polinomios de Hermite ($H_n(x)$) son soluciones de la ecuación diferencial ¹⁰

$$\ddot{y}(x) - 2xy'(x) + 2ny(x) = 0 \quad (3.15)$$

donde n es un entero no negativo.

Los polinomios forman un conjunto ortogonal en el intervalo $-\infty < x < \infty$ con respecto a la función peso e^{-x^2} . Puede probarse que

$$\frac{1}{2^n n! \sqrt{\pi}} \int_{-\infty}^{\infty} e^{-x^2} H_n(x) H_m(x) dx = \delta_{n,m} \quad (3.16)$$

donde $\delta_{n,m}$ es el delta de Kronecker. Entonces es posible construir una base de $L_2(-\infty, \infty)$ de la forma

$$\phi_n(x) = \frac{1}{\sqrt{2^n n! \sqrt{\pi}}} e^{-x^2/2} H_n(x) \quad (3.17)$$

Se le puede agregar un parámetro de escala para adaptarse al ancho del QRS haciendo el cambio de variable $x = \frac{t}{b}$ y entonces se obtiene las funciones ortonormales base del *modelo paramétrico de Hermite*:

$$\phi_n^b(t) = \frac{e^{-t^2/2b^2}}{\sqrt{b 2^n n! \sqrt{\pi}}} H_n(t/b) \quad (3.18)$$

¹⁰La ecuación diferencial de Hermite es un caso especial del problema de frontera de Sturm-Liouville, y aparece al tratar armónicas en mecánica cuántica.

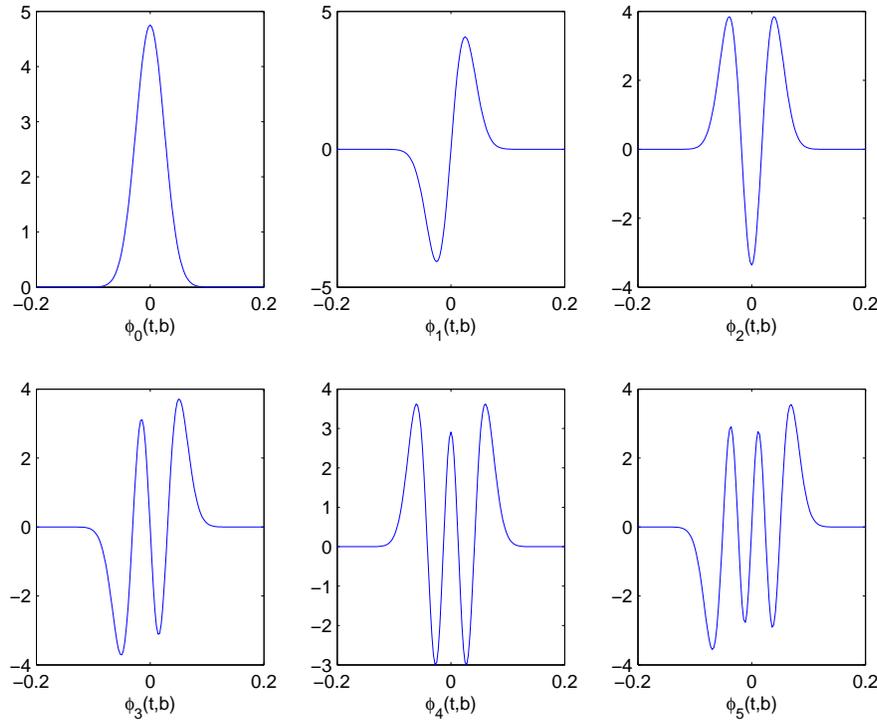


Figura 3.5: Primeras seis funciones base para $b=15$ ms.

De esta forma cualquier función de $s(t)$ de L_2 se puede expresar como

$$s(t) = \sum_{n=0}^{\infty} c_n(b) \phi_n^b(t) \quad (3.19)$$

donde $c_n(b) = \int_{-\infty}^{\infty} s(t) \phi_n^b(t) dt$, que no es más que la expansión en series de Fourier generalizada. La misma converge a $s(t)$ donde es continua y a $\frac{s(t^-) + s(t^+)}{2}$ en los puntos de discontinuidad.

En la figura 3.5 se pueden apreciar las primeras seis funciones base para $b = 15$ ms. Se puede observar la similitud de la primer función con la forma de un complejo QRS.

3.4.2. Procesamiento

Después de ser filtrada para atenuar el “*baseline drift*”, una ventana de ancho $t_0 = 200$ ms centrada en el pico R es extraída de la señal de ECG. A la misma se le resta el promedio de las primeras cinco y las últimas cinco muestras para que en los extremos se aproxime a cero. La señal es luego “extendida” con ceros al intervalo $(-t_0, t_0)$, y es expresada para un b fijo como una combinación lineal de N funciones base de Hermite ($\phi_0^b(t), \dots, \phi_{N-1}^b(t)$) más un término de error debido a la truncación.

$$s(t) \simeq \sum_{n=0}^{N-1} c_n(b) \phi_n^b(t) \quad (3.20)$$

N	3	4	5	6	7	8	9	10
max b (ms)	62	55	51	47	44	42	40	38

Tabla 3.1: Máximo b admisible para N dado

donde los coeficientes se calculan aproximando la integral sobre $(-\infty, \infty)$ como

$$c_n(b) = \sum_{i=-t_0}^{t_0} s(i)\phi_n^b(i)/fs \quad (3.21)$$

siendo fs la frecuencia de muestreo.

Obviamente, al considerar ventanas de un ancho determinado la propiedad de ortogonalidad (sección 3.4.1), y por lo tanto la expansión en series tal como se presentó, no es válida, especialmente a medida que b y n crecen. Esto es necesario ya que hay un costo computacional asociado a considerar vectores más largos y debe haber un compromiso. Sin embargo, si se considera aceptablemente cercano a cero un valor de

$$|\phi_n^b(-t_0)| = |\phi_n^b(t_0)| < 0,1 \max_{t \in [-t_0, t_0]} |\phi_n^b(t)| \quad (3.22)$$

se obtiene un b máximo como muestra la tabla 3.1. Estos valores son suficientemente grandes para una buena representación de los latidos. Por ejemplo $\phi_i^b \cdot \phi_j^b / fs$ para $i \neq j$; $i, j < 10$ es del orden de 10^{-16} para $b = 20$ ms (un valor común en latidos normales en el canal 1 de la base MIT-BIH).

Tomando $H_0(x) = 1$ y $H_1(x) = 2x$, los polinomios de Hermite se pueden calcular recursivamente con la ecuación

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x) \quad (3.23)$$

Este resultado es utilizado en el algoritmo para calcular el valor de H_n ya que es más eficiente y hay que calcular H_i para $i < N$ de todas formas.

3.4.3. Método iterativo para encontrar el b óptimo

Para representar a cada complejo QRS se realiza la expansión de Hermite con un valor del parámetro b que minimice el error de truncamiento:

$$e(b) = \left\| \mathbf{s} - \sum_{n=0}^{N-1} c_n(b)\phi_n^b \right\|_2^2 \quad (3.24)$$

Para encontrar el valor óptimo de b se utiliza el algoritmo iterativo cuasi de segundo orden de Levenberg-Marquardt. El mismo fue diseñado para aproximar la velocidad de un método de segundo orden sin tener que calcular la matriz Hessiana. Cuando la función de performance tiene la forma de suma de cuadrados (como en este caso), la matriz Hessiana se puede aproximar como

$$H = J^T J \quad (3.25)$$

y el gradiente es entonces

$$\mathbf{g} = J^T \mathbf{e} \quad (3.26)$$

donde J es la matriz Jacobiana.

El algoritmo de Levenberg-Marquardt usa esta aproximación de la matriz Hessiana en el paso de actualización

$$b_{k+1} = b_k - [J^T J + \mu I]^{-1} J^T \mathbf{e} \quad (3.27)$$

Cuando μ es grande, la ecuación se convierte en la regla de Widrow-Hoff (descenso por gradiente) con un paso de actualización pequeño.

El algoritmo comienza en este caso, y después de cada iteración exitosa (el nuevo b disminuye el error de proyección) μ se decrementa. Al tender μ a 0, el algoritmo se aproxima a la actualización de Newton pero utilizando la aproximación de la Hessiana. Por el contrario si el nuevo b no mejora la proyección, el mismo es rechazado y se aumenta μ de manera de disminuir el paso en el siguiente intento. Por lo tanto, el error siempre se reduce con cada paso del algoritmo.

El algoritmo termina cuando la actualización de b es pequeña ($\delta b \leq 10^{-4}b$) o el paso se hace muy chico ($\mu \geq 10^{10}$).

En la figura 3.6(a) se puede apreciar la proyección en las primeras 5 bases de Hermite para el latido 20 del registro 200 (canal 1). Este es un latido normal y es similar a la primer base de Hermite (ver figura 3.5), lo cual lleva a que la mayor parte de la energía se concentre en el primer coeficiente. Esto se ve en la figura 3.7(b) que muestra los 6 coeficientes correspondientes a la proyección de este latido.

El error en la proyección en función de b para este latido se puede apreciar en la figura 3.7(a), donde se puede observar como en cada iteración el error desciende monótonamente.

En la figura 3.7(d) se puede apreciar como el parámetro b disminuye su valor a lo largo de las iteraciones, para ajustarse a una medida proporcional al ancho del latido¹¹. En la figura 3.6(b) se ve la proyección de un latido ventricular, usualmente más anchos que los latidos normales. En la evolución del parámetro b para este latido que muestra la figura 3.8(a), se puede apreciar que el mismo aumentó su valor. Por otro lado, también se puede apreciar que la forma de este latido no coincide, como ocurría con el latido normal, con la forma de la primer base de Hermite. Esto se confirma en la figura 3.8(b) donde la energía esta más repartida entre todos los coeficientes.

3.4.3.1. Derivada parcial del error con respecto al parámetro b

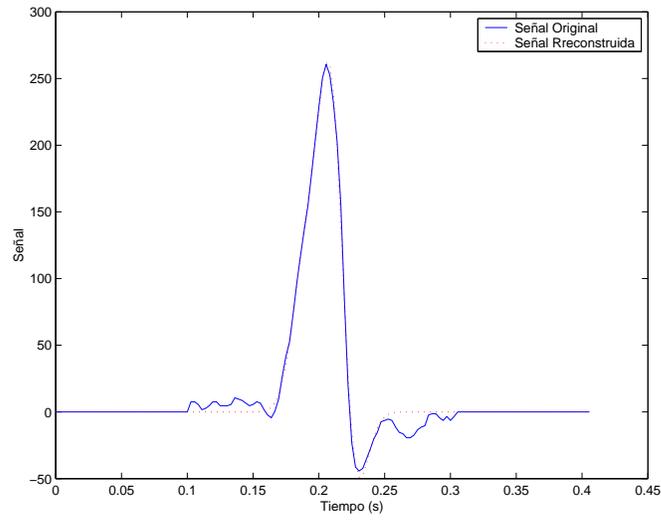
Recordando la definición del error (3.24),

$$e(b) = \left\| \mathbf{s} - \sum_{n=0}^{N-1} c_n(b) \phi_{\mathbf{n}}^{\mathbf{b}} \right\|_2^2$$

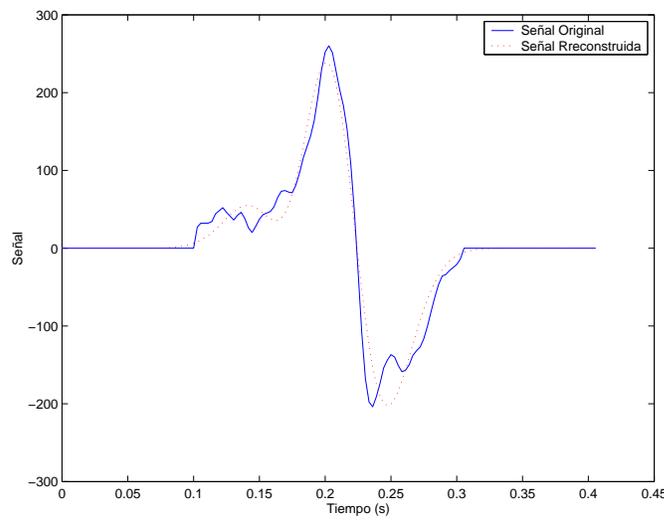
Al igual que el error, la matriz Jacobiana en este caso es un escalar, $J = \frac{de}{db}$. Entonces,

$$\frac{de}{db} = -2 \left(\mathbf{s} - \sum_{n=0}^{N-1} c_n \phi_{\mathbf{n}}^{\mathbf{b}} \right)^T \sum_{n=0}^{N-1} \left(c_n(b) \frac{\partial \phi_{\mathbf{n}}^{\mathbf{b}}}{\partial b} + \frac{dc_n(b)}{db} \phi_{\mathbf{n}}^{\mathbf{b}} \right) \quad (3.28)$$

¹¹El parámetro b regula el decaimiento de la exponencial de forma que cuando $t = b$, esta disminuye a un 60% de su valor de pico.



(a) Latido 20 del Registro 200



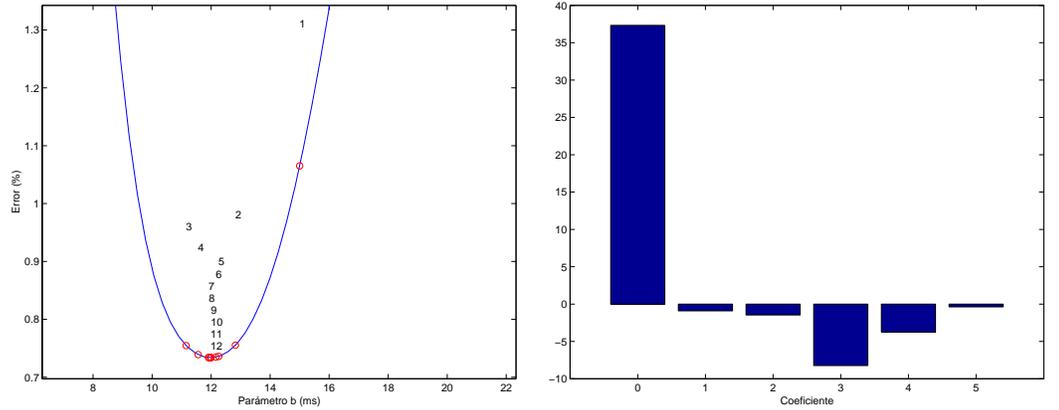
(b) Latido 137 del Registro 106

Figura 3.6: Proyección en las primeras 6 bases de Hermite

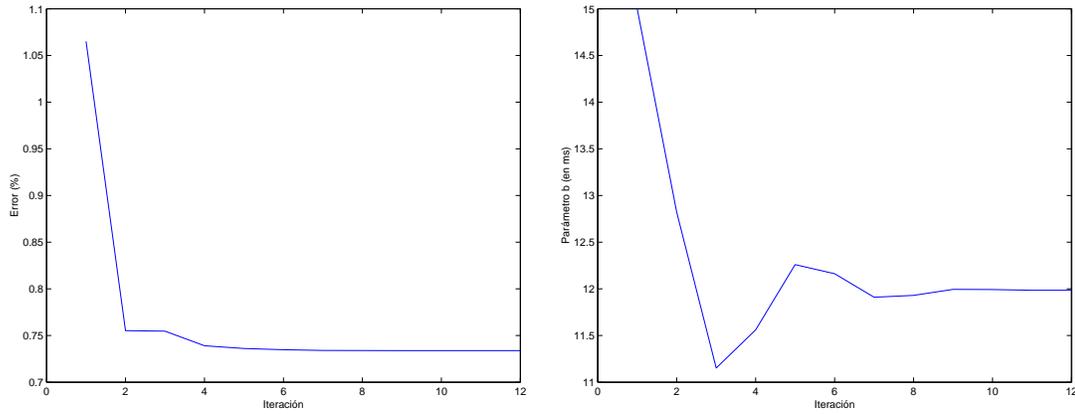
Pero, como

$$c_n(b) = \frac{1}{f_s} \mathbf{s} \cdot \phi_{\mathbf{n}} \quad (3.29)$$

$$c'_n(b) = \frac{1}{f_s} \mathbf{s} \cdot \frac{\partial \phi_{\mathbf{n}}}{\partial b} \quad (3.30)$$

(a) Evolución del error relativo en función de b

(b) Coeficientes



(c) Error vs Iteraciones

(d) Parámetro b vs Iteraciones**Figura 3.7: Latido 20 del Registro 200**

ambos términos de la última sumatoria de (3.28) son iguales y al sacar factor común.

$$\frac{de}{db} = -4 \left(\mathbf{s} - \sum_{n=0}^{N-1} c_n \phi_{\mathbf{n}}^{\mathbf{b}} \right)^T \sum_{n=0}^{N-1} c_n(b) \frac{\partial \phi_{\mathbf{n}}^{\mathbf{b}}}{\partial b} \quad (3.31)$$

En [33] se comete el error de suponer a c_n básicamente independiente de b y despreciar el término $\frac{dc_n}{db}$. Sin embargo, como se puede apreciar en (3.31) esto lleva a un error de escala lo que no afecta el descenso por gradiente.

Afortunadamente, se puede demostrar que la derivada parcial de las bases con respecto a b se puede calcular en términos de las bases mismas, haciendo el cálculo mucho más eficiente.

Para ello se utiliza el siguiente lema:

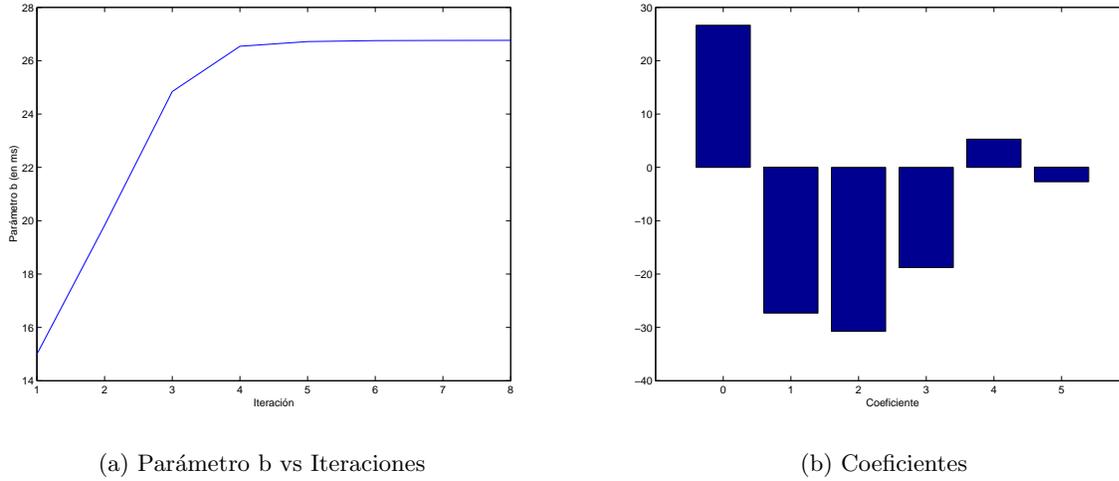


Figura 3.8: Latido 137 del Registro 106

Lema 1 Sea $H_n(x)$ un polinomio de Hermite. Entonces $H'_n(x) = 2nH_{n-1}(x)$.

La prueba del mismo se realiza utilizando inducción completa.

Se tiene entonces que encontrar la derivada parcial de las funciones base con respecto a b , $\frac{\partial \phi_n^b}{\partial b}$. Derivando la definición (3.18), sacando factor común y agrupando en los polinomios de Hermite se tiene,

$$\frac{\partial \phi_n^b(t)}{\partial b} = \frac{e^{-\frac{t^2}{2b^2}}}{2b\sqrt{b2^n n!}\sqrt{\pi}} \left[\left(2\frac{t^2}{b^2} - 1 \right) H_n(t/b) - 2\frac{t}{b} H'_n(t/b) \right] \quad (3.32)$$

Usando el Lema 1 (para $n \neq 0$),

$$\frac{\partial \phi_n^b(t)}{\partial b} = \frac{e^{-\frac{t^2}{2b^2}}}{2b\sqrt{b2^n n!}\sqrt{\pi}} \left[\left(2\frac{t^2}{b^2} - 1 \right) H_n(t/b) - 4n\frac{t}{b} H_{n-1}(t/b) \right] \quad (3.33)$$

Por último volviendo a usar (3.18),

$$\frac{\partial \phi_n^b(t)}{\partial b} = \frac{1}{2b} \left[\left(2\frac{t^2}{b^2} - 1 \right) \phi_n^b(t) - 2\sqrt{2n}\frac{t}{b} \phi_{n-1}^b(t) \right] \quad (3.34)$$

Queda aún por estudiar el caso $n = 0$. Por fortuna $H_0(x) = 1$ por lo que $H'_0(x) = 0$ y entonces,

$$\frac{\partial \phi_0^b(t)}{\partial b} = -\frac{1}{2b} \phi_0^b(t) \left(1 - 2\frac{t^2}{b^2} \right) \quad (3.35)$$

Es decir, para calcular la derivada parcial $\frac{\partial \phi_n^b}{\partial b}$ es necesario conocer solamente ϕ_n^b y ϕ_{n-1}^b que ya habían sido calculados previamente para realizar la proyección.

Usando (3.23) y el Lema 1, se puede probar que (3.34) se puede reescribir como,

$$\frac{\partial \phi_n^b(t)}{\partial b} = \frac{1}{2b} \left(-\sqrt{n(n-1)}\phi_{n-2}^b + \sqrt{(n+2)(n+1)}\phi_{n+2}^b \right) \quad (3.36)$$

lo que coincide con el resultado presentado en [33].

Aunque aparentemente más sencilla, la ecuación (3.36) presenta la necesidad de calcular la función base ϕ_{n+2} (e indirectamente también $H_{n+1}(x)$ para poder calcular $H_{n+2}(x)$), y es necesario calcular por separado los casos $\frac{\partial \phi_0^b}{\partial b}$ y $\frac{\partial \phi_1^b}{\partial b}$. Además, si se desea aumentar el orden de convergencia del algoritmo y utilizar C derivadas superiores, la evaluación de (3.36) precisa de las funciones base hasta $n + 2C$ mientras que (3.34) solo precisa las ya calculadas.

3.5. Resultados

En esta sección se presentará una comparación de las performances de las transformadas de Karhunen-Loève y Hermite. La comparación que aquí se realiza es en el sentido de cual de las dos representa mejor la señal, es decir, para las dos transformadas con la misma cantidad de vectores de la base, cual de la dos conserva la mayor cantidad de potencia de la señal. Por lo tanto el error en la representación se calcula como

$$E = \frac{\|s - \hat{s}\|^2}{\|s\|^2} \quad (3.37)$$

Cabe aclarar que estas dos transformadas no son directamente comparables, por varias razones:

- Por más que se compararon con el mismo número de vectores (para esta comparación se usaron 6), por la construcción de ambas bases, el número de coeficientes es distinto. Es decir, en el caso de Hermite se tienen 6 coeficientes más el parámetro b .
- La medida de error en ambos casos es distinta, en Karhunen-Loève la señal s es una señal de largo t_0 , en cambio en el caso de Hermite la señal tiene un largo $2t_0$, debido a la *extensión* de la misma que se mostró en la sección 3.4.2. De todas formas, la diferencia introducida por ese tramo de señal es despreciable.

En la tabla 3.5 se muestran los resultados obtenidos. Como se puede ver los resultados son levemente mejores para Hermite. A pesar de esta mejoría, y por las diferencias mencionadas anteriormente, se pueden hacer dos lecturas de esta tabla. Primero, que la diferencia de performance no es significativa, por lo cual se puede decir que desde el punto de vista de la representación de la señal ambas transformadas son igualmente apropiadas. Y segundo, que dado que el error es pequeño en ambos casos, se puede decir que ambas transformadas representan bien la señal con 6 coeficientes.

De todas formas, para los objetivos de este trabajo (que es de clasificación y no de compresión), no tiene sentido comparar las transformadas desde el punto de vista de la forma que represente la señal. La decisión final debe depender de cual de las dos maximiza la distancia (en el espacio de la transformada) entre latidos diferentes, y

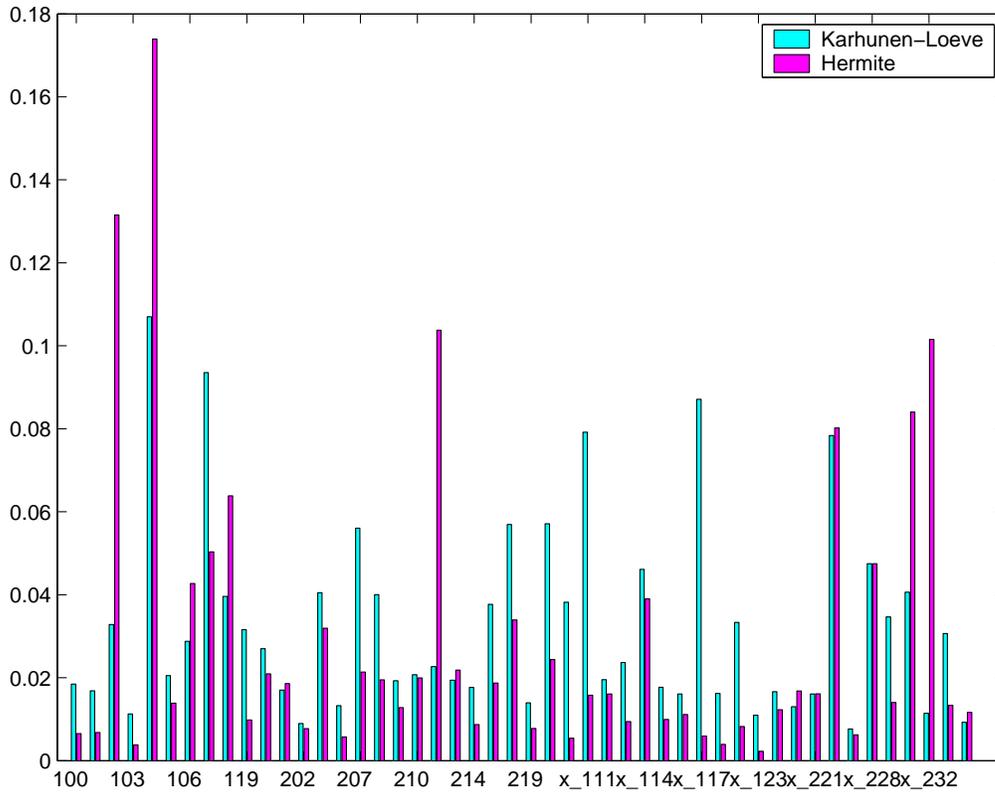


Figura 3.9: Comparacion de errores para Hermite y Karhunen-Loève.

minimiza la distancia entre latidos similares. Por lo tanto la decisión de cual transformada es mejor se postergará hasta el capítulo 4.

Existe otro aspecto a tener en cuenta en el momento de decidir por una transformada u otra: la eficiencia en el cálculo. Desde este punto de vista la, transformada K-L es mucho más eficiente que Hermite dado que lo único que se necesita para evaluar K-L es realizar productos escalares y la base se calcula fuera de línea, mientras que para Hermite es necesario calcular las bases para cada latido (adaptando el parámetro b) y evaluar las funciones recursivamente. La desventaja de K-L con respecto a Hermite es la incapacidad de adaptarse al paciente. Estas características son importantes en el momento de implementar los algoritmos en hardware para su funcionamiento en línea. Sin embargo en este trabajo se le dio más peso, a la hora de comparar los algoritmos, a la performance obtenida en la clasificación que a la velocidad de los mismos.

3.6. Generación de vectores de características

Además de la proyección en el espacio de cada transformada, se utilizó para la clasificación información temporal de la señal. En particular se consideraron las siguientes características:

1. **Intervalo RR instantáneo.** Calculado como,

$$RR_i = R_i - R_{i-1} \quad (3.38)$$

donde R_i representa la muestra en donde se estima se encuentra el pico R . Esta

Reg	núm. latidos	Error K -L	Error Hermite	Iteraciones
100	2271	1,9 %	0,7 %	15,9
101	1863	1,7 %	0,7 %	16,8
102	2185	3,3 %	13,2 %	18,8
103	2082	1,1 %	0,4 %	16,7
104	2227	10,7 %	17,4 %	17,3
105	2570	2,1 %	1,4 %	18,5
106	2025	2,9 %	4,3 %	17,4
107	2135	9,4 %	5,0 %	18,8
118	2276	4,0 %	6,4 %	11,2
119	1985	3,2 %	1,0 %	17,1
200	2599	2,7 %	2,1 %	15,5
201	1961	1,7 %	1,9 %	14,2
202	2134	0,9 %	0,8 %	14,6
203	2978	4,1 %	3,2 %	16,5
205	2654	1,3 %	0,6 %	17,7
207	2330	5,6 %	2,1 %	14,3
208	2953	4,0 %	2,0 %	16,5
209	3002	1,9 %	1,3 %	16,6
210	2648	2,1 %	2,0 %	17,9
212	2746	2,3 %	10,4 %	20,3
213	3249	1,9 %	2,2 %	18,0
214	2259	1,8 %	0,9 %	17,5
215	3361	3,8 %	1,9 %	15,0
217	2206	5,7 %	3,4 %	18,5
219	2152	1,4 %	0,8 %	17,0
x108	560	5,7 %	2,4 %	15,2
x109	855	3,8 %	0,5 %	16,9
x111	696	7,9 %	1,6 %	16,8
x112	852	2,0 %	1,6 %	15,8
x113	579	2,4 %	0,9 %	17,0
x114	554	4,6 %	3,9 %	18,4
x115	633	1,8 %	1,0 %	12,3
x116	795	1,6 %	1,1 %	14,8
x117	503	8,7 %	0,6 %	16,2
x121	607	1,6 %	0,4 %	14,3
x122	835	3,3 %	0,8 %	16,4
x123	504	1,1 %	0,2 %	19,1
x124	522	1,7 %	1,2 %	15,1
x220	697	1,3 %	1,7 %	19,7
x221	825	1,6 %	1,6 %	16,1
x222	737	7,8 %	8,0 %	14,5
x223	837	0,8 %	0,6 %	15,7
x228	695	4,8 %	4,8 %	14,9
x230	727	3,5 %	1,4 %	16,7
x231	503	4,1 %	8,4 %	16,8
x232	600	1,1 %	10,2 %	8,7
x233	1021	3,1 %	1,3 %	15,1
x234	919	0,9 %	1,2 %	11,9
Total	75988	3,3 %	3,0 %	16

Tabla 3.2: Comparación de Karhunen-Loève y Hermite

característica denota el ritmo cardíaco.

2. **Cociente entre el RR instantáneo y el medio.** Calculado como,

$$k_i = \frac{RR_i}{\overline{RR_i}} \quad (3.39)$$

donde $\overline{RR_i}$ se calcula como el promedio de los últimos ocho RR s. El k_i da una idea de cuan prematuro ($k_i < 1$) o retardado ($k_i > 1$) es el latido.

3. **Ancho del complejo QRS.** No hay una definición formal del ancho del QRS debido a la gran variabilidad de las morfologías. Se utilizó el ancho en el cual el latido cae a 75 % de su altura pico. Esta medida del ancho se usó para los vectores generados a partir de la transformada K-L, ya que en el caso de Hermite la información del ancho está dada por el parámetro b .

El vector generado de esta forma fue normalizado sobre toda la base del MIT-BIH restando la media y dividiendo entre la desviación estándar en cada dimensión. Dado que los coeficientes de las transformadas contienen diferente cantidad de información, los mismos se dividen por el mismo factor para preservar dicha relación. Como consecuencia de la normalización cada una de las características tiene el mismo peso relativo. Para darle mayor importancia a la morfología se le asignaron diferentes pesos. Los pesos utilizados en la implementación fueron: 0.25 al ancho, 0.5 al RR , 0.5 al k_i y 1 a los coeficientes de la transformada. La asignación de pesos menores a las características temporales responde a la idea de darle mayor importancia a la morfología, la cual se encuentra contenida en los coeficientes de la transformada.

Capítulo 4

CLASIFICACIÓN

*“Go to your bosom
Knock there and ask your heart
What it doth know”*

William Shakespeare

4.1. Introducción

Los algoritmos para clasificar ECG usualmente son desarrollados y evaluados sobre una extensa base de datos de señales provenientes de varios individuos que intenta cubrir un amplio espectro de posibles latidos. Los resultados existentes en la literatura son variados, pero todos enfrentan el mismo problema: la enorme variabilidad en la morfología de las señales electrocardiográficas, no solamente entre pacientes distintos, sino también debido a cambios en la actividad del propio individuo debido a sus movimientos, a la variación en la conductividad eléctrica del cuerpo, y otras variaciones. Esto tiene como consecuencia que, en su gran mayoría, estos algoritmos tienen un desempeño muy pobre cuando se los evalúa en una base de datos diferente a la usada para su entrenamiento, lo cual los hace inviables para el uso clínico. Un primer enfoque para solucionar este problema es construir una base de datos tan extensa como sea posible para entrenar el clasificador de ECG. Este enfoque presenta bastantes desventajas:

- No importa cuán amplia sea la base de datos, es imposible cubrir todas las posibles morfologías de latidos. Por lo tanto la performance siempre estará limitada por este hecho.
- La complejidad del clasificador crece a medida que el tamaño de la base de datos aumenta. Esto redundará en un clasificador complejo y costoso de entrenar y evaluar.
- Es prácticamente imposible lograr que el clasificador aprenda a corregir errores durante el uso normal. Lo cual hace que no sea útil para pacientes que presenten gran cantidad de latidos con una morfología que no estaba presente en el entrenamiento.

Una solución a este problema es desarrollar técnicas que permitan cierto nivel de adaptación al paciente. Es decir que una parte del clasificador pueda aprender ciertas características específicas del paciente. Para ello es necesario proporcionarle al clasificador un número suficiente de latidos del paciente previamente clasificados (manualmente por un especialista) para poder entrenarlo. El problema con los datos específicos es que el proceso de analizar y editar el ECG manualmente es muy costoso, por lo cual es importante que el tamaño de la base de datos específica del paciente no sea demasiado grande. Un posible alivio a este problema es el uso de un algoritmo de agrupamiento para ayudar al especialista a clasificar los latidos, esta idea se desarrollará más en la sección 4.4.

En torno a esta idea se han desarrollado algunos trabajos fundamentales como los de Hu *et al.* ([4]) y Watrous *et al.* ([5]). Watrous presenta un modelo basado en una red *feed-forward* en la cual se dejan un conjunto de neuronas sin entrenar con la base de datos genérica, para luego entrenarlas con los datos específicos del paciente. El problema con este enfoque es que al formar parte de la misma red, la actualización de sus pesos (debido al Back-Propagation) depende del resto de la red. Además, al formar parte de dicha red, el grado de adaptación al paciente se ve más limitado que si fuera una red independiente.

En CARDIDENT se utiliza el enfoque expuesto en [4]. El objetivo de este trabajo es lograr esta adaptación al paciente mediante la combinación de varios clasificadores.

Se trabajará con dos clasificadores (o expertos), uno general (llamado *Global Expert* o GE), entrenado de la manera usual: una amplia base de datos con registros de varios pacientes, y uno particular (*Local Expert* o LE) entrenado con registros específicos del paciente. En el caso de [4] los expertos se implementan mediante los algoritmos SOM y LVQ.

Estos dos expertos se combinan mediante el algoritmo llamado *Mezcla de expertos* (MOE), el cual consiste en una función de decisión que pesa la clasificación de cada uno de los expertos en base al vector de características presentado para lograr una decisión final. Este proceso es similar al proceso que ocurre cuando varios expertos en un tema llegan a un consenso.

Organización del capítulo

En la sección 4.2 se presentan diversos algoritmos de agrupamiento, sus problemas y soluciones y se comparan sus características. En la sección 4.3 se introducen los algoritmos de clasificación seleccionados para resolver el problema, y se comparan sus características. En las secciones 4.4, 4.5 y 4.6 se presentan algunos detalles de las características y la implementación de los algoritmos de clustering y de los expertos local y global respectivamente. Al final de cada sección se detallan los resultados obtenidos y se comparan con los resultados publicados en la literatura especializada. En la sección 4.7 se desarrolla en detalle la teoría y la implementación del algoritmo de mezcla de expertos y se presentan los resultados obtenidos. Finalmente, la sección 4.8 presenta las conclusiones de este capítulo y comenta posibles mejoras para el futuro.

4.2. Algoritmos de Clustering

El objetivo de esta sección es repasar los algoritmos de agrupamiento o *clustering* competitivos existentes en la literatura con el fin de determinar cual es el más apropiado para el problema de clasificación de ECG. Previo a la discusión de los algoritmos se definirá el problema de clustering y la notación a utilizar en el resto del capítulo.

El problema de clustering (o cuantizador vectorial VQ) se define como un mapeo $R^k \rightarrow Y \subset R^k$ donde $Y = y_1 \dots y_N$ son los llamados vectores representativos. En este trabajo también se le llamarán neuronas o *codevectors*. Asociado a cada una de las neuronas y_j existe una región del espacio S_j definida como

$$S_j = \{x \in X / d(x, y_j) \leq d(x, y_i) \forall i \neq j\} \quad (4.1)$$

donde

$$d(x, y_i) = \|x - y_i\|^r = \left(\sum_{j=1}^k |x_j - y_j|^2 \right)^{\frac{r}{2}} \quad (4.2)$$

tal que r denota la potencia de la norma Euclídea.

De la definición de S_j se desprende que

$$\bigcup_{j=1}^N S_j = R^k \quad (4.3)$$

$$S_j \cap S_i = \emptyset \quad \forall i \neq j \quad (4.4)$$

Por lo tanto el agrupador (o cuantizador) se puede definir como la función $Q(x)$ que cumple que

$$Q(x) = y_j \quad \text{si } x \in S_j \quad (4.5)$$

En la mayoría de los casos para el proceso de diseño de un agrupador se dispone de un conjunto $X = x_1 \dots x_T \subset T$ de T vectores de entrenamiento, el cual es utilizado repetidamente para encontrar los valores óptimos de los pesos de las neuronas y_j . La optimalidad se busca en el sentido de minimizar la *distorsión* definida como

$$D = \frac{1}{k} E\{d(x, Q(x))\} \quad (4.6)$$

que se puede escribir como

$$D = \frac{1}{k} \int_{R^k} d(x, Q(x)) p(x) dx \quad \text{en el caso continuo} \quad (4.7)$$

$$D = \frac{1}{k} \sum_i d(x_i, Q(x_i)) p_x(x_i) \quad \text{en el caso discreto} \quad (4.8)$$

4.2.1. Aprendizaje Competitivo

4.2.1.1. Regla Básica de aprendizaje

En los algoritmos competitivos (*competitive learning* CL) el vector de entrada $\mathbf{x} \in R^k$ es presentado a todas las unidades de procesamiento (neuronas), cada una de estas neuronas i tiene un vector de pesos $\mathbf{y}_i \in R^k$. Luego de presentado el vector de entrada se calcula la distancia, entre el vector de entrada \mathbf{x} y los pesos de la neurona \mathbf{y}_i , como en la ecuación 4.2.

Luego se elige una neurona *ganadora* c tal que tenga la mínima distancia.

$$d(x, y_c) \leq d(x, y_i) \quad \forall i \neq c \quad (4.9)$$

Esta regla es conocida como *vecino más cercano*. Para reducir la distancia media a la neurona ganadora, se calcula su nuevo vector de pesos $y_c(t+1)$ como

$$y_c(t+1) = y_c(t) + \eta(t)[x(t) - y_c(t)] \quad (4.10)$$

Los pesos de las neuronas perdedoras permanecen incambiables, por esta razón usualmente se le llama *winner-takes-all rule*. $\eta(t)$ es un parámetro no negativo que representa la velocidad de aprendizaje (o *learning rate*) para el ajuste de los pesos. Este parámetro se hace monótonamente decreciente a medida que el aprendizaje avanza, y controla la velocidad y la estabilidad del algoritmo, como se mostrará más adelante. En la práctica se dispone únicamente de un conjunto finito de vectores de entrenamiento $X = \{x_1, \dots, x_T\}$ con los cuales se actualizan los pesos de la neurona, mediante la presentación repetida de dicho conjunto.

Cada presentación del conjunto de entrenamiento se llama *epoch*, y habitualmente dicha presentación se hace eligiendo los vectores en forma aleatoria, la presentación de un sólo vector se llama iteración.

El algoritmo de entrenamiento es un descenso por el gradiente (estocástico) y se puede probar (teoría de aproximación estocástica, Monro 1951, ver [29]) que el algoritmo converge, en el sentido del error cuadrático medio, a un mínimo local con probabilidad uno. Esto ocurre siempre que inicialmente se encuentre en el dominio de atracción del algún punto de equilibrio (ver [34]).

Las condiciones para la convergencia son:

1. $\lim_{t \rightarrow \infty} \eta(t) = 0$
2. $\sum_{j=1}^{\infty} \eta(t) = \infty$
3. $\sum_{j=1}^{\infty} \eta^{\alpha}(t) < \infty$ con $\alpha > 1$

Debido a este algoritmo de entrenamiento (descenso por el gradiente) la performance final depende fuertemente de las condiciones iniciales.

Esta convergencia a mínimos locales generalmente lleva a que aparezcan neuronas que no ganan nunca, y por lo tanto la capacidad de la red es sub-utilizada. Como consecuencia de lo anterior la performance de este algoritmo no es la óptima.

A pesar de lo dicho anteriormente, en algunos casos, debido a la naturaleza estocástica del algoritmo, es posible que escape de los mínimos locales, pero esto último es algo que no se puede garantizar. En la sección 4.2.3 se mostrarán además algunas técnicas de escape como el *simulated annealing*.

Resumiendo, los principales problemas de este algoritmo son:

- La dependencia fuerte de las condiciones iniciales
- La convergencia a mínimos locales.
- La existencia de neuronas que nunca ganan (neuronas *muertas*), lo que produce una sub-utilización de la capacidad de la red.

A continuación se mostrarán algunas variantes de la regla básica de aprendizaje competitivo que atacan estos problemas, como los Mapas Auto-Organizativos (SOM) y el aprendizaje sensible a la frecuencia (FSCL).

4.2.1.2. Mapas Auto-Organizativos

El algoritmo SOM fue desarrollado originalmente por T. Kohonen en 1982 [35] como un modelo de la auto-organización de las conexiones de las neuronas de la corteza cerebral. Dicha auto-organización consiste en que estímulos similares se mapean en zonas adyacentes de la corteza. El algoritmo desarrollado por Kohonen es capaz de reproducir ese mecanismo y construir tales mapas, en este sentido se dice que el SOM preserva la topología del espacio de entrada.

La construcción del mapa se realiza presentando aleatoriamente los vectores de entrada, y actualizando los pesos de forma tal que los vectores de entrada cercanos se mapeen en neuronas cercanas. Entonces el algoritmo lleva a una representación organizada del espacio de entrada, partiendo de un desorden completo. Durante el proceso de aprendizaje se pueden observar dos etapas bien definidas: primero, una etapa de auto-organización (u ordenación) de las neuronas, y luego un ajuste (más fino) de los pesos para lograr una mejor cuantización del espacio de entrada (cuando el tamaño

del entorno se reduce a cero).

La regla de aprendizaje es similar a la de CL, un vector \mathbf{x} es presentado a la red y se ejecutan los siguientes pasos:

1. Se busca cual es la neurona *ganadora* (más cercana), de acuerdo a la regla (4.9).
2. Se actualizan todas las neuronas que se encuentran en un entorno de la ganadora, de modo que todas se muevan en dirección al vector de entrada. La actualización se realiza de acuerdo a la siguiente regla

$$y_n(t+1) = y_n(t) + \eta(t)h(n,c)[x(t) - y_n(t)] \quad (4.11)$$

donde $h(n,c)$ es la función que determina el entorno, que usualmente depende de la distancia $\|y_n - y_c\|$, y además es decreciente con el tiempo.

Esta noción de entorno (en contraste con la *winner takes all rule*) es la que permite que se formen los mapas que se mencionaron anteriormente.

También se puede ver que el CL es un caso particular del SOM cuando el entorno es nulo, es decir $h(n,c) = \delta_{n,c}$ (δ denota la función delta de Kronecker), por esta razón a veces se dice que las CL son mapas aleatorios (*random maps*) en contraste con la auto-organización del SOM.

Debido a la incorporación de entornos, esta técnica permite la actualización de todas las neuronas, eliminando por consiguiente, el problema de sub-utilización de neuronas.

Se puede probar ([34], [36]) que las condiciones para la convergencia del SOM son las mismas que las expuestas para CL (ver sección 4.2.1.1).

A pesar de que resuelve el problema de sub-utilización, este algoritmo presenta algunos problemas: primero, el uso de entornos es computacionalmente costoso; segundo, la elección de la topología correcta del entorno en dimensiones altas puede ser un problema complejo; tercero, al obligar a un grupo de neuronas a moverse en la misma dirección, se vuelve al algoritmo más rígido (dependiendo del tamaño del entorno) pudiendo a veces no llegar a la configuración óptima; y cuarto, sigue siendo un algoritmo de descenso por el gradiente, por lo cual sólo se puede asegurar convergencia a un mínimo local.

4.2.1.3. Aprendizaje Competitivo sensible a la frecuencia

El aprendizaje competitivo sensible a la frecuencia (FSCL) [34] es una variante que tiene como objetivo solucionar los problemas presentados por los algoritmos anteriormente presentados (CL y SOM). Para lograr lo anterior se usa una medida modificada de la distancia, que asegura que todas las neuronas se actualicen con la misma frecuencia a lo largo de las iteraciones. Esto se realiza asignando un contador $c_i(t)$ a cada neurona $y_i(t)$ que se incrementa cada vez que $y_i(t)$ gana. La ganadora se selecciona como aquella neurona que minimiza la nueva función de distancia. Esta última se calcula como el producto de la distancia usual y una función de *justicia* (o *fairness*).

$$F(c_i(t))\|y_i(t) - x(t)\| \quad (4.12)$$

La idea intuitiva de la función de *fairness* es que le de prioridad a las neuronas que ganan menos, frente a las que ganan más. Usualmente la función de *fairness* se calcula como

$$F(c) = c^{\beta(t)} \quad (4.13)$$

donde $\beta(t) > 0$ y $\lim_{t \rightarrow \infty} \beta(t) = 0$.

Entonces, al penalizar a las neuronas que ganan con más frecuencia se logra equiparar la chance de ganar de todas las neuronas. En una situación ideal, el espacio de entrada queda dividido en regiones equiprobables. Mediante este mecanismo se intenta lograr una mejor aproximación de la densidad de probabilidad subyacente. Por esta razón, usualmente se le llama principio de equiprobabilidad.

4.2.2. Aprendizaje Competitivo Selectivo

4.2.2.1. Introducción

En la sección 4.2.1.1 se mostró que las redes competitivas presentan problemas de sub-utilización (neuronas "muertas"), convergencia a mínimos locales y fuerte dependencia de las condiciones iniciales. En esta sección se mostrará un algoritmo basado en el principio de equidistorsión, que logra la performance óptima (mínimo global), evitando esos problemas. Se comenzará mostrando el principio de equidistorsión y luego se desarrollara el algoritmo CSL. Este algoritmo fue desarrollado por Ueda y Nakano (1994) en el trabajo fundamental [37].

4.2.2.2. Principio de Equidistorsión

El principio de equidistorsión se debe a Gerscho (1979), quien unificó y generalizó los resultados existentes sobre la optimalidad asintótica de los cuantizadores vectoriales. En su trabajo [38] se demuestra que, para una densidad de probabilidad "suave" $p(x)$, una cantidad grande de vectores representativos N y una medida Euclídea de la distorsión, el mínimo de la distorsión es

$$D^* = C(k, r) N^{-\beta} \left(\int p(x)^{\frac{1}{1+\beta}} dx \right)^{1+\beta} \quad (4.14)$$

donde r ó $\beta = \frac{r}{k}$ (normalizada por la dimensión) denotan la potencia de la norma Euclídea usada (ver ecuación (4.6)) y el término $C(k, r)$ es un coeficiente que depende solo de la dimensión k y de la potencia r . En general, para un k cualquiera, sólo se conoce una cota inferior del coeficiente, de todas formas este coeficiente no es esencial para el siguiente desarrollo, por lo cual se omitirá.

El mínimo de la distorsión (4.14) se alcanza sólo cuando la densidad de probabilidad de los vectores representativos es proporcional a $p(x)^{\frac{1}{1+\beta}}$ ¹². Gerscho (ver [38]) muestra que en ese caso todas las sub-distorsiones, D_i con $i = 1, 2, \dots, N$, en una partición óptima son iguales.

En el desarrollo anterior se asumió que la densidad de probabilidad subyacente era continua, lo cual no es una suposición muy realista, dado que en los casos prácticos

¹²La densidad de los vectores representativos es discreta, pero como se trata de un resultado asintótico, cuando N es grande se puede aproximar por una densidad continua.

en general se tiene un conjunto de clusters disjuntos de vectores de entrada. A continuación se extenderá el resultado de Gerscho para este caso.

Considere el caso que la distribución es un cluster (Ω) compuesto por M clusters disjuntos ($\Omega_1, \dots, \Omega_M$). Formalmente, el cluster Ω satisface:

1. $\Omega_i \subset R^k$
2. $i \neq j \Rightarrow \Omega_i \cap \Omega_j = \phi$
3. $\bigcup_{i=1}^M \Omega_i = \Omega \subset R^k$

Sea $p_j(x)$ una densidad de probabilidad continua para Ω_j . Entonces la condición (2) implica que $p_j(x) = 0$ fuera de Ω_j . Por esto, podemos decir que el caso anterior ($p(x)$ continua) corresponde al caso $M = 1$.

Sea $n_j = \frac{N_j}{N}$ la proporción de vectores representativos en el cluster j , obviamente $\sum_{j=1}^M n_j = 1$. Sea ahora $D^{(j)}$ la distorsión en el cluster j . Entonces, según la ecuación (4.14), la mínima distorsión para el cluster j , escrita en función de n_j , es

$$D^{(j)*}(n_j) = C(k, r)(n_j N)^{-\beta} \left(\int_{\Omega_j} p(x)^{\frac{1}{1+\beta}} dx \right)^{1+\beta} \quad \text{para } j = 1, \dots, M \quad (4.15)$$

Como k y N están fijos, resolviendo el problema de minimización planteado en la ecuación (4.16), se puede obtener la proporción óptima (n_j^*) de vectores representativos en el cluster j , en el sentido de minimizar la distorsión total sobre los clusters. Obviando la constante $C(k, r)$, se minimiza:

$$f(n_1, \dots, n_M) = \sum_{j=1}^M n_j^{-\beta} \left(\int_{\Omega_j} p(x)^{\frac{1}{1+\beta}} dx \right)^{1+\beta} \quad \text{sujeto a } \sum_{j=1}^M n_j = 1 \quad (4.16)$$

Este problema de minimización puede resolverse fácilmente usando el método de los multiplicadores de Lagrange. Entonces se debe buscar el mínimo de

$$F(n_1, \dots, n_M, \lambda) = \sum_{j=1}^M n_j^{-\beta} \left(\int_{\Omega_j} p(x)^{\frac{1}{1+\beta}} dx \right)^{1+\beta} + \lambda \left(\sum_{j=1}^M n_j - 1 \right) \quad (4.17)$$

Las condiciones para el extremo son:

$$\frac{\partial F}{\partial n_j} = 0 \quad \text{y} \quad \frac{\partial F}{\partial \lambda} = 0 \quad (4.18)$$

De la ecuación (4.18) se obtiene la proporción óptima n_j^* como

$$n_j^* = \frac{\int_{\Omega_j} p(x)^{\frac{1}{1+\beta}} dx}{\sum_{i=1}^M \int_{\Omega_i} p(x)^{\frac{1}{1+\beta}} dx} \quad (4.19)$$

y evaluando la matriz Hessiana en este punto, se puede ver que es un mínimo. Si se sustituye el valor de n_j^* hallado en (4.19) se obtiene que la mínima distorsión para el cluster j es

$$D^{(j)*}(n_j^*) = C(k, r)N^{-\beta} \left(\int_{\Omega_j} p(x)^{\frac{1}{1+\beta}} dx \right) \left(\sum_{i=1}^M \int_{\Omega_i} p(x)^{\frac{1}{1+\beta}} dx \right)^{\beta} \quad (4.20)$$

Por otro lado, recordando que $p_j(x) = 0$ fuera de Ω_j , se puede probar que se cumple la siguiente ecuación

$$D^* = \sum_{j=1}^M D^{(j)*}(n_j^*) \quad (4.21)$$

Como conclusión del desarrollo anterior se puede enunciar el siguiente lema

Lema 2 *Si la distribución de los vectores de entrada está constituida por M clusters disjuntos y la densidad $p_j(x)$ en cada cluster es "suave", la proporción óptima de vectores representativos en cada cluster que minimiza la distorsión total es la dada por la ecuación (4.19).*

Si existe la suficiente cantidad de vectores representativos por cluster, es decir, hay n_j^* vectores en el cluster j , y todas las regiones en el cluster j tienen la misma distorsión, entonces la distorsión total sobre todos los cluster se minimiza. Además se cumple que

$$\frac{D^{(j)*}}{n_j^*} = C(k, r)N^{-(1+\beta)} \left(\sum_{j=1}^M \int_{\Omega_j} p(x)^{\frac{1}{1+\beta}} dx \right)^{\beta} \quad (4.22)$$

y como se puede ver en la ecuación anterior, $\frac{D^{(j)*}}{n_j^*}$ es independiente de j , por lo tanto cuando la distorsión se hace mínima, no sólo todas las regiones tienen igual distorsión, sino también todos los clusters tienen la misma distorsión. Por lo cual se puede ver que el resultado asintótico de Gerscho se sigue cumpliendo para una distribución formada por clusters disjuntos. Este resultado se puede resumir en el siguiente teorema.

Teorema 1 *Para un número grande de vectores representativos N , aunque la distribución de probabilidad subyacente esté formada por clusters disjuntos, la distorsión es mínima cuando todas las regiones presentan la misma sub-distorsión.*

4.2.2.3. Algoritmo CSL

Mecanismo de Selección

De acuerdo al principio de equidistorsión, en los clusters con más sub-distorsión debe haber más neuronas y en aquellos con menor sub-distorsión debe haber menos. Para lograr esto se agrega un mecanismo de selección al algoritmo de aprendizaje competitivo tradicional.

Este mecanismo consiste en agregar neuronas en las zonas donde la distorsión es mayor, y eliminarlas donde la distorsión es menor. El número de neuronas a agregar(eliminar) se calcula mediante una función de *fitness* (que se explicará más adelante) calculada a partir de la sub-distorsión. De esta manera se puede ajustar el

número de neuronas en las regiones de acuerdo a sus sub-distorsiones sin cambiar el número total de neuronas. A medida que el aprendizaje avanza las sub-distorsiones se van igualando hasta llegar a la óptima.

Cuando se dispone de un conjunto finito de vectores de entrenamiento, éste se utiliza repetidamente para ajustar las neuronas hasta que el algoritmo converge. Como se dispone de un conjunto finito, el valor esperado de la distorsión se aproxima por la media de la misma sobre el conjunto de entrenamiento.

Además del ajuste usual, debido al aprendizaje competitivo, se realiza la selección cada T intervalos de entrenamiento. Para asegurar la convergencia de la selección, es necesario que el número de neuronas a seleccionar decrezca de forma monótona luego de cada paso de selección. Por lo tanto, la selección siempre se realiza en un número finito de pasos.

Resumen del algoritmo CSL

1. Inicialización
 - N : número de neuronas.
 - X : conjunto de entrenamiento.
 - Y^0 : conjunto de neuronas iniciales.
 - ϵ : criterio de parada (sobre la distorsión).
 - m : contador de *epochs* ($m = 1$).
 - $D(0) = \infty$: distorsión inicial.
 - $s(m)$: cantidad de neuronas a seleccionar en la iteración m .
2. Entrenamiento: Para todos los $x \in X$, ejecutar los pasos 2a y 2b que se explican a continuación, al igual que en el aprendizaje competitivo usual (sección 4.2.1).
 - a) Encontrar la neurona más cercana al vector x usando la ecuación (4.9)
 - b) Ajustar el peso de la neurona ganadora. usando la ecuación (4.10).
3. Selección: Si $s(m) > 2$, elegir $s(m)$ neuronas y ejecutar el algoritmo de selección para las neuronas elegidas. Incrementar el contador m y volver al paso 2. En caso contrario, avanzar al paso 4.
4. Test de convergencia:
 - a) Calcular la distorsión media $D(m)$ usando los pesos de las neuronas obtenidas en el paso 2.
 - b) Si $\frac{D(m)-D(m-1)}{D(m)} \leq \epsilon$ termina el algoritmo dando como resultado final las neuronas $Y^{(m)}$. En caso contrario, incrementar el contador m y volver al paso 2.

La distorsión media en el paso 4a en la iteración m se calcula como

$$D(m) = \frac{1}{|X|} \sum_{x \in X} \min_{y \in Y^{(m)}} d(x, y) \quad (4.23)$$

donde $Y^{(m)}$ es el conjunto de neuronas luego del paso 2 en la iteración m .

En el paso 3 se eligen las $s(m)$ neuronas que se encuentran en los extremos de la distorsión (las que tienen más y menos). De esta forma, es claro ver que eligiendo las neuronas de esta manera se disminuye más rápido la distorsión en cada una de las regiones.

Como se puede ver el algoritmo CSL es básicamente un aprendizaje competitivo básico, excepto que se realiza la selección cada cierto intervalo de tiempo. Como consecuencia de esto, durante el proceso de aprendizaje, no solamente la distorsión total se reduce debido a la actualización de las neuronas realizada por el paso competitivo (paso 2), sino que también se reduce la varianza de las sub-distorsiones en las distintas regiones debido al mecanismo de selección (3).

A continuación se presentará en detalle el algoritmo de selección.

Algoritmo de Selección

La selección puede ser interpretada como la multiplicación y desaparición de neuronas en cada región del espacio de entrada. Como es una función discontinua la selección puede ejecutarse correctamente incluso cuando un cluster de vectores de entrada pertenezca a varias regiones. Claramente con un sólo paso de selección no se logra la equidistorsión, pero luego de ejecutarla varias veces ésta es aproximadamente alcanzada.

Se tienen $s(m)$ neuronas a seleccionar, que han de ser repartidas de forma tal que en las regiones con mayor distorsión haya más neuronas. Para elegir cómo se distribuirán las neuronas, se calcula una medida no-lineal de la distorsión (función de *fitness* g_j). Esta función representa la fracción de las $s(m)$ neuronas que debería haber en la región j , es decir que $\sum_{j=1}^{s(m)} g_j = 1$.

Como el número de neuronas en cada región debe ser un número entero, la cantidad de neuronas u_j que habrá en la región j se calculará como la parte entera del producto $g_j s(m)$. Pero entonces $\sum_{j=1}^{s(m)} u_j \leq s(m)$, por lo tanto pueden sobrar $s'(m) = s(m) - \sum_{j=1}^{s(m)} u_j$ neuronas.

A cada región se le deberían asignar $g_j s(m)$ neuronas, pero en realidad se le asignan u_j , por lo tanto *queda* por repartir la fracción $u'_j = g_j s(m) - u_j$ de neurona. Entonces se le asigna una neurona más a cada una de las $s'(m)$ regiones cuyo u'_j es más cercano a uno.

Resumiendo, el algoritmo de selección es:

1. Calcular la función de *fitness* a partir de la subdistorsión D_j como

$$g_j = \frac{D_j^\gamma}{\sum_{i=1}^{s(m)} D_i^\gamma}$$

donde γ es una constante no negativa y menor que uno.

2. Calcular el número de neuronas a reproducir en cada región:

a) Calcular $u_j \equiv [g_j s(m)]$ ($[a]$ denota la parte entera del número a)

b) Ordenar las neuronas según el valor de $u'_j = g_j s(m) - u_j$.

Para las $s'(m) = s(m) - \sum_{j=1}^{s(m)} u_j$ neuronas que tengan el valor de u'_j más cercano a uno, sumar uno al u_j correspondiente.

3. Para cada j , la neurona y_j se reproducirá según u_j , de forma que:

- Si $u_j = 0$ la neurona desaparece.
- Si $u_j = 1$ la neurona permanece incambiada.
- Si $u_j > 1$ se agregan $u_j - 1$ neuronas en la región sumando a y_j una perturbación aleatoria δ_j tal que $\|\delta_j\| \ll \|y_j\|$.

Observaciones

La subdistorsión D_j en el paso 1 se calcula como

$$D_j = \frac{1}{T} \sum_{x \in S_j} d(x, y_j)$$

debe notarse que en el cálculo de D_j se usa la cantidad total de vectores de entrenamiento T y no la del cluster t_j . En [37] se muestra una justificación de esto.

En el paso 1 si $\gamma = 1$ la función g_j es lineal, y si además la varianza de las subdistorsiones es muy grande la selección será excesiva. Para evitar esto es conveniente tomar valores más pequeños por ejemplo $\gamma = 0,5$. En la práctica se puede observar que la selección es poco sensible al parámetro γ en un entorno de este valor.

4.2.3. Comparación

En esta sección se compararán los diferentes algoritmos presentados en las secciones anteriores. En la sección 4.2.1 se mostraron los problemas que presenta el aprendizaje competitivo (CL) y en las secciones siguientes se mostraron otros algoritmos (SOM y FSCL) que atacan algunos de ellos.

El algoritmo de Kohonen, a pesar de resolver el problema de la sub-utilización, sigue padeciendo el problema de los mínimos locales, y en general presenta performances más bajas que el algoritmo CL (sobre todo en un espacio de dimensión alta, como es el caso de este trabajo).

El algoritmo FSCL, en cambio, presenta una mejora sustancial en la mayoría de los casos. Esta mejora se debe fundamentalmente a la utilización del principio de

equiprobabilidad, porque se logra una mejor aproximación de la densidad de probabilidad del espacio de entrada, y prácticamente se elimina el problema de las neuronas *muertas*. Además con el uso de este mecanismo se logra una mayor robustez frente a variaciones en las condiciones iniciales. A pesar de la mejora que introduce este algoritmo, la performance no es la óptima porque, como se mostró en la sección 4.2.2.2, para lograr la performance óptima es necesaria la equidistorsión, no la equiprobabilidad. Solamente en el caso de que la probabilidad de que los vectores de entrada pertenezcan a una región sea uniforme, ambos principios coinciden, y por lo tanto el algoritmo alcanza el desempeño óptimo.

El algoritmo CSL, al utilizar el mecanismo de selección para lograr la equidistorsión, logra una performance cercana a la óptima. Una característica fundamental del mecanismo de selección es que permite al sistema escapar de los mínimos locales, a diferencia del aprendizaje competitivo tradicional que queda atrapado en los mismos. Existen técnicas, para solucionar este problema del algoritmo CL, como el *simulated annealing* (ver [29] y [39]), que permiten el escape de los mínimos locales, sin embargo, este escape se realiza de forma *ciega*, es decir, se perturba al sistema en forma aleatoria, sin información de hacia dónde debe moverse. En el mecanismo de selección, en cambio, el escape se realiza de forma determinística, y actualizando las neuronas con el objetivo de disminuir la distorsión. Dado que el mecanismo de selección logra evitar mínimos locales y obtener casi el mínimo global, es claro ver que logra que el algoritmo sea muy robusto frente a las condiciones iniciales.

4.3. Algoritmos de Clasificación

4.3.1. Learning Vector Quantization

El algoritmo LVQ es un algoritmo supervisado, basado en el aprendizaje competitivo, con la diferencia que se dispone de información sobre las clases de los vectores de entrenamiento. Esta información se puede incorporar en el aprendizaje de la siguiente manera: si el vector presentado es de la misma clase que la neurona, esta última se *mueve* hacia el vector presentado (según el descenso por el gradiente), y en caso contrario, se mueve en dirección opuesta.

Resumiendo la regla de aprendizaje del LVQ es

- $y_c(t + 1) = y_c(t) + \eta(t)[x(t) - y_c(t)]$ si x e y_c pertenecen a la misma clase
- $y_c(t + 1) = y_c(t) - \eta(t)[x(t) - y_c(t)]$ si x e y_c pertenecen a distintas clases

donde y_c denota la neurona ganadora según 4.9.

Al estar basado en el algoritmo CL, el LVQ tiene las mismas desventajas, por lo tanto para el problema supervisado se busca transformar el algoritmo CSL en supervisado, como se muestra en la siguiente sección.

4.3.2. Optimized Learning Vector Quantization

Este algoritmo es una modificación del LVQ original con el objetivo de asignar un *learning rate* $\eta_i(t)$ diferente a cada neurona y_i . Con esta idea se obtiene la siguiente regla de aprendizaje

- $y_c(t + 1) = y_c(t) + \eta_c(t)[x(t) - y_c(t)]$ si x e y_c pertenecen a la misma clase

- $y_c(t+1) = y_c(t) - \eta_c(t)[x(t) - y_c(t)]$ si x e y_c pertenecen a distintas clases

donde y_c denota la neurona ganadora según 4.9. Esta ecuación se puede reescribir de la forma

$$y_c(t+1) = y_c(t) + s(t)\eta_c(t)[x(t) - y_c(t)] \quad (4.24)$$

o también

$$y_c(t+1) = [1 - s(t)\eta_c(t)]y_c(t) + s(t)\eta_c(t)x(t) \quad (4.25)$$

donde $s(t)$ es una función que vale +1 si la clasificación fue correcta, y -1 si fue incorrecta.

$$y_c(t+1) = \alpha(t)y_c(t) + \beta(t)x(t) \quad (4.26)$$

donde se definió $\alpha(t) = 1 - s(t)\eta_c(t)$ y $\beta(t) = s(t)\eta_c(t)$.

Se puede ver ([40]) que la red obtenida luego del entrenamiento es óptima, si los efectos de las actualizaciones hechas en diferentes instantes de tiempo, referidas al final del entrenamiento, tienen el mismo peso. Hay que notar que $y_c(t_f)$ tiene una contribución de cada uno de los $x(t) \forall t \leq t_f$ presentados, ponderados por el factor $\alpha(t)$ que se muestra en la ecuación (4.25). Por ejemplo, si calculamos las actualizaciones correspondientes a $t = 0 \dots 2$ podemos ver que

$$y_c(1) = \alpha(0)y_c(0) + \beta(0)x(0) \quad (4.27)$$

$$y_c(2) = \alpha(1)\alpha(0)y_c(0) + \alpha(1)\beta(0)x(0) + \beta(1)x(1) \quad (4.28)$$

La idea es que los pesos de cada uno de los $x(t)$ sean los mismos, en este ejemplo la condición sería que $\alpha(1)\beta(0) = \beta(1)$. En el caso general, basta con plantear que los pesos de $x(t)$ y $x(t-1)$ de una iteración a la siguiente sean iguales, esto garantiza que al final del entrenamiento los pesos de todos los $x(t)$ serán iguales. En este caso la ecuaciones de actualización quedan

$$y_c(t) = \alpha(t-1)y_c(t-1) + \beta(t-1)x(t-1) \quad (4.29)$$

$$y_c(t+1) = \alpha(t)\alpha(t-1)y_c(t-1) + \alpha(t)\beta(t-1)x(t-1) + \beta(t)x(t) \quad (4.30)$$

y por lo tanto la condición es $|\alpha(t)\beta(t-1)| = |\beta(t)|$ y finalmente, recordando las definiciones de $\alpha(t)$ y $\beta(t)$ en la ecuación (4.26), se llega a que

$$[1 - s(t)\eta_c(t)]\eta_c(t-1) = \eta_c(t)$$

y despejando $\eta_c(t)$ queda

$$\eta_c(t) = \frac{\eta_c(t-1)}{1 + s(t)\eta_c(t)} \quad (4.31)$$

De esta forma el algoritmo se acelera mucho, pero hay que tener en cuenta que $\eta_c(t)$ eventualmente puede crecer, si una neurona pierde mucho ($s(t) < 0$). Por lo tanto hay que tener la precaución de que este parámetro no crezca mucho, por ejemplo imponiendo que no crezca más allá de su valor inicial.

4.3.3. Algoritmo CSL supervisado

Al igual que el algoritmo CL no supervisado, el LVQ presenta el problema de encontrar mínimos locales de los cuales no puede escapar. Para evitar este problema se extendió el algoritmo CSL al caso supervisado (SCSL).

A diferencia del LVQ, se optó por elegir la neurona ganadora como la más cercana *de la misma clase* al vector de entrenamiento, es decir que los vectores de entrenamiento de una clase son vistos solamente por las neuronas de dicha clase y son *invisibles* para las demás. De esta forma existe una diferencia del algoritmo SCSL con el no supervisado es la manera en que se calcula la sub-distorsión. Como cada neurona sólo es sensible a los vectores de su misma clase, la sub-distorsión de j -ésima neurona se calcula como

$$D_j = \frac{1}{T} \sum_{x \in S_j} d(x, y_j) \quad (4.32)$$

en donde S_j es el conjunto de vectores a los cuales responde la neurona. Por lo tanto se buscó minimizar la distorsión por clase global definida como

$$D = \sum_{j=1}^N D_j \quad (4.33)$$

El algoritmo de selección es el mismo que el del CSL con la excepción que, al ser ahora supervisado, las neuronas reproducidas a partir de una se las hace del mismo tipo. De esta forma las clases con mayor distorsión tienen mayor número de neuronas asociadas. Por otra parte, es ahora necesario revisar que la selección no elimine todas las neuronas de una clase de vectores. La regla de actualización de la neurona ganadora es la misma que para el algoritmo CL (4.10).

A medida que el algoritmo converge, las posiciones de las neuronas se aproximan a los centros de las regiones de cada clase. Sin embargo, es posible que la frontera de decisión por *vecino más cercano* (4.9) no sea la óptima. Para minimizar este problema el algoritmo oLVQ (sección 4.3.2) es ejecutado al final. Como esta etapa final no tiene como objeto minimizar la distorsión, se usa como criterio de parada la variación en norma de los vectores de la red, entre dos *epochs* consecutivos.

4.4. Agrupador

4.4.1. Introducción

El análisis de registros de ECG de larga duración no sólo es tedioso, sino que además depende del operador. Un algoritmo de agrupamiento no supervisado puede acelerar la tarea, ya que condensa los latidos en un número pequeño (con respecto a la cantidad de latidos) de clases. Esto permite un ahorro importante en el tiempo dedicado por el cardiólogo, ya que puede concentrarse en los casos complicados. Para que esto sea posible, es necesario que la performance del algoritmo sea muy alta para que el resultado sea confiable.

El análisis y los resultados del agrupamiento presentados en esta sección están basados en vectores de características generados a partir de la transformada K-L, que fue la transformada elegida para implementar el clasificador. Los resultados obtenidos utilizando Hermite y la justificación de la elección realizada se muestran la sección

4.4.5.

4.4.2. Implementación

El agrupador implementado utiliza el algoritmo CSL, realizando el entrenamiento con registros enteros, sin utilizar información sobre la clase de los latidos. Se trabajó sobre el espacio 12-dimensional formado por los vectores de características explicados en la sección 3.6, que se mapea en las neuronas de la red del agrupador. Se eligieron 25 neuronas para construir la red de forma tal de poder comparar los resultados con [1].

4.4.3. Entrenamiento

Usualmente los datos de interés constituyen un subconjunto *pequeño* del espacio de entrada, por lo tanto para la inicialización se buscó que los pesos de las neuronas se ubicaran cerca del centro de los datos de entrada, en este caso, como éstos fueron normalizados (sección 3.6), este centro estará cerca del origen. Por lo tanto se inicializaron los pesos de todas las neuronas en el origen. Esto no representa un problema dado que el algoritmo CSL es robusto frente a variaciones de las condiciones iniciales. Durante el entrenamiento los vectores de entrada son presentados en forma aleatoria, para facilitar el escape de mínimos locales (ver sección 4.2.1.1), actualizando los pesos con cada vector presentado. El *learning rate* se inicializa en 0.1 y se decrementa un 6% por cada *epoch*.

Para el algoritmo de selección se tomó un valor inicial de la función de selección $s(m)$ igual al número de neuronas menos uno. $s(m)$ se eligió de manera que se decremente una neurona por *epoch*. Si durante cinco *epoch* seguidas ninguna neurona fue modificada por el algoritmo de selección, éste finaliza.

Una vez finalizada la selección el algoritmo termina cuando el cambio relativo en la distorsión es menor que 10^{-5} o cuando se llega al número máximo de quince *epochs*.

4.4.4. Resultados

Para medir la performance de un algoritmo NO supervisado, es necesario conocer la clase a la que pertenece cada latido agrupado. Con esta información disponible es posible asociar a cada neurona una clase de latido. Esta clase se determina de acuerdo al tipo de latido presente en mayor proporción dentro de la región correspondiente. En este trabajo se discriminaron los latidos en los 20 tipos disponibles¹³ en la base MIT-BIH ([9]). Los códigos de los tipos de latidos pueden encontrarse en la tabla 1.1.

La performance del algoritmo se medirá en función de la *sensibilidad* y el *valor predictivo positivo*. Para definir estas magnitudes se considerarán la siguientes categorías de latidos (por clase)

TP_{*i*}: Verdaderos Positivos (True Positive) son los latidos de clase *i* asignados a la clase *i*.

FP_{*i*}: Falsos Positivos (False Positive) son los latidos de otra clase asignados a la clase *i*.

¹³Cabe aclarar que si bien existen 20 tipos de latidos definidos en la base, sólo aparecen en los registros 16 de dichos tipos

Clase	!	/	A	E	F	J	L	N	
Nro.Lats	472	7020	1273	106	786	31	4994	50407	
S_i	95,55 %	99,66 %	90,97 %	96,23 %	78,50 %	48,39 %	99,92 %	99,79 %	
$+P_i$	92,42 %	99,49 %	94,69 %	94,44 %	87,64 %	78,95 %	99,84 %	99,31 %	
Clase	Q	R	S	V	a	e	f	j	Total
Nro.Lats	33	5072	2	5561	147	5	982	16	76907
S_i	9,09 %	99,78 %	0,00 %	96,94 %	79,59 %	0,00 %	95,62 %	12,50 %	99,00 %
$+P_i$	100,00 %	99,53 %	x	98,14 %	99,15 %	x	95,43 %	100,00 %	98,98 %

Tabla 4.1: Performance por clase del agrupador de CARDIDENT sobre la base de datos.

FN_i : Falsos Negativos (False Negative) son los latidos de clase i asignados a otra clase.

Entonces se define,

Valor Predictivo Positivo ($+P_i$): Relación de los latidos de clase i correctamente clasificados sobre el total de latidos asignados a la clase i por el algoritmo. Representa que tan seguro se puede estar de que el latido pertenece al grupo asignado por el agrupador, y no a otro.

$$+P_i = \frac{TP_i}{TP_i + FP_i} \quad (4.34)$$

Sensibilidad (S_i): Relación de los latidos de la clase i correctamente clasificados sobre el total de latidos efectivamente de clase i . Representa que tan seguro se puede estar de que los latidos de esa clase sean asignados correctamente por el algoritmo.

$$S_i = \frac{TP_i}{TP_i + FN_i} \quad (4.35)$$

De la misma forma se definen las performances globales¹⁴

Valor Predictivo Positivo (+P): Es el promedio de las predictividades positivas por clase, pesadas por el número de latidos de cada clase

Sensibilidad (S): Es el promedio de las sensibilidades por clase, pesadas por el número de latidos de cada clase.

En la tabla 4.1 se pueden apreciar las performances por clase obtenidas por el algoritmo sobre toda la base MIT-BIH. La sensibilidad global fue 99.00 % y el valor predictivo positivo global 98.98 %. En la tabla se puede observar que los tipos más comunes de latidos son los que presentan las mayores sensibilidades. Esto se debe a la naturaleza del algoritmo CSL, que asigna un mayor número de neuronas a las regiones con mayor densidad. Por otro lado, al haber tantas neuronas asignadas a estas clases, algunos latidos de clases minoritarias caen en las regiones correspondientes a

¹⁴ Una vez que se tienen las performances por clase por registro existen dos formas de calcular las globales: la primera es calcular las performances totales por registro y luego promediar sobre todos los registros, y la segunda es hacerlo a la inversa, primero sumando sobre todos los registros y luego promediando sobre las clases. En este trabajo se usó la segunda.

Tipo	!	/	A	E	F	J	L	N	Q	R	S	V	a	e	f	j	$\bar{\Sigma}$
!	451	0	0	0	0	0	0	0	0	1	0	36	0	0	0	0	37
/	0	6996	0	0	0	0	0	0	0	0	0	5	0	0	31	0	36
A	2	0	1158	0	0	1	1	52	0	1	0	1	7	0	0	0	65
E	0	0	0	102	0	0	2	0	0	3	0	1	0	0	0	0	6
F	0	0	1	0	617	0	0	26	1	0	0	55	4	0	0	0	87
J	0	0	1	0	0	15	0	0	0	3	0	0	0	0	0	0	4
L	1	0	0	1	0	0	4990	0	2	0	0	4	0	0	0	0	8
N	0	0	104	1	124	1	0	50299	9	1	2	65	15	5	10	14	351
Q	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
R	0	0	6	2	1	14	0	1	0	5061	0	0	0	0	0	0	24
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V	18	0	2	0	44	0	1	27	2	2	0	5391	4	0	2	0	102
a	0	0	1	0	0	0	0	0	0	0	0	0	117	0	0	0	1
e	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f	0	24	0	0	0	0	0	2	16	0	0	3	0	0	939	0	45
j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
$\bar{\Sigma}$	21	24	115	4	169	16	4	108	30	11	2	170	30	5	43	14	766

Tabla 4.2: Agrupamiento de los latidos sobre la base de datos, por clase.

dichas neuronas. Por esta razón generalmente la predictividad positiva de clases más comunes es ligeramente menor que su sensibilidad.

A la inversa, las clases minoritarias, presentan una baja sensibilidad y un alto valor predictivo positivo, en términos relativos.

Por ejemplo, la mayoría de los latidos son normales (65.6%) y consecuentemente la sensibilidad en el agrupamiento es alto (99.8%), y el valor predictivo positivo es de 99.3%. Los latidos *J* (latidos prematuros nodales) representan el 0.04% de toda la base, y presentan una sensibilidad de 48.4% y un valor predictivo de 79.0%.

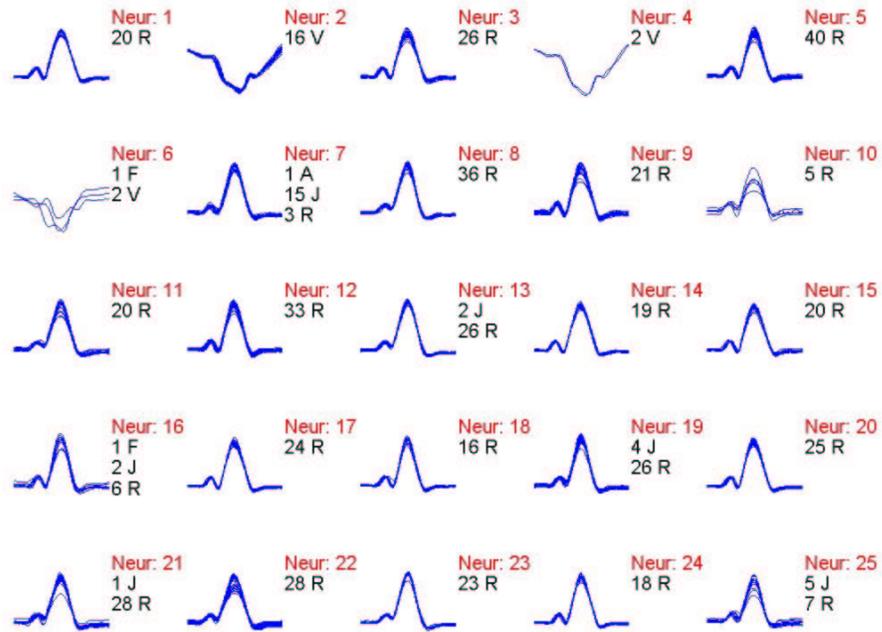
En la tabla 4.3 se muestra con mayor detalle el agrupamiento de los latidos sobre toda la base de datos. El eje horizontal representa la clase verdadera del vector y el vertical la clase asignada por el agrupador. Por ejemplo, hay 18 latidos *flutter* ventriculares (!) que son asignados incorrectamente a latidos ventriculares prematuros (V). La diagonal representa los latidos correctamente agrupados. La columna y la fila $\bar{\Sigma}$ denota la suma en sentido horizontal y vertical respectivamente de los latidos erróneamente asignados (excluyendo entonces la diagonal). El error en sentido horizontal representa los latidos de otras clases asignados a ésta, y en sentido vertical los latidos de esta clase asignados a otras.

Observando esta tabla se puede verificar lo afirmado anteriormente sobre las performances de agrupamiento en las clases mayoritarias. Por ejemplo hay 108 latidos de clase normal (N) asignados a otras clases y 351 latidos de otras clase asignados a la clase normal. En el caso de la clase *J*, las cantidades son 16 y 4 respectivamente.

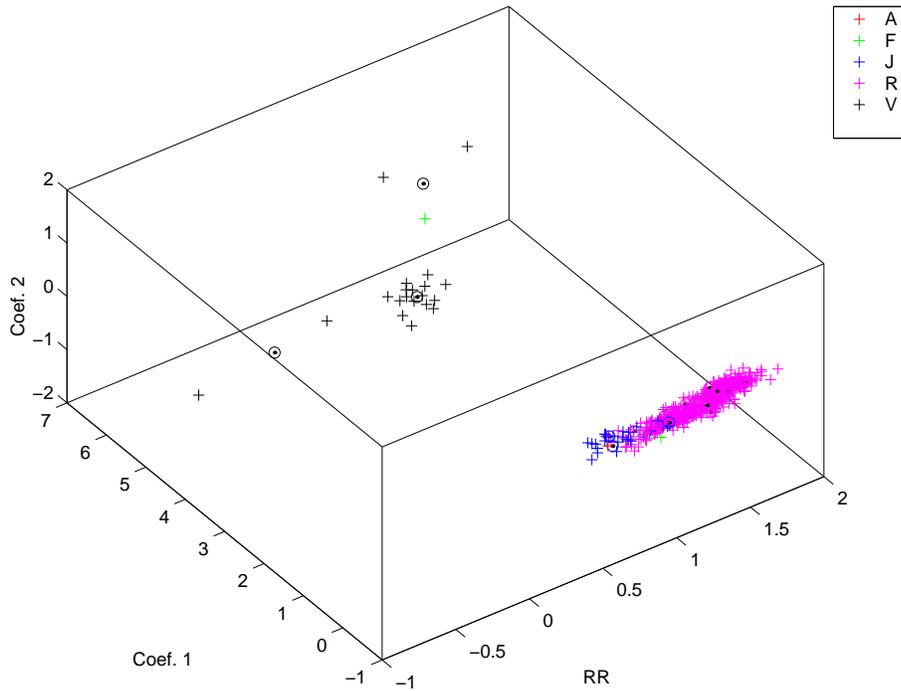
En la tabla 4.2 se muestra la performance global por registro. La sensibilidad global varía de 96.1% (registro 213) hasta 100% (varios registros). El valor predictivo global lo hace entre 95.3% (x_124) y 100% (varios). En la figura 4.1(a) se puede apreciar la superposición de los complejos QRS (después de filtrados) asociados a cada neurona para el registro x_124. Este registro presenta la performance más pobre de

Reg.	Clase	total	Reg.	Clase	total	Reg.	Clase	total
100	Nro.Lats	2271	208	Nro.Lats	2953	x_116	Nro.Lats	795
	S	100,00 %		S	96,95 %		S	100,00 %
	+P	100,00 %		+P	96,89 %		+P	100,00 %
101	Nro.Lats	1863	209	Nro.Lats	3002	x_117	Nro.Lats	503
	S	99,95 %		S	98,60 %		S	100,00 %
	+P	99,95 %		+P	98,56 %		+P	100,00 %
102	Nro.Lats	2185	210	Nro.Lats	2648	x_121	Nro.Lats	607
	S	98,49 %		S	98,83 %		S	100,00 %
	+P	98,26 %		+P	98,75 %		+P	100,00 %
103	Nro.Lats	2082	212	Nro.Lats	2746	x_122	Nro.Lats	835
	S	99,90 %		S	99,96 %		S	100,00 %
	+P	99,81 %		+P	99,96 %		+P	100,00 %
104	Nro.Lats	2227	213	Nro.Lats	3249	x_123	Nro.Lats	504
	S	97,49 %		S	96,12 %		S	99,80 %
	+P	96,67 %		+P	95,88 %		+P	99,60 %
105	Nro.Lats	2570	214	Nro.Lats	2259	x_124	Nro.Lats	522
	S	99,69 %		S	99,82 %		S	96,17 %
	+P	99,49 %		+P	99,69 %		+P	95,27 %
106	Nro.Lats	2025	215	Nro.Lats	3361	x_220	Nro.Lats	697
	S	99,90 %		S	99,88 %		S	99,71 %
	+P	99,90 %		+P	99,86 %		+P	99,71 %
107	Nro.Lats	2135	217	Nro.Lats	2206	x_221	Nro.Lats	825
	S	100,00 %		S	99,59 %		S	100,00 %
	+P	100,00 %		+P	99,59 %		+P	100,00 %
118	Nro.Lats	2276	219	Nro.Lats	2152	x_222	Nro.Lats	737
	S	99,65 %		S	99,40 %		S	98,10 %
	+P	99,67 %		+P	99,03 %		+P	97,43 %
119	Nro.Lats	1985	x_108	Nro.Lats	560	x_223	Nro.Lats	837
	S	100,00 %		S	99,64 %		S	97,37 %
	+P	100,00 %		+P	99,31 %		+P	96,00 %
200	Nro.Lats	2599	x_109	Nro.Lats	855	x_228	Nro.Lats	695
	S	98,38 %		S	99,88 %		S	99,86 %
	+P	97,19 %		+P	99,92 %		+P	99,86 %
201	Nro.Lats	1961	x_111	Nro.Lats	696	x_230	Nro.Lats	727
	S	97,86 %		S	99,86 %		S	100,00 %
	+P	98,25 %		+P	99,71 %		+P	100,00 %
202	Nro.Lats	2134	x_112	Nro.Lats	852	x_231	Nro.Lats	503
	S	98,50 %		S	100,00 %		S	99,60 %
	+P	98,47 %		+P	100,00 %		+P	99,41 %
203	Nro.Lats	2978	x_113	Nro.Lats	579	x_232	Nro.Lats	600
	S	98,15 %		S	100,00 %		S	99,83 %
	+P	97,91 %		+P	100,00 %		+P	99,83 %
205	Nro.Lats	2654	x_114	Nro.Lats	554	x_233	Nro.Lats	1021
	S	99,47 %		S	99,28 %		S	99,22 %
	+P	99,19 %		+P	98,60 %		+P	98,57 %
207	Nro.Lats	2330	x_115	Nro.Lats	633	x_234	Nro.Lats	919
	S	96,91 %		S	100,00 %		S	100,00 %
	+P	96,77 %		+P	100,00 %		+P	100,00 %

Tabla 4.3: Performance global por registro del agrupador CSL de CARDI-DENT.



(a) Agrupamiento de complejos QRS



(b) Visualización del espacio de entrada

Figura 4.1: Agrupamiento del registro x_124.

toda la base de datos, con una sensibilidad de 96.2% y un valor predictivo de 95.3%. En la figura se ve que la mayor parte de los errores corresponde a latidos prematuros nodales (J) asignados a bloqueos de rama derecha (R). La causa de esto es que la morfología de ambos tipos de QRS es la misma en el canal 1 para este paciente. También fue confundido un latido prematuro atrial (A). Este tipo de latidos tiene una morfología similar a los otros latidos presentes, pero tiene una mayor frecuencia cardíaca. El agrupador estimó que esta diferencia no amerita la asignación de una clase entera. Por último, un latido fusión entre ventricular y normal (F) fue asignado a un grupo de latidos ventriculares (V), y otro a un grupo de bloqueos de rama derecha (R).

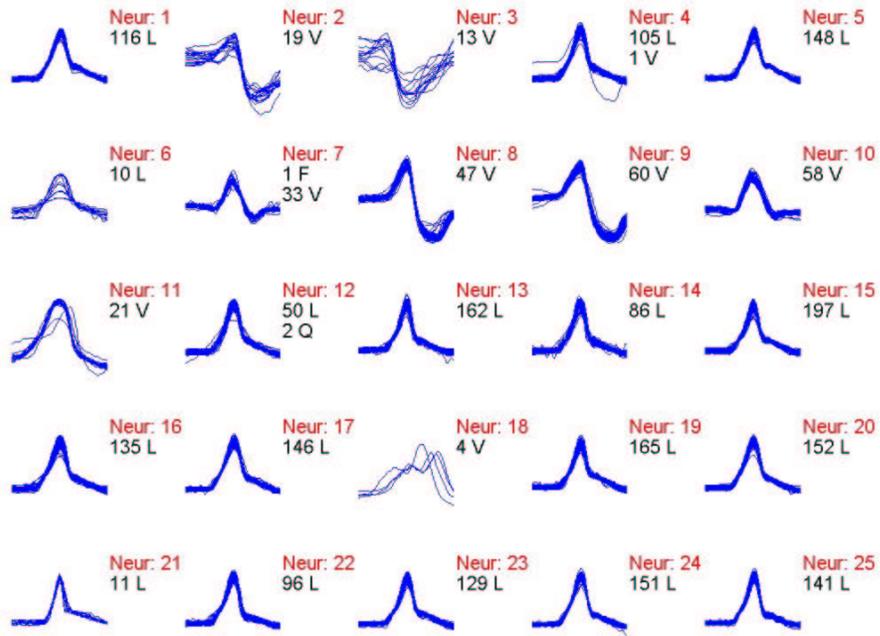
En la figura 4.1(b) se puede observar la estructura de los latidos en el espacio de entrada a la red (la salida de la extracción de características). Obviamente no se pueden representar las 12 dimensiones al mismo tiempo, por lo que se optó por hacerlo con las dos primeras direcciones de la transformada (que deberían tener la mayor parte de la información) y el período cardíaco instantáneo (RR). Las cruces de diferentes colores representan distintos tipos de latido, mientras que los círculos son las neuronas. Las observaciones hechas en el párrafo anterior pueden aclararse al apreciar la figura: algunos de los latidos prematuros (J) se confunden en su período con los bloqueos de rama derecha (R). Los latidos ventriculares (V) y el fusión (F) están separados mirando el primer coeficiente de la transformada.

Como contraste, en la figura 4.2(a) se presenta el resultado del agrupador para un registro de mejor performance, el 214 (sensibilidad 99.8%, predictividad 99.7%). En la misma se puede apreciar que los errores corresponden a un latido fusión entre ventricular y normal (F) asignado a un grupo de ventriculares (V), un V a un grupo de bloqueo de rama izquierda (L), y dos inclasificables (Q) a uno de L. En la figura 4.2(b) se puede apreciar el espacio de entrada para este registro. En la misma se observa que el latido ventricular (V) mal clasificado en la neurona 4 se debe a que tiene un período (RR) muy distinto al del resto de los demás ventriculares.

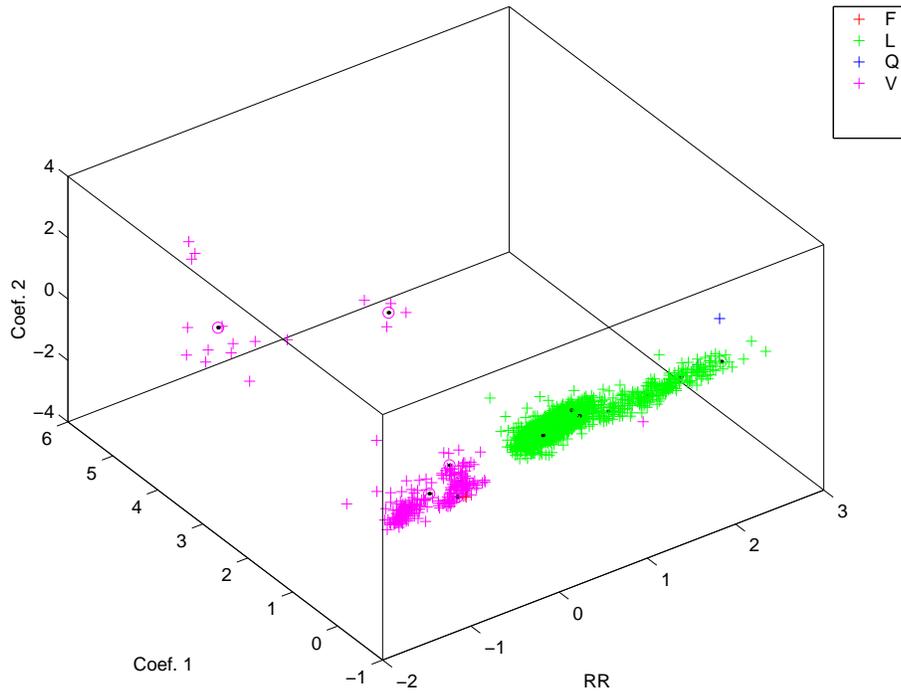
Es digno de mención que la inclusión de la medida del ancho no mejoró significativamente la performance de la clasificación en la transformada K-L (la diferencia es del orden de 0.1%). Esto se debe a que, indirectamente, la información del ancho ya se encuentra en los coeficientes. Además el cálculo e inclusión del ancho agrega costo de cálculo adicional y aumenta la dimensión del problema de clasificación. Por estas razones, si es necesario por razones computacionales se puede evitar el uso del mismo sin sacrificar el desempeño.

4.4.5. Resultados del agrupador basado en Hermite

A los efectos de comparar ambas representaciones de los latidos desde el punto de vista de la clasificación se implementó un agrupador basado en la transformada de Hermite. La metodología para el desarrollo de los algoritmos es exactamente la misma que se utilizó para la transformada K-L explicada en las secciones anteriores. A efectos de comparar se proyectó sobre una base de 9 funciones, es decir que se utilizaron 9 coeficientes más el parámetro b . Además, como se mostró en la sección 3.6 se utilizaron los parámetros temporales (menos el ancho, para el cual se utilizó el parámetro b). Los pesos asignados en este caso fueron los mismos que para la transformada K-L. Los resultados globales obtenidos con este algoritmo fueron 98.88% de sensibilidad y 98.85% de predictividad positiva. Como se puede ver los resultados son ligeramente



(a) Agrupamiento de complejos QRS



(b) Visualización del espacio de entrada

Figura 4.2: Agrupamiento del registro 214.

más bajos, sin embargo se puede afirmar que ambos algoritmos logran desempeños comparables. Por lo tanto, debido a que las bases de Hermite debe calcularse *en línea*, y las de K-L *fuera de línea* es preferible usar la transformada K-L y evitar así el costo de cálculo adicional, como se explicó en la sección 3.5.

4.4.6. Comparación con otros algoritmos

En esta sección se comparará la performance del agrupador desarrollado con la de otros algoritmos similares en la literatura.

Clustering ECG complexes using Hermite functions and self-organizing maps

Lagerholm *et al.* [1] usan la base de Hermite para extraer características de los complejos QRS y un mapa auto-organizado (sección 4.2.1.2) de 25 neuronas para realizar el agrupamiento. En este trabajo se reporta una performance global (sensibilidad) sobre *toda* la base de datos MIT-BIH de 98.5 % frente a 99.00 % de CARDIDENT. No se calcula el valor predictivo positivo. Cabe aclarar que en dicho trabajo la performance se calcula también dividiendo los latidos en los 16 tipos no vacíos que aparecen en la base de datos.

El trabajo de Lagerholm *et al.* presenta las siguientes diferencias básicas, que es necesario aclarar previo a comparar performances:

- Se utilizó la base de datos de 48 registros completos.
- No se usaron en este trabajo las anotaciones de la base de datos para ubicar los picos R sino un algoritmo de detección que tiene una sensibilidad de 99.7 % (aparentemente se ignoran en todo el estudio los Falsos Positivos del detector). Por lo tanto, la performance *real* del algoritmo en las condiciones de CARDIDENT está acotada por 98.8 %, que se daría en el caso en que todos los latidos que no fueron detectados fueran agrupados correctamente.
- En este trabajo se usaron ambos canales de la base de datos, CARDIDENT utiliza el canal 1 solamente.

No se reportaron las performances por registro. Sin embargo, para comparar los resultados con Hu *et al.* [4] (lo que no tiene sentido ya que es un algoritmo supervisado) se calcula la medida de performance: ¹⁵,

$$R^{SOM} = \frac{\text{No. de latidos V y E correctamente agrupados}}{\text{No. total de latidos}} \quad (4.36)$$

En la tabla 4.4 se puede apreciar la comparación de esta medida para el trabajo citado [1] y para CARDIDENT.

Classifying Multichannel ECG Patterns with an Adaptive Neural Network

En este trabajo S. Barro *et al.* [2] aplican una red neuronal ART multicanal para agrupar los QRS utilizando las muestras temporales. Se reporta una sensibilidad global de 98.72 % y una predictividad positiva de 98.78 % frente al valor predictivo

¹⁵Es de notar que aunque entre los VEB (latidos ectópicos ventriculares) de [4] se encuentran los latidos prematuros ventriculares R-sobre-T que Lagerholm *et al.* ignoran al definir la medida ya que no hay ninguno en la base de datos. Los *flutter* ventriculares no se colocan en este grupo (aunque deberían pertenecer) ya que [4] los ignora.

Reg.	Lagerholm <i>et al.</i>		CARDIDENT	
	Latidos	R^{SOM}	Latidos	R^{SOM}
200	2605	98,1 %	2599	99,6 %
201	1858	99,9 %	1961	100,0 %
202	2038	99,4 %	2134	100,0 %
203	2882	96,8 %	2978	98,3 %
205	2491	99,9 %	2654	99,9 %
207	2215	96,9 %	2330	97,2 %
208	2823	98,5 %	2953	98,9 %
209	2860	100,0 %	3002	100,0 %
210	2502	98,3 %	2648	99,3 %
213	3071	97,6 %	3249	98,7 %
214	2134	99,9 %	2259	99,9 %
215	3172	99,9 %	3361	100,0 %
219	2156	99,6 %	2152	99,8 %
221	2302	100,0 %	825	100,0 %
223	2480	99,1 %	837	99,3 %
228	1978	100,0 %	695	99,9 %
230	2303	100,0 %	727	100,0 %
231	1877	99,9 %	503	100,0 %
233	2936	99,8 %	1021	99,4 %
234	2577	100,0 %	919	100,0 %
Tot	49260	99,1 %	39807	99,5 %

Tabla 4.4: Comparación con algoritmo SOM de Lagerholm *et al.* [1].

de 98.98 % de CARDIDENT. Es necesario hacer varias aclaraciones sobre el trabajo de Barro:

- Utiliza 20 registros completos de la base MIT-BIH, y todos tienen latidos ventriculares.
- El número de neuronas¹⁶ no es fijo sino que se adapta a cada registro, llegando a más de 80 en el registro 108 frente a las 25 neuronas fijas en CARDIDENT.
- Se usan ambos canales de la base y se le asignan créditos a cada uno basado en el ruido estimado, mientras que CARDIDENT utiliza el canal 1 solamente.
- Es un algoritmo *online*; el agrupador de CARDIDENT es *offline*.

En la tabla 4.5 se muestran las performances en los registros donde son comparables en ambos trabajos. Como los autores no incluyeron el número de latidos por registro, se usaron los de la base de datos. Es de notar que los valores de sensibilidad y valor predictivo global no coinciden con el marcado por Barro, lo que lleva a pensar que no se tomó la misma cantidad de clases y deja un interrogante acerca de la validez de la comparación.

¹⁶En realidad es el número de clases del algoritmo, se usa el nombre *neurona* por asociación a CARDIDENT.

Reg.		MART	CARD.	Reg.		MART	CARD.
105	Nro.Lats	2570	2570	208	Nro.Lats	2953	2953
	S	99,53 %	99,69 %		S	97,53 %	96,95 %
	+P	99,53 %	99,49 %		+P	97,63 %	96,89 %
106	Nro.Lats	2025	2025	210	Nro.Lats	2648	2648
	S	94,97 %	99,90 %		S	98,17 %	98,83 %
	+P	93,34 %	99,90 %		+P	98,18 %	98,75 %
108	Nro.Lats	1738	560	213	Nro.Lats	3249	3249
	S	99,94 %	99,64 %		S	99,93 %	96,12 %
	+P	99,94 %	99,31 %		+P	99,93 %	95,88 %
114	Nro.Lats	1869	554	214	Nro.Lats	2259	2259
	S	98,11 %	99,28 %		S	99,22 %	99,82 %
	+P	99,95 %	98,60 %		+P	98,44 %	99,69 %
119	Nro.Lats	1985	1985	217	Nro.Lats	2206	2206
	S	100,00 %	100,00 %		S	97,29 %	99,59 %
	+P	100,00 %	100,00 %		+P	96,47 %	99,59 %
123	Nro.Lats	1518	504	219	Nro.Lats	2152	2152
	S	100,00 %	99,80 %		S	99,91 %	99,40 %
	+P	100,00 %	99,60 %		+P	99,91 %	99,03 %
200	Nro.Lats	2599	2599	221	Nro.Lats	2302	825
	S	96,85 %	98,38 %		S	99,83 %	100,00 %
	+P	96,21 %	97,19 %		+P	99,81 %	100,00 %
201	Nro.Lats	1961	1961	223	Nro.Lats	2480	837
	S	99,95 %	97,86 %		S	100,00 %	97,37 %
	+P	99,95 %	98,25 %		+P	100,00 %	96,00 %
202	Nro.Lats	2134	2134	228	Nro.Lats	1978	695
	S	99,95 %	98,50 %		S	99,65 %	99,86 %
	+P	99,95 %	98,47 %		+P	99,80 %	99,86 %
203	Nro.Lats	2978	2978				
	S	93,96 %	98,15 %				
	+P	95,55 %	97,91 %				
205	Nro.Lats	2654	2654	Total	Nro.Lats	46258	38348
	S	100,00 %	99,47 %		S	98,64 %	98,70 %
	+P	97,19 %	99,19 %		+P	98,49 %	98,48 %

Tabla 4.5: Comparación con algoritmo MART de Barro *et al.* [2].

4.5. Experto Local

4.5.1. Introducción

El experto local (LE) es la parte del sistema que se encarga de la adaptación al paciente. Esto se lleva a cabo entrenando la red con un conjunto reducido de latidos específicos del paciente. Cada registro de la base MIT-BIH contiene datos de un mismo paciente y por lo tanto antes de clasificar un registro, el LE se entrenará con los primeros latidos del mismo para poder tener información específica de ese paciente. Para poder llevar a cabo este entrenamiento es necesario disponer de información a priori de los tipos de latidos con los que se efectúa el entrenamiento, por lo que en la práctica el intervalo destinado al entrenamiento debe ser clasificado por un cardiólogo.

4.5.2. Implementación

El LE se implementó utilizando el algoritmo SCSL, realizando el entrenamiento para cada registro con sus primeros 2.5 minutos y utilizando las anotaciones del MIT-BIH como la clase verdadera de cada latido. Se trabajó sobre el mismo espacio 12-dimensional (sección 3.6) que se utilizó en el agrupador (ver sección 4.4.2). En este caso el mapeo es sobre una red de 25 neuronas al igual que en el agrupador.

4.5.3. Entrenamiento

El entrenamiento del algoritmo SCSL se realizó con los mismos parámetros y criterios que el CSL utilizado para el agrupador.

La diferencia con dicho algoritmo es que al finalizar la etapa de aprendizaje selectivo supervisado, se ejecuta el algoritmo oLVQ. Para el mismo se toman como pesos iniciales de sus neuronas, los obtenidos de la etapa selectiva. En este caso también los vectores de entrada fueron presentados al clasificador de forma aleatoria, actualizando los pesos con cada vector presentado.

El *learning rate* inicial se tomó 0.1, y se va actualizando en cada iteración de acuerdo a la regla presentada en la ecuación (4.31). Además se decrementó el mismo en un 15% luego de finalizada cada *epoch*.

Como criterio de parada se estableció que la norma de la variación absoluta de los pesos de las neuronas (entre dos *epochs* consecutivas) sea menor que 5×10^{-3} o que se llegue al número máximo de 5 *epochs*.

4.5.4. Resultados

Para evaluar el clasificador local se utilizaron los latidos correspondientes al intervalo comprendido entre el minuto 5 hasta el final del registro, de acuerdo a los criterios de la AAMI (ver sección 1.3).

Para medir la performance del algoritmo se utilizaron los mismos criterios que para el agrupador mostrados en la 4.4.4. Los latidos fueron discriminados en las 6 clases explicadas en la sección 1.3.

Los resultados de esta sección se presentarán en las siguientes tablas. En la tabla 4.6 se pueden ver los resultados obtenidos por el experto local por clase sobre la base de datos. La tabla 4.7 muestra los resultados obtenidos por el LE, discriminado por registros. Además la tabla 4.8 muestra la clasificación por clases de todos los latidos, especificando a que clase fueron asignados. Finalmente en el Apéndice B se presenta una tabla detallada de los resultados de la clasificación por registro y por clase.

En estas tablas se puede observar el mismo fenómeno que aparece en el agrupador (ver sección 4.4), es decir que los latidos más comunes presentan una sensibilidad alta y un valor predictivo positivo menor. En general valen las mismas observaciones realizadas para el agrupador, sin embargo aparece un nuevo efecto que no se presentaba anteriormente. Debido a la forma de entrenar el LE con los primeros minutos de registro, este clasificador puede fallar completamente en identificar una clase si ésta no se encuentra presente en el período de entrenamiento. Un ejemplo claro de éste fenómeno se puede apreciar en la performance del registro 207 para los latidos clase S (ver tabla B.1). Esto tiene como consecuencia que la sensibilidad y el valor predictivo positivo para el registro 207 bajen (ver tabla 4.7), y contribuye a que, como se ve en la tabla 4.8, aproximadamente el 50% de los latidos clase S de la base se clasifiquen como normales. Este fenómeno aparece también en los registros x_124, 201 y 217 pero

Clase	N	S	V	F	/	Q	Total
Nro.Lats	45212	1175	4882	1431	5934	28	58662
Se	98,98 %	53,96 %	88,04 %	89,24 %	99,48 %	0,00 %	96,93 %
+P	97,79 %	93,79 %	92,47 %	78,25 %	99,43 %	0,00 %	96,91 %

Tabla 4.6: Performance por clase del LE de CARDIDENT sobre la base de datos

con otras clases de latidos.

Los resultados del LE obtenidos sobre toda la base de datos fueron una sensibilidad de 96.93 % y un valor predictivo positivo de 96.91 %. Estos resultados validan al algoritmo para su uso clínico y son comparables con la literatura especializada, como se mostrará en la siguiente sección.

4.5.5. Comparación con otros algoritmos

En esta sección se compararán los resultados de CARDIDENT con los resultados presentados en la literatura. Dadas las condiciones en las cuales se mide el desempeño es natural comparar con [4].

A Patient Adaptable ECG Beat Classifier using a Mixture of Experts Approach [4]

Previo a presentar y comparar los resultados con el trabajo de Hu *et al* [4] es necesario hacer algunas aclaraciones:

- El objetivo de dicho trabajo es distinto, puesto que se propone clasificar una clase específica de latidos (V). Y por lo tanto define magnitudes como la *especificidad* que en CARDIDENT no se utilizan (el concepto de TN no fue definido), puesto que no se planteó una clase objetivo.
- Para realizar la comparación anterior se deben separar los latidos en V o no-V y calcular la cantidad de FP, TP, FN y TN. La clase V considerada por *Hu et al.* no incluye las ondas flutter ventriculares (!), y la de CARDIDENT si los incluye, por lo que los resultados no son comparables en sentido estricto. De todas formas esta comparación proporciona una buena idea sobre la performance de ambos algoritmos.
- Otra diferencia significativa entre ambos es que en [4] se eliminan algunos registros. Esto se hace por dos razones: se eliminan los que no contienen latidos V (porque es el objetivo del trabajo) y los que contienen latidos de marcapasos (/) lo cual es una practica aceptada por la AAMI. A los efectos de la comparación se presentarán los resultados solamente sobre los registros utilizados por [4].
- Hay que destacar también (tabla 4.9) que en la mayoría de los casos *Hu et al.* presenta una cantidad significativamente mayor que el presente trabajo. Los valores presentados en [4] coinciden con el número total de latidos por registro, lo cual no deja claro si evalúan con todo el registro y por lo tanto no respetan las recomendaciones de la AAMI. O si, en cambio, dichas cantidades no representan la cantidad de latidos usados para la evaluación, sino simplemente la cantidad

Reg.	Total		Reg.	Total		Reg.	Total	
100	Nro.Lats	1902	208	Nro.Lats	2437	x_116	Nro.Lats	402
	S	99,68 %		S	94,67 %		S	100,00 %
	+P	99,65 %		+P	94,77 %		+P	100,00 %
101	Nro.Lats	1523	209	Nro.Lats	2518	x_117	Nro.Lats	254
	S	99,67 %		S	92,57 %		S	100,00 %
	+P	99,34 %		+P	92,91 %		+P	100,00 %
102	Nro.Lats	1821	210	Nro.Lats	2204	x_121	Nro.Lats	306
	S	97,97 %		S	98,05 %		S	100,00 %
	+P	98,17 %		+P	97,10 %		+P	100,00 %
103	Nro.Lats	1729	212	Nro.Lats	2285	x_122	Nro.Lats	415
	S	99,88 %		S	100,00 %		S	100,00 %
	+P	99,77 %		+P	100,00 %		+P	100,00 %
104	Nro.Lats	1857	213	Nro.Lats	2700	x_123	Nro.Lats	257
	S	97,95 %		S	92,48 %		S	99,61 %
	+P	97,54 %		+P	94,29 %		+P	99,22 %
105	Nro.Lats	2155	214	Nro.Lats	1878	x_124	Nro.Lats	272
	S	99,72 %		S	97,55 %		S	85,66 %
	+P	99,50 %		+P	97,44 %		+P	73,38 %
106	Nro.Lats	1696	215	Nro.Lats	2795	x_220	Nro.Lats	345
	S	99,88 %		S	99,89 %		S	99,42 %
	+P	99,88 %		+P	99,82 %		+P	99,42 %
107	Nro.Lats	1784	217	Nro.Lats	1845	x_221	Nro.Lats	420
	S	99,72 %		S	85,96 %		S	99,76 %
	+P	99,72 %		+P	79,93 %		+P	99,76 %
118	Nro.Lats	1916	219	Nro.Lats	1773	x_222	Nro.Lats	372
	S	99,06 %		S	99,04 %		S	97,85 %
	+P	98,98 %		+P	98,66 %		+P	95,75 %
119	Nro.Lats	1661	x_108	Nro.Lats	279	x_223	Nro.Lats	433
	S	100,00 %		S	98,92 %		S	93,07 %
	+P	100,00 %		+P	98,58 %		+P	96,71 %
200	Nro.Lats	2168	x_109	Nro.Lats	424	x_228	Nro.Lats	347
	S	98,34 %		S	100,00 %		S	100,00 %
	+P	96,99 %		+P	100,00 %		+P	100,00 %
201	Nro.Lats	1521	x_111	Nro.Lats	350	x_230	Nro.Lats	332
	S	81,26 %		S	99,71 %		S	100,00 %
	+P	71,16 %		+P	99,43 %		+P	100,00 %
202	Nro.Lats	1871	x_112	Nro.Lats	426	x_231	Nro.Lats	212
	S	96,85 %		S	100,00 %		S	100,00 %
	+P	94,26 %		+P	100,00 %		+P	100,00 %
203	Nro.Lats	2481	x_113	Nro.Lats	292	x_232	Nro.Lats	307
	S	95,00 %		S	100,00 %		S	99,67 %
	+P	95,54 %		+P	100,00 %		+P	99,68 %
205	Nro.Lats	2201	x_114	Nro.Lats	281	x_233	Nro.Lats	505
	S	96,41 %		S	98,93 %		S	99,60 %
	+P	92,95 %		+P	97,99 %		+P	99,21 %
207	Nro.Lats	1932	x_115	Nro.Lats	319	x_234	Nro.Lats	459
	S	93,89 %		S	100,00 %		S	100,00 %
	+P	88,88 %		+P	100,00 %		+P	100,00 %

Tabla 4.7: Performances totales del LE de CARDIDENT por registro

Clase	N	S	V	F	/	Q	$\bar{\Sigma}$
n	44751	521	413	64	0	14	1012
s	20	634	22	0	0	0	42
v	269	19	4298	61	0	1	350
f	172	1	139	1277	30	13	355
/	0	0	10	24	5903	0	34
q	0	0	0	5	1	0	6
$\bar{\Sigma}$	461	541	584	154	31	28	1799

Tabla 4.8: Clasificación del LE por clase. Las clases anotadas en la base se muestran horizontalmente y la clasificación hecha por el experto verticalmente

total de latidos en el registro, lo cual es coherente con lo desarrollado por los autores.

Para el experto local Hu *et al.* utilizaron una combinación de los algoritmos LVQ y SOM implementados con los paquetes de software LVQ_PAK y SOM_PAK¹⁷. La clasificación se realizó en una red de 150 neuronas sobre 4 clases siguiendo las recomendaciones de la AAMI explicadas en la sección 1.3. Como se explicó el objetivo de Hu *et al.* es clasificar la clase V, por lo que las cantidades TP, FP, FN y TN tienen significados levemente distintos a los utilizados en este trabajo o directamente no están definidas:

TP : Verdaderos Positivos (True Positive) son los latidos V clasificados como V

FP : Falsos Positivos (False Positive) son los latidos no-V clasificados como V

FN : Falsos Negativos (False Negative) son los latidos V clasificados como no-V

TN : Verdaderos Negativos (True Negative) son los latidos no-V clasificados como no-V

A partir de estas definiciones se realizan las siguientes medidas de la performance, algunas de las cuales ya fueron definidas pero considerando un clasificador con una clase objetivo varían levemente su significado:

Sensibilidad Es la fracción de latidos V correctamente clasificados.

$$SE = \frac{TP}{TP + FN}$$

Especificidad Es la fracción de latidos no-V correctamente clasificados como no-V.

$$SP = \frac{TN}{TN + FP}$$

Valor Predictivo Positivo Es la fracción de latidos V correctamente clasificados entre todos los latidos clasificados como V.

$$PP = \frac{TP}{TP + FP}$$

¹⁷Universidad de Helsinki, Finlandia. <ftp://cochlea.hut.fi/pub/>

reg	Hu et al.					CARDIDENT				
	No lat	SE	SP	PP	CR	No lat	SE	SP	PP	CR
200	2605	93,0 %	98,7 %	97,0 %	97,0 %	2168	99,3 %	99,9 %	99,7 %	99,7 %
201	1858	0,0 %	100,0 %	NaN	89,3 %	1521	0,0 %	100,0 %	NaN	87,0 %
202	2038	0,0 %	100,0 %	NaN	99,1 %	1871	93,3 %	99,3 %	50,0 %	99,2 %
203	2882	88,0 %	94,7 %	73,0 %	93,7 %	2481	87,9 %	97,3 %	85,4 %	95,9 %
205	2491	0,0 %	100,0 %	NaN	97,1 %	2201	0,0 %	100,0 %	NaN	97,0 %
207	2215	0,0 %	100,0 %	NaN	92,2 %	1932	97,6 %	99,9 %	99,8 %	99,4 %
208	2823	95,0 %	94,4 %	89,0 %	94,5 %	2437	95,8 %	97,6 %	95,4 %	97,0 %
209	2860	0,0 %	100,0 %	NaN	100,0 %	2518	0,0 %	100,0 %	NaN	100,0 %
210	2502	96,0 %	96,3 %	67,0 %	96,2 %	2204	89,1 %	99,8 %	96,7 %	99,0 %
213	3071	0,0 %	100,0 %	NaN	92,8 %	2700	64,1 %	99,4 %	89,3 %	96,9 %
214	2134	69,0 %	98,5 %	86,0 %	95,2 %	1878	79,8 %	99,9 %	99,4 %	97,7 %
215	3172	99,0 %	97,0 %	62,0 %	97,1 %	2795	99,2 %	100,0 %	100,0 %	100,0 %
219	2156	89,0 %	93,1 %	25,0 %	93,0 %	1773	82,4 %	100,0 %	100,0 %	99,5 %
x_221	2302	98,0 %	94,9 %	79,0 %	95,4 %	420	98,8 %	100,0 %	100,0 %	99,8 %
x_223	2480	66,0 %	96,9 %	83,0 %	91,0 %	433	58,5 %	99,7 %	96,0 %	95,8 %
x_228	1978	97,0 %	98,1 %	91,0 %	97,9 %	347	100,0 %	100,0 %	100,0 %	100,0 %
x_230	2303	0,0 %	100,0 %	NaN	100,0 %	332	NaN	100,0 %	NaN	100,0 %
x_231	1877	0,0 %	100,0 %	NaN	99,9 %	212	NaN	100,0 %	NaN	100,0 %
x_233	2936	95,0 %	97,3 %	93,0 %	96,7 %	505	100,0 %	99,2 %	97,8 %	99,4 %
x_234	2577	0,0 %	100,0 %	NaN	99,9 %	459	NaN	100,0 %	NaN	100,0 %
Tot.	49260	79,0 %	98,0 %	83,0 %	95,9 %	31187	86,0 %	99,5 %	95,9 %	97,9 %

Tabla 4.9: Comparación de performance por registro del LE de CARDIDENT contra el LE de HU *et al.*

Tasa de Clasificación Es la fracción de latidos correctamente clasificados entre todos los latidos.

$$CR = \frac{TN + TP}{TN + TP + FN + FP}$$

La tabla 4.9 muestra claramente que la performance de CARDIDENT es superior en todas las medidas de performance y en algunos casos, como en el valor predictivo positivo, existe una diferencia importante. En la tasa de clasificación, que muestra como se comportan los clasificadores frente a todos los tipos de latidos, se puede apreciar el mejor desempeño de CARDIDENT frente a este trabajo fundamental.

4.6. Experto Global

4.6.1. Introducción

La idea de un experto global es utilizar la información disponible en bases de datos de gran tamaño para poder abarcar la mayor variedad posible de tipos y morfologías de latidos. Es posible así evitar problemas como los que presenta el LE (sección 4.5) cuando debe clasificar un latido de un tipo que no apareció en el período de entrenamiento. Sin embargo, esta estrategia tiene como desventaja que la base de datos debe ser muy grande y variada para lograr performances aceptables. De todas maneras, como se mencionó antes, nunca es lo suficientemente variada.

4.6.2. Implementación

El GE se implementó entrenando con el algoritmo SCSL sobre una red de 150 neuronas. Para evaluar el GE desarrollado no es posible utilizar los mismos registros para entrenar y evaluar. Por lo tanto, al disponer de una única base de datos, hay que dividir la misma en dos partes: un grupo de registros se utilizó como base de datos para el entrenamiento y otro grupo se utilizó para evaluar la performance. En la tabla 4.10 se muestra que registros se utilizaron para el entrenamiento y cuáles para la evaluación.

Así dispuesta la división, los registros de entrenamiento presentan una mayor cantidad de latidos normales y las señales más limpias, mientras que los registros de evaluación presentan una mayor cantidad de latidos no-normales y episodios que presentan mayores dificultades, como fue explicado en la introducción (1.3). La idea es simular un caso realista para el GE, ya que ninguna base por amplia que sea puede incluir todos los casos.

En el conjunto de registros de entrenamiento existen aproximadamente 30.000 latidos, esto hace que cada *epoch* sea muy costosa computacionalmente y por lo tanto consume mucho tiempo. De la misma forma si se de agrega un registro nuevo al conjunto entrenamiento es necesario realizar nuevamente el proceso completo con la totalidad de los latidos. Además, existe mucha información redundante al utilizar todos los latidos de un mismo registro. Para evitar este problema, se redujo el número de vectores de entrenamiento a 2500, que es un número más manejable y que corresponde al promedio típico de latidos presentes en un registro. Estos 2500 vectores se elegirán de manera que cada registro esté bien representado, lo cual se logra eligiendo un número de vectores por registro proporcional a la cantidad total en ese registro. A su vez, dentro de cada registro, se utiliza el mismo criterio para lograr que cada clase esté bien representada. La manera más sencilla de hacerlo es simplemente tomar uno de cada N latidos, pero es una elección que carece de un criterio sobre la cantidad de información que se pierde. Un método más adecuado para seleccionar los vectores en cada registro, es realizar el entrenamiento de un clasificador para obtener un cierto número de vectores representativos. Es importante destacar que estos vectores no corresponden exactamente a latidos y que además el algoritmo garantiza que la cantidad de vectores de cada clase concuerde con las proporciones presentes en el registro. Una vez obtenidos los vectores representativos por registro, se juntan en un conjunto, de 2500 vectores, que es el que se utiliza para el entrenamiento del Global Expert.

La proporción de latidos normales presentes en la base de datos es muy grande con respecto al resto de las clases. Esto tiene como consecuencia que si se entrena con estas proporciones el clasificador funcionará muy bien con los normales y tendrá un desempeño muy pobre en el resto de las clases. La solución planteada a este problema fue utilizar (sólo para el entrenamiento) uno de cada cuatro latidos normales en cada registro, con lo cual se logra una representación más uniforme de los latidos. Este procedimiento se efectuó previo a la elección de vectores representativos mostrada anteriormente.

4.6.3. Entrenamiento

Los parámetros empleados en el algoritmo SCSL, así como los criterios de parada, fueron los mismos usados para entrenar el LE (sección 4.5.3).

Grupo	Registros
Entrenamiento	100, 101, 102, 103, 104, 105, 106, 107, 118, 119, x_108, x_109, x_111, x_112, x_113, x_114, x_115, x_116, x_117, x_121, x_122, x_123, x_124
Evaluación	200, 201, 202, 203, 205, 207, 208, 209, 210, 212, 213, 214, 215, 217, 219, x_220, x_221, x_222, x_223, x_228, x_230, x_231, x_232, x_233, x_234

Tabla 4.10: División de la base para el entrenamiento y evaluación del GE

Clase	N	S	V	F	/	Q	Total
No.Lat	29356	1035	3872	830	1240	8	36341
S	87,70 %	24,44 %	87,65 %	6,51 %	73,71 %	12,50 %	83,54 %
+P	94,21 %	24,73 %	68,43 %	2,96 %	92,98 %	0,44 %	87,34 %

Tabla 4.11: Performances totales del GE de CARDIDENT por clase

4.6.4. Resultados

De acuerdo a los criterios de la AAMI la evaluación se realizó a partir del minuto 5 del grupo de registros correspondientes. Los resultados se pueden ver en las tablas 4.11, 4.12 y 4.13. Las mismas muestran que la performance del GE es inferior a la del LE ya que muestra una sensibilidad de 83.54 % y una predictividad positiva de 87.34 %. La principal razón para esta baja en la performance es que el conjunto de registros utilizados como base de entrenamiento es insuficiente y cualquier base de datos para entrenamiento de un clasificador de uso clínico debe ser mucho más amplia. Es de esperar que al usar una base de datos más completa la performance del algoritmo se incremente. De todas formas el resultado es aceptable y como se verá en la sección 4.6.5 está a la par de los resultados alcanzados por otras publicaciones.

Se puede apreciar en la tabla 4.12 que hay grandes diferencias en las performances por registro. Mientras que en la mayoría de los registros la performance está por encima del 80 %, en algunos se presentan caídas importantes. Por ejemplo, en los registros 203 (sensibilidad 68.4 %), 217 (69.6 %), y especialmente en los registros x_223 (29.6 %) y x_232 (23.5 %). Esto se debe a la presencia de gran cantidad de morfologías particulares a los pacientes frente a las cuales el GE tiene un pobre desempeño. Estos registros muestran la necesidad de adaptarse al paciente.

4.6.5. Comparación con otros Algoritmos

En esta sección se comparará el GE desarrollado en CARDIDENT con el que desarrollaron Hu *et al.* en [4].

A Patient Adaptable ECG Beat Classifier using a Mixture of Experts Approach [4]

Las diferencias planteadas en la comparación de los algoritmos para expertos locales se mantienen en esta comparación. En particular es necesario evaluar la performance del GE como un clasificador de latidos V y considerar las definiciones de TP, FP, TN y FN y de las medidas de performance explicadas en la sección 4.5.5. Dejando de lado el hecho, ya comentado, de que Hu *et al.* descartan algunos registros (sección 4.5.5), la división de la base MIT-BIH en una base de entrenamiento y otra

Reg			Reg			Reg		
200	Nro.Lats	2168	212	Nro.Lats	2285	x_223	Nro.Lats	433
	Se	87,92 %		Se	97,94 %		Se	29,56 %
	+P	96,65 %		+P	100,00 %		+P	74,30 %
201	Nro.Lats	1521	213	Nro.Lats	2700	x_228	Nro.Lats	347
	Se	71,20 %		Se	88,37 %		Se	98,56 %
	+P	70,92 %		+P	89,68 %		+P	99,71 %
202	Nro.Lats	1871	214	Nro.Lats	1878	x_230	Nro.Lats	332
	Se	73,44 %		Se	71,78 %		Se	99,10 %
	+P	95,98 %		+P	99,43 %		+P	100,00 %
203	Nro.Lats	2481	215	Nro.Lats	2795	x_231	Nro.Lats	212
	Se	68,40 %		Se	99,50 %		Se	98,58 %
	+P	90,26 %		+P	99,46 %		+P	100,00 %
205	Nro.Lats	2201	217	Nro.Lats	1845	x_232	Nro.Lats	307
	Se	99,27 %		Se	69,59 %		Se	23,45 %
	+P	98,97 %		+P	86,07 %		+P	84,60 %
207	Nro.Lats	1932	219	Nro.Lats	1773	x_233	Nro.Lats	505
	Se	82,35 %		Se	75,47 %		Se	92,48 %
	+P	83,14 %		+P	97,86 %		+P	99,20 %
208	Nro.Lats	2437	x_220	Nro.Lats	345	x_234	Nro.Lats	459
	Se	81,86 %		Se	98,84 %		Se	100,00 %
	+P	87,88 %		+P	98,85 %		+P	100,00 %
209	Nro.Lats	2518	x_221	Nro.Lats	420			
	Se	93,09 %		Se	84,76 %			
	+P	93,20 %		+P	100,00 %			
210	Nro.Lats	2204	x_222	Nro.Lats	372		Nro.Lats	36341
	Se	79,81 %		Se	93,82 %		Se	83,54 %
	+P	95,63 %		+P	98,57 %		+P	87,34 %

Tabla 4.12: Performances totales del GE de CARDIDENT por registro

Clase	N	S	V	F	/	Q	$\tilde{\Sigma}$
n	25745	358	300	599	321	3	1581
s	753	253	15	2	0	0	770
v	1022	415	3394	123	4	2	1566
f	1753	6	9	54	0	2	1770
/	3	2	23	41	914	0	69
q	80	1	131	11	1	1	224
$\tilde{\Sigma}$	3611	782	478	776	326	7	5980

Tabla 4.13: Clasificación del GE por clases. Las clases anotadas en la base se muestran horizontalmente y la clasificación hecha por el experto verticalmente

reg	Hu et al.					CARDIDENT				
	No lat	SE	SP	PP	CR	No lat	SE	SP	PP	CR
200	2605	53,6 %	97,4 %	89,6 %	62,3 %	2168	97,9 %	99,6 %	99,1 %	99,0 %
201	1858	95,0 %	80,6 %	36,7 %	72,4 %	1521	20,7 %	84,1 %	16,3 %	75,8 %
202	2038	94,4 %	66,8 %	2,6 %	66,7 %	1871	66,7 %	87,1 %	4,0 %	86,9 %
203	2882	48,1 %	81,7 %	30,4 %	33,9 %	2481	98,4 %	73,3 %	39,5 %	77,1 %
205	2491	59,2 %	100,0 %	100,0 %	81,2 %	2201	93,9 %	100,0 %	100,0 %	99,8 %
207	2215	6,2 %	94,0 %	7,6 %	80,6 %	1932	56,8 %	90,2 %	63,6 %	82,4 %
208	2823	59,6 %	87,3 %	69,5 %	60,5 %	2437	98,8 %	93,1 %	88,0 %	95,0 %
209	2860	0,0 %	99,9 %	0,0 %	21,9 %	2518	100,0 %	100,0 %	50,0 %	100,0 %
210	2502	69,7 %	92,8 %	42,5 %	88,1 %	2204	88,5 %	97,5 %	74,1 %	96,8 %
213	3071	58,3 %	78,6 %	17,5 %	74,4 %	2700	89,2 %	98,4 %	80,9 %	97,7 %
214	2134	92,1 %	92,8 %	62,4 %	92,4 %	1878	97,2 %	99,9 %	99,5 %	99,6 %
215	3172	13,7 %	99,9 %	95,2 %	3,7 %	2795	94,7 %	99,8 %	96,1 %	99,6 %
219	2156	18,2 %	100,0 %	100,0 %	97,1 %	1773	100,0 %	96,7 %	47,2 %	96,8 %
x_221	2302	96,1 %	92,5 %	70,8 %	58,9 %	420	100,0 %	100,0 %	100,0 %	100,0 %
x_223	2480	69,2 %	17,8 %	16,4 %	26,7 %	433	48,8 %	98,0 %	71,4 %	93,3 %
x_228	1978	81,7 %	96,1 %	80,7 %	93,5 %	347	99,0 %	100,0 %	100,0 %	99,7 %
x_230	2303	100,0 %	75,3 %	0,2 %	72,4 %	332	NaN	100,0 %	NaN	100,0 %
x_231	1877	0,0 %	99,8 %	0,0 %	98,5 %	212	NaN	98,6 %	0,0 %	98,6 %
x_233	2936	66,9 %	5,7 %	20,3 %	20,9 %	505	97,7 %	99,5 %	98,5 %	99,0 %
x_234	2577	66,7 %	99,5 %	13,2 %	97,4 %	459	NaN	100,0 %	NaN	100,0 %
Tot.	49260	63,5 %	83,5 %	31,9 %	62,2 %	31187	87,4 %	94,7 %	69,3 %	93,9 %

Tabla 4.14: Comparación de performance por registro del GE de CARDIDENT contra el GE de HU *et al.*

de evaluación es la misma.

En la tabla 4.14 se comparan ambos algoritmos según los parámetros utilizados por Hu *et al.* y se puede observar a simple vista que CARDIDENT supera ampliamente el desempeño alcanzado en [4]. Aparecen inclusive diferencias del orden de 30 puntos porcentuales (Se: 63.5 % 87.4 %, +P: 31.9 % 69.3 %, CR: 62.2 % 93.9 %) lo que marca una mejora sustancial más allá de las diferencias en la evaluación.

4.7. Mezcla de Expertos

4.7.1. Introducción

El problema que presentan la mayoría de los clasificadores basados exclusivamente en el entrenamiento sobre una amplia base de datos, como el experto global presentado en la sección 4.6, es que muchas veces no logran performances aceptables a pesar del tamaño o variedad de la base utilizada ya que en el caso de los ECG los latidos presentan una gran variabilidad en las morfologías. La alternativa de un experto local, presentada en la sección 4.5, no requiere de una base de datos general, ya que utiliza para su entrenamiento información específica del paciente. Sin embargo esta otra estrategia presenta problemas cuando no aparece ningún latido de un cierto tipo en el conjunto de entrenamiento . Éste fue el caso del registro 201, tal como ya fue comentado en la sección 4.5.4.

El enfoque de la Mezcla de Expertos (MOE, Mixtures of Experts) fue propuesto originalmente por Jacobs *et al.* en 1988 y fue utilizado en diversos trabajos publicados en la literatura especializada ([41], [42], [43]). Su idea básica es que la combinación lineal de varias estimaciones estadísticas tendrán una mejor performance que la de

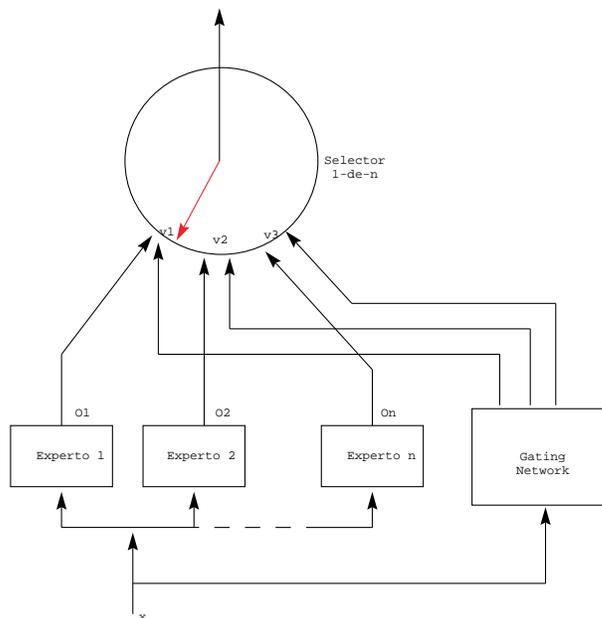


Figura 4.3: Sistema MOE genérico formado por un conjunto de expertos y una gating network que realiza una selección $1-de-n$ ponderada por el vector de características

cualquiera de éstas individualmente. Esta idea se basa en el hecho general de que un panel de expertos suelen obtener un mejor diagnóstico que cualquier experto individualmente, ya que en el panel cada experto puede contribuir desde su propio punto de vista. De esta manera se puede utilizar la información complementaria que aportan el LE y el GE para obtener un mejor resultado que el que se logra con los algoritmos utilizados independientemente.

4.7.2. Fundamento Teórico

El algoritmo de *Mixture of experts* (MOE) es una técnica que permite generar un clasificador a partir de la combinación de varios clasificadores llamados *expertos*. En esta sección se presentará el fundamento teórico del MOE, y por simplicidad de la exposición, se realizará el desarrollo en el caso de dos expertos, aunque todos los razonamientos presentados aquí son válidos para un mayor número de expertos.

Sean $o_i(x)$ la salida del experto i con $i = 1, 2$, donde x es el vector de características de entrada. Esta salida es un vector columna, de tamaño igual al número de clases, en el cual cada componente representa el grado de proximidad del vector de entrada a la clase correspondiente.

La idea genérica de la mezcla de expertos se ilustra en la figura 4.3, donde se puede ver que existe una *gating network* que combina las salidas de ambos expertos pesada por una función del vector de entrenamiento. Esto se puede ver como una selección ponderada de uno de los n expertos.

En el caso del presente trabajo, la mezcla de expertos se implementó de acuerdo al modelo que se muestra en la figura 4.4. En ella se puede ver que las salidas de los expertos son pesadas (multiplicadas) por las salidas de la gating network (g_1 y g_2)

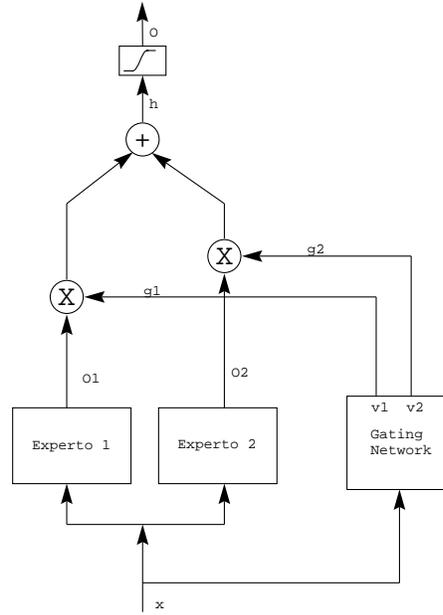


Figura 4.4: Detalle de la implementación realizada del MOE formado por dos expertos y una gating network.

y luego sumadas para obtener la función de activación h . Finalmente se calcula la salida o pasando través de una función de transferencia no-lineal y diferenciable. Resumiendo, la función de activación del MOE (para la clase m) se calcula como

$$h_m(x) = g_{1m}(x)o_{1m}(x) + g_{2m}(x)o_{2m}(x) \quad (4.37)$$

donde g_{1m} y g_{2m} son las funciones de peso de cada experto de la *gating network* definidos como

$$g_{jc} = \frac{e^{v_{jm}^t x}}{\sum_{j=1}^2 e^{v_{jm}^t x}} \quad (4.38)$$

donde los v_{ij} son los pesos de la *gating network*. Como se puede ver en 4.38 se cumple que $\sum_{j=1}^2 g_{jc} = 1$ y finalmente la salida es

$$o_m(x) = f(h_m(x)) \quad (4.39)$$

donde f es alguna función de umbral no-lineal y diferenciable (en este caso una sigmoide).

Teorema 2 Sea $R[o_i(x)] = \{x \in X/o_i = z(x)\}$ con $i = 1, 2$ donde $z(x)$ es la salida correcta. Este conjunto $R[o_i(x)]$ es la sub-región del espacio de características donde el clasificador i clasifica correctamente. Sea $R[o(x)]$ definido de la misma forma para el MOE. Se toma la hipótesis que $o_i(x) \in 0, 1$ y $z(x) \in 0, 1$. Entonces,

$$R[o(x)] \subseteq R[o_1(x)] \cup R[o_2(x)] \quad (4.40)$$

Prueba:

Para realizar esta prueba es necesario probar únicamente que si ambos expertos clasifi-

can mal, entonces el MOE no puede clasificar correctamente. Como $z(x)$, $o_1(x)$, $o_2(x)$ y $o(x)$ son vectores que toman valores binarios de la misma dimensión, si ambos expertos clasifican mal un vector x perteneciente a la clase m , se tiene que, para el componente m de los vectores binarios

$$z_m(x) \oplus o_{1m}(x) = z_m(x) \oplus o_{1m}(x) = 0 \quad (4.41)$$

Como $g_{1m} + g_{2m} = 1$, se puede concluir que $o_m(x) = 0$, si $o_1(x) = o_2(x) = 0$ y $o_m(x) = 1$, si $o_1(x) = o_2(x) = 1$. Por lo tanto $z_m(x) \oplus o_m(x) = 0$. En otras palabras $o(x)$ debe clasificar mal el vector de características x para cualquier elección de las funciones de peso $g_{jm}(x)$. Es decir que, si $x \in R[o_1(x)]$ $x \in R[o_2(x)]$ entonces $o(x) \in R[o_1(x)]$.

El teorema anterior implica que la máxima mejora de performance alcanzable usando el enfoque de MOE se logra cuando $R[o_1(x)] \cap R[o_2(x)] = \phi$. Un ejemplo de esto es cuando cada uno de los clasificadores es capaz de identificar una clase distinta de vectores de entrada.

La suposición de que todas las señales implicadas toman valores binarios es fundamental para este teorema. En caso de que no fuera cierto y las señales tomen valores continuos en $[0, 1]$ se puede encontrar un contraejemplo, en el cual el MOE puede clasificar bien aunque los dos expertos se equivoquen.

Ejemplo

Sea $o_1(x) = [0,1 \ 0,5 \ 0,4]$, $o_2(x) = [0,5 \ 0,1 \ 0,4]$ $z(x) = [0 \ 0 \ 1]$. Por lo tanto tomando pesos iguales para todas las clases $g_{jm} = 0,5$. Entonces $o(x) = [0,1g_{11} + 0,6g_{21} \ 0,6g_{12} + 0,1g_{22} \ 0,2(g_{13} + g_{23})]$, por lo tanto $o(x) = [0,3 \ 0,3 \ 0,4]$ lo cual es una clasificación correcta.

Los autores de [4] presentan además otro teorema que pretende justificar que si los dos expertos clasifican correctamente, el MOE también lo hará. A continuación se presenta el mencionado teorema y se mostrará que la afirmación es incorrecta salvo en el caso binario.

Teorema 3 Sea $o_i \in [0, 1]$. Entonces $R[o_1(x)] \cap R[o_2(x)] \subseteq R[o(x)]$.

La demostración presentada en [4] es incorrecta porque supone que la función de peso de cada experto es igual para todas las clases. Un contraejemplo de esto es considerar $o_1(x) = [0,1 \ 0,5 \ 0,4]$, $o_2(x) = [0,1 \ 0,5 \ 0,4]$ y $z(x) = [0 \ 1 \ 0]$ Si además los pesos luego del entrenamiento son $g_1 = [0,1 \ 0,1 \ 0,5]$ y $g_2 = [0,1 \ 0,1 \ 0,5]$ la salida del MOE sería $o_2(x) = [0,01 \ 0,05 \ 0,4]$, es decir que estaría clasificando mal al vector de entrada. Si los vectores fueran binarios este problema no se presentaría porque por más pequeños que sean los pesos correspondientes a las clases correctas, todos los demás valores son cero, por lo cual nunca pueden superar a los correctos.

Basados en los dos teoremas anteriores los autores de [4] afirman que en el caso que ambos expertos clasifican bien un latido, no es necesario usarlo en el entrenamiento. En el caso del presente trabajo esto no se realizará dado que, como se dijo antes, los teoremas no son válidos.

Como se desprende de lo mencionado anteriormente sobre ambos teoremas, en el caso continuo (que es el de interés) no se puede afirmar nada sobre la performance del

MOE en base a la de los expertos.

De todas maneras, si el entrenamiento se realiza de forma correcta, es decir *moviendo* los pesos de forma tal de dar más peso a las clasificaciones correctas, y menos a las incorrectas, se puede lograr un aumento de la performance de la clasificación de cada uno de los expertos.

Algoritmo de Entrenamiento

El objetivo del algoritmo entrenamiento es estimar los pesos v_{ic} óptimos, considerando que los clasificadores $o_i(x)$ están fijos. Sea $X = x_1 \dots x_K$ el conjunto de entrenamiento utilizado para buscar las funciones de peso óptimas g_{1m} y g_{2m} en el sentido de minimizar el error cuadrático medio definido como

$$E_m = \frac{1}{2K} \sum_{k=1}^K \|z_{kc} - o(v_{1m}, v_{2m}, x_k)\|^2 \quad (4.42)$$

Cabe aclarar que ésta no es la única forma de entrenar este algoritmo. Existen otras técnicas, como la utilización de mínimos cuadrados ([43]) que en algunos casos puede ser igualmente apropiado para el entrenamiento. Pero el algoritmo más interesante, desde el punto de vista teórico, es el presentado por Moerland en [43] y por Jacobs en [42] donde se parte de una medida distinta de error, basada en el principio de máxima similitud de los datos de entrada. Usando esta medida de error, se proponen dos alternativas, el descenso según el gradiente clásico (con esta nueva medida de error), y el algoritmo de *Expectation Maximization* (EM). Este último se ha estudiado en varios trabajos presentes en la literatura ([42], [44],[43] y [45]). La ventaja de este algoritmo frente al descenso por el gradiente es que la estimación de cada uno de los parámetros de los expertos se realiza en forma *desacoplada* de los otros.

Jacobs ([42]) propone además una estrategia de *Divide and Conquer*, organizando la mezcla de expertos jerárquicamente. Todas estas variantes mencionadas requieren tener una buena estimación de la probabilidad *a posteriori* de que un experto i , clasifique el vector de entradas x como de clase m . Esto supone aproximar dicha probabilidad por un modelo Gaussiano o Multinomial entre otros. Como los datos de este problema no se ajustan a estos modelos, esta aproximación no resulta una tarea sencilla. Por esta razón en el caso del presente trabajo se optó por el enfoque (más sencillo) presentado en [4]. A continuación se desarrollará el algoritmo de entrenamiento utilizado.

Para encontrar un mínimo (local) de (4.42) se puede encontrar calculando el descenso por el gradiente de según

$$v_{ic}(t+1) = v_{ic}(t) - \eta(t) \nabla_{V_i} E_m \quad (4.43)$$

donde $V_i = (v_{i1} \dots v_{iC})$ y M es el número de clases.

El cálculo que permite llegar a la expresión exacta de $\nabla_{V_i} E_m$ se muestra en el Apéndice C. El resultado es

$$\nabla_{V_i} E_m = \frac{1}{K} \sum_{k=1}^K \sum_{m=1}^M x_k \bullet [o(x_k) - z_{km}] \bullet f'(h_m(x_k)) \bullet [o_{im}(x_k) - o_{jm}(x_k)] \bullet g_{1m}(x_k) \bullet g_{2m}(x_k) \quad (4.44)$$

Como se puede ver en el resultado anterior, se presentan todos los vectores de características, se acumula el error durante toda la *epoch* (suma en los K vectores de entrenamiento) y luego se actualizan los pesos. Otra opción podría ser realizar una actualización por iteración de los pesos, es decir actualizar los v_{ij} con cada vector de entrenamiento presentado. En este caso la suma en K desaparece y se obtiene

$$\nabla_{v_{ic}} E_m = x_k \bullet [o(x_k) - z_{kc}] \bullet \text{diag}\{f'(h(x_k))\} \bullet [o_{ic}(x_k) - o_{jc}(x_k)] \bullet g_{1m}(x_k) \bullet g_{2m}(x_k) \quad (4.45)$$

Tomando en cuenta (C.6) se puede ver que $\nabla_{v_1} E_m = -\nabla_{v_2} E_m$, lo cual muestra que la superficie de decisión entre los vectores de parámetros v_{jm} es el hiperplano $(v_{1m} - v_{2m})^t x = 0$.

4.7.3. Implementación

El algoritmo MOE se implementó discriminando los latidos en 6 clases (con los mismos criterios presentados en la sección 1.3) combinando el LE entrenado con 6 clases en los 2.5 primeros minutos de cada registro y el GE entrenado con el primer grupo de latidos completos (*los 100*) de la base de datos, como se explicó en la sección 1.3 y se ilustra en la tabla 4.10. Como se mencionó anteriormente (sección 4.7.2), la salida de cada experto debe ser un vector cuyos componentes representan el grado de proximidad del vector presentado a cada una de las clases. Dado que ambos clasificadores entregan como resultado el número de clase del vector de entrada, es necesario realizar una estimación de la proximidad de dicho vector a cada clase. Esta estimación se realizó en función de la distancia media. Es decir, primero se buscó la neurona de cada clase más cercana al vector x , una vez determinada la neurona n_j se calcula la distancia σ_m normalizada por la distancia media a todos los vectores de entrenamiento¹⁸ y se estima el grado de proximidad a la clase mediante una Gaussiana de la forma

$$o_m = e^{-\alpha \frac{\|x - y_j\|^2}{2\sigma_m^2}} \quad (4.46)$$

donde α es un parámetro de ajuste.

4.7.4. Entrenamiento

Para el entrenamiento del MOE se utilizaron los segundos 2.5 minutos de cada registro de forma tal de no exceder los 5 minutos estipulados por la AAMI (1.3). Durante el período de ajuste ambos expertos se mantienen fijos y se les presentan los vectores de características. Luego, la salida de ambos expertos (utilizando un valor de α de 0.01), junto con el vector de entrada, se le pasa al MOE y se ajustan los pesos

¹⁸Es importante destacar que la distancia media calculada es desde la neurona a los vectores de entrenamiento de los expertos y no a los vectores de evaluación. Por lo tanto debe estar calculada *a priori* y disponible junto con los pesos de cada neurona. Para darle generalidad, la distancia media se calculó como el promedio de las distancias medias de las regiones de entrenamiento del GE.

Clase	N	S	V	F	/	Q	Total
No.Lat	29.356	1.035	3.872	830	1.240	8	36.341
Se_i	98,79 %	47,34 %	91,35 %	73,13 %	99,84 %	0,00 %	95,96 %
$+P_i$	97,19 %	94,78 %	92,30 %	71,08 %	96,27 %	0,00 %	95,96 %

Tabla 4.15: Performance por clase del MOE de CARDIDENT sobre la base de datos

según la regla 4.43. Inicialmente los pesos de la *gating network* se establecieron como los centroides de las clases de los vectores de entrada. Los mismos se presentaron repetidamente en forma *determinística*, y se actualizaron los pesos una vez por *epoch*. El *learning rate* usado para dicha actualización fue de 1, y se mantuvo constante con las iteraciones. El criterio de parada utilizado fue que el cambio relativo en el error (definido en 4.42) fuera menor a 10^{-5} , o que el número de *epochs* llega a un máximo de 1200.

4.7.5. Resultados

Al igual que con los expertos local y global, se evaluó la performance del MOE después de los primeros 5 minutos, sobre los mismos registros que la evaluación del GE (tabla 4.10). Se evaluó de acuerdo a los criterios presentados en la evaluación del Global Expert, utilizando las magnitudes definidas en la sección 4.4.4.

En la tabla 4.15 se puede apreciar la performance por clase de la mezcla de expertos. La sensibilidad global del MOE fue de 95.96 % (frente a 95.51 % del LE¹⁹ y 83.54 % del GE sobre los mismos registros) y el valor predictivo positivo de 95.96 % (frente a 95.58 % y 87.34 %).

Una vez más podemos apreciar que los tipos más comunes de latidos son los que presentan una sensibilidad alta, y una predictividad positiva un poco menor. Y las menos frecuentes presentan en comportamiento opuesto. Además de lo dicho anteriormente, son válidos todos los análisis realizados anteriormente para los expertos.

Como se puede ver en la tabla 4.16 aunque la sensibilidad y la predictividad positiva aumentaron, cuando se discrimina por clases se puede ver que algunas clases tuvieron el comportamiento opuesto. Los latidos normales y los ventriculares aumentan y el resto de las clases disminuyen su desempeño.

En la tabla 4.17 se muestran los resultados globales de la clasificación discriminados por registro. Como se puede ver en dicha tabla, el MOE no siempre mejora la performance de ambos expertos. Esto se debe a que muchas veces el segmento de registro usado para el entrenamiento no es muy representativo. En el único caso que debería aparecer una tendencia clara a mejorar ambas performances es en el caso de que se evalúe sobre los 2.5 minutos usados para el entrenamiento.

A pesar de esto este algoritmo presenta una propiedad interesante: actúa como regulador entre ambos expertos. Es decir, cuando uno de ellos presenta una performance muy baja el MOE no permite que baje demasiado. Un ejemplo de este fenómeno se puede ver en la tabla 4.19, donde se ve que en el registro 201 ambos expertos presentan un desempeño muy pobre (sensibilidad 81.26 % y 71.20 %) y sin embargo

¹⁹La sensibilidad del LE de 96.93 % y valor predictivo de 95.51 % presentados en la tabla 4.6 es sobre todos los registros de la base, no sólo sobre los de evaluación del MOE.

Clase	No Lats	Se			+P		
		LE	GE	MOE	LE	GE	MOE
N	29356	98,46 %	87,70 %	98,79 %	96,83 %	94,21 %	97,19 %
S	1035	50,43 %	24,44 %	47,34 %	95,26 %	24,73 %	94,78 %
V	3872	86,00 %	87,65 %	91,35 %	90,71 %	68,43 %	92,30 %
F	830	85,78 %	6,51 %	73,13 %	69,46 %	2,96 %	71,08 %
/	1240	99,92 %	73,71 %	99,84 %	99,44 %	92,98 %	96,27 %
Q	8	0,00 %	12,50 %	0,00 %	x	0,44 %	0,00 %
Total	36341	95,51 %	83,54 %	95,96 %	95,58 %	87,34 %	95,96 %

Tabla 4.16: Comparación de performance del MOE y los expertos (por clase).

el MOE presenta un mejor desempeño (86.72%). Otro ejemplo es el registro x_232 donde, a pesar del que el GE presenta una sensibilidad de 23.45%, el MOE mantiene la sensibilidad de 99.67% que tenía el LE.

Otra posible causa de que el aumento en el desempeño no sea importante es la forma de entrenamiento utilizada. La medida de la performance no es derivable y por eso se optó en el entrenamiento por usar la medida (4.42) de la distancia entre la salida actual y la deseada. Es decir que una disminución en el error no implica automáticamente un aumento en el desempeño del clasificador, aunque es razonable que en la mayoría de los casos así sea.

4.7.6. Comparación con otros algoritmos

En esta sección se comparará el GE desarrollado en CARDIDENT con el que desarrollaron Hu *et al.* en [4].

A Patient Adaptable ECG Beat Classifier using a Mixture of Experts Approach [4]

En la tabla 4.20 se muestran los resultados obtenidos utilizando los mismos criterios que Hu *et al.* (4.6.5) y se comparan con los presentados por dicho trabajo. En términos globales CARDIDENT presenta un mejor desempeño que el logrado por Hu *et al.* en todas las magnitudes medidas. Se debe notar que el desempeño global de la mezcla de expertos de CARDIDENT es superior al de ambos expertos, lo cual no es cierto para el trabajo de Hu *et al.* Sin embargo cuando se analizan las diferentes medidas por registro se pueden notar algunos fenómenos interesantes. En el caso del CR los resultados de CARDIDENT son mayores en todos los registros con excepción del 201, donde el resultado de [4] es un 3% mayor. En la sensibilidad, en cambio, aparecen resultados más interesantes dado que el desempeño sobre la base es más variado. En varios registros (201, 203, 214 y x_223) la performance del presente trabajo es menor que la de HU *et al.*. El registro 201 en particular tiene una sensibilidad de 45.5%, que se debe a que las performances de ambos expertos son 0% y 20.7% respectivamente.

Los valores *NaN* que aparecen en la tabla se deben a la ausencia de latidos ventriculares en estos registros, ya que no se contó con la base de datos completa.

Reg.		Total	Reg.		Total	Reg.		Total
200	No.Lat	2168	212	No.Lat	2285	x_223	No.Lat	433
	Se	98,29 %		Se	100,00 %		Se	92,15 %
	+P	96,94 %		+P	100,00 %		+P	96,65 %
201	No.Lat	1521	213	No.Lat	2700	x_228	No.Lat	347
	Se	86,72 %		Se	94,15 %		Se	100,00 %
	+P	86,64 %		+P	93,97 %		+P	100,00 %
202	No.Lat	1871	214	No.Lat	1878	x_230	No.Lat	332
	Se	95,83 %		Se	98,14 %		Se	100,00 %
	+P	94,89 %		+P	98,11 %		+P	100,00 %
203	No.Lat	2481	215	No.Lat	2795	x_231	No.Lat	212
	Se	93,99 %		Se	99,93 %		Se	100,00 %
	+P	94,99 %		+P	99,86 %		+P	100,00 %
205	No.Lat	2201	217	No.Lat	1845	x_232	No.Lat	307
	Se	99,23 %		Se	85,47 %		Se	99,67 %
	+P	98,82 %		+P	89,58 %		+P	99,68 %
207	No.Lat	1932	219	No.Lat	1773	x_233	No.Lat	505
	Se	93,79 %		Se	99,10 %		Se	99,60 %
	+P	88,79 %		+P	98,71 %		+P	99,21 %
208	No.Lat	2437	x_220	No.Lat	345	x_234	No.Lat	459
	Se	96,35 %		Se	99,42 %		Se	100,00 %
	+P	96,39 %		+P	99,42 %		+P	100,00 %
209	No.Lat	2518	x_221	No.Lat	420			
	Se	92,06 %		Se	100,00 %			
	+P	92,68 %		+P	100,00 %			
210	No.Lat	2204	x_222	No.Lat	372	No.Lat		36.341
	Se	97,78 %		Se	97,85 %	Se		95,96 %
	+P	96,52 %		+P	95,75 %	P		95,96 %

Tabla 4.17: Performances totales del MOE de CARDIDENT por registro

Clase	N	S	V	F	/	Q	$\tilde{\Sigma}$
N	29002	501	206	123	0	7	837
S	6	490	21	0	0	0	27
V	188	42	3537	63	1	1	295
F	153	1	93	607	0	0	247
/	0	0	12	36	1238	0	48
Q	7	1	3	1	1	0	13
$\tilde{\Sigma}$	354	545	335	223	2	8	1467

Tabla 4.18: Clasificación del MOE por clase. Las clases anotadas en la base se muestran horizontalmente y la clasificación hecha por el experto verticalmente.

Reg.		LE	GE	MOE	Reg.		LE	GE	MOE
200	No.Lat	2168	2168	2168	217	No.Lat	1845	1845	1845
	S	98,34 %	87,92 %	98,29 %		S	85,96 %	69,59 %	85,47 %
	+P	96,99 %	96,65 %	96,94 %		+P	79,93 %	86,07 %	89,58 %
201	No.Lat	1521	1521	1521	219	No.Lat	1773	1773	1773
	S	81,26 %	71,20 %	86,72 %		S	99,04 %	75,47 %	99,10 %
	+P	71,16 %	70,92 %	86,64 %		+P	98,66 %	97,86 %	98,71 %
202	No.Lat	1871	1871	1871	x_220	No.Lat	345	345	345
	S	96,85 %	73,44 %	95,83 %		S	99,42 %	98,84 %	99,42 %
	+P	94,26 %	95,98 %	94,89 %		+P	99,42 %	98,85 %	99,42 %
203	No.Lat	2481	2481	2481	x_221	No.Lat	420	420	420
	S	95,00 %	68,40 %	93,99 %		S	99,76 %	84,76 %	100,00 %
	+P	95,54 %	90,26 %	94,99 %		+P	99,76 %	100,00 %	100,00 %
205	No.Lat	2201	2201	2201	x_222	No.Lat	372	372	372
	S	96,41 %	99,27 %	99,23 %		S	97,85 %	93,82 %	97,85 %
	+P	92,95 %	98,97 %	98,82 %		+P	95,75 %	98,57 %	95,75 %
207	No.Lat	1932	1932	1932	x_223	No.Lat	433	433	433
	S	93,89 %	82,35 %	93,79 %		S	93,07 %	29,56 %	92,15 %
	+P	88,88 %	83,14 %	88,79 %		+P	96,71 %	74,30 %	96,65 %
208	No.Lat	2437	2437	2437	x_228	No.Lat	347	347	347
	S	94,67 %	81,86 %	96,35 %		S	100,00 %	98,56 %	100,00 %
	+P	94,77 %	87,88 %	96,39 %		+P	100,00 %	99,71 %	100,00 %
209	No.Lat	2518	2518	2518	x_230	No.Lat	332	332	332
	S	92,57 %	93,09 %	92,06 %		S	100,00 %	99,10 %	100,00 %
	+P	92,91 %	93,20 %	92,68 %		+P	100,00 %	100,00 %	100,00 %
210	No.Lat	2204	2204	2204	x_231	No.Lat	212	212	212
	S	98,05 %	79,81 %	97,78 %		S	100,00 %	98,58 %	100,00 %
	+P	97,10 %	95,63 %	96,52 %		+P	100,00 %	100,00 %	100,00 %
212	No.Lat	2285	2285	2285	x_232	No.Lat	307	307	307
	S	100,00 %	97,94 %	100,00 %		S	99,67 %	23,45 %	99,67 %
	+P	100,00 %	100,00 %	100,00 %		+P	99,68 %	84,60 %	99,68 %
213	No.Lat	2700	2700	2700	x_233	No.Lat	505	505	505
	S	92,48 %	88,37 %	94,15 %		S	99,60 %	92,48 %	99,60 %
	+P	94,29 %	89,68 %	93,97 %		+P	99,21 %	99,20 %	99,21 %
214	No.Lat	1878	1878	1878	x_234	No.Lat	459	459	459
	S	97,55 %	71,78 %	98,14 %		S	100,00 %	100,00 %	100,00 %
	+P	97,44 %	99,43 %	98,11 %		+P	100,00 %	100,00 %	100,00 %
215	No.Lat	2795	2795	2795	No.Lat		36341	36341	36.341
	S	99,89 %	99,50 %	99,93 %		S	95,51 %	83,54 %	95,96 %
	+P	99,82 %	99,46 %	99,86 %		P	95,58 %	87,34 %	95,96 %

Tabla 4.19: Comparación de las performances totales del MOE con la de los expertos (por registro).

reg	Hu et al.					CARDIDENT				
	No lat	SE	SP	PP	CR	No lat	SE	SP	PP	CR
200	2605	66,9 %	93,3 %	80,6 %	81,0 %	2168	99,1 %	99,8 %	99,6 %	99,6 %
201	1858	99,5 %	99,3 %	94,5 %	95,4 %	1521	45,5 %	99,4 %	91,8 %	92,4 %
202	2038	94,4 %	72,1 %	3,0 %	71,7 %	1871	93,3 %	97,4 %	22,2 %	97,3 %
203	2882	91,0 %	90,8 %	62,2 %	87,4 %	2481	88,7 %	95,6 %	78,3 %	94,6 %
205	2491	33,8 %	100,0 %	100,0 %	97,1 %	2201	93,9 %	100,0 %	100,0 %	99,8 %
207	2215	20,0 %	98,7 %	54,7 %	88,3 %	1932	97,1 %	99,9 %	99,8 %	99,3 %
208	2823	95,2 %	93,4 %	87,5 %	90,5 %	2437	99,0 %	98,4 %	96,9 %	98,6 %
209	2860	0,0 %	99,9 %	0,0 %	98,7 %	2518	100,0 %	100,0 %	100,0 %	100,0 %
210	2502	70,3 %	98,6 %	79,0 %	93,4 %	2204	87,9 %	99,7 %	95,4 %	98,8 %
213	3071	14,2 %	99,8 %	83,8 %	91,9 %	2700	64,1 %	99,3 %	87,4 %	96,7 %
214	2134	89,1 %	99,7 %	97,7 %	98,2 %	1878	85,0 %	99,9 %	99,5 %	98,2 %
215	3172	97,9 %	98,5 %	76,4 %	98,4 %	2795	100,0 %	100,0 %	99,2 %	100,0 %
219	2156	16,4 %	99,9 %	90,0 %	97,4 %	1773	82,4 %	100,0 %	100,0 %	99,5 %
x_221	2302	98,9 %	100,0 %	100,0 %	99,0 %	420	100,0 %	100,0 %	100,0 %	100,0 %
x_223	2480	91,9 %	95,7 %	83,4 %	94,4 %	433	53,7 %	100,0 %	100,0 %	95,6 %
x_228	1978	100,0 %	99,9 %	99,7 %	99,9 %	347	100,0 %	100,0 %	100,0 %	100,0 %
x_230	2303	0,0 %	99,5 %	0,0 %	99,1 %	332	NaN	100,0 %	NaN	100,0 %
x_231	1877	50,0 %	100,0 %	100,0 %	99,8 %	212	NaN	100,0 %	NaN	100,0 %
x_233	2936	93,5 %	99,5 %	98,8 %	97,8 %	505	100,0 %	99,5 %	98,5 %	99,6 %
x_234	2577	66,7 %	100,0 %	100,0 %	99,8 %	459	NaN	100,0 %	NaN	100,0 %
Tot.	49260	82,6 %	97,1 %	77,7 %	94,0 %	31187	91,1 %	99,2 %	94,2 %	98,3 %

Tabla 4.20: Comparación de performance del MOE desarrollado con el de Hu *et al.* utilizando la medida de performance este último

4.8. Conclusiones

Los algoritmos desarrollados en este capítulo presentan desempeños superiores a las referencias bibliográficas consultadas. Los resultados muestran que la elección del algoritmo de aprendizaje competitivo *selectivo* (CSL) para la implementación de los clasificadores fue un acierto. Este algoritmo superó la performance de otros algoritmos de aprendizaje tanto para la implementación de algoritmos supervisados como no supervisados.

El agrupador desarrollado en CARDIDENT mostró una sensibilidad del 99.00 % y un valor predictivo positivo de 98.98 %, ligeramente superior a los desempeños reportados en trabajos similares ([1], [2]).

Los expertos local y global también presentan desempeños superiores a trabajos de clasificación presentes en la literatura. El experto local logró excelentes resultados, sin embargo la necesidad de que el segmento de entrenamiento sea representativo persiste. Por otra parte, el desempeño del experto global mostró que contar con una base de datos extensa, con la cual no se pudo contar en este trabajo, es un requisito para obtener resultados satisfactorios. Vale la pena mencionar que la elección de los registros de entrenamiento y evaluación fue en detrimento del desempeño del mismo, ya que para entrenar se utilizó el grupo que no presenta características particulares, dejando el resto para la evaluación.

La mezcla de expertos desarrollada funcionó como regulador de la clasificación que realizaba cada experto y presentó una performance superior a cada uno de ellos. Este aumento, sin embargo, fue poco significativo ya que en la mayoría de los casos fue menor al 1 %. El trabajo de referencia de *Hu et al.* ([4]) no alcanza el nivel logrado

en CARDIDENT, dado que presenta resultados inferiores en todas las magnitudes consideradas (SE, PP, etc.). Además, la mezcla de expertos de *Hu et al.* muestra desempeños inferiores a los de uno de los expertos.

A pesar de los prometedores resultados existe una serie de limitaciones en el enfoque expuesto, que en el capítulo 6 se analizan y se proponen posibles soluciones.

Capítulo 5

IMPLEMENTACIÓN DE CARDIDENT

“He who hasn’t hacked assembly language as a youth has no heart. He who does as an adult has no brain.”

John Moore

5.1. Introducción

Los algoritmos desarrollados en este trabajo pueden ser divididos en dos grupos. Aquellos que forman la implementación de las tres etapas de este trabajo (Detección, Extracción y Clasificación) y aquellos cuya función es el entrenamiento de redes neuronales utilizadas por los algoritmos del primer grupo y que por consiguiente están destinados a funcionar de manera *Off-line* desde un PC. Para este grupo de funciones alcanza con compilar el código MatLab que puede ser ejecutado independientemente en un PC estándar. El primer grupo, sin embargo, deberá funcionar de manera *On-line* desde un dispositivo portable, implementado posiblemente con un microprocesador digital.

Esto requiere, primero, la implementación de un único programa en MatLab que integre las 3 etapas de manera de validar el funcionamiento del sistema completo. Luego es necesario implementar este grupo de algoritmos, desarrollados en MatLab, en un lenguaje de bajo nivel que pueda ser compilado en cualquier procesador. Esta implementación permite además, llevar a cabo una estimación adecuada del poder de procesamiento necesario para ejecutar en línea (*On-line*) estos algoritmos.

Para esta implementación se utilizó lenguaje C que es aceptado por todos los fabricantes como lenguaje de programación en los microprocesadores digitales como por ejemplo los DSP.

Organización del capítulo

Este capítulo está dividido en dos partes. Una primer parte (sección 5.2) donde se comenta que problemas se encontraron y los resultados obtenidos con la implementación completa de CARDIDENT en Matlab. En la segunda parte se muestra que problemas hay y que decisiones se tomaron al respecto en la implementación de CARDIDENT en lenguaje C (sección 5.3) y se estudian los requerimientos de precisión y memoria (sección 5.4) y los requerimientos de tiempo y capacidad de procesamiento (sección 5.5) de esta implementación.

5.2. Implementación de CARDIDENT en Matlab

5.2.1. Introducción

El diseño de las tres etapas que componen el sistema se realizó de forma independiente, buscando en cada etapa la mejor solución posible. En esta sección se expondrán los resultados obtenidos en la integración de dichas etapas.

Para independizar los resultados de las etapas de extracción de características y clasificación se trabajó sobre las anotaciones hechas por los especialistas del MIT y no sobre las realizadas por el detector de CARDIDENT. El motivo de esto es poder evaluar el desempeño de la clasificación sin depender de los errores introducidos por la detección²⁰. Al momento de implementar esta integración se realizan algunas consideraciones. La primera proviene del hecho de que algunos complejos QRS pueden no ser detectados (FN) y por otro lado, aunque si lo sean, los picos R detectados pueden no coincidir exactamente con los anotados por el MIT. Por último, algunas formas de onda son confundidas con latidos (FP).

La sensibilidad se define como el cociente de los latidos correctamente clasificados

²⁰Esto contribuye además a lograr una comparación objetiva de los resultados con los trabajos publicados, ya que si cada uno utilizara su propio detector, los resultados se estarían comparando sobre datos distintos.

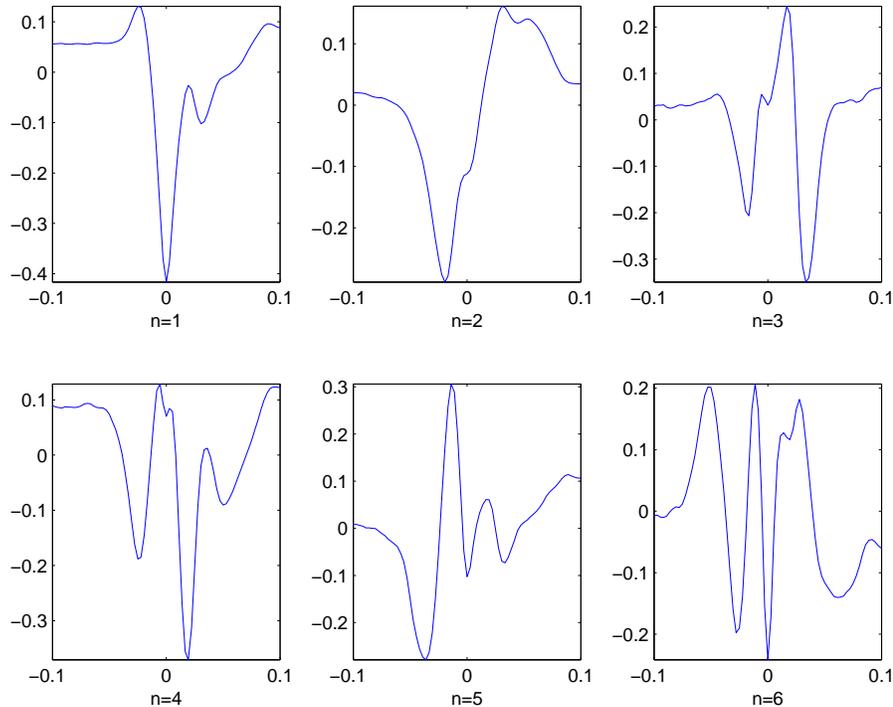


Figura 5.1: Primeras seis funciones base de la transformada KL en la integración (abscisas en seg).

entre el total de latidos, entonces los FN la afectan ya que no son clasificados pero contribuyen al total. En el caso de los FP la magnitud afectada es la predictividad positiva (+ P), dado que estas formas de onda son asignadas a una de las clases existentes y la predictividad se define como el cociente de los latidos correctamente atribuidos a una clase entre el total de asignados a la misma.

El efecto del corrimiento en los picos detectados por CARDIDENT afecta la generación de la base de Karhunen-Loève. En la figura 5.1 se puede apreciar las primeras seis bases generadas en las nuevas condiciones.

5.2.2. Resultados

La transformación y clasificación fueron aplicados sobre los datos entregados por el detector, siguiendo las mismas condiciones que las explicadas en los capítulos 3 y 4. Los resultados detallados de cada clasificador se pueden apreciar en la tabla 5.1. En la misma se observa que la performance global de la clasificación de los expertos local y global, paradójicamente, no disminuye sino que aumenta. La explicación de este hecho se puede encontrar en las tablas 5.2 y 5.3, que disgregan los resultados por clase. La presencia de falsos negativos y positivos se ve compensada por un aumento en la tasa de clasificación de los latidos. Esto se debe probablemente a que la elección de los picos R es ahora más uniforme (antes había sido hecha a mano por los especialistas) lo que puede ayudar en la comparación de latidos similares.

Sin embargo, el desempeño de la mezcla de expertos no sufre este efecto. La merma en la performance es pequeña, por lo que los resultados de la integración de las tres

etapas se consideran satisfactorios.

5.3. Implementación de CARDIDENT en C

Para el desarrollo del código en C y la utilización de la base de datos del MIT-BIH [9] se utilizó la Biblioteca WFDB de funciones C provista con el "*WFDB Software Package*" [16]. Esta biblioteca permite un acceso fácil y limpio a señales digitalizadas, anotadas y almacenadas en diversos formatos. Si bien estas funciones fueron originalmente diseñadas para ser usadas con señales de electrocardiogramas disponibles en algunas bases de datos como la utilizada en este trabajo, la biblioteca ha evolucionado para permitir su uso en un gran número de bases de datos que incluyen todo tipo de señales biológicas como por ejemplo presión sanguínea, respiración, electroencefalogramas y por supuesto ECG. Con el paquete de programas ("*WFDB Software Package*") se incluye el software necesario para evaluar analizadores de ECG de acuerdo a los protocolos estándar de evaluación: ANSI/AAMI EC38:1998 y ANSI/AAMI EC57:1998 que se comentaron en el capítulo 1. Uno de los aportes del estándar EC38 es la definición de la interfaz entre el diseñador de los algoritmos de análisis y el evaluador, de manera que ninguno tenga que develar información acerca de sus algoritmos. Esta interfaz es el archivo de anotaciones.

El archivo de anotaciones (*Annotation file*) contiene una etiqueta para cada latido y para ciertos eventos presentes en las señales. Estas etiquetas se almacenan en orden y el "tiempo" de una etiqueta se refiere a la muestra de la señal con la que la misma está asociada. La biblioteca WFDB incluye funciones C (`getann` y `putann`) que permiten leer y escribir etiquetas (anotaciones). En el programa C, las etiquetas aparecen como estructuras de datos que contienen una campo de 32 bits `time` junto a un par de campos de 8 bits que permiten codificar el tipo y subtipo de etiqueta (`anntyp` y `subtyp`, el primero listado en la tabla 1.1). También esta disponible un campo `aux` de largo variable utilizado para guardar texto. Estas estructuras son guardadas en archivos de anotaciones en forma compacta, ocupando un promedio de 16 bits por etiqueta.

Por más información acerca de la evaluación de analizadores de ECG y los archivos de anotaciones, se puede consultar el Apéndice B: "Evaluating ECG Analyzers" de la *WFDB Application Guide* que es parte de la documentación del *WFDB Software Package*.

5.3.1. Solución Propuesta

La estructura planteada para la implementación de los algoritmos en C es la misma que se explica a lo largo del trabajo. Por ello este apéndice se limitará a explicar como fueron resueltos los problemas que dependían exclusivamente de una implementación de bajo nivel.

Este programa funciona *On-line*, por lo que es necesario separar la función de detección que toma la señal de entrada y que debe trabajar muestra a muestra y las funciones de extracción de características y clasificación que procesan complejos QRS y que por lo tanto deben trabajar entre latidos. El manejo de las ejecuciones de cada función debe considerar la llegada de cada muestra y procesarla, pero permitir mientras no llegan muestras la ejecución de las funciones de análisis de los latidos. Este manejo probablemente deba ser implementado mediante interrupciones al procesador,

Reg.		LE	GE	MOE	Reg.		LE	GE	MOE
200	Nro.Lats	2167	2167	2167	217	Nro.Lats	1845	1845	1845
	Se	97,51 %	87,45 %	96,86 %		Se	85,15 %	81,03 %	84,55 %
	+P	96,07 %	96,73 %	96,69 %		+P	77,99 %	88,55 %	85,92 %
201	Nro.Lats	1521	1521	1521	219	Nro.Lats	1773	1773	1773
	Se	81,46 %	76,33 %	88,36 %		Se	99,32 %	92,50 %	99,32 %
	+P	72,62 %	79,42 %	90,31 %		+P	98,93 %	98,15 %	98,93 %
202	Nro.Lats	1871	1871	1871	x_220	Nro.Lats	345	345	345
	Se	96,79 %	73,38 %	96,47 %		Se	99,42 %	98,84 %	99,42 %
	+P	94,05 %	96,01 %	93,90 %		+P	99,42 %	98,85 %	99,42 %
203	Nro.Lats	2481	2481	2481	x_221	Nro.Lats	420	420	420
	Se	95,89 %	66,10 %	95,53 %		Se	100,00 %	95,48 %	100,00 %
	+P	96,33 %	83,25 %	96,25 %		+P	100,00 %	100,00 %	100,00 %
205	Nro.Lats	2201	2201	2201	x_222	Nro.Lats	372	372	372
	Se	96,37 %	98,55 %	99,05 %		Se	97,85 %	92,47 %	96,24 %
	+P	93,08 %	98,95 %	98,69 %		+P	95,75 %	98,45 %	95,71 %
207	Nro.Lats	1856	1856	1856	x_223	Nro.Lats	433	433	433
	Se	92,83 %	77,59 %	92,83 %		Se	91,45 %	17,09 %	90,07 %
	+P	87,46 %	78,03 %	87,24 %		+P	97,01 %	69,19 %	96,34 %
208	Nro.Lats	2436	2436	2436	x_228	Nro.Lats	347	347	347
	Se	96,80 %	78,78 %	95,32 %		Se	99,14 %	93,08 %	99,14 %
	+P	96,92 %	92,00 %	95,30 %		+P	99,42 %	98,18 %	98,85 %
209	Nro.Lats	2518	2518	2518	x_230	Nro.Lats	332	332	332
	Se	92,34 %	97,06 %	88,60 %		Se	100,00 %	97,59 %	100,00 %
	+P	92,32 %	97,15 %	89,74 %		+P	100,00 %	100,00 %	100,00 %
210	Nro.Lats	2203	2203	2203	x_231	Nro.Lats	212	212	212
	Se	97,96 %	94,37 %	96,46 %		Se	100,00 %	98,11 %	99,53 %
	+P	97,35 %	94,71 %	96,16 %		+P	100,00 %	100,00 %	100,00 %
212	Nro.Lats	2284	2284	2284	x_232	Nro.Lats	307	307	307
	Se	100,00 %	88,40 %	99,61 %		Se	99,67 %	19,22 %	99,67 %
	+P	100,00 %	100,00 %	100,00 %		+P	98,71 %	76,49 %	97,76 %
213	Nro.Lats	2699	2699	2699	x_233	Nro.Lats	505	505	505
	Se	94,63 %	88,22 %	96,04 %		Se	99,21 %	92,67 %	99,21 %
	+P	94,93 %	85,27 %	96,06 %		+P	99,21 %	97,21 %	99,21 %
214	Nro.Lats	1878	1878	1878	x_234	Nro.Lats	459	459	459
	Se	98,35 %	77,85 %	98,62 %		Se	100,00 %	99,78 %	100,00 %
	+P	98,16 %	97,17 %	98,42 %		+P	100,00 %	100,00 %	100,00 %
215	Nro.Lats	2795	2795	2795		No.Lat	36260	36260	36260
	Se	99,89 %	94,81 %	99,86 %		Se	95,73 %	84,85 %	95,68 %
	+P	99,86 %	99,82 %	99,86 %		+P	95,90 %	88,34 %	95,52 %

Tabla 5.1: Desempeños por registro y globales de los expertos al integrar las tres etapas.

Clase	N	S	V	F	/	Q	FP _{DET}	$\tilde{\Sigma}$
n	28930	480	372	68	0	2	26	948
s	18	525	20	0	0	0	8	46
v	71	5	3328	36	0	3	20	135
f	309	1	48	690	0	0	2	360
/	0	0	2	30	1240	0	1	33
q	0	0	0	0	0	0	0	0
FN _{DET}	25	24	23	6	1	3		
$\tilde{\Sigma}$	423	510	465	140	1	8		

Tabla 5.2: Clasificación del LE por clase. Las clases anotadas en la base se muestran horizontalmente y la clasificación hecha por el experto verticalmente

Clase	N	S	V	F	/	Q	FP _{DET}	$\tilde{\Sigma}$
n	26096	304	484	454	10	3	14	1269
s	965	383	41	5	0	0	2	1013
v	865	302	3139	101	160	2	29	1459
f	1212	8	18	79	0	0	2	1240
/	15	1	68	36	1070	0	0	120
q	175	13	20	149	0	0	10	367
FN _{DET}	25	24	23	6	1	3		
$\tilde{\Sigma}$	3257	652	654	751	171	8		

Tabla 5.3: Clasificación del GE por clase. Las clases anotadas en la base se muestran horizontalmente y la clasificación hecha por el experto verticalmente

sin embargo esa parte del diseño del sistema ya escapa a los alcances de este trabajo. Los códigos desarrollados incluyen por lo tanto, las funciones de detección, extracción y clasificación para ser utilizadas en un sistema final y un programa que simula la interfaz con el dispositivo real (que aún no existe) que lee la señal de la base de datos, entrega las muestras una a una y recoge los resultados en un archivo de salida.

El principal problema que se encuentra al programar en este nivel es que, si bien las funciones no pueden utilizar información contenida en muestras futuras, hay que considerar que las mismas necesitan acceso a los últimos segundos de señal original y de las señales obtenidas de su procesamiento, por lo que determinar esa ventana de tiempo determinará el orden de cantidad de memoria de datos necesaria para el sistema integrado.

La manera más elegante de almacenar las muestras que se reciben y se generan con el procesamiento en un espacio finito de almacenamiento es mediante colas circulares (*buffers*). Las mismas consisten en un espacio de memoria destinado para un número dado de muestras y un puntero de carga que permite saber cuál es la última muestra recibida. Así ante cada nueva muestra recibida, se sobre-escribe la muestra más antigua (que es la siguiente a la que hasta ese momento era la muestra actual) y se avanza el puntero de carga para marcar la posición de la muestra más reciente. Un puntero de descarga señala la muestra pedida al buffer, cuando este puntero alcanza (o supera) el puntero de carga se genera un código de error debido a que se están pidiendo muestras que aún no han sido adquiridas o generadas. El tamaño del espacio

Clase	N	S	V	F	/	Q	FP _{DET}	$\tilde{\Sigma}$
n	29002	501	206	123	0	7	26	863
s	6	490	21	0	0	0	8	35
v	188	42	3537	63	1	1	20	315
f	153	1	93	607	0	0	2	249
/	0	0	12	36	1238	0	1	49
q	7	1	3	1	1	0	0	13
FN _{DET}	25	24	23	6	1	3		
$\tilde{\Sigma}$	379	569	358	229	3	11		

Tabla 5.4: Clasificación del MOE por clase. Las clases anotadas en la base se muestran horizontalmente y la clasificación hecha por el experto verticalmente

de memoria destinado al buffer determina cuantas muestras se pueden guardar y por lo tanto de cuanto tiempo atrás se dispone de señal. La determinación del tamaño de los buffers se explicará en la sección 5.3.2.

5.3.2. Determinación del tamaño de los Buffers

Para determinar el tamaño de los buffers de las distintas señales que se generan y de la señal original, es necesario considerar de cuanto tiempo atrás es la información que van a requerir los algoritmos. Además, el tamaño debe ser una potencia de dos, ya que cada buffer debe formar una cola circular en un sistema binario.

En el caso de los filtros de preprocesamiento del detector, el orden de los filtros es el que determina el tamaño. Los mismos incluyen un filtro FIR de orden 67 y un filtro IIR de orden 4, por lo que el buffer de la señal para el FIR debe ser de por lo menos 128 muestras y el que utiliza el IIR de 4 muestras. Finalmente en el preprocesamiento existe una media móvil de 54 muestras que precisa un buffer de 64 muestras (ver capítulo 2).

La señal sin *base line drift* (deriva de la línea base) que se utiliza para el extractor de características, se obtiene a partir de un filtro FIR de orden 512 (ver capítulo 3), por lo que el buffer de esta señal es de por lo menos de 512 muestras.

Ahora, durante la detección aparece un caso donde es necesario revisar información de varios segundos en el pasado. Cuando el detector efectúa una búsqueda hacia atrás (*SearchBack*, ver capítulo 2) revisa la señal de energía (que se obtiene del preprocesamiento del detector) desde varios segundos en el pasado y ubica complejos QRS en la señal original que no fueron detectados en una primera revisión. Este salto de tiempo depende del ritmo cardíaco que se viene registrando según la ecuación 2.11:

$$RR_{max} = 1,66 \times RRAV_2$$

donde $RRAV_2$ es un promedio de los últimos intervalos RR detectados que cumplen con cierto criterio de regularidad (ver sec. 2.3.3).

El peor caso se da cuando se tiene un ritmo regular muy lento donde el promedio $RRAV_2$ es aproximadamente igual al último intervalo RR antes de que el detector declare que es necesario un *Searchback* y retroceda RR_{max} muestras. Para determinar cuan lento puede ser ese ritmo para un tamaño de buffer dado que cubra ese salto, el

Largo Buffer	$RR_{min}(\frac{lat}{min})$
512	66
1024	19
2048	8

Tabla 5.5: Ritmo cardíaco mínimo que soporta el sistema para distintos tamaños de buffer

mismo puede calcularse como:

$$RR\left(\frac{lat}{min}\right) = \frac{f_s\left(\frac{samp}{seg}\right) \times 60\frac{seg}{min}}{N_{samp}\left(\frac{samp}{lat}\right) \times 1,66} \quad (5.1)$$

donde N_{samp} es el número máximo de muestras del buffer entre dos latidos.

El detector necesita que tres buffers de señal cumplan este requerimiento: El buffer de la señal original (donde se ubica el pico R), el de energía (donde se realiza el análisis para determinar donde hay un latido) y el de la señal sin *base line drift* (de donde se va a extraer el complejo QRS). El más perjudicado es el de la señal original, ya que además de tener en cuenta el *SearchBack* debe considerar los retardos introducidos tanto por el filtro del detector como por el filtro pasa-altos que elimina el *base line drift*. Por lo tanto, $N_{samp} = L_{buffer} - DELAY_{filDet} - DELAY_{filHP}$, donde $DELAY_{filDet} = 57$ muestras y $DELAY_{filHP} = 256$ muestras.

Con estos datos podemos armar la tabla 5.5, donde se ve claramente que 512 muestras son insuficientes ya que permitirían como mínimo un ritmo de 66 latidos por minuto que es un ritmo perfectamente normal. El otro extremo, 2048 muestras, permite un funcionamiento correcto del sistema para ritmos de hasta 8 latidos por minuto. Este caso, probablemente, cubre cualquier situación, pero dado que para un buffer de la mitad de tamaño (1024 muestras) el ritmo mínimo permitido para un correcto funcionamiento sigue siendo bajo (19 latidos por minuto), se opta por este último para mantener los requerimientos de memoria bajos.

5.4. Requerimientos de Precisión y Memoria

Antiguamente los procesadores trabajaban con precisión de punto fijo, lo cual obligaba a los algoritmos a intentar trabajar con múltiplos de dos e impedía la realización de filtros agudos con varios ordenes de magnitud de diferencia entre sus coeficientes. Claramente una precisión de punto fijo es insuficiente para CARDIDENT, por lo que una precisión de punto flotante es indispensable para la implementación de los filtros, principalmente el utilizado para quitar la deriva de la línea base (*base line drift*). Los algoritmos de extracción de características y clasificación también necesitan una precisión mayor a la que se consigue con sistemas de punto fijo. En consecuencia, para la implementación de los códigos en C se utilizó una precisión de punto flotante de 8 bytes²¹.

El sistema requiere una cierta cantidad de memoria para su correcto funcionamiento. Durante el tiempo de ejecución el sistema necesita disponer de los buffers, cuyo tamaño fue determinado en cuanto al número de muestras que deben aceptar. Para saber cuanta memoria medida en bytes es requerida basta con saber la precisión con

²¹double float

que se mide cada muestra. Repasando lo visto en la tabla 5.5 se puede apreciar que los tres buffers de mayor tamaño utilizarían cada uno 8Kb ($8 \times 1024 = 8192$ bytes), por lo que los tres utilizarían 24Kb (24576 bytes).

Por otro lado los resultados del análisis que efectúa el sistema deben ser almacenados en archivos de anotaciones como se explicó en la sección 5.3. Estos ocupan en promedio unos 16 bits por etiqueta (o anotación), esto corresponde a 2 bytes por complejo QRS detectado y clasificado, por lo que 100.000 latidos (aprox. 24 horas) ocuparían solamente 200Kb. Este cálculo no incluye, por supuesto, un posible almacenamiento de secciones de señal completa, sin embargo da una idea que mientras el analizador no detecte actividad anormal (o el paciente no lo solicite con un registrador de eventos) las necesidades de almacenamiento son ínfimas.

La opción de almacenar secciones de señal no fue implementada en este trabajo, sin embargo la biblioteca WFDB provee las funciones C necesarias para almacenar señal digitalizada en un formato estándar. La proporción de memoria-tiempo de señal utilizado por este formato es de aproximadamente 1Mb cada 30 minutos de registro para señales de dos canales muestreadas a 360Hz.

A modo de ejemplo un sistema de análisis implementado con CARDIDENT con capacidad de almacenar la estadística de 5 días de actividad y hasta aproximadamente 30 minutos de secciones de señal (2 canales) que el sistema o el paciente consideren necesario registrar, requeriría de 2Mb de memoria. Considerando que en 5×24 horas de señal en 2 canales, muestreados a 360Hz, ocupan 240Mb, el factor de compresión sería de 120:1 o de 240:1 si no se guardan 30 minutos de señal.

5.5. Requerimientos de Tiempo y Capacidad de Procesamiento

Los requerimientos de tiempo fueron comentados en las secciones 5.1 y 5.3.1. En ellas se detalla que los algoritmos deben funcionar en línea y que específicamente hay un grupo de algoritmos que debe funcionar muestra a muestra y otro que debe funcionar entre latidos.

El primer grupo está formado por el algoritmo de detección y los de preprocesamiento y filtrado de la señal. Este grupo de algoritmos, que como se dijo funciona entre muestras consecutivas, tiene su límite de tiempo fijado por la frecuencia a la que se encuentra muestreada la señal digitalizada. En el caso de CARDIDENT esta frecuencia es de 360 muestras por segundo.

El segundo grupo incluye los algoritmos de extracción del complejo QRS (elección de pico R y medida de parámetros temporales), extracción de características (transformada KL) y clasificación. Para este grupo el límite de tiempo para finalizar el procesamiento no está definido ya que éstos deben trabajar entre latidos consecutivos y por lo tanto el tiempo está dado por el ritmo cardíaco instantáneo. La capacidad de procesamiento del microprocesador utilizado fijará el límite superior del ritmo cardíaco instantáneo capaz de ser manejado por esta implementación de CARDIDENT.

Una métrica natural para medir la carga de trabajo computacional en algoritmos de procesamiento de señales es el número de operaciones de punto flotante (*"flops"*) necesarias. Para realizar este conteo es necesario determinar cuantas operaciones de punto flotante requieren cada operación aritmética. La tabla 5.6 [46] establece una serie de reglas para llevar a cabo esta cuenta.

Operación	Flops	Comentarios
$A[i+2]=B[j-1]+1.5*C-2$	3	Sumas, restas o multiplicaciones cuentan como 1 flop; Aritmética de índices y asignaciones en operaciones no se cuentan
$X=Y$	1	Asignaciones aisladas cuentan como 1 flop
$\text{if } (x < y) \text{ max}=2*X$	2	Las comparaciones cuentan como 1 flop
$x=y/3+\text{sqrt}(z)$	9	Divisiones y raíces cuadradas cuentan como 4 flops
$x=\text{sin}(y)-\text{exp}(z)$	17	Senos, exponenciales, etc. cuentan como 8 flops

Tabla 5.6: Reglas para contar operaciones de punto flotante[46]

Algoritmo	Sumas	Prod	Div	Comp	TOTAL
Preprocesamiento Det.	250	126	2	125	
Filtro Baseline	1024	512	0	512	
Total Grupo 1	1274	638	2	637	
Flops x Tipo de Op	1	1	4	1	
Total FLOPS G1	1274	638	8	637	2557

Tabla 5.7: Resumen de operaciones de punto flotante para los algoritmos que deben funcionar muestra a muestra

En la tabla 5.7 se puede apreciar la cantidad de operaciones de punto flotante necesarias en CARDIDENT para realizar el preprocesamiento y el filtrado de la señal de acuerdo a las reglas de la tabla 5.6. Como se dijo, el tiempo disponible para llevar a cabo estos cálculos es el disponible entre muestras: $1/360Hz = 2,88mseg$. Sin embargo, si el procesador utiliza todo el tiempo entre muestras en ejecutar los algoritmos de este primer grupo, no queda tiempo disponible para ejecutar el resto de los algoritmos, ya que las muestras siguen llegando entre latidos y deben ser procesadas. Por lo tanto, se puede ver que la velocidad de procesamiento necesaria está determinada a partir del tiempo que precisa el segundo grupo de algoritmos para procesar un latido, descontando el tiempo necesario para procesar todas las muestras que llegan entre dos latidos.

Se pueden deducir las siguientes ecuaciones para mostrar la relación entre la velocidad del procesador ($v_{\mu P}$, en millones de operaciones de punto flotante por segundo) y la

Algoritmo	Sumas	Prod	Div	Comp	Asig	\sqrt{x}	e^x	TOTAL
Detección	471	136	32	79	90	1		
Extracción de Carac.	318	963	88	85		1		
Local Expert	450	312	6	181	32	300	6	
Global Expert	2706	1812	6	1056	157	1800	6	
Mixture of Experts	174	156	18	6	12	0	18	
SubTotal Clasificación	3330	2280	30	1243	201	2100	30	
Total Grupo 2	4119	3379	150	1407	291	2102	30	
Flops x Tipo de Op	1	1	4	1	1	4	8	
Total FLOPS G2	4119	3379	600	1407	291	8408	240	18444

Tabla 5.8: Resumen de operaciones de punto flotante para los algoritmos que deben funcionar entre latidos consecutivos

$f_{cMAX} \left(\frac{lats}{seg} \right)$	$f_{cMAX} \left(\frac{lats}{min} \right)$	$v_{\mu P} \left(\frac{Mflops}{seg} \right)$
3.33	200	1.080
3.66	220	1.087
4.00	240	1.094
4.33	260	1.101
4.67	280	1.107
5.00	300	1.114

Tabla 5.9: Frecuencia cardíaca máxima contra velocidad de procesamiento

frecuencia cardíaca máxima (f_{cMAX}):

$$v_{\mu P} \left(\frac{Mflops}{seg} \right) = N_{G2} \times f_{cMAX} + N_{G1} \times f_s \quad (5.2)$$

$$f_{cMAX} \left(\frac{lats}{seg} \right) = \frac{v_{\mu P} - N_{G1} \times f_s}{N_{G2}} \quad (5.3)$$

donde N_{G1} y N_{G2} son el número de *flops* totales para cada grupo de algoritmos (ver tablas 5.7 y 5.8) y f_s es la frecuencia de muestreo de la señal digitalizada (360Hz en CARDIDENT). La tabla 5.9 muestra el resultado de evaluar la ecuación 5.2 para un rango de frecuencias cardíacas que va de $200lats/min$ a $300lats/min$ considerando un factor de error de 10%. En la misma se puede ver que en todo caso la potencia necesaria en el procesador a utilizar, estará en el orden de 1.1 millones de operaciones de punto flotante por segundo, la cual es fácilmente alcanzable por un procesador (con capacidad de realizar operaciones de punto flotante) con una velocidad de algunas decenas de MHz.

5.6. Líneas de código utilizadas

Es interesante comentar también cuantas líneas de código se generaron para llevar a cabo este proyecto. En particular se puede observar en la tabla 5.10 la cantidad de líneas que se escribieron de código para MatLab que fue el ambiente principal de desarrollo de este proyecto.

Por otro lado, la tabla 5.11 muestra la cantidad de líneas en lenguaje C escritas para la implementación de CARDIDENT en un sistema real.

5.7. Conclusiones

Este capítulo muestra los resultados de utilizar CARDIDENT en un sistema integrado y resumió los problemas de implementar CARDIDENT en un sistema real. Asimismo se comentaron algunas de las soluciones propuestas para los estos problemas para poder también estimar que requerimientos de hardware conlleva CARDIDENT. Las secciones 5.4 y 5.5 muestran claramente que estos requerimientos no son exigentes y demuestran que este sistema es factible de implementar sin necesidad de tecnología de punta ya que bastan unos pocos megabytes de memoria y algunas decenas de MHz de velocidad de procesamiento para lograr un funcionamiento en línea del sistema. La

Función		Líneas	Sin comentarios
Detección		1107	815
Extr. Carac.	K-L	303	191
	Herm.	1670	1325
	Util.	217	161
Clasific.	Agrup.	550	368
	LE	511	350
	GE	355	238
	MOE	1731	1141
	Alg.	2599	1892
	Util.	2976	2010
Integración		2611	1893
Útiles		4099	3207
Total		22881	16984

Tabla 5.10: Cantidad de líneas de código de MatLab empleadas, separadas por función. En el total se incluyen también las líneas de código creadas durante la etapa de investigación que finalmente no se usaron.

Función	Lineas
Detección	1783
Ext. de Carac.	311
Clasificación	481
Prog. Principal	314
Otros	470
Total	3359

Tabla 5.11: Cantidad de líneas generadas en código C

capacidad de trabajar con precisión de punto flotante acaso es el único requerimiento exigente.

Capítulo 6

CONCLUSIONES

“Le coeur a ses raisons que la raison ne connaît point.”

Blaise Pascal

6.1. Introducción

En el presente trabajo se investigó el problema del análisis automático de los electrocardiogramas. En particular se estudió la capacidad de algunos algoritmos de detección y clasificación de complejos QRS, y se propuso un enfoque a partir de esto. Esta investigación abarcó diversas áreas del conocimiento, en tratamiento de señales, reconocimiento de patrones, redes neuronales, programación, sistemas digitales y electrocardiografía.

Asimismo se realizó una revisión exhaustiva de la bibliografía disponible sobre el tema, lo que permitió tener una visión global de las técnicas existentes. Se adoptó como metodología de trabajo la utilización de criterios claros de evaluación de desempeño y la presentación objetiva de resultados para facilitar su reproducción, así como la comparación con los resultados publicados en la bibliografía consultada.

Finalmente, se propuso un conjunto de técnicas que es posible implementar en un sistema portable con microprocesadores de potencia y consumo limitado.

Organización del Capítulo

En las secciones 6.2, 6.3 y 6.4 se resumen las conclusiones y los resultados de las tres etapas principales de este proyecto para presentar los resultados de este trabajo. Un breve análisis de la integración de estas etapas y la implementación de CARDIDENT se hace en la sección 6.5. Finalmente en la sección 6.6 se detallan posibles mejoras y líneas de investigación para continuar.

6.2. Detección

El detector de complejos QRS es una etapa básica para el funcionamiento del sistema completo. La correcta y precisa ubicación de los complejos es fundamental para una buena clasificación. El diseño requiere, entonces, obtener un algoritmo eficiente que funcione *on-line* y que obtenga resultados similares a los publicados de manera de reducir el factor de error que introduce una detección errónea o imprecisa. Para ello, el detector desarrollado en este trabajo se basó en el trabajo de Pan y Tompkins ([14]), que es considerado estándar en la literatura. Además de incorporar las modificaciones sugeridas por Afonso ([13]), se re-diseñaron los filtros y especialmente se introdujo una nueva manera de ubicar con mayor precisión los picos R en la señal, lo cual permite que los complejos QRS estén centrados en la ventana correctamente. Con estas modificaciones se obtuvo desempeños de detección ligeramente superiores a los de ambos trabajos, con lo cual se cumplieron satisfactoriamente los objetivos de esta etapa.

6.3. Extracción de características

El objetivo de la extracción de características es doble: reducir la dimensión del problema y construir una representación de los latidos en un espacio adecuado para la clasificación. El primero de ellos hace que los algoritmos de clasificación sean más robustos frente al ruido, además de reducir el costo de cálculo y almacenamiento. El segundo permite que en dicho espacio la distancia entre latidos similares se minimice y entre latidos distintos se maximice. En este trabajo se utilizaron los algoritmos usuales en la literatura, Hermite y Karhunen-Loève. La investigación en esta etapa

se centró en el ajuste de la cantidad de coeficientes, la incorporación de parámetros temporales, y la importancia relativa asignada a cada característica. El desempeño de ambas técnicas es similar desde el punto de vista de la representación de la señal, la diferencia es el costo de cálculo necesario para cada una: la construcción de la base de Hermite se debe realizar en línea y la de K-L se realiza fuera de línea. La comparación de performance en este sentido tiene como único fin la validación de estas técnicas para su posterior uso en la clasificación, y no para decidir cuál es más adecuada para el sistema. Para este último fin, el criterio de comparación utilizado en este trabajo fue medir su desempeño según los resultados obtenidos por una clasificación sobre cada una de las representaciones en idénticas condiciones. Nuevamente, con este criterio, se arribó a la misma conclusión: ambas transformadas tienen desempeños comparables y por lo tanto son igualmente apropiadas para la tarea propuesta. Se optó por la transformada K-L debido no sólo a su ligera ventaja en cuanto a desempeño sino también a su bajo costo computacional (en línea). En cuanto a los parámetros temporales, se pudo comprobar la importancia de agregar las medidas del intervalo RR que son fundamentales al momento de clasificar latidos prematuros. En el caso de la medida del ancho del QRS, su utilidad fue muy limitada debido a que en ambas transformadas ya se cuenta (implícita o explícitamente) con información sobre el mismo.

6.4. Clasificación

En esta etapa se concentra el mayor esfuerzo de investigación y el aporte más importante de este trabajo a la solución de este problema. El objetivo planteado en esta etapa fue investigar nuevas técnicas de clasificación para el problema de clasificación de ECG. En este sentido se logró un gran avance debido a la incorporación del algoritmo CSL, y el posterior desarrollo del algoritmo SCSL. A los autores de este trabajo les sorprendió que el algoritmo CSL sea muy poco utilizado en la literatura (apenas un trabajo de todos los consultados) y nunca fue utilizado para clasificación de ECG.

Previamente a la incorporación de este algoritmo se realizó una comparación exhaustiva de la mayoría de los algoritmos competitivos utilizados habitualmente en problemas de reconocimiento de patrones. Como resultado de esta comparación se concluyó que el algoritmo CSL era superior en la gran mayoría de los casos. Una vez agotada esta etapa, se desarrolló un agrupador con el fin de probar el algoritmo CSL en el problema de análisis de ECG. Los resultados obtenidos con este algoritmo fueron excelentes y estuvieron a la altura de todos los trabajos consultados sobre el tema. Estos resultados tienen varias lecturas: primero, tiene valor en si mismo como algoritmo de *clustering* de ECG. Segundo la performance obtenida lo valida para su posterior uso en un clasificador. Y tercero, permite evaluar y seleccionar una de las transformadas para el mismo fin.

El desarrollo de un agrupador tiene interés, además de los mencionado anteriormente, desde el punto de vista práctico debido a que libera a un especialista de realizar análisis repetitivos de señales y le permite enfocarse en las características importantes. También tiene algunos usos accesorios, pero no menos importantes como la reducción de tamaño en el conjunto de entrenamiento del experto global (GE) y la asistencia en el análisis de los primeros cinco minutos de la señal. Para la implementación de los clasificadores se desarrolló una variante supervisada del algoritmo CSL, llamada en este trabajo SCSL de forma que tal que en el entrenamiento, para el ajuste de los

pesos, las neuronas de clases distintas sean *invisibles*. De esta manera se minimiza la distorsión por clase global. Después de esto se ajustan las fronteras de decisión por vecino más cercano con un algoritmo de cuantización vectorial competitivo. Esta modificación incorporada fue validada sobre la base del MIT-BIH logrando muy buenos resultados tanto en el experto local como en el global, con guarismos comparables y hasta superiores en algunos casos a la literatura consultada. A pesar de esto el resultado del clasificador global es muy pobre, lo cual lo hace inviable para su uso clínico. Esto muestra varios aspectos del problema para su implementación en un sistema real de monitoreo y análisis de ECG: primero, la necesidad de contar con una base de datos mucho más amplia en número de latidos y variedad de sujetos. Segundo que por amplia que sea la base de datos, siempre será insuficiente, y por lo tanto se torna imprescindible la necesidad de adaptación al paciente. Este hecho se termina de confirmar con la implementación de la mezcla de expertos, la cual a pesar de no representar una mejora sustancial de performance, presenta propiedades interesantes de regulación de ambos clasificadores y un enorme potencial de investigación para lograr aumentos aún más significativos de performance.

6.5. Integración e implementación

Los resultados de la integración de las tres etapas fueron coherentes con lo obtenido en cada una de ellas, lo que valida la separación propuesta del problema en capas. Por esta razón, se considera que el desempeño global del sistema es satisfactorio. Los algoritmos a ejecutarse en tiempo real escritos en lenguaje C, mostraron la factibilidad de ser implementados en un dispositivo móvil. Esto es debido a que los requerimientos de memoria y procesamiento son alcanzables con hardware de mediano porte.

6.6. Mejoras

El enfoque presentado anteriormente presenta algunas limitaciones que restringen su utilización en aplicaciones médicas reales. En esta sección se presentarán diversas ideas que podrían contribuir a mejorar los resultados de este trabajo y algunas modificaciones imprescindibles para su utilización en un producto de uso clínico.

- Para realizar un sistema confiable y con fines clínicos es necesario llevar a cabo una validación mucho más exigente que la realizada, en particular es imprescindible medir la performance de cada una de las etapas sobre una base de datos mucho más amplia. Un primer paso es disponer de la base de datos del MIT-BIH completa, lo cual permitiría una comparación más fiel con los trabajos de referencia. Además de la validación, cada etapa tiene particularidades que hacen necesaria además una base de datos muy extensa y variada. En primer lugar, en el detector una base extensa es necesaria para poder validarlo como analizador automático de registros de 24 hs o más. En segundo lugar, en la extracción de características, es importante porque la transformada K-L se basa en una buena estimación de la estadística de la señal, para lo cual es necesaria una cantidad importante de latidos. Y por último, como se explicó en el capítulo 4 y mostraron las bajas performances obtenidas por el clasificador global, es fundamental contar con una base de datos mucho más amplia para lograr un conjunto de entrenamiento más representativo del espacio de entrada.

- En el caso de implementar estos algoritmos en un sistema de registros de larga duración es imprescindible aumentar el período de entrenamiento de cinco minutos. En el caso de extenderlo proporcionalmente, para 24 hs. sería necesario un período de 30 minutos aproximadamente. Además es necesario crear las condiciones necesarias para poder reproducir en el período de entrenamiento una variedad de morfologías representativas del paciente.
- Una mejora importante para la clasificación es la utilización de más canales en todo el proceso. Esto evidentemente implica una mejora en la performance, dado que se dispone de más información sobre un mismo latido. Además permite detectar eventos en un canal que pueden no aparecer en otros. En general, la calidad de la señal entre los diferentes canales es variable, por lo cual la incorporación de más de un canal permitiría que el sistema presentara una mayor inmunidad frente al ruido.
En [3] se utiliza esta idea en un agrupador multicanal, y se puede ver que en la implementación de esta técnica es necesario incorporar una medida de calidad (o nivel de ruido) en cada uno para poder pesar ambas clasificaciones con esa medida. Una desventaja que presenta este enfoque es el aumento del costo de cálculo asociado al procesamiento en paralelo de varios canales, sin embargo, en vista de los resultados mostrados en el apéndice 5 y el constante aumento en la potencia de cálculo de los procesadores, la implementación de esta clase de algoritmos es viable.
- Una modificación fundamental en el caso de implementarse un algoritmo de clasificación en un producto de uso clínico es disponer de criterios de rechazo de latidos. Cuando se presenta un latido muy diferente (es decir, muy lejano a las neuronas de la red) es necesario declararlo como desconocido y no incluirlo en una de las clases existentes, porque el clasificador no tiene ninguna certeza sobre la pertenencia del latido a alguna de las clases. Esto se torna fundamental para el experto local, donde en general aparecen morfologías que no estuvieron presentes en el período de entrenamiento. Otro caso en el cual es necesario determinar un criterio de rechazo es cuando las distancias de un latido a dos clases difieren muy poco, por lo que el clasificador no puede estar seguro a cual de las dos pertenece. En el trabajo [6] se muestran resultados obtenidos aplicados criterios de rechazo similares a los mencionados aquí.
- Dado que la tarea de clasificar manualmente los primeros cinco minutos es tediosa e insume mucho tiempo al especialista, se puede implementar un agrupador sobre ese segmento para facilitar la tarea de etiquetado de los latidos. En cinco minutos de señal a un ritmo cardíaco normal aparecen aproximadamente 350 latidos, los cuales se pueden agrupar en un número de clases adecuado y mucho menor que la cantidad de latidos, por ejemplo 25 que es un número adecuado para representar todas las clases existentes en la base de datos del MIT-BIH. Se obtuvo con estos parámetros una sensibilidad de 99.33 % con un agrupador como el descrito en la sección 4.4 sobre los primeros 5 minutos de cada registro.
- En la versión implementada, el algoritmo CSL es adaptivo en el número de neuronas por clase, pero el número total permanece constante. Como se pudo ver en las diferentes pruebas, la performance se ve alterada significativamente según

el número total de neuronas. Por lo tanto es razonable pensar que permitiendo variar el número total de neuronas, se puedan lograr performances más altas en los clasificadores.

En la literatura existen varios trabajos en torno a esta idea, por ejemplo [47] y [3]. En la implementación de un algoritmo de esta clase se corre el riesgo de que el número de neuronas crezca indefinidamente, por lo cual es necesario implementar un mecanismo de control.

- Una mejora interesante es continuar la investigación sobre otros algoritmos de entrenamiento para la mezcla de expertos, para intentar un aumento más significativo del desempeño. De los algoritmos existentes en la literatura, el que presenta un mayor atractivo, desde el punto de vista teórico, es el algoritmo de Expectation Maximization (EM) presentado en los trabajos [44] y [45]. La ventaja de este algoritmo frente al descenso por el gradiente es que la estimación de cada uno de los parámetros de los expertos se realiza en forma *desacoplada* de los demás.

- Siguiendo con la mezcla de expertos, en [42] se presenta una estructura jerárquica de MOE, que presenta la ventaja de subdividir el problema en varios problemas de clasificación más sencillo. Una forma de implementar esta clase de algoritmos es clasificar los latidos por etapas, primero utilizar un clasificador que divida el espacio en un número reducido de clases y luego un clasificador para cada una de ellas que las sub-divida en clases más especializadas. Por ejemplo una primera etapa que divida entre ventriculares y no-ventriculares, y posteriormente discriminarlos en las sub-clases restantes. Desde el punto de vista del costo computacional, este enfoque en general, insume tiempos de convergencia en los entrenamiento mucho menores.

- Para finalizar con la mezcla de expertos, un tema crucial a resolver es buscar una mejor estimación de la probabilidad a posteriori asignada a cada uno de los expertos. Esto es fundamental ya que esta medida es utilizada independientemente de la elección del algoritmo de entrenamiento.

- Por último, al apreciar las tablas de operaciones de punto flotante del capítulo 5 es claro que el prefiltrado de la señal consume gran parte de los recursos computacionales. Como fue mencionado en el capítulo 3, una alternativa de menor costo computacional es emplear un *spline cúbico* para eliminar el desplazamiento de la línea de base. Otra alternativa para disminuir el número de operaciones necesarias es bajar la frecuencia de muestreo, que a 360Hz es excesiva tanto como registro de la señal como para realizar la extracción de características.

Apéndice A

MANUAL DE USUARIO CARDIDENT DEMO

A.1. Introducción

El programa CARDIDENT DEMO permite analizar los resultados obtenidos en este trabajo en un entorno gráfico de fácil uso. De esta manera el usuario puede elegir el registro de la base de datos que desea analizar y ver el desempeño de los algoritmos sobre el propio electrocardiograma, al poder navegar entre los errores que comete el algoritmo, ver la energía de la señal (cuando se analiza la detección) o ver la distribución de los latidos en un subespacio tridimensional del espacio de extracción de características. También da la posibilidad de generar un reporte de performance para el registro bajo análisis.

A.2. Instalación

El programa se instala desde la unidad de CD-ROM. Para ello, inserte el disco del programa y ejecute el programa `setup.exe` que se encuentra en el disco.

El programa de instalación lo guiará por el proceso de instalación. Cuando termine reinicie y busque el icono CARDIDENT en su escritorio o en los grupos de programa del menú de inicio.

La primera vez que ejecute el programa, éste le pedirá que le indique la ruta de acceso al ejecutable del Internet Explorer. Este archivo se llama `iexplore.exe` y generalmente se encuentra en `C:\Archivos de Programa\Internet Explorer` aunque esta ubicación puede variar de acuerdo al idioma de su versión de Windows.

A.3. Estructura del Programa

En la ventana principal el usuario elige si el desea evaluar la detección o la clasificación. En la figura A.1(a) se ve esta ventana, donde se presenta un breve resumen de cada algoritmo.

Antes de empezar es necesario elegir la configuración de los algoritmos que se va a utilizar. Si bien hay una única opción (configuración *cardidentv100*), esto permite darle capacidad al programa para en un futuro poder elegir entre distintas pruebas o versiones del trabajo.

A continuación se presentan los pasos para abrir la evaluación del Detector o la del Clasificador:

1. Ir al Menu Archivo y elegir la opción Abrir
2. Elegir la configuración “*cardidentv100*” con el archivo `cardidentv100.mat`(figura A.1(b))
3. Elija la Evaluación
 - a) Detector: Presione Evaluar Detección
 - b) Clasificador: Presione Evaluar Clasificación

A.3.1. Evaluar Detección

La ventana de evaluación del detector (figura A.2) permite recorrer el electrocardiograma, y ver donde se ubican los latidos (y el tipo), donde se ubica la detección



(a) Elección de Evaluadores y Menú Archivo

(b) Menú Archivo/Abrir

Figura A.1: Ventana Principal

realizada por el algoritmo y sus errores. Además permite ver el canal 2 o bien la energía de la señal en esa ventana de tiempo. Los ejes horizontales miden el número de muestras (200 muestras por marca) y el eje vertical del ECG se encuentra en mV.

1. Ventana 1: ECG (Canal 1)
 - a) Tipos de Latido según la Referencia MIT-BIH
 - b) Tipo de Error
 - c) Marcador de Referencia MIT-BIH
 - d) Marcador de Detección
 - e) Marcador de Error
2. Ventana 2: ECG (Canal 2) / Energía
3. Selector de Ventana 2
4. Barra de Desplazamiento
5. Cuadro 1:
 - a) Selector de Registro de la base MIT-BIH
 - b) Activa/Desactiva Marcadores de Referencia MIT-BIH
 - c) Activa/Desactiva Marcadores de Detección
 - d) Activa/Desactiva Marcadores de Error
 - e) Navegador de Errores
6. Cuadro 2: Navegador

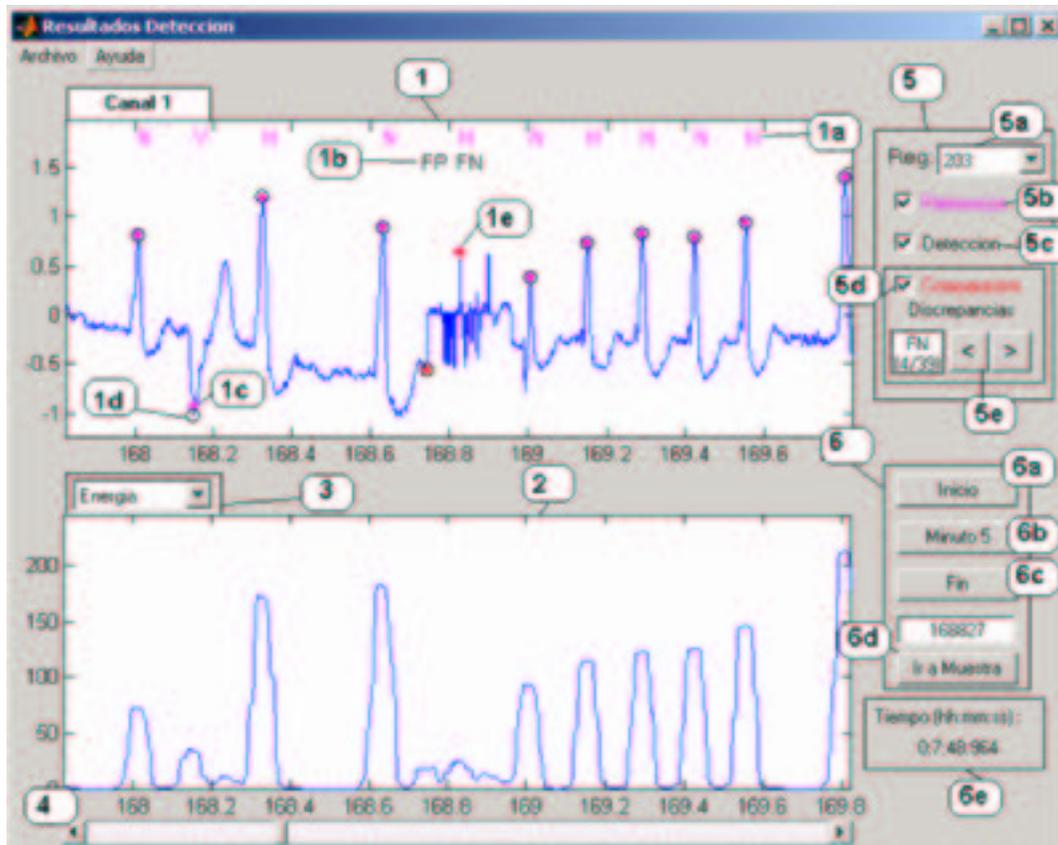


Figura A.2: Ventana de Evaluación del Detector

- a) Ir a Inicio
- b) Ir al Minuto 5
- c) Ir al Fin
- d) Navegador por Muestra
- e) Tiempo de la muestra actual en el centro de la ventana

A.3.2. Evaluar Clasificación

La ventana de evaluación del clasificador, al igual que la ventana del detector, permite recorrer el electrocardiograma y ver los tipos de latido, la clasificación en 6 clases que realiza el algoritmo y permite ver y navegar por los errores. Además da la posibilidad de elegir el clasificador, efectuar dos análisis extra y generar un reporte HTML.

El primer análisis muestra la agrupación que el algoritmo CSL realiza sobre el registro. El segundo permite ver un subespacio tridimensional del espacio de la extracción de características, donde se proyectan los vectores de cada latido del registro y las neuronas del clasificador.

La figura A.3 muestra una ventana de ejemplo:

1. Ventana ECG: Canal 1

- a) Tipos de Latido según la Referencia MIT-BIH
 - b) Tipos de Latido según el Algoritmo
 - c) Tipo de Error
 - d) Marcador de Referencia MIT-BIH
 - e) Marcador del Algoritmo
 - f) Marcador de Error
2. Selector de Algoritmo de Clasificación
 3. Barra de Desplazamiento
 4. Cuadro 1:
 - a) Selector de Registro ECG de la base MIT-BIH
 - b) Activa/Desactiva Marcadores de Referencia MIT-BIH
 - c) Activa/Desactiva Marcadores del Algoritmo
 - d) Activa/Desactiva Marcadores de Error
 - e) Navegador de Errores
 5. Cuadro 2: Navegador
 - a) Ir a Inicio
 - b) Ir al Minuto 5
 - c) Ir al Fin
 - d) Navegador por Muestra
 - e) Tiempo de la muestra actual en el centro de la ventana
 6. Cuadro 3: Análisis
 - a) Agrupador CSL sobre el registro
 - b) Visualizador de los latidos y neuronas en un subespacio configurable del espacio de extracción de características. (figura A.4)
 7. Generador de Reportes HTML (figura A.5)
 8. Leyendas: Clases y los códigos de los tipos de latidos que caen en cada clase.

A.3.3. Estructura ECG: Visualizador de latidos y neuronas

La ventana de estructura ECG (ver figura A.4) tiene la finalidad de dar una idea acerca de como se encuentran distribuidos los latidos y las neuronas del clasificador en el espacio de la extracción de características. Dado que éste es un espacio 12-dimensional no es posible mostrarlo realmente, por ello se da la opción de elegir 3 características entre los parámetros temporales y los 9 coeficientes de la transformada KL para armar un subespacio tridimensional representable gráficamente. Esto permite ver como afecta en la distribución de los latidos características como el RR que diferencia claramente los latidos normales de los prematuros. Para poder apreciar con todo detalle el subespacio elegido es posible girar los ejes con

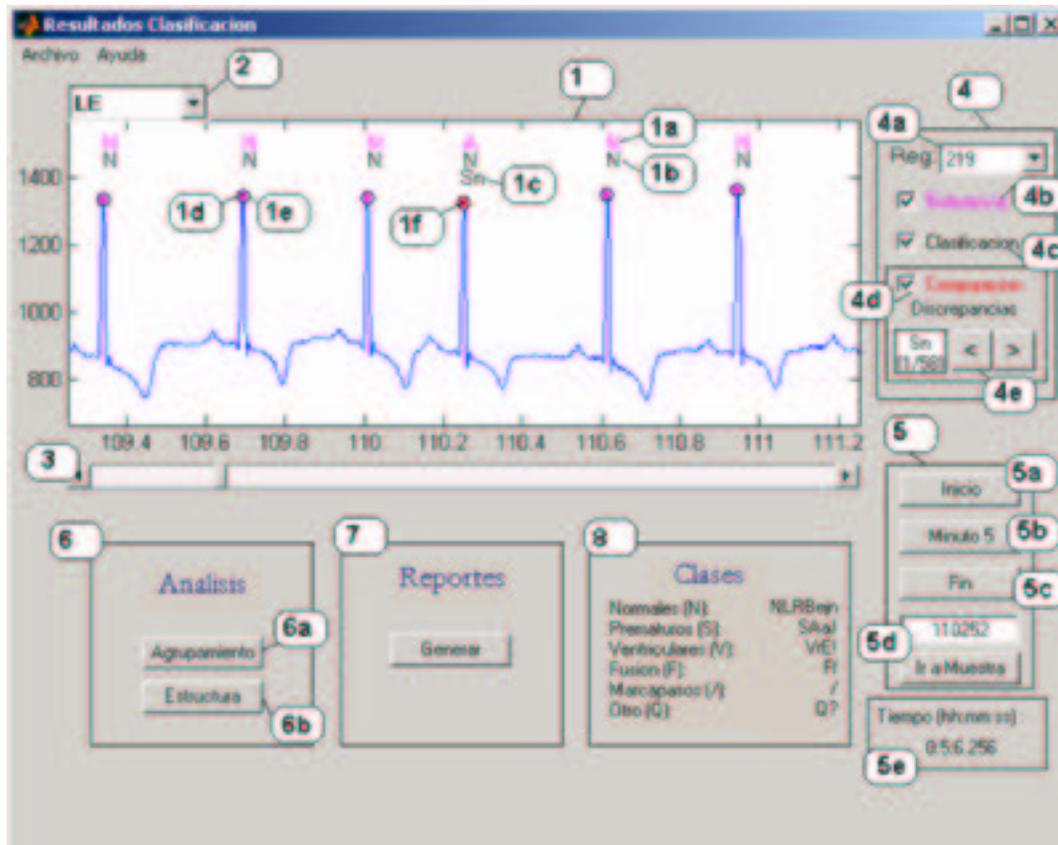


Figura A.3: Ventana Evaluador de la Clasificación

el mouse haciendo click con el botón izquierdo en algún punto del cubo y moviendo el mismo con el mouse mientras se mantiene presionado el botón.

Finalmente la ventana incluye un cuadro con una matriz de correlación donde es posible apreciar la relación estadística que existe entre cada par de características.

1. Subespacio Elegido
 - a) Latidos
 - b) Neuronas (Los puntos negros con un círculo del color de una de las clases son neuronas pertenecientes a esa clase)
 - c) Leyenda
2. Matriz de correlación
3. Escala de la matriz de correlación
4. Cuadro de elección de coeficientes
 - a) Elección del coeficiente para el Eje X
 - b) Elección del coeficiente para el Eje Y
 - c) Elección del coeficiente para el Eje Z

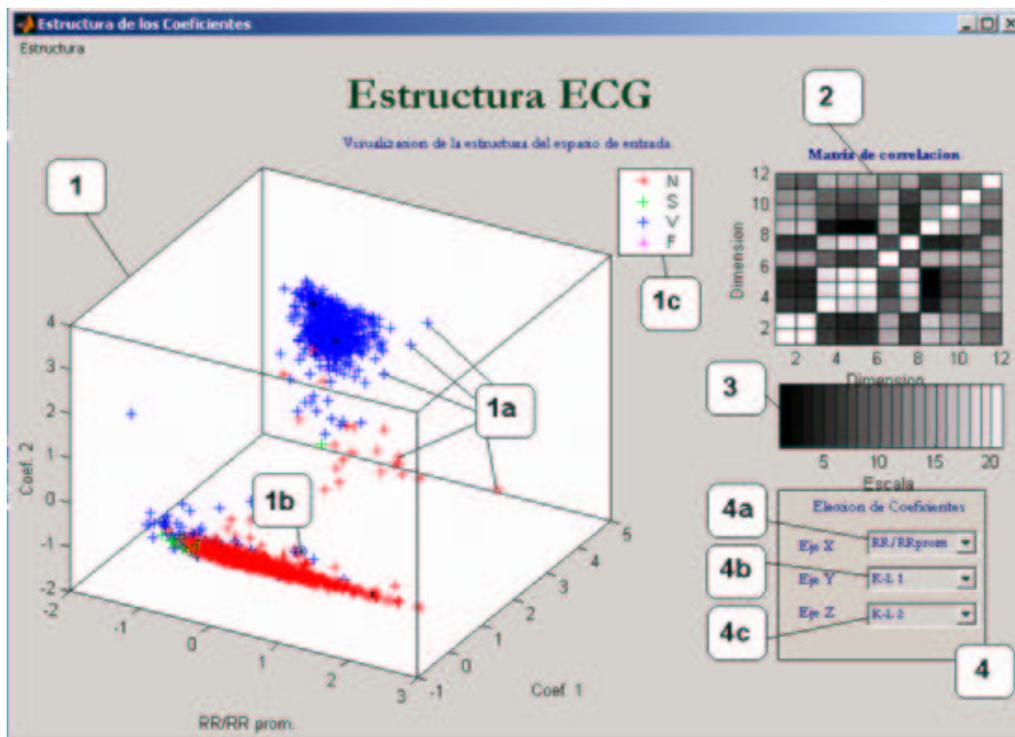


Figura A.4: Estructura para Visualización del espacio de los latidos y neuronas

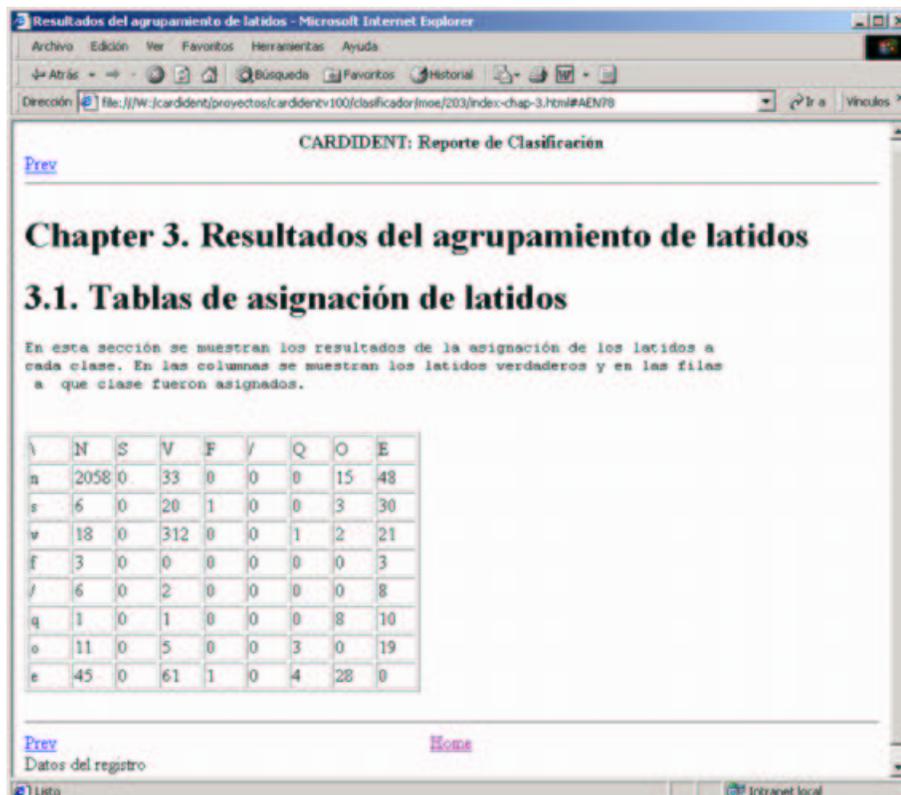


Figura A.5: Ejemplo de los reportes generados

Apéndice B

TABLAS DE RESULTADOS

Tabla B.1: Performance del LE por clase y registro

Reg	Clase	N	S	V	F	/	Q	Total
100	Nro.Lats	1872	29	1	0	0	0	1902
	Se_i	99,84 %	93,10 %	0,00 %	x	x	x	99,68 %
	$+P_i$	99,89 %	87,10 %	x	x	x	x	99,65 %
101	Nro.Lats	1518	3	0	0	0	2	1523
	Se_i	100,00 %	0,00 %	x	x	x	0,00 %	99,67 %
	$+P_i$	99,67 %	x	x	x	x	x	99,34 %
102	Nro.Lats	1	0	3	32	1785	0	1821
	Se_i	100,00 %	x	66,67 %	53,13 %	98,82 %	x	97,97 %
	$+P_i$	50,00 %	x	100,00 %	44,74 %	99,16 %	x	98,17 %
103	Nro.Lats	1727	2	0	0	0	0	1729
	Se_i	100,00 %	0,00 %	x	x	x	x	99,88 %
	$+P_i$	99,88 %	x	x	x	x	x	99,77 %
104	Nro.Lats	98	0	1	563	1182	13	1857
	Se_i	100,00 %	x	0,00 %	97,34 %	99,24 %	0,00 %	97,95 %
	$+P_i$	96,08 %	x	x	96,31 %	99,41 %	0,00 %	97,54 %
105	Nro.Lats	2121	0	29	0	0	5	2155
	Se_i	99,95 %	x	100,00 %	x	x	0,00 %	99,72 %
	$+P_i$	99,81 %	x	93,55 %	x	x	x	99,50 %
106	Nro.Lats	1236	0	460	0	0	0	1696
	Se_i	99,92 %	x	99,78 %	x	x	x	99,88 %
	$+P_i$	99,92 %	x	99,78 %	x	x	x	99,88 %
107	Nro.Lats	0	0	57	0	1727	0	1784
	Se_i	x	x	91,23 %	x	100,00 %	x	99,72 %
	$+P_i$	x	x	100,00 %	x	99,71 %	x	99,72 %
118	Nro.Lats	1817	86	13	0	0	0	1916
	Se_i	99,72 %	95,35 %	30,77 %	x	x	x	99,06 %
	$+P_i$	99,83 %	87,23 %	57,14 %	x	x	x	98,98 %
119	Nro.Lats	1296	0	365	0	0	0	1661
	Se_i	100,00 %	x	100,00 %	x	x	x	100,00 %
	$+P_i$	100,00 %	x	100,00 %	x	x	x	100,00 %
200	Nro.Lats	1438	28	700	2	0	0	2168
	Se_i	99,93 %	0,00 %	99,29 %	0,00 %	x	x	98,34 %
	$+P_i$	97,76 %	x	99,57 %	x	x	x	96,99 %
201	Nro.Lats	1195	126	198	2	0	0	1521
	Se_i	100,00 %	32,54 %	0,00 %	0,00 %	x	x	81,26 %
	$+P_i$	80,96 %	91,11 %	x	x	x	x	71,16 %
202	Nro.Lats	1800	55	15	1	0	0	1871
	Se_i	99,89 %	0,00 %	93,33 %	0,00 %	x	x	96,85 %
	$+P_i$	97,56 %	x	50,00 %	x	x	x	94,26 %
203	Nro.Lats	2103	0	373	1	0	4	2481
	Se_i	96,39 %	x	88,47 %	0,00 %	x	0,00 %	95,00 %
	$+P_i$	98,11 %	0,00 %	82,29 %	x	x	x	95,54 %
205	Nro.Lats	2122	2	66	11	0	0	2201
	Se_i	100,00 %	0,00 %	0,00 %	0,00 %	x	x	96,41 %

Tabla B.1: (cont.)

Reg	Clase	N	S	V	F	/	Q	Total
207	$+P_i$	96,41 %	x	x	x	x	x	92,95 %
	Nro.Lats	1377	106	449	0	0	0	1932
	Se_i	99,93 %	0,00 %	97,55 %	x	x	x	93,89 %
208	$+P_i$	92,16 %	x	99,77 %	x	x	x	88,88 %
	Nro.Lats	1308	2	824	301	0	2	2437
	Se_i	97,40 %	0,00 %	94,17 %	85,38 %	x	0,00 %	94,67 %
209	$+P_i$	97,62 %	x	97,12 %	77,18 %	x	x	94,77 %
	Nro.Lats	2146	371	1	0	0	0	2518
	Se_i	99,77 %	51,21 %	0,00 %	x	x	x	92,57 %
210	$+P_i$	92,17 %	97,44 %	x	x	x	x	92,91 %
	Nro.Lats	2010	20	165	9	0	0	2204
	Se_i	100,00 %	0,00 %	89,09 %	44,44 %	x	x	98,05 %
212	$+P_i$	98,43 %	x	94,84 %	57,14 %	x	x	97,10 %
	Nro.Lats	2285	0	0	0	0	0	2285
	Se_i	100,00 %	x	x	x	x	x	100,00 %
213	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	2209	27	195	269	0	0	2700
	Se_i	95,70 %	85,19 %	62,56 %	88,48 %	x	x	92,48 %
214	$+P_i$	99,11 %	92,00 %	89,05 %	58,77 %	x	x	94,29 %
	Nro.Lats	1662	0	213	1	0	2	1878
	Se_i	100,00 %	x	79,81 %	0,00 %	x	0,00 %	97,55 %
215	$+P_i$	97,36 %	x	99,42 %	x	x	x	97,44 %
	Nro.Lats	2662	1	131	1	0	0	2795
	Se_i	100,00 %	0,00 %	99,24 %	0,00 %	x	x	99,89 %
217	$+P_i$	99,89 %	x	100,00 %	x	x	x	99,82 %
	Nro.Lats	235	0	141	229	1240	0	1845
	Se_i	0,00 %	x	95,04 %	93,01 %	99,92 %	x	85,96 %
219	$+P_i$	x	x	39,88 %	80,99 %	99,44 %	x	79,93 %
	Nro.Lats	1715	7	51	0	0	0	1773
	Se_i	100,00 %	0,00 %	80,39 %	x	x	x	99,04 %
x_108	$+P_i$	99,02 %	x	100,00 %	x	x	x	98,66 %
	Nro.Lats	275	0	3	1	0	0	279
	Se_i	100,00 %	x	33,33 %	0,00 %	x	x	98,92 %
x_109	$+P_i$	98,92 %	x	100,00 %	x	x	x	98,58 %
	Nro.Lats	419	0	5	0	0	0	424
	Se_i	100,00 %	x	100,00 %	x	x	x	100,00 %
x_111	$+P_i$	100,00 %	x	100,00 %	x	x	x	100,00 %
	Nro.Lats	349	0	1	0	0	0	350
	Se_i	100,00 %	x	0,00 %	x	x	x	99,71 %
x_112	$+P_i$	99,71 %	x	x	x	x	x	99,43 %
	Nro.Lats	426	0	0	0	0	0	426
	Se_i	100,00 %	x	x	x	x	x	100,00 %
x_113	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	289	3	0	0	0	0	292
	Se_i	100,00 %	100,00 %	x	x	x	x	100,00 %

Tabla B.1: (cont.)

Reg	Clase	N	S	V	F	/	Q	Total
x_114	$+P_i$	100,00 %	100,00 %	x	x	x	x	100,00 %
	Nro.Lats	256	0	22	3	0	0	281
	Se_i	100,00 %	x	100,00 %	0,00 %	x	x	98,93 %
x_115	$+P_i$	100,00 %	x	88,00 %	x	x	x	97,99 %
	Nro.Lats	319	0	0	0	0	0	319
	Se_i	100,00 %	x	x	x	x	x	100,00 %
x_116	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	373	0	29	0	0	0	402
	Se_i	100,00 %	x	100,00 %	x	x	x	100,00 %
x_117	$+P_i$	100,00 %	x	100,00 %	x	x	x	100,00 %
	Nro.Lats	254	0	0	0	0	0	254
	Se_i	100,00 %	x	x	x	x	x	100,00 %
x_121	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	306	0	0	0	0	0	306
	Se_i	100,00 %	x	x	x	x	x	100,00 %
x_122	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	415	0	0	0	0	0	415
	Se_i	100,00 %	x	x	x	x	x	100,00 %
x_123	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	256	0	1	0	0	0	257
	Se_i	100,00 %	x	0,00 %	x	x	x	99,61 %
x_124	$+P_i$	99,61 %	x	x	x	x	x	99,22 %
	Nro.Lats	233	17	20	2	0	0	272
	Se_i	100,00 %	0,00 %	0,00 %	0,00 %	x	x	85,66 %
x_220	$+P_i$	85,66 %	x	x	x	x	x	73,38 %
	Nro.Lats	322	23	0	0	0	0	345
	Se_i	100,00 %	91,30 %	x	x	x	x	99,42 %
x_221	$+P_i$	99,38 %	100,00 %	x	x	x	x	99,42 %
	Nro.Lats	339	0	81	0	0	0	420
	Se_i	100,00 %	x	98,77 %	x	x	x	99,76 %
x_222	$+P_i$	99,71 %	x	100,00 %	x	x	x	99,76 %
	Nro.Lats	364	8	0	0	0	0	372
	Se_i	100,00 %	0,00 %	x	x	x	x	97,85 %
x_223	$+P_i$	97,85 %	x	x	x	x	x	95,75 %
	Nro.Lats	370	21	41	1	0	0	433
	Se_i	99,73 %	42,86 %	60,98 %	0,00 %	x	x	93,07 %
x_228	$+P_i$	96,85 %	100,00 %	96,15 %	0,00 %	x	x	96,71 %
	Nro.Lats	251	0	96	0	0	0	347
	Se_i	100,00 %	x	100,00 %	x	x	x	100,00 %
x_230	$+P_i$	100,00 %	x	100,00 %	x	x	x	100,00 %
	Nro.Lats	332	0	0	0	0	0	332
	Se_i	100,00 %	x	x	x	x	x	100,00 %
x_231	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	212	0	0	0	0	0	212
	Se_i	100,00 %	x	x	x	x	x	100,00 %

Tabla B.1: (cont.)

Reg	Clase	N	S	V	F	/	Q	Total
x_232	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	69	238	0	0	0	0	307
	Se_i	98,55 %	100,00 %	x	x	x	x	99,67 %
x_233	$+P_i$	100,00 %	99,58 %	x	x	x	x	99,68 %
	Nro.Lats	371	0	132	2	0	0	505
	Se_i	100,00 %	x	100,00 %	0,00 %	x	x	99,60 %
x_234	$+P_i$	100,00 %	x	98,51 %	x	x	x	99,21 %
	Nro.Lats	459	0	0	0	0	0	459
	Se_i	100,00 %	x	x	x	x	x	100,00 %
	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
Nro.Lats		45212	1175	4882	1431	5934	28	58662
S		98,98 %	53,96 %	88,04 %	89,24 %	99,48 %	0,00 %	96,93 %
P		97,79 %	93,79 %	92,47 %	78,25 %	99,43 %	0,00 %	96,91 %

Tabla B.2: Performance del GE por clase y registro

Reg	Clase	N	S	V	F	/	Q	Total
200	Nro.Lats	1438	28	700	2	0	0	2168
	S_i	84,77 %	7,14 %	97,86 %	0,00 %	x	x	87,92 %
	$+P_i$	97,36 %	4,88 %	99,13 %	0,00 %	x	0,00 %	96,65 %
201	Nro.Lats	1195	126	198	2	0	0	1521
	S_i	86,95 %	1,59 %	20,71 %	50,00 %	x	x	71,20 %
	$+P_i$	86,95 %	5,88 %	16,27 %	2,86 %	0,00 %	0,00 %	70,92 %
202	Nro.Lats	1800	55	15	1	0	0	1871
	S_i	75,72 %	1,82 %	66,67 %	0,00 %	x	x	73,44 %
	$+P_i$	99,63 %	3,33 %	4,00 %	0,00 %	x	0,00 %	95,98 %
203	Nro.Lats	2103	0	373	1	0	4	2481
	S_i	63,24 %	x	98,39 %	0,00 %	x	0,00 %	68,40 %
	$+P_i$	99,48 %	0,00 %	39,50 %	0,00 %	0,00 %	0,00 %	90,26 %
205	Nro.Lats	2122	2	66	11	0	0	2201
	S_i	99,95 %	100,00 %	93,94 %	0,00 %	x	x	99,27 %
	$+P_i$	99,48 %	66,67 %	100,00 %	0,00 %	0,00 %	x	98,97 %
207	Nro.Lats	1377	106	449	0	0	0	1932
	S_i	97,02 %	0,00 %	56,79 %	x	x	x	82,35 %
	$+P_i$	95,91 %	0,00 %	63,59 %	0,00 %	0,00 %	0,00 %	83,14 %
208	Nro.Lats	1308	2	824	301	0	2	2437
	S_i	88,38 %	50,00 %	98,79 %	7,97 %	x	0,00 %	81,86 %
	$+P_i$	86,20 %	0,75 %	88,00 %	96,00 %	0,00 %	0,00 %	87,88 %
209	Nro.Lats	2146	371	1	0	0	0	2518
	S_i	99,44 %	56,33 %	100,00 %	x	x	x	93,09 %
	$+P_i$	92,98 %	94,57 %	50,00 %	x	x	x	93,20 %
210	Nro.Lats	2010	20	165	9	0	0	2204
	S_i	80,20 %	0,00 %	88,48 %	11,11 %	x	x	79,81 %
	$+P_i$	98,77 %	0,00 %	74,11 %	0,27 %	x	0,00 %	95,63 %
212	Nro.Lats	2285	0	0	0	0	0	2285
	S_i	97,94 %	x	x	x	x	x	97,94 %
	$+P_i$	100,00 %	0,00 %	0,00 %	x	x	x	100,00 %
213	Nro.Lats	2209	27	195	269	0	0	2700
	S_i	100,00 %	7,41 %	89,23 %	0,37 %	x	x	88,37 %
	$+P_i$	89,07 %	100,00 %	80,93 %	100,00 %	0,00 %	0,00 %	89,68 %
214	Nro.Lats	1662	0	213	1	0	2	1878
	S_i	68,59 %	x	97,18 %	0,00 %	x	50,00 %	71,78 %
	$+P_i$	99,56 %	x	99,52 %	0,00 %	x	33,33 %	99,43 %
215	Nro.Lats	2662	1	131	1	0	0	2795
	S_i	99,81 %	0,00 %	94,66 %	0,00 %	x	x	99,50 %
	$+P_i$	99,70 %	0,00 %	96,12 %	x	x	x	99,46 %
217	Nro.Lats	235	0	141	229	1240	0	1845
	S_i	89,79 %	x	93,62 %	11,79 %	73,71 %	x	69,59 %
	$+P_i$	30,67 %	0,00 %	96,35 %	84,38 %	95,71 %	0,00 %	86,07 %
219	Nro.Lats	1715	7	51	0	0	0	1773
	S_i	74,87 %	42,86 %	100,00 %	x	x	x	75,47 %

Tabla B.2: (cont.)

Reg	Clase	N	S	V	F	/	Q	Total
x_220	$+P_i$	99,77 %	0,80 %	47,22 %	0,00 %	x	x	97,86 %
	Nro.Lats	322	23	0	0	0	0	345
	S_i	100,00 %	82,61 %	x	x	x	x	98,84 %
x_221	$+P_i$	98,77 %	100,00 %	x	x	x	x	98,85 %
	Nro.Lats	339	0	81	0	0	0	420
	S_i	81,12 %	x	100,00 %	x	x	x	84,76 %
x_222	$+P_i$	100,00 %	0,00 %	100,00 %	0,00 %	x	x	100,00 %
	Nro.Lats	364	8	0	0	0	0	372
	S_i	93,96 %	87,50 %	x	x	x	x	93,82 %
x_223	$+P_i$	100,00 %	33,33 %	0,00 %	0,00 %	x	x	98,57 %
	Nro.Lats	370	21	41	1	0	0	433
	S_i	28,92 %	4,76 %	48,78 %	0,00 %	x	x	29,56 %
x_228	$+P_i$	78,68 %	6,25 %	71,43 %	0,00 %	x	0,00 %	74,30 %
	Nro.Lats	251	0	96	0	0	0	347
	S_i	98,41 %	x	98,96 %	x	x	x	98,56 %
x_230	$+P_i$	99,60 %	0,00 %	100,00 %	0,00 %	x	x	99,71 %
	Nro.Lats	332	0	0	0	0	0	332
	S_i	99,10 %	x	x	x	x	x	99,10 %
x_231	$+P_i$	100,00 %	0,00 %	x	x	x	x	100,00 %
	Nro.Lats	212	0	0	0	0	0	212
	S_i	98,58 %	x	x	x	x	x	98,58 %
x_232	$+P_i$	100,00 %	x	0,00 %	x	x	x	100,00 %
	Nro.Lats	69	238	0	0	0	0	307
	S_i	98,55 %	1,68 %	x	x	x	x	23,45 %
x_233	$+P_i$	31,48 %	100,00 %	0,00 %	x	x	x	84,60 %
	Nro.Lats	371	0	132	2	0	0	505
	S_i	91,11 %	x	97,73 %	0,00 %	x	x	92,48 %
x_234	$+P_i$	100,00 %	0,00 %	98,47 %	0,00 %	x	x	99,20 %
	Nro.Lats	459	0	0	0	0	0	459
	S_i	100,00 %	x	x	x	x	x	100,00 %
	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
Nro.Lats		29356	1035	3872	830	1240	8	36341
S		87,70 %	24,44 %	87,65 %	6,51 %	73,71 %	12,50 %	83,54 %
P		94,21 %	24,73 %	68,43 %	2,96 %	92,98 %	0,44 %	87,34 %

Tabla B.3: Performance del MOE por clase y registro

Reg	Clase	N	S	V	F	/	Q	Total
200	Nro.Lats	1438	28	700	2	0	0	2168
	S_i	99,93 %	0,00 %	99,14 %	0,00 %	x	x	98,29 %
	$+P_i$	97,69 %	x	99,57 %	x	x	x	96,94 %
201	Nro.Lats	1195	126	198	2	0	0	1521
	S_i	100,00 %	26,19 %	45,45 %	50,00 %	x	x	86,72 %
	$+P_i$	87,16 %	73,33 %	91,84 %	100,00 %	0,00 %	0,00 %	86,64 %
202	Nro.Lats	1800	55	15	1	0	0	1871
	S_i	98,83 %	0,00 %	93,33 %	0,00 %	x	x	95,83 %
	$+P_i$	98,45 %	x	22,22 %	0,00 %	x	x	94,89 %
203	Nro.Lats	2103	0	373	1	0	4	2481
	S_i	95,15 %	x	88,74 %	0,00 %	x	0,00 %	93,99 %
	$+P_i$	98,18 %	0,00 %	78,25 %	0,00 %	x	0,00 %	94,99 %
205	Nro.Lats	2122	2	66	11	0	0	2201
	S_i	100,00 %	0,00 %	93,94 %	0,00 %	x	x	99,23 %
	$+P_i$	99,39 %	x	100,00 %	0,00 %	0,00 %	x	98,82 %
207	Nro.Lats	1377	106	449	0	0	0	1932
	S_i	99,93 %	0,00 %	97,10 %	x	x	x	93,79 %
	$+P_i$	92,04 %	x	99,77 %	x	x	x	88,79 %
208	Nro.Lats	1308	2	824	301	0	2	2437
	S_i	99,46 %	0,00 %	99,03 %	76,74 %	x	0,00 %	96,35 %
	$+P_i$	96,30 %	x	96,91 %	96,65 %	0,00 %	0,00 %	96,39 %
209	Nro.Lats	2146	371	1	0	0	0	2518
	S_i	99,95 %	46,36 %	100,00 %	x	x	x	92,06 %
	$+P_i$	91,51 %	99,42 %	100,00 %	x	x	x	92,68 %
210	Nro.Lats	2010	20	165	9	0	0	2204
	S_i	100,00 %	0,00 %	87,88 %	0,00 %	x	x	97,78 %
	$+P_i$	98,00 %	x	95,39 %	x	x	0,00 %	96,52 %
212	Nro.Lats	2285	0	0	0	0	0	2285
	S_i	100,00 %	x	x	x	x	x	100,00 %
	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
213	Nro.Lats	2209	27	195	269	0	0	2700
	S_i	99,77 %	66,67 %	64,10 %	72,49 %	x	x	94,15 %
	$+P_i$	97,05 %	94,74 %	87,41 %	73,31 %	x	0,00 %	93,97 %
214	Nro.Lats	1662	0	213	1	0	2	1878
	S_i	100,00 %	x	84,98 %	0,00 %	x	0,00 %	98,14 %
	$+P_i$	98,11 %	x	99,45 %	x	x	0,00 %	98,11 %
215	Nro.Lats	2662	1	131	1	0	0	2795
	S_i	100,00 %	0,00 %	100,00 %	0,00 %	x	x	99,93 %
	$+P_i$	99,96 %	x	99,24 %	x	x	x	99,86 %
217	Nro.Lats	235	0	141	229	1240	0	1845
	S_i	9,36 %	x	97,87 %	78,17 %	99,84 %	x	85,47 %
	$+P_i$	100,00 %	x	61,33 %	55,59 %	97,10 %	0,00 %	89,58 %
219	Nro.Lats	1715	7	51	0	0	0	1773
	S_i	100,00 %	0,00 %	82,35 %	x	x	x	99,10 %

Tabla B.3: (cont.)

Reg	Clase	N	S	V	F	/	Q	Total
x_220	$+P_i$	99,08 %	x	100,00 %	x	x	x	98,71 %
	Nro.Lats	322	23	0	0	0	0	345
	S_i	100,00 %	91,30 %	x	x	x	x	99,42 %
x_221	$+P_i$	99,38 %	100,00 %	x	x	x	x	99,42 %
	Nro.Lats	339	0	81	0	0	0	420
	S_i	100,00 %	x	100,00 %	x	x	x	100,00 %
x_222	$+P_i$	100,00 %	x	100,00 %	x	x	x	100,00 %
	Nro.Lats	364	8	0	0	0	0	372
	S_i	100,00 %	0,00 %	x	x	x	x	97,85 %
x_223	$+P_i$	97,85 %	x	x	x	x	x	95,75 %
	Nro.Lats	370	21	41	1	0	0	433
	S_i	99,46 %	38,10 %	53,66 %	100,00 %	x	x	92,15 %
x_228	$+P_i$	96,34 %	100,00 %	100,00 %	4,76 %	x	x	96,65 %
	Nro.Lats	251	0	96	0	0	0	347
	S_i	100,00 %	x	100,00 %	x	x	x	100,00 %
x_230	$+P_i$	100,00 %	x	100,00 %	x	x	x	100,00 %
	Nro.Lats	332	0	0	0	0	0	332
	S_i	100,00 %	x	x	x	x	x	100,00 %
x_231	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	212	0	0	0	0	0	212
	S_i	100,00 %	x	x	x	x	x	100,00 %
x_232	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
	Nro.Lats	69	238	0	0	0	0	307
	S_i	98,55 %	100,00 %	x	x	x	x	99,67 %
x_233	$+P_i$	100,00 %	99,58 %	x	x	x	x	99,68 %
	Nro.Lats	371	0	132	2	0	0	505
	S_i	100,00 %	x	100,00 %	0,00 %	x	x	99,60 %
x_234	$+P_i$	100,00 %	x	98,51 %	x	x	x	99,21 %
	Nro.Lats	459	0	0	0	0	0	459
	S_i	100,00 %	x	x	x	x	x	100,00 %
	$+P_i$	100,00 %	x	x	x	x	x	100,00 %
Nro.Lats		29.356	1.035	3.872	830	1.240	8	36.341
S		98,79 %	47,34 %	91,35 %	73,13 %	99,84 %	0,00 %	95,96 %
P		97,19 %	94,78 %	92,30 %	71,08 %	96,27 %	0,00 %	95,96 %

Apéndice C

DESARROLLOS AUXILIARES

En este Apéndice se muestra el cálculo de $\nabla_{V_i} E_m$ necesario para construir la regla de actualización de pesos del MOE (sección 4.7). Recordando que el error se definió como

$$E_m = \frac{1}{2K} \sum_{k=1}^K \|z_{km} - o_m(v_{1m}, v_{2m}, x_k)\|^2 \quad (\text{C.1})$$

el gradiente se puede expresar como

$$\nabla_{V_i} E_m = \frac{\not{2}}{2K} \sum_{k=1}^K [o_m(v_{1m}, v_{2m}, x_k) - z_{km}] \bullet \nabla_{V_i} o_m(v_{1m}, v_{2m}, x_k) \quad (\text{C.2})$$

entonces

$$\nabla_{V_i} E_m = \frac{1}{K} \sum_{k=1}^K \left\{ [o_m(x_k) - z_{km}] \bullet f'(h_m(x_k)) \bullet \sum_{j=1}^2 g_{jm}(x_k) \bullet \nabla_{V_i} o_m(v_{1m}, v_{2m}, x_k) \right\} \quad (\text{C.3})$$

Por otro lado calcularemos

$$\nabla_{v_{ip}} g_{jm}(v_{1m}, v_{2m}, x_k) = \frac{\nabla_{v_{ip}} e^{v_{jm}^t x_k}}{\sum_{j=1}^2 e^{v_{jm}^t x_k}} \quad (\text{C.4})$$

$$\nabla_{v_{ip}} g_{jm}(v_{1m}, v_{2m}, x_k) = \frac{\nabla_{v_{ip}} e^{v_{jm}^t x_k} \bullet \sum_{m=1}^2 e^{v_{pm}^t x_k} - e^{v_{jm}^t x_k} \nabla_{v_{ip}} e^{v_{im}^t x_k}}{\left[\sum_{j=1}^2 e^{v_{jm}^t x_k} \right]^2} \quad (\text{C.5})$$

$$\nabla_{v_{im}} g_{jm}(v_{1m}, v_{2m}, x_k) = (-1)^{i-j} x_k \bullet g_{1m}(x_k) \bullet g_{2m}(x_k) \quad (\text{C.6})$$

Combinando las ecuaciones C.3 y C.6 se obtiene la expresión final para el gradiente

$$\nabla_{V_i} E_m = \frac{1}{K} \sum_{k=1}^K \sum_{m=1}^M x_k \bullet [o(x_k) - z_{km}] \bullet f'(h_m(x_k)) \bullet [o_{im}(x_k) - o_{jm}(x_k)] \bullet g_{1m}(x_k) \bullet g_{2m}(x_k) \quad (\text{C.7})$$

Apéndice D

ESPECIFICACIÓN INICIAL

Descripción preliminar de las tareas a realizar para satisfacer los requerimientos de la materia Proyecto, Plan 91' (60 créditos).

Introducción

Actualmente existe una necesidad real de tener equipos portátiles de análisis automático de señales electrocardiográficas (ECG). Los equipos portátiles existentes se reducen, por razones tecnológicas, al registro de la señal durante un lapso de tiempo, el cual se ve limitado por la capacidad de almacenamiento de los dispositivos. Una señal típica de 24 horas de duración consta de aproximadamente 100.000 latidos los cuales pueden ser agrupados morfológicamente en un número mucho menor de clases. En la mayoría de estas clases donde la actividad cardíaca es normal, es suficiente registrar el número de latidos que la componen y un latido representativo de la misma. En el lapso de tiempo donde la actividad cardíaca presenta anomalías o síntomas de enfermedades, es necesario el registro completo de la señal. Esto sólo es posible si el equipo portátil además de registrar la señal, también la analiza.

Dada una cierta capacidad de almacenamiento del dispositivo, el tiempo de observación de la actividad cardíaca aumenta considerablemente implementando el análisis propuesto anteriormente. La relación de tiempos de observación entre los métodos existentes y el propuesto, depende claramente del tipo de actividad cardíaca desarrollada por el paciente durante el lapso de tiempo considerado. Otra ventaja del análisis en línea es la posibilidad de alertar ante la presencia de anomalías en la señal que signifiquen riesgo para el paciente, en forma local o remota. La acción más simple es la del registro detallado del episodio de riesgo, disponible hoy en los llamados registradores de eventos cardíacos”.

Sobre la base de estas consideraciones, pensamos que el estudio e investigación de la implementación de este método de análisis puede ser de utilidad para la clínica.

Implementación

CARDIDENT realizará la extracción, clasificación e identificación en línea de complejos QRS de señales electrocardiográficas, en un proceso que puede dividirse en tres etapas: detección, extracción de características y clasificación. La primer etapa consiste en la detección de los complejos QRS, lo que se hará con algoritmos de detección en tiempo real estándares como Pan-Tompkins [14].

Existe una gran variabilidad entre las señales cardíacas de distintos pacientes, por lo que resulta sumamente costoso computacionalmente clasificarlas en el dominio temporal. Para ello se buscará representar los complejos QRS en un espacio en el cual se reduzca la dimensionalidad de los datos manteniendo sus características principales. Se investigará la viabilidad de la implementación de las transformadas más comúnmente utilizadas, como son las de Karhunen-Loève [21] y Hermite [20].

La clasificación se realizará en el espacio de las transformadas mencionadas anteriormente, mediante la implementación de algoritmos de agrupamiento (clustering”) tanto supervisados como no supervisados. Se investigarán y desarrollarán algoritmos basados en redes neuronales competitivas como los Mapas Auto-Organizativos (SOM) [35], Learning Vector Quantization (LVQ) o Competitive Selective Learning [37], así como también combinaciones de clasificadores (Mixture Of Experts, MOE) [4].

En la programación se buscará la modularidad mediante la implementación de los algoritmos en tres capas, que se corresponden con las etapas anteriormente descritas, con interfaces claramente definidas. De esta forma se independiza una etapa de otra, y permite el ensayo de diferentes algoritmos en cada etapa. En la primer etapa de investigación y desarrollo de los algoritmos se utilizará MatLab. Una vez elegidos los algoritmos de mejor performance, los mismos se implementarán en C para optimizar su funcionamiento y evaluar la potencia de procesamiento necesaria para permitir su ejecución en tiempo real.

Validación

Los resultados obtenidos se compararán con técnicas existentes en la literatura. Con este fin se trabajará con bases de datos estándar previamente clasificadas (MIT-BIH) [9], y se utilizarán los criterios de la AAMI (Association for the Advancement of Medical Instrumentation) [10] para medir la performance de clasificadores automáticos de ECG.

Tiempos y costos

Se desarrollará CARDIDENT en un lapso de un año, en las instalaciones del NIB en el Hospital de Clínicas.

Apéndice E

TIEMPOS Y COSTOS

E.1. Tiempos

Este trabajo se desarrolló a lo largo de 54 semanas entre abril de 2001 y mayo de 2002. El mismo implicó 2918 horas hombre que se detallan, discriminando por integrante del grupo y a lo largo de la duración de este proyecto, en la tabla E.1 y en la figura E.1.

En la tabla E.1 se identifica el trabajo previo al comienzo oficial como “Semana 0”. Este trabajo previo consistió básicamente en búsqueda de bibliografía y se realizó en los meses previos a Abril de 2001. Por otro lado la figura E.1 permite apreciar que durante los primeros meses el ritmo de trabajo fue irregular debido mayormente a períodos de examen y/o parciales, mientras que a partir de la semana 28 aproximadamente se logró, sin contar enero, un ritmo de trabajo de aproximadamente 30 horas semanales por persona.

Tabla E.1: Horas trabajadas por persona y por semana.

Semana	Fecha	Juan	Pablo	Nicolás	Total
0		60	60	60	180
1	23-Abr	26	17	15	58
2	30-Abr	25	22,5	21	68,5
3	07-May	3	3	12	18
4	14-May	0	0	0	0
5	21-May	16	4	7	27
6	28-May	5,5	5,5	2,5	13,5
7	04-Jun	15,5	15,5	13	44
8	11-Jun	5	7	7	19
9	18-Jun	12	10	12	34
10	25-Jun	16,5	25	29	70,5
11	02-Jul	20	19	18,5	57,5
12	09-Jul	16	15	16	47
13	16-Jul	0	0	0	0
14	23-Jul	0	0	22	22
15	30-Jul	5	0	22	27
16	06-Ago	5	0	21	26
17	13-Ago	5	0	0	5
18	20-Ago	0	0	0	0
19	27-Ago	17	13	0	30
20	03-Sep	22	18	0	40
21	10-Sep	17	13	30	60
22	17-Sep	12	12	40	64
23	24-Sep	8	8	10	26
24	01-Oct	27,5	27,5	27	82
25	08-Oct	17	0	17	34
26	15-Oct	0	0	0	0
27	22-Oct	5	19	14	38
28	29-Oct	43	33	28	104
29	05-Nov	23	23	24	70
30	12-Nov	26	26	26	78

Tabla E.1: (cont.)

Semana	Fecha	Juan	Pablo	Nicolás	Total
31	19-Nov	21	21	21	63
32	26-Nov	32	32	32	96
33	03-Dic	38	38	38	114
34	10-Dic	35	35	35	105
35	17-Dic	25	25	25	75
36	24-Dic	12	12	12	36
37	31-Dic	0	0	0	0
38	07-Ene	0	0	0	0
39	14-Ene	0	0	0	0
40	21-Ene	0	0	0	0
41	28-Ene	0	0	25	25
42	04-Feb	18	18	33	69
43	11-Feb	20	37	0	57
44	18-Feb	37	37	31	105
45	25-Feb	38	38	35	111
46	04-Mar	30	30	30	90
47	11-Mar	40	40	40	120
48	18-Mar	30	30	30	90
49	25-Mar	0	0	0	0
50	01-Abr	30	0	30	60
51	08-Abr	36	36	36	108
52	15-Abr	40	40	30	110
53	22-Abr	29	29	29	87
54	29-Abr	19	16	19	54
	Totales	983	910	1025	2918

En el trabajo realizado a lo largo de este proyecto se pueden distinguir diversas tareas, agrupadas por área, que resumen distintos aspectos del mismo:

Tareas Administrativas Incluyen todas las horas utilizadas en reuniones con el tutor y con diversos colaboradores de este proyecto.

Estudio :

Bibliografía Búsqueda y clasificación de bibliografía

Algoritmos Investigación de los algoritmos de clasificación y agrupamiento presentes en la literatura

Proyecto :

Estructura Desarrollo de la estructura general de este proyecto

Detector Desarrollo de los algoritmos de Detección

Hermite Desarrollo de la transformada Hermite y sus algoritmos

KL Desarrollo de la transformada Karhunen-Loève

Clasificador Desarrollo de los algoritmos de clasificación

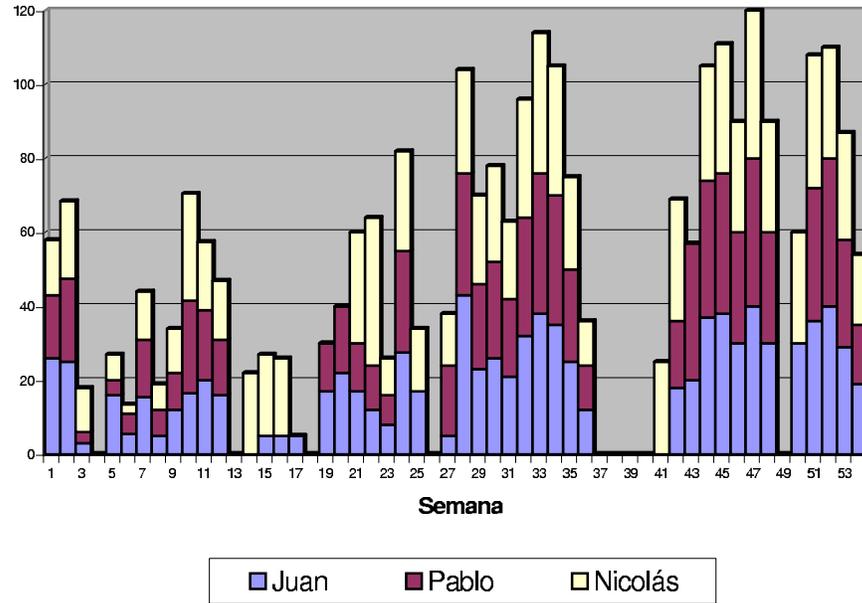


Figura E.1: Distribución de horas por Semana y por Persona

Documentación Este documento

Imp. y DEMO :

Implementación Desarrollo de la implementación de CARDIDENT en Matlab y C

DEMO Desarrollo del programa CARDIDENT DEMO

Cada uno de estos grupos requirió una cierta dedicación que se puede observar en la tabla E.2. Las figuras E.2 y E.3 muestran como se repartió el trabajo a lo largo del tiempo entre cada uno de estos grupos y la figura E.4 muestra que proporción del trabajo total significó cada tarea. En particular en esta figura se puede apreciar que el desarrollo del clasificador, junto con la investigación de los algoritmos, significó gran parte del trabajo realizado. La documentación del proyecto fue otra de las tareas que insumió un tiempo considerable.

T.Adm.	Estudio	Proyecto	Doc.	Imp. y Demo.	Total
76,5	501	1406,5	626,5	307,5	2918
3 %	17 %	48 %	21 %	11 %	100 %

Tabla E.2: Horas trabajadas discriminadas por área

E.2. Costos

Los costos de desarrollar CARDIDENT se pueden dividir en 2 clases: Equipamiento y Mano de Obra. El equipo utilizado consta de 2 computadoras de escritorio y 1 computadora portátil conectadas en red y de una impresora a chorro de tinta. La

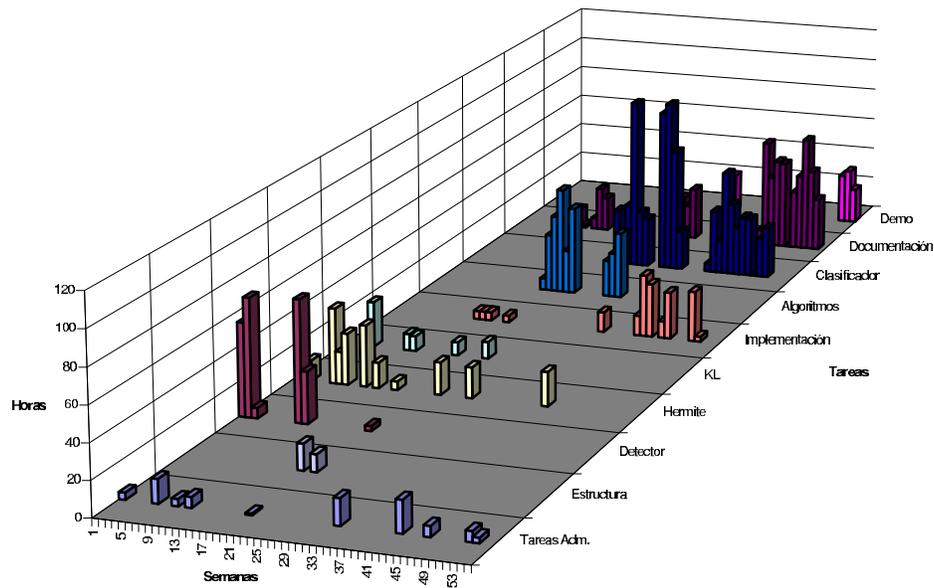


Figura E.2: Distribución de las tareas realizadas a lo largo del tiempo detallando las horas dedicadas

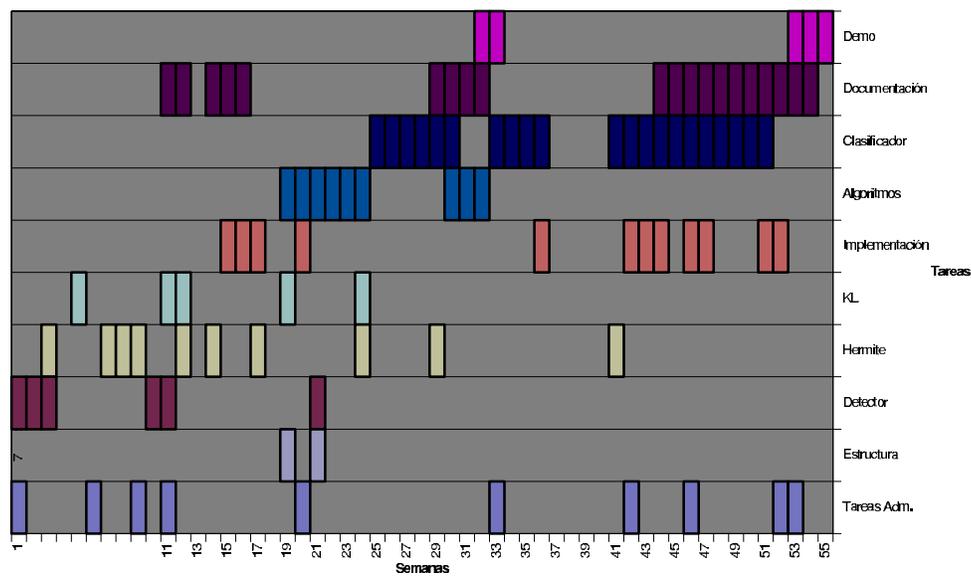


Figura E.3: Distribución de las tareas realizadas a lo largo del tiempo

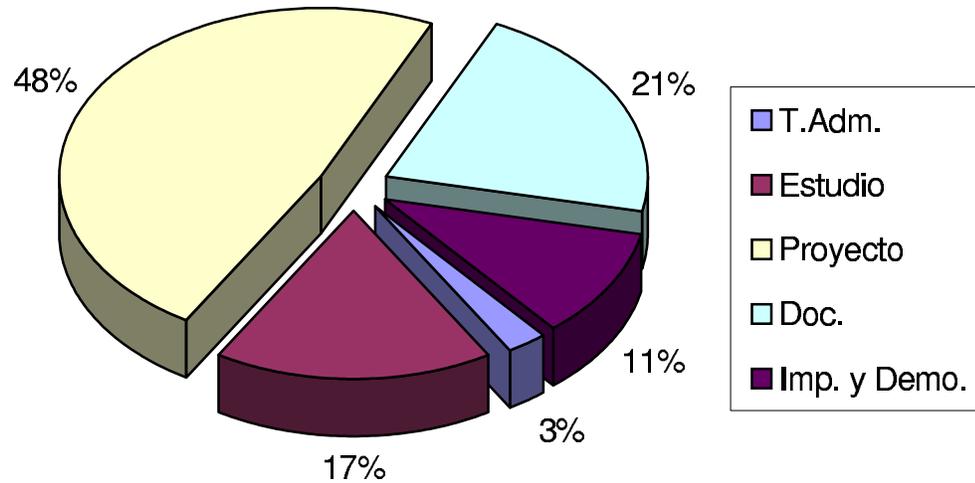


Figura E.4: Distribución de horas por área

tabla E.3 detalla las características de cada equipo así como su precio estimado en la fecha de comienzo del proyecto.

Los costos de la mano de obra se determinan a partir del costo de la hora-hombre que se establezca. A modo de ejemplo el costo horario de un Ingeniero Electricista en un entorno comercial productivo se estima en US\$20 la hora. Como se vió en la sección E.1 el desarrollo de CARDIDENT demandó 2918 horas-hombre, esto significaría un costo de mano de obra de aproximadamente US\$58360.

En la tabla E.4 se puede observar entonces que el costo total del desarrollo de CARDIDENT sería de US\$62520²².

A los costos totales de este proyecto se les deberá agregar los costos de implementar este sistema en un dispositivo real. De esta manera se podrá realizar un estudio de amortización del costo de esta investigación de acuerdo al número de unidades que se deseen producir.

²²Compartir un año de trabajo en equipo, no tiene precio

	Computadora 1	Computadora 2:
	Procesador AMD Athlon de 1GHz 256Mb de memoria RAM 40Gb de disco duro Lectora CD-ROM 56X Tarjeta de Red Ethernet 10/100	Procesador Intel III de 800MHz 384Mb de memoria RAM Dos Discos Duros: 30Gb y 10Gb Lectora/Grabadora CD-RW Lectora CD-ROM 52X Tarjeta de Red Ethernet 10/100
Precio Abril 2001	U\$\$ 1000	U\$\$ 1000
	Computadora Portátil	Equipo de Red
	Procesador AMD Athlon 4 de 1GHz 256Mb de memoria RAM 20Gb de disco duro Lectora/Grabadora CD-RW Tarjeta de Red Ethernet 10/100	Hub Ethernet 10/100 Cables Cat 5
Precio Abril 2001	U\$\$ 1700	U\$\$ 50
	Impresora	Varios
	Modelo EPSON Stylus Color 777	4 Paquetes de Hojas A4 3 cartuchos de tinta negra 1 cartucho de tinta color
Precio Abril 2001	U\$\$ 200	U\$\$ 210
	Total	U\$\$ 4160

Tabla E.3: Equipo utilizado en el proyecto y su precio estimado a Abril del 2001

	Costo (U\$\$)
Equipo	4160
Mano de Obra	58360
Total	62520

Tabla E.4: Costos de CARDIDENT

Bibliografía

- [1] M. Lagerholm; C. Peterson; G. Braccini; L. Edenbrandt; L. Sörnmo. Clustering ECG complexes using hermite functions and self-organizing maps. *Submitted to IEEE Trans. on Neural Networks*, 1998.
- [2] S. Barro; M. Fernandez-Delgado; J.A. Vila-Sobrino; C. V. Regueiro; and E. Sanchez. Classifying multichannel ECG patterns with an adaptive neural network. *IEEE Eng. in Med. and Bio.*, January/February 1998:45–55.
- [3] M. Fernandez-Delgado and S. Barro. Mart: A multichannel ART-based neural network. *IEEE Trans. on Neural Networks*, 9:139–150.
- [4] Y. H. Hu; S. Palreddy; and W. J. Tompkins. A patient-adaptable ECG beat classifier using a mixture of experts approach. *IEEE Trans. on Biomed. Eng.*, Vol. 44, No. 9:891–900, 1997.
- [5] R. L. Watrous; and G. Towell. A patient-adaptive neural network ECG patient monitoring algorithm. *Computers in Cardiology, Vienna, Austria*, pages 229–232, September 10-13, 1995.
- [6] R. Silipo; and C. Marchesi. Artificial neural networks for automatic ECG analysis. Unpublished Draft. Dept. Sistemi e Informatica, Università di Firenze, Italy, 1995.
- [7] R. L. Watrous; G. Towell; M. S. Glassman; M. S. Shahraray and D. Theivanayagam. Synthesize, optimize, analyze, repeat (SOAR): Application of neural network tools to ECG patient monitoring. *Proceedings of the Third International Congress on Air- and Structure-Borne Sound and Vibration, Montreal*, pages 997–1004, 1994.
- [8] Y. H. Hu; W. J. Tompkins; J. L. Urrusty; V. X. Alfonso. Applications of artificial neural networks for ecg signal detection and classification. *Journal Electrocardiology*, 1994.
- [9] Massachusetts Institute of Technology. MIT-BIH Database Distribution. <http://www.physionet.org/physiobank/database/mitdb/>.
- [10] AAMI Association for the Advancement of Medical Instrumentation. <http://www.aami.org>.
- [11] G. B. Moody. ECG database programmer's guide. Harvard-Massachusetts Institute of Technology (MIT) Division of Health Science and Technology, 1989.
- [12] N.V. Thakor; J.G. Webster; W.J. Tompkins. Optimal QRS detection. *Medical and Biological Engineering*, pages 343–350, 1983.
- [13] W.J.Tompkins; V.X.Afonso. *Biomedical Digital Signal Processing: C-Language Examples and Laboratory Experiments for the IBM PC*, chapter 12, pages 236–264. Englewood Cliffs,NJ: Prentice-Hall, 1993.

- [14] J.Pan; W.J.Tompkins. A real-time QRS detection algorithm. *IEEE Trans. Biomed. Eng.*, BME-32:230–236, 1985.
- [15] F.Sarutte;J.Alfaro;T.Forcellati. CLASICAR: Clasificador de complejos QRS. Proyecto de fin de carrera, Facultad de Ingeniería, UdelaR, Marzo 2001.
- [16] PhysioNet. WFDB Software Package. <http://www.physionet.org>.
- [17] V.X. Afonso; W.J. Tompkins; T.Q. Nguyen; Shen Luo. ECG beat detection using filter banks. *IEEE Tran. on Biomedical Engineering*, 46(2):192–202, Feb. 1999.
- [18] S.Kadambe; R.Murray; G.F. Boudreaux-Bartels. Wavelet transform-based QRS complex detector. *IEEE Tran. on Biomedical Engineering*, 46(7):838–848, Jul. 1999.
- [19] T. Kohama; S. Nakamura; H. Hoshino. An efficient R-R interval detection for ECG monitoring system. *IEICE Trans. Inf. and Syst.*, E82-D(10):1425–1432, Oct. 1999.
- [20] R. Jané; S. Olmos; P. Laguna and P. Caminal. Adaptive hermite models for ECG data compression: Performance and evaluation with automatic wave detection. *Computers in Cardiology, IEEE Computer Society Press, London*, pages 389–392, 1993.
- [21] P. Laguna; G. B. Moody; R. Jané; P. Caminal and R.G. Mark. The KL transform as a tool to analyze the ST segment: Comparison with QT interval. *Journal of Electrocardiology*, vol. 28 (suplement):41–49, 1995.
- [22] P. Laguna; G. B. Moody and R. G. Mark. Analysis of the cardiac repolarization period using the KL transform: Aplicaciones on the ST-T database. *Computers in Cardiology, IEEE Computer Society Press, Bethesda*, pages 233–236, 1994.
- [23] R. Jané; S. Olmos. Análisis de la repolarización cardiaca en el ECG, registrado durante angioplastia, mediante la transformada de KL. *XIII Congreso Anual de la Asociación Española de Bioingeniería, Barcelona*, pages 41–42, 1995.
- [24] P. Laguna; G.B. Moody; J. Garcia; A.L. Goldberger and R.G. Mark. Analysis of the ST-T complex using the KL transform: Adaptive monitoring and alternans detection. *Medical and Biological Engineering and Computing*, vol. 37:175–189, 1999.
- [25] S. Olmos; J. Garcia; R. Jane and P. Laguna. Análisis de la señal ECG con expansiones ortogonales de reducido número de coeficientes. *Comunicaciones del XV Congreso anual de la Sociedad Española de Ingeniería Biomedica (SEIB), Valencia*, pages 89–92, 1997.
- [26] I. Roncal and P. Laguna. Detector de isquemia utilizando la transformada de karhunen-loève. *Comunicaciones del XIV Congreso anual de la Sociedad Española de Ingeniería Biomédica, Pamplona*, pages 91–93, 1996.

- [27] S. Olmos; D. Echeverria; R. Jane and P. Laguna. Compresion de señal ECG multiderivacionales mediante expansiones ortogonales optimas (in spanish). *Comunicaciones del XV Congreso anual de la Sociedad Española de Ingenieria Biomedica (SEIB), Valencia*, pages 321–324, 1997.
- [28] S. Olmos; M. Millan; J. Garcia and P. Laguna. ECG data compression with the KL transform. *Computers in Cardiology, IEEE Computer Society Press, Indianapolis*, 1996.
- [29] S. Haykin. *NEURAL NETWORKS: A comprehensive foundation*. Prentice-Hall, Englewood Cliffs,NJ, second edition, 1999.
- [30] L. Sörnmo; P. O. Börjesson; M. E. Nygård and O. Pahlm. A method for evaluation of QRS shape features using a mathematical model for the ECG. *IEEE Trans. on Biomed. Eng.*, BME-28:713–717, 1981.
- [31] R. Jané; S. Olmos. A comparative study of adaptive algorithms for ECG data compression using hermite models. *Proc. Annual Int. Conf. fo the IEEE Engineering in Medicine and Biology Society , Baltimore*, pages 1262–1263, 1994.
- [32] R. Jané; S. Olmos; P. Laguna; and P. Caminal. ECG data compression using an adaptive hermite model. *Proc. of the 2nd European Conf. on Eng. and Med., Stuttgart*, pages 362–363, 1993.
- [33] P. Laguna; R. Jané and P. Caminal. Adaptive estimation of QRS complex by the hermite model for classification and ectopic beat detection. *Medical and Biological Engineering and Computing*, vol. 34:58–68, 1996.
- [34] A. S. Galanopoulos; J. E. Fowler; S. C. Ahalt. Vector quantization using artificial neural network models. Department of Electrical Engineering. The Ohio State University.
- [35] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biol. Cyb.*, 43:59–69, 1982.
- [36] M. Cottrell; J. C. Fort; G. Pages. Convergence conditions for frequency sensitive competitive learning. Samos. Université Paris.
- [37] N. Ueda; R. Nakano. A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers. *Neural Networks*, 7(8):1211–1227, January 1994.
- [38] A. Gerscho. Asymptotically optimal block quantization. *IEEE Trans. on Information Theory*, IT-25(4):1211–1227, July 1979.
- [39] J. A. Freeman; D. M. Skapura. *NEURAL NETWORKS. Algorithms, Applications, and Programming Techniques*. Prentice-Hall, Englewood Cliffs,NJ, 1999.
- [40] T. Kohonen; Jussi Hynninen; Jari Kangas; Jorma Laaksonen; and Kari Torkkola. LVQPACK: the learning vector quantization program package. Report A30; Helsinki University of Technology; Faculty of Information Technology; Laboratory of Computer and Information Science.

- [41] R. Jacobs; M. Jordan. Adaptive mixtures of local experts. *Neural Computation*, 3(79-87), 1991.
- [42] R. Jacobs; M. Jordan. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(181-214), 1994.
- [43] P. Moerland. Some methods for training mixtures of experts. *IDIAP-COM*, (97-05), 1995.
- [44] Z. Ghahramani; M. Jordan. Supervised learning from incomplete data via an EM approach. *Advances in Neural Information Processing Systems.*, (6), 1994.
- [45] S. Russell. The EM algorithm. *Machine Learning*, 1998.
- [46] Rasit Eskicioglu. Introduction to parallel computation course. Department of Computer Science. University of Manitoba, Manitoba, Canada, Set-Dec 2001.
- [47] D. Alahakoon; S. Halgamuge; and B. Srinivasan. Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Trans. on Neural Networks*, 11(3), May 2000.
- [48] The MathWorks Inc. MatLab R12. www.mathworks.com.
- [49] G.S. Furno; W.J. Tompkins. QRS Detection using automata theory in a battery-powered microprocessor system. *IEEE Frontiers of Engineering in Health Care*, 4:155–158, 1982.
- [50] P. Moerland. Mixtures of experts: estimating a posteriori probabilities. *Proc. International Conference on Neural Networks*, (97-07):499–504, October 1997.
- [51] D. L. Hudson; M. E. Cohen. *Neural Networks and Artificial Intelligence for Biomedical Engineering*. IEEE Press Series in Biomedical Engineering, Piscataway,NJ, 2000.
- [52] Jurgen Schurmann. *PATTERN CLASIFICATION: an unified view of statistical and neural approaches*. Wiley-InterScience, 605 Third Avenue, NY, 1996.
- [53] N. V. Thakor; J. G. Webster and W. J. Tompkins. Estimation of QRS complex power spectra for design of a QRS filter. *IEEE Trans. Biomed. Eng.*, BME-31:702–05, 1984.
- [54] R. Jané; P. Laguna and P. Caminal. Automatic detection of wave boundaries in multilead ECG signals: Validation with the CSE database. *Computers and Biomedical Research*, Vol. 27:45–60, 1994.
- [55] R. Mark and R. Wallen. AAMI-recommended practice: Testing and reporting performance results of ventricular arrhythmia detection algorithms. Association for the Advancement of Medical Instrumentation, Arrhythmia Monitoring Subcomitee, Tech. Rep. AAMI ECAR-1987.
- [56] J. Garcia; S. Olmos; G. Moody; R. Mark and P. Laguna. Adaptive estimation of the KL series applied to the study of ischemic ECG records. *Computers in Cardiology, IEEE Computer Society Press. Indianapolis.*, 1996.

- [57] L. Senhadji; G. Carrault; J. J. Bellanger and G. Passariello. Comparing wavelet transforms for recognizing cardiac patterns. *IEEE Engineering in Medicine and Biology*, March/April 1995:167–173.
- [58] M. L. Hilton. Wavelet and wavelet packet compression of electrocardiograms. *IEEE Trans. on Biomed. Eng.*, Vol. 44, No. 5:394–402, 1997.
- [59] A. Tewfik; D. Sinha; and P. Jorgensen. On the optimal choice of a wavelet for signal representation. *IEEE Trans. on Information Theory*, Vol. 38:747–765, 1992.
- [60] R. R. Coifman and M. V. Wickerhauser. Entropy based algorithms for best basis selection. *IEEE Trans. on Information Theory*, Vol. 38, 1992.
- [61] D. Pollen. Parameterization of compactly supported wavelets. 1989.
- [62] K. Anant; F. Dowla; and G. Rodrigue. Vector quantization of ecg wavelet coefficients. *IEEE Signal Processing Lett.*, vol. 2:129–131, 1995.
- [63] S. G. Miaou; H.L. Yen. Quality driven gold washing adaptive vector quantization and its application to ECG data compression. *IEEE Trans. on Biomed. Eng.*, Vol. 47, No. 2:209–218, 2000.
- [64] A. Tuzman; M. Acosta; R. Bartesaghi; and T. Hobbins. Compression of long ECG data records using a wavelet based heartbeat dictionary, IIE, Facultad de Ingeniería, Universidad de la República.
- [65] A. Tuzman; and S. Chialanza. Design of wavelet basis for ecg signal processing, IIE, Facultad de Ingeniería, Universidad de la República.
- [66] N. Maglaveras; T. Stamkopoulos; C. Pappas; M. G. Strintzis. An adaptive backpropagation neural network for real-time ischemia episodes detection: Development and performance analysis using the european ST-T database. *IEEE Trans. on Biomed. Eng.*, Vol. 45, No. 7:805–813, 1998.
- [67] A. S. Galanopoulos; R. M. Moses; S. C. Ahalt. Convergence conditions for frequency sensitive competitive learning. Department of Electrical Engineering. The Ohio State University.