Métodos de extracción de características en el ECG: análisis comparativo

J.E. Neto[1,2,3], A. A. Suarez-Leon[4], C. R. Vázquez-Seisdedos[4], N. A. López-Mora[5], J. C. Leite[6], y R. C. L. Oliveira[3],

[1] Universidade Paulista do Estado do Amazonas, Manaus, Brasil
[2] Universidade Paulista do Estado do Amazonas, Manaus, Brasil
[3]Faculdade de Engenharia Elétrica/Universidade Federal do Pará, Belém, Brasil
[4]Facultad de Ingeniería Eléctrica/ Universidad de Oriente, Santiago de Cuba, Cuba
[5] Instituto Universitario de Tecnología/Departamento de Informática, Valencia, Venezuela
[6]Instituto de Tecnologia e Educação da Amazônia, Manaus, Brasil

Abstract— **ECG Feature Extraction plays a significant role in diagnosing most of the cardiac diseases. In this paper, a comparison between three ECG feature extraction methods is presented. The methods are: Linear Principal Components Analysis (PCA), Discrete Cosine Transformation (DCT) and Kernel Principal Components Analysis (KPCA). A Multilayer Perceptron is used as classifier and beats for training and validation of the classifier are extracted from twelve MIT – BIH Arrhythmia Database registers. The performance of the three classifiers is discussed and a simple execution time evaluation is performed.**

Palabras claves— **ECG classifiers, DCT, PCA, Kernel PCA.**

I. Introducción

En los últimos años se han desarrollado múltiples algoritmos de clasificación (clasificadores) que usan diversos métodos de Inteligencia Artificial (IA). Entre estos se destacan las redes neuronales artificiales, y en especial las redes *Perceptron* Multicapas (MLP) [1], la lógica difusa, los modelos ocultos de Harkov, los algoritmos genéticos, las Máquinas de Vectores Soporte (SVM), los algoritmos genéticos, los algoritmos híbridos Neuro-Fuzzy, entre otros.

Los esquemas para la clasificación automática de señales de ECG usan generalmente el esquema de la Figura 1 donde el clasificador, generalmente, está compuesto por una etapa de extracción de características y una variante de las técnicas mencionadas anteriormente.

La extracción de características (EC) es la etapa encargada de transformar el segmento de la señal a analizar, con el fin de obtener un número reducido de coeficientes con la información clínica relevante. Los principales métodos de extracción se resumen a continuación:

1. Características morfológicas extraídas directamente de la señal [2], como amplitud de ondas, medidas de segmentos, y áreas, entre otras.

2. Uso de modelos matemáticos para representar las curvas de la señal de ECG [3]. Entre los más usados están los modelos auto regresivos, los coeficientes de predicción lineal, el uso de funciones para la aproximación de tramos de la señal mediante funciones definidas (constante, recta, triangular, etc.), entre otros.

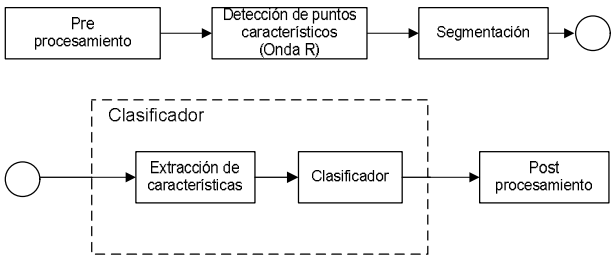

Figura 1: Etapas de un sistema clasificador de señales de ECG.

3. Uso de diversas transformadas [4]: (a) *Karhunen-Loeve* o método de análisis de componentes principales (PCA), (b) discreta del coseno (DCT), (c) wavelets, (d) distribuciones tiempo-frecuencia y (e) las funciones de Hermite, entre otras.

En [5] se comparan cualitativamente diversos métodos de EC, en cuanto a simplicidad, exactitud y valor predictivo positivo. En este trabajo no se analizan aspectos importantes como son el tiempo de ejecución (da idea del costo computacional), ni el desempeño de métodos de EC de amplio uso, como son la DCT, y el PCA lineal y no lineal.

En este artículo se realiza una comparación del desempeño de los tres métodos de extracción de características siguientes: Transformada Discreta del Coseno (DCT), PCA Lineal y Kernel PCA (o PCA no Lineal) [6], con el objetivo final de clasificar los latidos en dos clases: los generados o no, en el nodo sinusal. En los tres casos se usa como clasificador patrón, a una red neuronal del tipo MLP.

II. Materiales y métodos

Para realizar la comparación y análisis de los métodos propuestos se utilizaron 2800 latidos extraídos de 12 registros de la base de datos *MIT–BIH Arrhythmia Database* [7]

cuya frecuencia de muestreo es de 360 Hz. Se consideraron un total de 8 morfologías distintas: latido normal (NRB), bloqueo de rama izquierda (LBBB), bloqueo de rama derecha (RBBB), mixto normal + paced (PNB), mixto normal + ventricular (VNB), contracción ventricular prematura (PVCB), contracción auricular prematura (APCB) y paced (PCB). En la Tabla 1 se muestran los registros y la cantidad y tipo de latidos extraídos de cada uno.

Tabla 1 Registros seleccionados para la clasificación.

Registros	Latidos	Clasificación
201	**200**	**NRB**
202	**200**	**NRB**
203	**200**	**NRB**
210	**100**	**NRB**
213	**400**	**NRB(200), VNB (200)**
214	**200**	**LBBB**
217	**500**	**NRB(100),PCB(200), PNB(200)**
230	**200**	**NRB**
231	**300**	**NRB(100), RBBB(200)**
232	**200**	**APCB(200)**
233	**200**	**PVCB(200)**
234	**100**	**NRB(100)**
201	**200**	**NRB**
202	**200**	**NRB**
203	**200**	**NRB**
210	**100**	**NRB**
213	**400**	**NRB(200), VNB (200)**
214	**200**	**LBBB**

Los 2800 latidos se dividen en dos clases: latidos normales o generados en el nodo sinusal (NRMB) y latidos identificados como artefactos fisiológicos (ARTF), que son aquellos que no se originan en dicho nodo y por ende no son válidos para el análisis de la HRV. La división se realiza aleatoriamente pero conservando en cada clase el mismo número de subclases de manera que cada grupo tiene 700 NRB, 100 LBBB, 100 RBBB, 100 PVCB, 100 VNB, 100 APCB, 100 PCB y 100 PNB.

Las señales de ECG se han filtrado con un filtro bidireccional de Butterworth de orden 6, pasa bandas, con frecuencia de corte inferior de 0,6 Hz y superior de 45 Hz. La segmentación de latidos que se emplea es centrada alrededor del punto R con ventana simétrica de ±90 (180) muestras. Como pre-procesamiento final se elimina en cada latido, la media del conjunto y se normaliza con respecto a la desviación típica.

La red neuronal es del tipo MLP con arquitectura dirigida hacia delante (*feedforward*), con una capa oculta. En forma matricial esta red se modela como:

$$\mathbf{O} = \mathbf{g}\left(\mathbf{W}_{\mathbf{jk}}^{\mathbf{T}} \cdot \mathbf{f}(\mathbf{W}_{\mathbf{ij}}^{\mathbf{T}} \cdot \mathbf{I})\right) \quad (1)$$

Donde **I** y **O** son los vectores de entrada y salida respectivamente, **Wij** es la matriz de los pesos cuyos elementos **wij** relacionan la entrada *i* con la neurona *j* de la capa oculta. **Wjk** es la matriz cuyos elementos **wjk** relacionan la salida de la neurona *j* de la capa oculta con la neurona *k* de la capa de salida, **f** y **g** son funciones vectoriales generalmente de tipo sigmoidal o lineal.

El algoritmo utilizado para el entrenamiento es el *back-propagation* Levenberg - Marquardt con el criterio del error mínimo cuadrático (MSE).

La red que se ha implementado tiene 10 neuronas en la capa de entrada, 20 neuronas en la capa oculta y 1 neurona en la capa de salida (10 – 20 – 1). La función de activación en todas las capas es la tangente hiperbólica:

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (2)$$

La evaluación del desempeño del clasificador MLP para los tres métodos de extracción se realizó mediante el cálculo: (a) del rendimiento usando varios indicadores extraídos de la matriz de confusión para las 2 clases definidas: NRMB y ARTF (Figura 2) y (b) de los tiempos de ejecución de los casos mejor, peor y medio. Las columnas de la matriz de confusión expresan el valor real de cada clase y las filas expresan el valor predicho de estas. Para cada método de extracción se usaron 700 latidos para el entrenamiento e igual número para la validación.

NRMB	VP %VP	FP %FP	VPP (%)
ARTF	FN %FN	VN %VN	VPN (%)
(Predicción)	S (%VP)	E (% VN)	ACC(%)
	NRMB (Real)	**ARTF (Real)**	

Fig. 2: Matriz de confusión para evaluar el rendimiento del clasificador MLP para cada método de extracción de características. VP: verdaderos positivos, %VP y %VN: Porcentaje de latidos de tipo VP y VN, respectivamente; VN: verdaderos negativos, FP: Falsos positivos, FN: Falsos negativos, S: Sensibilidad, E: Especificidad, ACC: exactitud, VPP: Valor predictivo positivo, VPN: Valor predictivo negativo.

DCT – La transformada discreta del coseno (DCT) es la forma discreta de la transformada de Fourier del coseno (FCT) [8]. Matemáticamente se define discretizando la FCT dentro de una ventana rectangular finita de altura unitaria, la matriz de transformación se obtiene de:

$$[C]_{m\times n} = \sqrt{\frac{2}{N}}\left\{k_m k_n \cos\left(\frac{mn\pi}{N}\right)\right\} \quad m,n = 0,1,\mathrm{K}\,,N \quad (3)$$

En forma matricial:

$$\mathbf{X}_{\mathbf{C}} = \mathbf{C} \cdot \mathbf{x} \quad (4)$$

Los vectores $\mathbf{X_c}$ y $\mathbf{x}$ se conocen como pares de la DCT. La DCT tiene la propiedad de compactar la energía de la señal en un número de coeficientes menor; es por ello que ha sido ampliamente utilizada como método de extracción de características y en algoritmos de compresión implementados en software y en hardware. En este trabajo, se tomaron las 10 primeras componentes del vector **Xc** obtenido con MATLAB.

PCA LINEAL [9] – Es una de las técnicas de extracción de características más usadas. Consiste en obtener las proyecciones de un vector de datos en el sub-espacio generado por los *M* vectores propios que corresponden a los *M* mayores autovalores de la matriz de covarianza de los datos. Esto da lugar a una base ortogonal de manera que cualquier vector **x** del espacio de entrada puede ser representado de la forma:

$$\mathbf{x} = \sum_{i=1}^{d} \alpha_i \cdot \mathbf{v_i} : \quad \mathbf{x} \in \Re^{\mathbf{N}} \tag{5}$$

Su obtención a partir de un conjunto de *N* vectores **x** de *N* componentes (con media igual a 0), se realiza inicialmente mediante el cálculo de la matriz de covarianza **C**:

$$\mathbf{C} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{x_j} \mathbf{x_j^T} \tag{6}$$

A continuación se plantea y se resuelve el problema de autovalores:

$$\mathbf{C} \cdot \mathbf{v} = \lambda \mathbf{v} \tag{7}$$

Finalmente se seleccionan los *M* vectores ($M < N$) asociados a los mayores valores propios, obteniéndose un subespacio de $\mathbf{R^{\hat{N}}}$ sobre el cual se proyecta el vector de entrada. Esto permite reducir dimensionalmente los datos de entrada y a la vez conservar la información relevante que poseen. En este trabajo, el número de componentes seleccionado fue $M = 10$.

KERNEL PCA (KPCA) – Este método fue propuesto por Schölkopf y otros [6] y está basado en expresar el problema de auto valores del PCA lineal en forma de producto escalar en cierto espacio vectorial (*feature space*) de dimensión mayor que el espacio de entrada, relacionado con este último por medio de una aplicación no lineal:

$$\begin{aligned} \mathbf{\Phi} &: \Re^{\mathbf{n}} \rightarrow \mathbf{F} \\ \mathbf{x} &\rightarrow \mathbf{X} \end{aligned} \tag{8}$$

El método propone realizar un PCA lineal en este espacio lo que equivale a realizar un PCA no lineal en el espacio de entrada. Es necesario calcular la matriz de covarianza en el espacio F, que queda definida como:

$$\overline{\mathbf{C}} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{\Phi}(\mathbf{x_j}) \mathbf{\Phi}(\mathbf{x_j})^{\mathbf{T}} \tag{9}$$

El problema de autovalores en este espacio queda expresado como:

$$M\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha} \tag{10}$$

Donde K es tal matriz en la cual:

$$k_{ij} = \mathbf{\Phi}(\mathbf{x_i}) \cdot \mathbf{\Phi}(\mathbf{x_j}) \tag{11}$$

El procedimiento para extraer las 10 componentes principales usando el método KPCA, consta de los siguientes pasos:

1. Calcular k_{ij} según la expresión (11)
2. Resolver la expresión (10)
3. Normalizar según $1 = \lambda_k (\alpha^k \alpha^k)$
4. Extraer las primeras 10 componentes del vector ***x*** resolviendo la siguiente expresión:

$$(kPC)_n(x) = (\mathbf{V^n} \cdot \mathbf{\Phi}(\mathbf{x})) = \sum_{i=1}^{M} \alpha_i^n k(\mathbf{x_i}, \mathbf{x}) \tag{12}$$

Para la comparación de la eficiencia en la ejecución se ha tomado solamente el proceso de extracción de propiedades en sí mismo y no la etapa previa de resolver el problema de autovalores en PCA y KPCA. La computadora donde se realizaron las pruebas es una IBM PC con procesador Pentium™ IV, 2.66 GHz, 1 GB de memoria RAM DDR400 de tipo *Dual Channel*.

III. Resultados y discusión

Las Tablas 2–7 muestran los indicadores de desempeño del entrenamiento y la validación del clasificador MLP para los 3 métodos de extracción de características descriptos.

Tabla 2 Matriz de confusión del clasificador DCT + MLP (entrenamiento).

NRMB	**690 49.3%**	**47 3.4%**	**93.6%**
ARTF	**10 0.7%**	**653 46.6%**	**98.5%**
	98.6%	**93.3%**	**95.9%**
	NRMB	**ARTF**	

Tabla 3 Matriz de confusión del clasificador DCT + MLP (entrenamiento).

NRMB	**687 49.1%**	**40 2.9%**	**98.1%**
ARTF	**13 0.9%**	**660 47.1%**	**98.1%**
	98.1%	**94.3%**	**96.2%**
	NRMB	ARTF	

Los tiempos de ejecución (en segundos) en los casos mejor, peor y medio se muestran en la Tabla 8.

Tabla 4 Matriz de confusión del clasificador PCA + MLP (entrenamiento).

NRMB	**679 48.5%**	**76 5.4%**	**89.9%**
ARTF	**21 1.5%**	**624 44.6%**	**96.7%**
	97.0%	**89.1%**	**93.1%**
	NRMB	**ARTF**	

Tabla 5 Matriz de confusión del clasificador PCA + MLP (validación).

NRMB	**671 47.9%**	**69 4.9%**	**90.7%**
ARTF	**29 2.1%**	**631 45.1%**	**95.6%**
	95.9%	**90.1%**	**93.0%**
	NRMB	ARTF	

Tabla 6 Matriz de confusión del clasificador PCA + MLP (validación).

NRMB	**696 49.7%**	**8 0.6%**	**98.9%**
ARTF	**4 0.3%**	**692 49.4%**	**99.4%**
	99.4%	**98.9%**	**99.1%**
	NRMB	**ARTF**	

Tabla 7 Matriz de confusión del clasificador KPCA + MLP (validación).

NRMB	**690 49.3%**	**8 0.6%**	**98.9%**
ARTF	**10 0.7%**	**692 49.4%**	**98.6%**
	98.6%	**98.9%**	**98.7%**
	NRMB	ARTF	

Tabla 8 Tiempo de ejecución (en segundos) de los clasificadores.

	Mejor	**Peor**	**Medio**
DCT	**1,67·10-4**	**2,17·10-2**	**3,44·10-4**
PCA	**5,20·10-5**	**7,47·10-2**	**1,16·10-4**
KPCA	**1,19·10-1**	**1,58**	**1,27·10-1**

El método Kernel PCA tiene el mejor rendimiento pero también el peor tiempo de ejecución. Esto es consecuencia de que para extraer las componentes principales de un vector arbitrario es necesario evaluar *M* veces, la función Kernel, donde *M* es el número de observaciones. No obstante, la extracción de las componentes principales de un vector como promedio sólo emplea 127 ms aproximadamente. En [6] se dan algunas recomendaciones para aminorar este problema, que son objeto de investigación actual de los autores.

La contraparte es el método de PCA Lineal que es el más rápido, pues sólo requiere de una multiplicación de matrices, pero a la vez es el que peor rendimiento exhibe con un 93% en cada caso. La DCT es el caso medio con una exactitud promedio del 96 %, y un tiempo medio de menos de 0,4 ms por vector.

IV. Conclusiones

El compromiso entre los indicadores de rendimiento y el tiempo de ejecución para los tres métodos analizados sugiere el uso del método de Kernel PCA para la clasificación de señales de ECG, en lugar de otros métodos muy establecidos como son: el PCA Lineal y la Transformada Discreta del Coseno. A pesar de ser el método de mayor tiempo de ejecución, puede ser implementado con recursos relativamente modestos teniendo en cuenta las tecnologías computacionales actuales.

Reconocimientos

A la Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM), a la Universidade Paulista (UNIP), y el Ministerio de Educación Superior de Cuba por el financiamiento para la realización de este trabajo.

Referencias

1. Hosseini H G, Luo D, Reynolds K J (2006) The comparison of different feed forward neural network architectures for ECG signal diagnosis", Medical Engineering & Physics 28: 372–378.
2. Mitra S, Mitra, M Chaudhuri, B B. (2009) Pattern defined heuristic rules and directional histogram based online ECG parameter extraction, Measurement 42: 150–156.
3. Olszewsky R. (2001) Generalized feature extraction for structural pattern recognition in time-series data. PhD Thesis, School of Computer Science, Carnegie Mellon University, Pittsburg, PA 15213.
4. Cuiwei L, Chongxun Z, Changfeng T. (1995) Detection of ECG Characteristic Points using Wavelet Transforms, IEEE Trans. Biomedical Engineering 42: 21–28.
5. Karpagachelvi S, Arthanari M, Sivakumar M. (2010) ECG Feature Extraction Techniques-A Survey Approach, International Journal of Computer Science and Information Security 8: 76–80.
6. Schölkopf B, Smola A J, Müller, K R. (1998) Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation, 10: 1299–1319.
7. Moody G B, Mark R G (1990) The MIT-BIH Arrhythmia Database on CD-ROM and software for use with it, Computers in Cardiology, 17: 185–188.
8. Pat Y. (2000) Sine and cosine transforms, In: The transforms and applications handbook. Ed. Poularikas A., D. CRC Handbook Published in Cooperation with IEEE Press, pp. 33 – 40.
9. Jolliffe I T. (1986) Principal Component Analysis, Springer, Berlin.

Autor: João Evangelista Neto
Instituto: Universidadede Paulista.
Calle: Recife s/n
Ciudad: Manaus
País: Brasil
E-mail: joao_evangelista_neto@yahoo.com