

Agrupador de complejos QRS basado en Aprendizaje Competitivo Selectivo

Pablo Aguirre, Juan Cardelino, Nicolás Loeff & Franco Simini
Instituto de Ingeniería Eléctrica,
Facultad de Ingeniería,
Universidad de la República.

Resumen

El complejo QRS es una de las señales más importantes presentes en el ECG en cuanto a la detección de anomalías cardíacas. El algoritmo presentado descompone cada complejo en sus primeros componentes principales y le incorpora información temporal. Los complejos QRS son agrupados en este espacio utilizando el algoritmo de aprendizaje competitivo selectivo (Competitive Selective Learning, CSL) el cual está basado en un resultado óptimo de cuantización vectorial. Este procedimiento presentó una tasa de clasificación del 99.0% al agrupar los complejos de la base de datos MIT/BIH en 25 grupos.

Palabras clave: complejo QRS, agrupamiento, CSL, aprendizaje competitivo, base de datos MIT/BIH.

1. Introducción

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 a pesar de la aparición de técnicas más modernas.

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. Para que esto sea posible, es necesario que la performance del algoritmo sea muy alta para que el resultado sea confiable.

El presente trabajo se centrará en el estudio del complejo QRS¹ que esta 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 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 la características de conducción eléctrica del cuerpo, etc.

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

¹El latido cardíaco tiene una expresión bio-eléctrica compuesta por varias ondas, llamadas P, Q, R, S y T respectivamente. En este trabajo se utiliza la expresión *complejo QRS* como referencia a la señal formada por las tres ondas centrales de cada latido.

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

Cuadro 1: Códigos de anotaciones de la base MIT-BIH y sus explicaciones.

de acondicionamiento de la señal (pre-procesamiento), ya que la calidad de la señal a la entrada del clasificador tiene impacto directo sobre el desempeño del mismo.

El enfoque propuesto en este trabajo consiste en la extracción de cada complejo QRS un vector n -dimensional de características que describen al mismo. Este vector se obtiene truncando el desarrollo en componentes principales de la señal, y agregando información temporal del mismo. De esta manera se logra representar los latidos en un espacio conveniente para el agrupamiento, reducir el costo computacional y lograr una mayor inmunidad al ruido. Los vectores así obtenidos son agrupados en 25 clases utilizando una variante del aprendizaje competitivo (CL), desarrollada por Ueda y Nakano ([1]), denominada aprendizaje competitivo selectivo (CSL). Este algoritmo resuelve la muchos de los problemas usuales del CL, principalmente la convergencia a mínimos locales, y por lo tanto, en general, presenta un mejor desempeño. La validación del algoritmo se realiza sobre la base de datos de arritmias MIT-BIH ([2]). A continuación se presenta una breve descripción de esta.

Base de datos MIT-BIH

La misma fue construida a partir de un conjunto de unos 4000 registros Holter. Consta de 48 registros de aproximadamente 30 minutos de duración, de los cuales se encuentran disponibles gratuitamente 25 registros completos y los primeros diez minutos de los restantes. Los registros están numerados como 1XX o 2XX y aquellos incompletos se identifican con el prefijo 'x.'. Las señales fueron tomadas de hombres y mujeres entre los 23 y 89 años. Las mismas fueron registradas analógicamente con dos canales por el mismo modelo de instrumento en todos los casos, y más tarde digitalizadas con una frecuencia de muestreo de 360 Hz y resolución de 11 bits (para ± 5 mV).

Los registros fueron anotados por dos grupos de cardiólogos independientemente, y las discrepancias fueron resueltas por consenso. Las anotaciones de la base se consideraron en este estudio como el verdadero tipo de arritmia y los distintos tipos presentes en la base se pueden apreciar en la tabla 1.

Organización del artículo

En la sección 2 se describe el proceso de filtrado y extracción de características. El algoritmo CSL es presentado en la sección 3. En la sección 4 se explica la implementación del agrupador de complejos QRS. Los resultados del mismo son discutidos en la sección 5, y son comparados con algoritmos similares presentes en la literatura. Por último, las conclusiones se desarrollan en la sección 6.

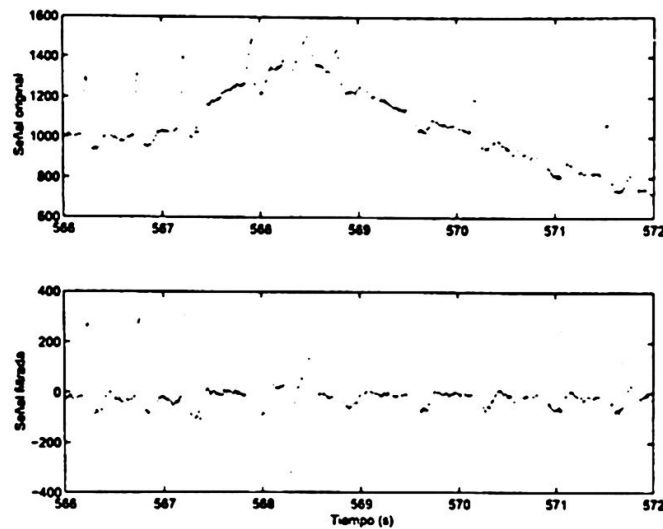


Figura 1: Filtrado del *baseline drift* en el registro 203

2. Preprocesamiento

2.1. Filtrado

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 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 fase lineal, de 512 retardos, con una ganancia de 1 en la banda pasante, frecuencia de -3dB en 1.25 Hz y una ganancia de -50 dB en 0.1 Hz. Un ejemplo de este filtrado se puede apreciar en la figura 1 donde se muestra su comportamiento en una sección del registro 203.

La ubicación del pico R es determinada a partir de las anotaciones de la base de datos y el complejo QRS es extraído del canal 1 de la señal en una ventana de ancho $t_0 = 200$ ms (73 muestras a 360Hz) centrada en el pico R.

2.2. Transformada de Karhunen-Loève

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 ([3, 4, 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 dado que su efectividad depende de la elección de los umbrales.

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.

Es común en la literatura para problemas de clasificación de complejos QRS trabajar con la transformada de Karhunen-Loève y la de Hermite ([8, 9]).

La transformada de Karhunen-Loève (K-L, también conocida como *Análisis de Componentes Principales*) 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 ([10, 11, 12, 13, 14, 15]), compresión de ECG ([16, 17]) y clasificación de latidos ([18]). Dado que los vectores correspondientes a valores propios más altos representan mayor proporción de energía proyectada, un extractor de características de orden N debe utilizar los primeros vectores propios ordenados por valor propio descendente.

En [19] se muestra que la representación obtenida con 6 coeficientes de K-L presenta un error cuadrático medio de aproximadamente 3% sobre la base de datos MIT-BIH.

2.3. Generación de vectores de características

Para generar los vectores de características se utilizaron los primeros nueve coeficientes de la proyección en el espacio de la transformada, y además se agregó información temporal de la señal. En particular se consideraron las siguientes características:

- Intervalo RR instantáneo. Calculado como,

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

donde R_i representa la muestra en donde se estima se encuentra el pico R . Esta característica denota el ritmo cardíaco.

- Cociente entre el RR instantáneo y el medio. Calculado como,

$$k_i = \frac{RR_i}{\overline{RR}_i} \quad (2)$$

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

- 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.

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. Este factor es la desviación estándar del primer coeficiente.

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, asignándose pesos menores a las características temporales ya que la mayor parte de la información morfológica se encuentra contenida en los coeficientes de la transformada. De acuerdo a esta idea los pesos utilizados en la implementación fueron: 0.25 al ancho, 0.5 al RR y 0.5 al K_i y 1 a los coeficientes de la Karhunen-Loève.

3. Algoritmo CSL

3.1. Introducción

El problema de cuantización vectorial se define como el mapeo de un espacio de entrada k -dimensional en un conjunto Y de N vectores representativos (*code-vectors*)

$$R^k \rightarrow Y \subset R^k \text{ donde } Y = \{y_1 \dots y_N\}$$

A cada *code-vector* se le asocia una región S_i del espacio de entrada definido a partir de la regla del vecino más cercano:

$$S_i = \{x \in R^k / d(x, y_i) \leq d(x, y_j) \forall j \neq i\} \quad (3)$$

donde, por ejemplo para la norma Euclídea,

$$d(x, y_i) = \|x - y_i\|_2^r \quad (4)$$

tal que r denota la potencia de la norma.

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

$$Q(x) = y_i \text{ si } x \in S_i \quad (5)$$

En la mayoría de los casos para el proceso de diseño de un cuantizador se dispone de un conjunto $X = x_1 \dots x_T \subset R^k$ de T vectores de entrenamiento, el cual es utilizado repetidamente para encontrar los valores óptimos de los *code-vectors* y_j . Cada presentación completa de los T vectores se denomina *epoch*. La optimalidad se busca en el sentido de minimizar la *distorsión* definida como

$$D = \frac{1}{k} E\{d(x, Q(x))\} \quad (6)$$

que se puede escribir como

$$D = \frac{1}{k} \int_{R^k} d(x, Q(x)) p(x) dx \text{ en el caso continuo} \quad (7)$$

$$D = \frac{1}{k} \sum_i d(x_i, Q(x_i)) p_x(x_i) \text{ en el caso discreto} \quad (8)$$

Los algoritmos tradicionales de aprendizaje competitivo presentan cada vector de entrada $x \in R^k$ a todos los *code-vector* para calcular la distancia entre ellos y elegir al ganador como el *code-vector* y_c más cercano. Para reducir la distancia media al ganador, el nuevo *code-vector* se calcula como

$$y_c(t+1) = y_c(t) + \eta(t)[x(t) - y_c(t)] \quad (9)$$

donde $\eta(t)$, denominado *learning rate*, es un parámetro no negativo monótonamente decreciente que controla la convergencia del algoritmo. La ecuación 9 es conocida usualmente como la regla *winner-takes-all*.

Debido a este entrenamiento de descenso por gradiente, el resultado final es altamente dependiente de las condiciones iniciales y lleva al algoritmo a converger a mínimos locales. Así, algunos *code-vectors* puede que nunca resulten ganadores, sub-utilizando la capacidad del algoritmo. Diversas propuestas han aparecido para atenuar este problema, incluyendo aprendizaje competitivo sensible a la frecuencia (FS-CL) y Mapas Auto-Organizados (SOM). Sin embargo, ninguno de estos enfoques logran un desempeño óptimo.

3.2. Principio de Equidistorsión

El principio de equidistorsión fue propuesta por Gerscho en 1979, quien unificó y generalizó los resultados existentes sobre la optimalidad asintótica de los cuantizadores vectoriales. En su trabajo [20] 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

$$D^* = C(k, r) N^{-\beta} \left(\int p(x)^{\frac{1}{1+\beta}} dx \right)^{1+\beta} \quad (10)$$

donde r y $\beta = \frac{r}{k}$ (normalizada por la dimensión) denotan la potencia de la norma Euclídea usada (6) y el término $C(x, k)$ es un coeficiente que depende solo de la dimensión k y de la potencia r .

El mínimo de la distorsión (10) se alcanza sólo cuando la densidad de probabilidad de los vectores de entrenamiento es proporcional a $p(x)^{\frac{1}{1+\beta}}$. Gerscho ([20]) muestra que en ese caso todas las sub-distorsiones, D_i con $i = 1, 2, \dots, N$, en una partición óptima son iguales. Donde la sub-distorsión D_i se define como:

$$D_i = \frac{1}{k} E\{d(x, Q(x))\}, \text{ para } x \in S_i \quad (11)$$

de esta forma $D = \sum_i D_i$.

En el desarrollo anterior se asumió que la densidad de probabilidad subyacente era de soporte conexo, lo cual no es una suposición muy realista, dado que en los casos prácticos en general se tiene un conjunto

²La densidad de los vectores de entrenamiento es discreta, pero como T es grande se puede aproximar por una densidad continua.

1. Inicialización

- N : número de *code-vectors*.
- X : conjunto de entrenamiento.
- Y^0 : conjunto de *code-vectors* iniciales.
- ϵ : criterio de parada (sobre la distorsión).
- m : contador de *epochs* ($m = 1$).
- $D(0) = \infty$: distorsión inicial.
- $s(m)$: función de selección de la cantidad de *code-vectors* a seleccionar en el paso m .

2. Entrenamiento: Para todos los $x \in X$, ejecutar 2a y 2b al igual que en el aprendizaje competitivo usual.

- a) Encontrar el *code-vector* más cercana al vector x .
- b) Calcular el nuevo *code-vector* usando la ecuación 9.

3. Selección: Si $s(m) > 2$, elegir $s(m)$ *code-vectors* y ejecutar el algoritmo de selección para los *code-vectors* elegidos. 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 *code-vectors* obtenidos en el paso 2.
- b) Si $\left| \frac{D(m) - D(m-1)}{D(m)} \right| \leq \epsilon$ termina el algoritmo dando como resultado final los *code-vectors* $Y^{(m)}$. En caso contrario, incrementar el contador m y volver al paso 2.

Figura 2: Resumen de Algoritmo CSL

de clusters disjuntos de vectores de entrada. Ueda y Nakano ([1]) extendieron el resultado de Gerscho para este caso.

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

1. $\Omega_i \subset R^k$
2. $i \neq j \Rightarrow \Omega_i \cap \Omega_j = \emptyset$
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 ($p(x)$ de soporte conexo) corresponde al caso $M = 1$.

Este resultado se puede resumir en el siguiente teorema.

Teorema 1 (Principio de Equidistorsión) Para un número grande de vectores representativos N , aunque la densidad de probabilidad subyacente este formada por clusters disjuntos, la distorsión es mínima cuando todas las regiones presentan la misma sub-distorsión.

3.3. Algoritmo CSL

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

Este mecanismo consiste en quitar *code-vectors* de las zonas donde la distorsión es menor para agregarlos a las zonas donde la distorsión es mayor. El número de *code-vectors* 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 *code-vectors* en las regiones de acuerdo a sus sub-distorsiones sin cambiar el número total de *code-vectors*. A medida que el aprendizaje avanza las sub-distorsiones se van igualando hasta aproximar la óptima.

Cuando se dispone de un conjunto finito de vectores de entrenamiento, éste se utiliza repetidamente para ajustar los *code-vectors* hasta que el algoritmo converge. Asimismo los vectores se eligen aleatoriamente para facilitar el escape de los mínimos locales. Además, como se dispone de un conjunto discreto, 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 en cada *epoch*. Para asegurar la convergencia de la selección, es necesario que el número de *code-vectors* 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

La figura 2 muestra un resumen del algoritmo CSL. La distorsión media en el paso 4a en la iteración m se calcula como

$$D(m) = \frac{1}{T} \sum_{x \in X} \min_{y \in Y^{(m)}} d(x, y) \quad (12)$$

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

En el paso 3 se eligen los $s(m)$ *code-vectors* que se encuentran en los extremos de la distorsión (los que tienen más y menos). De esta forma, es claro ver que eligiendo los *code-vectors* de esta manera se equipara más rápido la distorsión de cada región.

Como se puede ver el algoritmo CSL es básicamente un aprendizaje competitivo clásico, 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 los *code-vectors* 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 *code-vectors* 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 esta es aproximadamente alcanzada.

En el algoritmo se tienen $s(m)$ *code-vectors* a seleccionar, que han de ser repartidos de forma de tal que en la regiones con mayor distorsión haya más *code-vectors*. Para elegir como se distribuirán los *code-vector*, se calcula una medida no-lineal de la distorsión (función de *fitness* g_i). Esta función representa la fracción de los $s(m)$ *code-vectors* que corresponden en la región i , es decir que $\sum_{i=1}^{s(m)} g_i = 1$.

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

A cada región se le deberían asignar $g_i s(m)$ *code-vectors*, pero en realidad se le asignan u_i , por lo tanto queda por repartir la fracción $u'_i = g_i s(m) - u_i$ de *code-vector*. Entonces se le asigna un *code-vector* más a cada una de las $s'(m)$ regiones cuyo u'_i es más cercano a uno.

En la figura 3 se puede apreciar un resumen del algoritmo de selección.

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

$$g_i = \frac{D_i^\gamma}{\sum_{l=1}^{s(m)} D_l^\gamma}$$

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

2. Calcular el número de *code-vectors* a reproducir en cada región:

- a) Calcular $u_i \equiv [g_i s(m)]$ ($[a]$ denota la parte entera del número a)
- b) Ordenar los *code-vectors* según el valor de $u'_i = g_i s(m) - u_i$.
Para los $s'(m) = s(m) - \sum_{i=1}^{s(m)} u_i$ *code-vectors* que tengan el valor de u'_i más cercano a uno, sumar uno al u_i correspondiente.

3. Para cada i , el *code-vector* y_i se reproducirá según u_i , de forma que:

- Si $u_i = 0$ el *code-vector* desaparece.
- Si $u_i = 1$ el *code-vector* permanece sin cambios.
- Si $u_i > 1$ se agregan $u_i - 1$ *code-vectors* en la región sumando a y_i una perturbación aleatoria δ_i tal que $\|\delta_i\| \ll \|y_i\|$.

Figura 3: Resumen del algoritmo de Selección

Tipo	!	/	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 %

Tipo	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 %

Cuadro 2: Performance por clase del agrupador sobre la base de datos.

Observaciones

La subdistorsión D_i en el paso 1 se calcula como

$$D_i = \frac{1}{T} \sum_{x \in S_i} d(x, y_i)$$

debe notarse que en el cálculo de D_i se usa la cantidad total de vectores de entrenamiento T y no la de la región, t_i . Esto se desprende de la definición de subdistorsión (11).

En el paso 1 si $\gamma = 1$ la función g_i es lineal, y si además la varianza de las sub-distorsiones 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. Agrupador

4.1. 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 2.3, que se mapea en los *code-vectors*

Tipo	!	/	A	E	F	J	L	N	Q	R	S	V	a	e	f	j	Σ
!	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
Σ	21	24	115	4	169	16	4	108	30	11	2	170	30	5	43	14	766

Cuadro 3: Agrupamiento de los latidos sobre la base de datos, por tipo.

del agrupador. Los latidos se agruparon en 25 clases de forma tal de poder comparar los resultados con [8].

4.2. Entrenamiento

Usualmente los datos de interés constituyen un subconjunto *pequeño* del espacio de entrada, por lo tanto para la inicialización se buscó que los vectores representativos se ubicaran cerca del centro de los datos de entrada, en este caso como éstos fueron normalizados, este centro estará cerca del origen. Por lo tanto se inicializaron los *code-vectors* 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, actualizando los *code-vectors* 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 *code-vectors* menos uno. $s(m)$ se eligió de manera que se decremente un *code-vector* por *epoch*. Si durante cinco *epoch* seguidas ningún *code-vector* fue modificado 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*.

5. 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 *code-vectors* 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 disponibles³ en la base MIT-BIH ([2]). Los códigos de los tipos de latidos pueden encontrarse en la tabla 1.

La performance del algoritmo se medirá en función de la *sensibilidad* (S) y el *valor predictivo positivo* (+P). 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*.

³Cabe aclarar que si bien existen 20 tipos de latidos definidos en la base, sólo aparecen en los registros 16 de dichos tipos

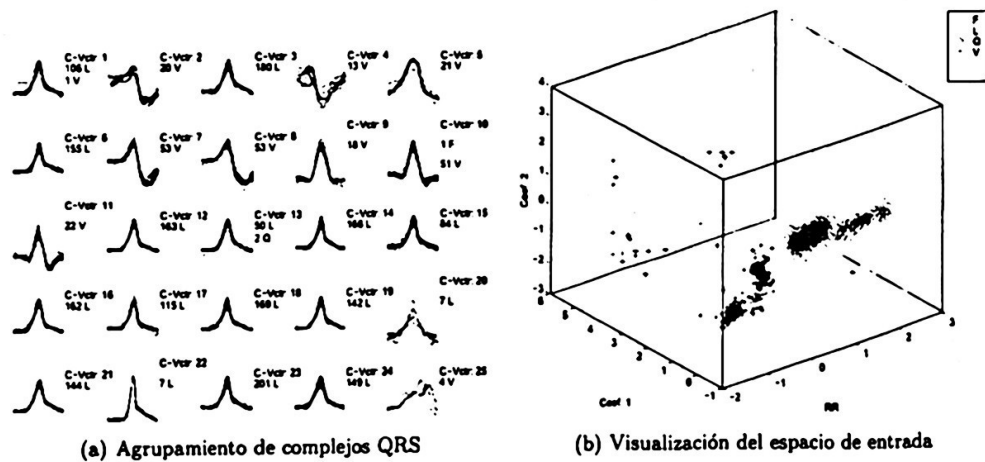


Figura 4: Agrupamiento del registro 214.

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

Entonces se definen sensibilidad y valor predictivo positivo por clase (S_i y $+P_i$) como

+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.

$$+P_i = \frac{TP_i}{TP_i + FP_i} \quad (13)$$

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.

$$S_i = \frac{TP_i}{TP_i + FN_i} \quad (14)$$

De la misma forma se definen las performances globales⁴

+P: Es el promedio de las predictividades positivas por clase, pesadas por el número de latidos de cada clase

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

En la tabla 2 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 *code-vectors* a las regiones con mayor densidad. Por otro lado, al haber tantos *code-vectors* asignados a estas clases, algunos latidos de clases minoritarias caen en las regiones correspondientes a dichos *code-vectors*. 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

⁴Una 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.

prematurados 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 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 Σ 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 una clase, y en sentido vertical los latidos de una 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 5 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).

Como ejemplo, en la figura 4(a) se puede apreciar la superposición de los complejos QRS (después de filtrados) asociados a cada *code-vector* para el registro 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(b) se puede observar la estructura de los latidos en el espacio de entrada al agrupador (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). En la leyenda de la misma se pueden observar los distintos tipos de latidos representados, mientras que los *code-vectors* figuran como estrellas (*). Las observaciones hechas en el párrafo anterior pueden aclararse. En la misma se observa que el latido ventricular (V) mal clasificado en el *code-vector* 4 se debe a que tiene un período (RR) muy distinto al del resto de los demás ventriculares.

5.1. 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.* [8] utilizan la transformada de Hermite para extraer características de los complejos QRS y un mapa auto-organizado de 25 *code-vectors* para realizar el agrupamiento. Dichos autores reportan una performance global (sensibilidad) sobre toda la base de datos MIT-BIH de 98.5 % frente a 99.00 % de este trabajo. 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 de Lagerholm en las condiciones de nuestro trabajo está acotada por 98.8 %, que se daría en el caso en que todos los latidos que no fueron detectados fueran agrupados correctamente.
- Lagerholm *et al.* utilizan ambos canales de la base de datos, mientras que este trabajo utiliza el canal 1 solamente.

No se reportaron las performances por registro. Sin embargo, para comparar los resultados con el trabajo de Hu *et al.* [18] (que es un algoritmo supervisado) se calcula la medida de performance: ⁵,

$$R = \frac{\text{No. de latidos V y E correctamente agrupados}}{\text{No. total de latidos}} \quad (15)$$

En la tabla 4 se puede apreciar la comparación de esta medida para este trabajo con el trabajo citado [8]. Como se puede apreciar en la misma en los registros completos (200-219), que son directamente comparables, la performance obtenida por este trabajo es ligeramente superior a la reportada por Lagerholm *et al.*. En los demás registros se sigue la misma tendencia.

Reg.	Lagerholm <i>et al.</i>		CSL	
	Latidos	R	Latidos	R
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 %

Cuadro 4: Comparación con algoritmo SOM de Lagerholm *et al.*[8]

Classifying Multichannel ECG Patterns with an Adaptive Neural Network

En este trabajo S. Barro *et al.* [3] 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 de 98.98 % de este trabajo. Es necesario hacer varias aclaraciones sobre el trabajo de Barro:

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

Debido a estas diferencias y dado que los autores no incluyeron el número de latidos por registro, no se efectuó una comparación por registro con este trabajo.

⁵Es de notar que aunque entre los VEB (latidos ectópicos ventriculares) de [18] 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 [18] los ignora.

Reg.	Tipo	total	Reg.	Tipo	total	Reg.	Tipo	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 %

Cuadro 5: Performance global por registro del agrupador CSL.

6. Conclusiones

En el presente trabajo se investigó el problema de agrupamiento de complejos QRS, y en particular se estudió la aplicabilidad del algoritmo CSL al mismo, lo cual constituye el aporte fundamental de este trabajo, dado que no se encontró en la literatura ningún trabajo que lo utilizara en este tipo de problemas.

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.

El agrupador desarrollado mostró una sensibilidad de 99.00% y un valor predictivo positivo de 98.98%, ligeramente superior a los desempeños reportados en trabajos similares ([3, 8]). Estos resultados muestran que utilizar el algoritmo de aprendizaje competitivo selectivo (CSL) en el desarrollo de un agrupador es una alternativa válida para enfrentar este problema. Esto es coherente con los resultados teóricos que muestran la superioridad de CSL frente a los demás algoritmos de aprendizaje competitivo.

Los resultados obtenidos sugieren la posibilidad de aplicar una variante supervisada del algoritmo CSL en problemas de clasificación. Una posible implementación de esta idea, desarrollada por los autores, se puede encontrar en [19].

Existen algunas mejoras, que escapan al alcance de este trabajo, que pueden contribuir a mejorar el desempeño obtenido. Por ejemplo la utilización de más canales en todo el proceso ([3, 4]). Esto implica que se dispone de más información sobre un mismo latido, lo cual puede influir favorablemente en la tasa de clasificación.

Por otra parte, el algoritmo CSL es adaptivo en el número de *code-vectors* por clase, pero el número total permanece constante. Se puede ver [19] que la performance se ve alterada significativamente según el número total de *code-vectors*. Por lo tanto es razonable pensar que permitiendo variar el número total del *code-vectors*, se puedan lograr performances más altas en los clasificadores. En la literatura existen varios trabajos en torno a esta idea, por ejemplo [4, 21, 22] y [23].

Referencias

- [1] 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.
- [2] Massachusetts Institute of Technology. MIT-BIH Database Distribution. <http://www.physionet.org/physiobank/database/mitdb/>.
- [3] 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.
- [4] M. Fernandez-Delgado and S. Barro. Mart: A multichannel ART-based neural network. *IEEE Trans. on Neural Networks*, 9:139-150.
- [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. *IEEE Trans. Signal Processing*, Vol. 46, no. 5:1417-1425, 1998.
- [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] M. Lagerholm; C. Peterson; G. Braccini; L. Edenbrandt; L. Sörnmo. Clustering ECG complexes using hermite functions and self-organizing maps. *IEEE Trans. on Biomedical Engineering*, pages pp. 838-848, 2000.
- [9] 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.

- [10] 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.
- [11] P. Laguna; G. B. Moody and R. G. Mark. Analysis of the cardiac repolarization period using the KL transform: Applications on the ST-T database. *Computers in Cardiology, IEEE Computer Society Press, Bethesda*, pages 233-236, 1994.
- [12] R. Jané; S. Olmos. Análisis de la repolarización cardíaca 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.
- [13] 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.
- [14] S. Olmos; J. Garcia; R. Jane and P. Laguna. Análisis de la señal ECG con expansiones ortogonales de reducido número de coeficientes (in spanish). *Comunicaciones del XV Congreso anual de la Sociedad Española de Ingeniería Biomédica (SEIB), Valencia*, pages 89-92, 1997.
- [15] 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.
- [16] 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 Ingeniería Biomédica (SEIB), Valencia*, pages 321-324, 1997.
- [17] 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.
- [18] 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.
- [19] Pablo Aguirre; Juan Cardelino; Nicolás Loeff. Sistema de detección, clasificación e indentificación en línea de complejos QRS. Tesis de grado, Facultad de Ingeniería, Universidad de la República, Mayo 2002.
- [20] A. Gerscho. Asymptotically optimal block quantization. *IEEE Trans. on Information Theory*, IT-25(4):1211-1227, July 1979.
- [21] 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.
- [22] 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.
- [23] Stergios Papadimitriou; Seferina Mavroudi; Liviu Vladutu; and Anastasios Bezerianos. Ischemia detection with a self-organizing map supplemented by supervised learning. *IEEE Trans. on Neural Networks*, vol. 12. No. 3:503-515.