

Reconocimiento de Patrones

Informe proyecto final

Introducción	1
Adquisición y pre-procesado	2
Extracción y selección de características	3
Clasificadores	5
Combinación de clasificadores	7
Conclusiones	9
Referencias	9

Fabio Lima Gainzarain

4250238-1

1. Introducción

Los ECG es una de las señales biológicas más utilizadas en la medicina, permitiendo al médico entender el funcionamiento del corazón humano y detectar distintas patologías. La detección temprana de arritmias y anomalías es crucial en el tratamiento de un paciente y prevención de ataques cardíacos. Por esta razón el procesado informático de ECGs para detección automática de anomalías ha sido un área de interés para la comunidad científica, habiendo numerosos trabajos sobre el tema.

El reconocimiento de ECGs presenta distintos problemas, al ser series en el tiempo el clasificar patrones en base a los datos en crudos no suele ser efectivo. Además las distintas anomalías presentan variación en un mismo paciente (dos latidos de la misma clase pueden diferir según la frecuencia cardíaca) y entre pacientes (dos ECGs "diferentes" pertenecientes a dos pacientes pueden corresponder a la misma anomalía). Todo esto desemboca en un problema de reconocimiento que es difícil de resolver incluso para el humano entrenado.

En este trabajo se probaran modelos para clasificar cinco clases de latidos: Normal (N), arritmia auricular (A), arritmia ventricular (V), bloqueo de rama izquierda (L), bloqueo de rama derecha (R). El trabajo se estructuró de la siguiente manera: Pre-procesado, Extracción de características, Selección de características, Clasificación individual de clases, Combinación y clasificación completa.

Una aclaración con respecto al informe de avances, por error en ese informe se trabajaba con un set reducido de pacientes, reduciendo la variabilidad entre pacientes. Ahora se pasó a trabajar con la base de 44 pacientes por lo que el desempeño en general disminuyó. Sin embargo como se verá, al combinar clasificadores se lograron resultados similares.

2. Adquisición y pre-procesado

Los patrones utilizados corresponden a la base de datos de ECG de MIT-BIH, esta base es la más ampliamente utilizada en trabajos de reconocimientos de ECG. La base consta de 102000 latidos correspondientes a 47 pacientes, la base está en 47 registros correspondientes a cada paciente, cada registro con una duración de 1000 segundos. Aproximadamente la mitad de los registros corresponden a pacientes en situación normal, mientras que el resto fueron seleccionados de pacientes que presentaban fenómenos poco comunes, de manera de lograr un registro más extensivo de las anomalías. Las clases seleccionadas para el trabajo corresponden a aquellas que presenten al menos 2500 instancias en la base de datos. Todos los latidos de la base fueron etiquetados por dos médicos trabajando por separado, y decidiendo por consenso en caso de diferencias.

Un latido normal presenta 5 complejos bien definidos, llamados P Q-R-S y T, para la clasificación de un latido normalmente se toma la información desde el complejo R correspondiente al latido anterior hasta el complejo R del latido posterior. Por esta razón la base se reestructuro de manera que cada latido se represente en un vector con las muestras del R anterior al R posterior. Si bien el reconocimiento de complejos R no es trivial, hay abundantes métodos efectivos estudiados para ellos (Rodríguez et al. 2015). En este trabajo no se realiza un reconocimiento de los complejos R, sino que se toma su ubicación de las anotaciones de la base original.

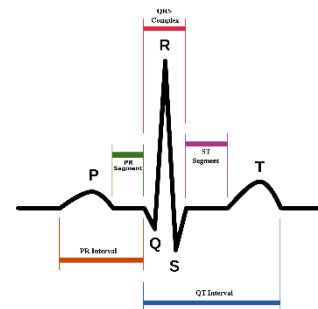


Figura 1 ECG, fuente: [wikicommons.org](https://commons.wikimedia.org/wiki/File:ECG.svg)

Para igualar el largo de los vectores se realizó un submuestreo en cada vector (previa aplicación de un filtro pasabajo) para igualar el largo de cada vector a 200 muestras, se espera que esto también aplaque el efecto de la frecuencia cardíaca variable. Para remover el baseline se aplicó un filtro de continua. No se busca un filtrado estricto del ruido.

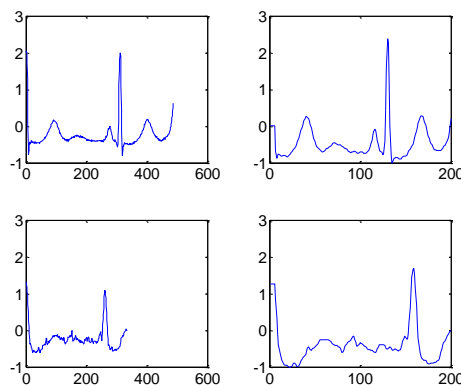


Figura 2 Ejemplo del submuestreo

Por último hay que aclarar que para todas las pruebas se tomaron las clases como equiprobables, ya que la idea es estudiar la efectividad de los métodos y no hacer un análisis riesgo/costo.

3. Extracción y selección de características

Las señales ECG muestran las descargas eléctricas en distintas regiones del corazón, de esta manera podría pensarse que el ECG es la suma de distintos canales independientes. Por esta razón un primer candidato para extraer componentes es usar ICA, este método de extracción fue utilizado en (Yu & Chou 2008) para clasificar 8 clases con un rendimiento de 98.7%, este es el mejor rendimiento que he encontrado en clasificación de ECGs.

Otro método ampliamente utilizado en varios problemas de reconocimiento por su sencillez y buen desempeño en general son los PCA, por lo que también se probarán aquí.

Por último dado que los ECG son tratados por el humano de forma visual, se pensó en aplicar algún método de extracción de características utilizado para reconocimiento de imágenes. El método seleccionado fue sparse coding (que puede aplicarse a señales también) utilizando la implementación de (Honglak Lee, Alexis Battle, Rajat Raina 2006), para construir el diccionario y calcular la codificación de los vectores se utilizó el script de matlab provisto por los autores en la web de Stanford. Se probará la clasificación con sparse, y con sparse+PCA.

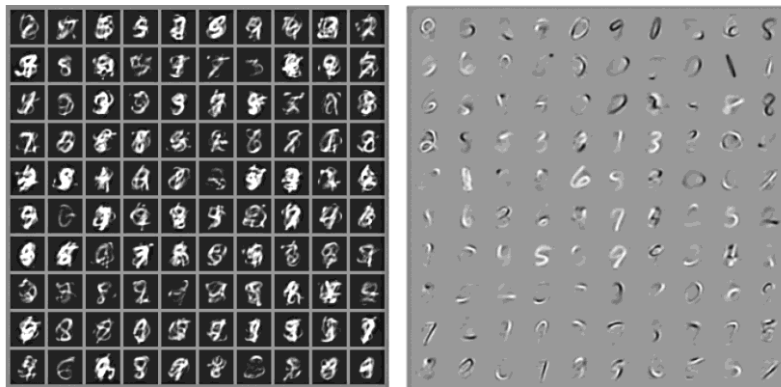


Figura 3 Ejemplo de sparse coding

Tanto para PCA como para ICA se toman los 36 primeros valores y luego se realizó una selección de características. La selección de características se hace con un método secuencial SFS (sequentialfs en MatLab), que tiene un rendimiento sub-óptimo pero tiene una buena relación entre el aumento del rendimiento y el tiempo de selección. La medida de rendimiento para los subconjuntos es la clasificación de clases mediante Bayes y se usa cross-validation con $k = 5$. Además de las características seleccionadas se agrega en todos los casos una nueva variable que indica la posición del complejo R en relación con los complejos R anterior y posterior.

Para evaluar el rendimiento de los distintos métodos se mide la performance de clasificar cada una de las clases por separado (frente a una clase de rechazo) utilizando una red neuronal y la clasificación de todas las clases utilizando Bayes. La capa oculta de la red neuronal consta de $(input + 5) * (2/3)$ de nodos. Para construir los clasificadores se usó el set de 12500 muestras el cual fue dividido en 40% entrenamiento, 20% validación, 40% test.

PERFORMANCE	ICA (35 COMP)	PCA (22 COMP)	SPARSE	SPARSE+PCA (8 COMP)
BAYES	76,2%	84,0%	64.4%	78.5%
RED CLASE N	84,4%	94,1%	68.6%	80.8%
RED CLASE A	92,3%	97,0%	83.7%	85.2%
RED CLASE V	84,4%	97,2%	93.5%	92.7%
RED CLASE L	94,8%	98,3%	90.6%	91.7%
RED CLASE R	89,1%	97,6%	88.9%	89.7%

Se observa como el mejor rendimiento se obtiene con PCA, por lo tanto se van a probar los clasificadores con este método. Un dato curioso es que el uso de ICA no alcanza ni cerca el rendimiento indicado en el artículo de Yu & Chou, hice varias pruebas de obtener ICAs con distintos pre-procesados y distintos sets pero el rendimiento no aumenta. También probé distintos scripts para eliminar posibles errores de programación.

El sparse coding aunque visualmente bueno (ver diccionario en figura 4) no logra buenos resultados por sí sólo, posiblemente por la gran cantidad de elementos nulos en cada vector. Este problema se mitiga al reducir la dimensionalidad con PCA pero sigue sin ser la mejor extracción. Otro dato interesante son las pocas características seleccionadas en sparse+PCA, pasando de 36 características a 8.

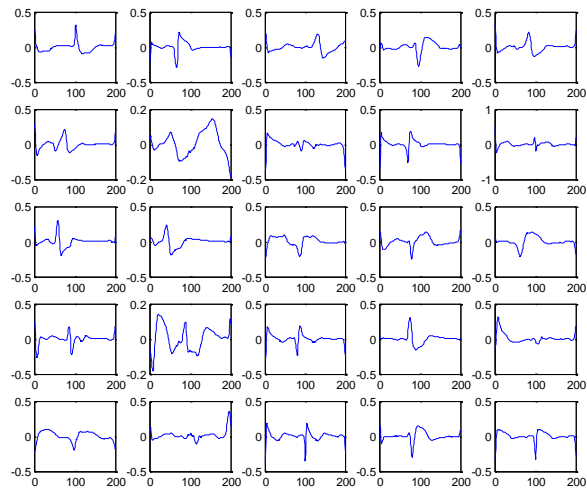


Figura 4 Parte del diccionario obtenido con sparse coding

Vale aclarar que los rendimientos presentados son el porcentaje de patrones correctamente clasificados, no el porcentaje de patrones pertenecientes a la clase indicada correctamente clasificados. Para la parte siguiente se tomarán en cuenta ambos desempeños.

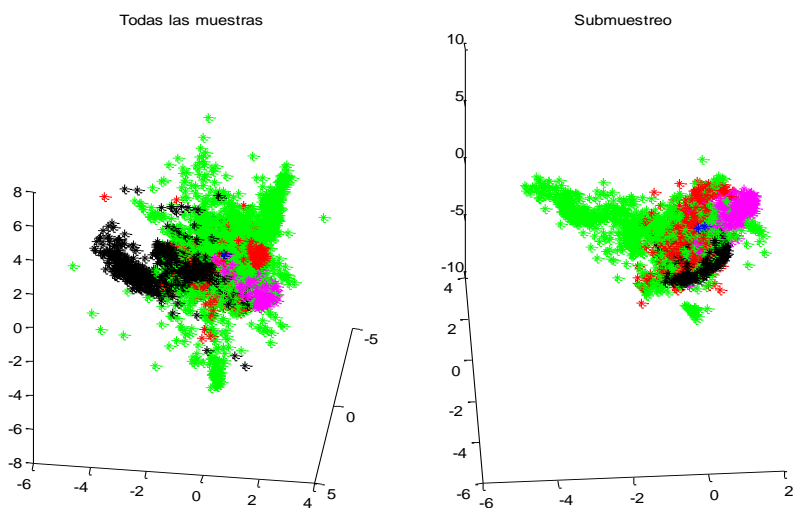


Figura 5 Tres PCA con mejor discriminación

4. Clasificadores

Para la construcción de los clasificadores se tomaron sets de 12500 muestras, 2500 por clase. El set a su vez se divide 40% para la extracción de características y entrenamiento, 20% para validación (en el caso de las ANNs) y el resto para test. Durante la extracción de los PCA en el set de rendimiento se guarda la matriz U de la descomposición svd y los parámetros del blanqueo. De esta forma se extraen las características del set de test.

```
sigmaMat = repmat(sigma,1,nt);
latidos_todos = latidos_todos - repmat(mu,1,nt);
latidos_todos = U' * (latidos_todos);
latidos_todos = latidos_todos./sigmaMat;
```

Como se vió en la figura 5 las regiones correspondientes a cada clase son de difícil separación, por lo que las ANNs son buenas candidatas, aunque se probarán otras opciones. La estrategia de clasificación será construir clasificadores individuales para cada clase y luego combinarlos. Los clasificadores individuales probados serán: ANNs, 1-NN, y Bayes. Las ANNs se entrenan utilizando el método del gradiente conjugado (Meiller 1993) y el toolbox de MatLab para redes neuronales. En principio se probaron árboles pero debido al gran costo computacional de su construcción y su bajo rendimiento en los pocos que se construyeron (menor a 70%) se descartó continuar.

%VP/%TOTAL	N	A	V	L	R
Bayes	86.2/71.0	61.0/89.9	93.6/93.1	91.2/83.8	81.5/95.9
ANN	81.6/94.8	88.5/96.1	94.0/98.2	93.0/97.8	95.1/98.5
1-NN	93.6/97.2	95.3/97.8	94.0/98.4	98.6/99.0	97.5/99.2

Aquí surge algo curioso, se esperaba a priori que las ANNs tuvieran un mejor desempeño a la regla del vecino más cercano, sin embargo para todas las clases el vecino más cercano arrojó el mejor desempeño. Por otro lado se esperaba que los mejores clasificadores de clase no utilicen siempre el mismo método, el resultado es lo contrario. Para todas las clases 1-NN fue la mejor, seguida de ANN y por último Bayes.

También se probaron métodos de clustering para separar las clases, los cuales no aportaron nada a la discriminabilidad de los patrones. Se probó k-means para 5 clases, y un clustering jerárquico (figura 6), una rápida inspección al dendrograma del clustering jerárquico permite observar que no existe ningún tipo de separación en 5 clases equiprobables. En la figura 7 se muestra el clustering obtenido con k-means, se ve como el algoritmo no encuentra distribuciones similares a las presentadas en la figura 5. De hecho el algoritmo tiende a reducir el número de clases a tres.

Antes de pasar a la combinación se muestra el rendimiento de clasificación global para Bayes, k-NN, y ANNs.

CLASIFICADOR	PERFORMANCE(%)
BAYES	76.6
ANN	93.4
1-NN	95.8
2-NN	95.8
3-NN	93.2
4-NN	93.5
5-NN	92.1

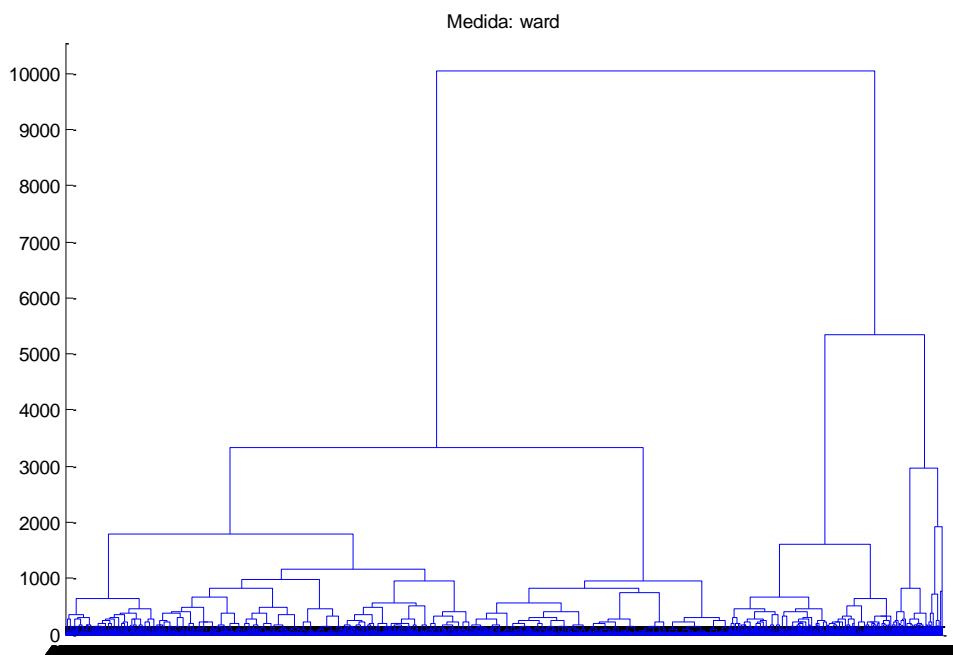


Figura 7 Dendrograma obtenido en el clustering jerárquico

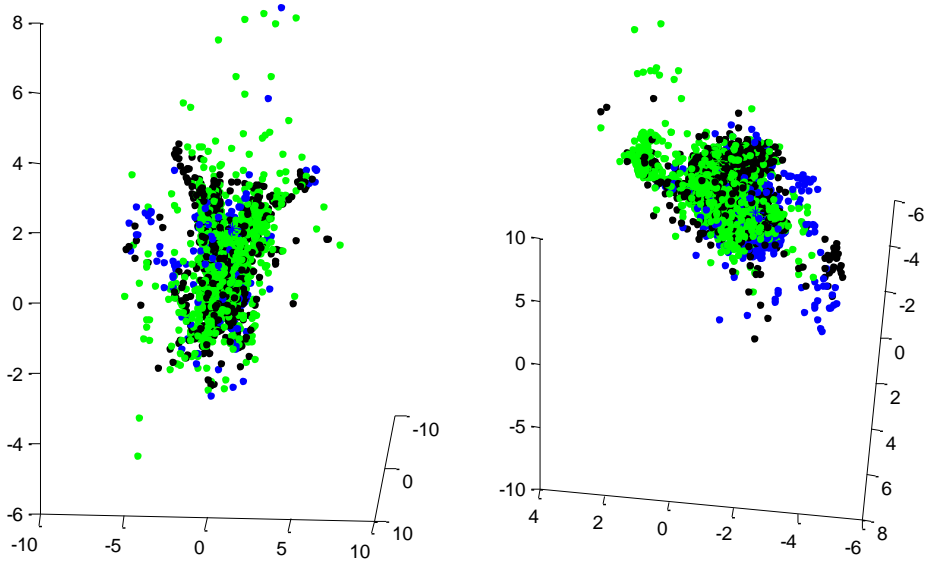


Figura 6 Clustering obtenido con k-means

5. Combinación

En principio (como se había mostrado en el informe de avances) se ideaba combinar clasificadores expertos por clase, esto no tiene sentido si todos los clasificadores expertos son 1-NN ya que nunca va a haber dos positivos distintos para una misma muestra. Por esto se cambió la estrategia y se crearon tres clasificadores (Bayes, 1-NN y ANN) para utilizar una combinación en paralelo. Se utilizaron tres criterios de combinación:

- 1) Votación simple eligiendo la clase con más votos (en caso de empate se elige el voto del 1-NN).
- 2) Se idea un ANN combinador cuyas entradas sean las salidas de los clasificadores de nivel 0. La idea conceptual de esta red es que podría aprender cuando los clasificadores se equivocan. Se espera tener un mejor desempeño que ante la votación simple.
- 3) Gating con 1-NN y ANN, en caso de clasificar distinto se elige la clase que tenga más verosimilitud en el clasificador bayesiano.

COMBINADOR	PERFORMANCE
VOTACION	95.9%
ANN	93.9%
GATING	98.0%

Tiene sentido que el combinador por ANN tenga mal rendimiento, ya que estrictamente está clasificando atributos nominales (la clase indicada en el nivel 0) a las que se les dieron un valor numérico. La votación apenas mejora el rendimiento del clasificador 1-NN. Por otro lado el combinador con gating sí logra una mejora significativa en el desempeño.

Ya que la mayoría de las clasificaciones erróneas se dan cuando el objetivo es la clase "N", se modificó el gating para que decidiera "N" si el clasificador 1-NN así lo decidía, esto empeoró el rendimiento llevándolo a 96.5%.

No se han presentado en este informe los rendimientos presentados en el informe de avances, ya que como se dijo antes estos tenían por error un número reducido de pacientes (Por lo que se lograban rendimientos cercanos al 98%). Aún así el último resultado usando gating logra un rendimiento mayor sobre 44 pacientes frente a los 24 pacientes usados anteriormente.

Confusion Matrix

Output Class	1	2436 19.5%	40 0.3%	16 0.1%	19 0.2%	14 0.1%	96.5% 3.5%
	2	27 0.2%	2431 19.4%	32 0.3%	2 0.0%	17 0.1%	96.9% 3.1%
	3	12 0.1%	14 0.1%	2443 19.5%	5 0.0%	3 0.0%	98.6% 1.4%
	4	23 0.2%	14 0.1%	8 0.1%	2469 19.8%	0 0.0%	98.2% 1.8%
	5	2 0.0%	1 0.0%	1 0.0%	5 0.0%	2466 19.7%	99.6% 0.4%
			97.4% 2.6%	97.2% 2.8%	97.7% 2.3%	98.8% 1.2%	98.6% 1.4%
		Target Class					
		1	2	3	4	5	

Figura 8 Matriz de confusión para Gating

Visualizando algunos de los patrones mal clasificados pueden obtenerse algunas conclusiones. En la figura 9 se muestra un patrón mal clasificado, se observa que el ruido de la señal es tan grande que la detección visual de los complejos es casi imposible. En la figura 10 se muestra un latido normal que fue clasificado en la clase L, más allá de que el latido sea normal se observa un complejo T muy grande lo cual es frecuente en la clase L, el modelo utilizado no tuvo la versatilidad suficiente para clasificarlo correctamente.

Aunque en general los patrones mal clasificados son casos extremos, sigue habiendo casos donde la mala clasificación es evidente (figura 11). Esto da a entender que el modelo puede seguir mejorando.

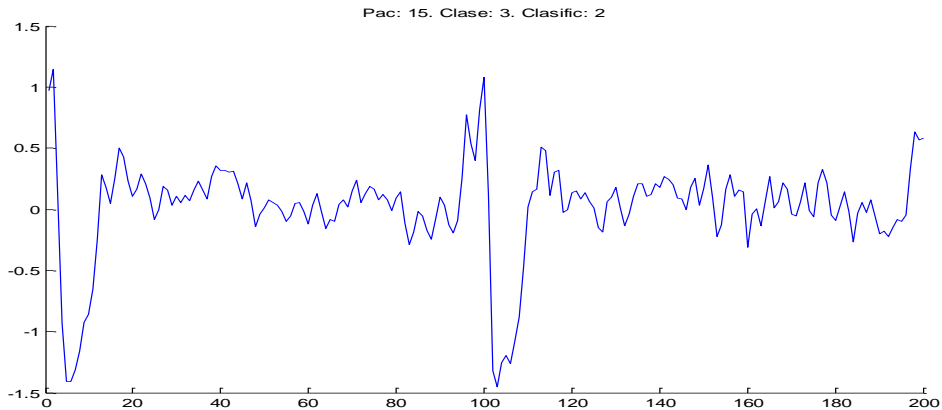


Figura 9 Clase R, Clasificado L

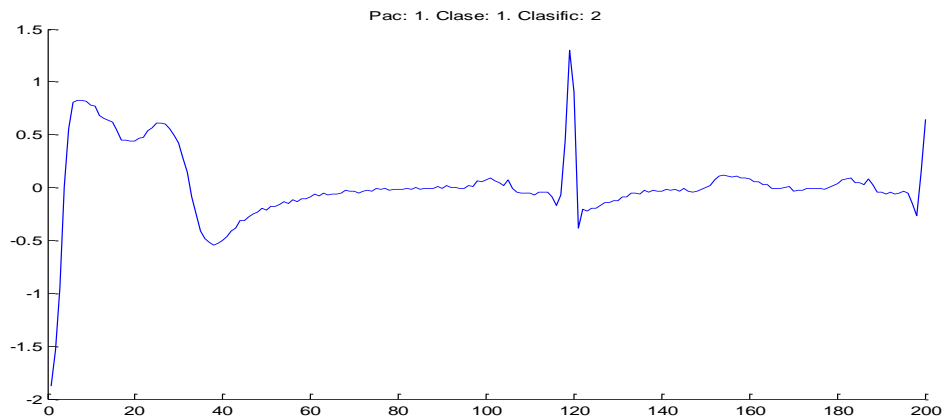


Figura 10 Clase 8, Clasificado 3

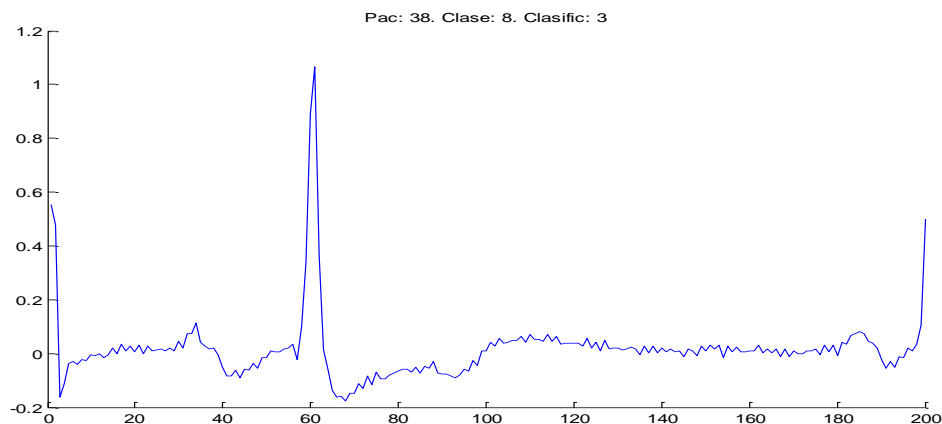


Figura 11 Clase N, Clasificado L

6. Conclusiones

Es posible lograr un rendimiento aceptable en señales temporales utilizando una buena extracción y selección de características. Más allá de las diferencias obtenidas con los distintos clasificadores, si se clasifica la señal en bruto se obtienen performances cercanas al 20% (igual probabilidad de acertar que de errar) lo cual fue ampliamente mejorado en cualquiera de los métodos utilizados para extraer características, luego incluso utilizando clasificadores simples como Bayes se obtuvieron rendimientos mayores a 75%. En cuanto a la extracción mediante sparse coding, si bien con los clasificadores utilizados no se lograron buenos resultados, los diccionarios obtenidos aparentan ser útiles visualmente, debería estudiarse mejor su implementación.

Es importante también el correcto pre-procesamiento de las señales, en este caso el filtrado de continua contribuyó a aumentar el rendimiento, mientras que el submuestreo efectivamente permitió mitigar el efecto de la frecuencia cardíaca.

En cuanto a los clasificadores utilizados sorprendió ver que 1-NN obtenía el mejor rendimiento para todas las clases. Debido a esto tuvo que descartarse la estrategia inicial de tener clasificadores expertos por cada clase y combinarlos de forma paralela. En su lugar se utilizó un pseudo-gating para combinar dos clasificadores fuertes (ANN y 1-NN), utilizando como desempate Bayes, el desempeño obtenido fue el mejor de los probados.

Los métodos de clustering probados no lograron obtener clases similares a las buscadas, por lo que se descartó usarlos para la clasificación.

Queda abierta la posibilidad de utilizar AdaBoost con clasificadores débiles y estudiar su desempeño.

7. Referencias

- Honglak Lee, Alexis Battle, Rajat Raina, A.Y.N., 2006. Efficient Sparse coding algorithms. *Advances in neural information processing systems*, pp.801–808.
- Meiller, M.F., 1993. ORIGINAL CONTRIBUTION A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. , 6, pp.525–533.
- Rodríguez, R. et al., 2015. Feature Extraction of Electrocardiogram Signals by Applying Adaptive Threshold and Principal Component Analysis. *Journal of Applied Research and Technology*, 13(2), pp.261–269. Available at: <http://linkinghub.elsevier.com/retrieve/pii/S1665642315000103>.
- Yu, S.N. & Chou, K.T., 2008. Integration of independent component analysis and neural networks for ECG beat classification. *Expert Systems with Applications*, 34(4), pp.2841–2846.