

# Control de mouse através de señales EOG y algoritmos de Boosting

Diana Mahecha, Fernando Lozano, Elkin García,

**Resumen**—La creación de alternativas de acceso a la tecnología orientadas a personas discapacitadas, es un área de gran interés en la actualidad. El desarrollo de software asociado a estos elementos tiende a la implementación de interfaces gráficas, cuyo principal acceso es el cursor controlado por el Mouse. Crear una interfaz entre un usuario discapacitado y este control, es un paso muy importante para permitir su interacción exitosa con diferentes tecnologías. Este documento describe el diseño y la implementación de un control de Mouse manejado por potenciales bioeléctricos, dirigido a personas con algún tipo de discapacidad física-motora. Se centra en la adquisición de señales electrooculográficas (EOG) y su clasificación mediante técnicas del área de *Machine Learning*, más específicamente, por medio de la aplicación de algoritmos de *Boosting* Multivariable usando un clasificador lineal como aprendizaje débil. Se obtienen resultados competitivos en la generalización.

**Index Terms**—Electrooculografía, Boosting, Aprendizaje, Adquisición, Boosting Multivariable, Mouse

## I. INTRODUCCIÓN

ACTUALMENTE, el uso creciente de los computadores ha llevado a crear alternativas de acceso a los mismos para personas con diferentes grados de discapacidad motora. Existen distintos tipos de mouse alternativos al tradicional, cuyo desarrollo consiste principalmente en 3 áreas: El reconocimiento de imágenes de video, el uso de elementos físicos fáciles de manejar con partes del cuerpo diferentes a las manos y el análisis de potenciales bioeléctricos [1] [2] [3] [4]. De acuerdo al grado de discapacidad se debe escoger la opción que brinde más comodidad al usuario.

En el análisis de potenciales bioeléctricos se usan diferentes técnicas: electromiografía (EMG), electroencefalografía (EEG) y electrooculografía (EOG). Este trabajo se enfoca en el análisis EOG y el uso técnicas de clasificación del área de Machine Learning (*Boosting* Multivariable) para desarrollar un control de mouse por medio del análisis EOG.

A continuación, en la sección II se explican los conceptos generales de la técnica de EOG y el aprendizaje supervisado con *boosting*. En la sección III se desarrolla la implementación y diseño de la solución. En la sección IV se presentan los resultados y finalmente en la sección V las conclusiones.

## II. PRELIMINARES

### A. Electrooculografía

El análisis electrooculográfico obtiene una apreciación directa de la movilidad ocular mediante la medición de voltajes diferenciales alrededor del ojo [5]. Hay 4 movimientos oculares entre los cuales están los sacádicos [6]. Estos son desplazamientos angulares muy rápidos y precisos que se producen al observar

D. Mahecha. Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Bogotá, Colombia, email: d-mahech@uniandes.edu.co

F. Lozano. Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Bogotá, Colombia, email: flozano@uniandes.edu.co

E. García. Departamento de Ingeniería Eléctrica y Electrónica, Universidad de los Andes, Bogotá, Colombia, email: elkin-ga@uniandes.edu.co



Figura 1. Comportamiento del ojo humano como dipolo (derecha) y ubicación de electrodos para el análisis EOG (izquierda). Los electrodos B y C toman la señal vertical, los electrodos D y E toman la señal horizontal y el electrodo A sirve de referencia. Imagen tomada de [1]

con detalle algún objeto y son los únicos movimientos que se controlan voluntariamente.

La fuente de la señal EOG es el potencial continuo existente entre la cornea y la base del ojo. De esta forma el ojo se comporta como un dipolo que se mueve en un conductor de volumen no homogéneo (la cabeza). Con el uso de 5 electrodos puestos alrededor de los ojos (Figura 1) se pueden medir estos potenciales y ser usados en la detección de la dirección de la mirada respecto a la cabeza. [7]

Muchos estudios han intentado determinar la relación entre el voltaje del EOG y el ángulo de mirada. Rafael Barea [6] realizó un amplio estudio acerca de este tema y se resaltan los siguientes resultados:

- La señal EOG varía entre 50 y 3500  $\mu V$  con un rango de frecuencias de DC-35Hz y tiene un comportamiento lineal para ángulos entre  $\pm 30^\circ$ .
- La señal EOG se ve perturbada por otros biopotenciales (EEG, EMG) y las perturbaciones del sistema de adquisición.
- El EEG y el EMG pueden producir errores de aproximadamente un grado.
- La forma del EOG para movimientos sacádicos es similar en cualquier persona, por eso al usar una normalización de los datos se obtiene un modelo EOG universal.
- Los movimientos sacádicos pueden detectarse en función de la derivada del EOG.

### B. Aprendizaje supervisado y Boosting

El aprendizaje supervisado está fundamentado en una programación por ejemplos, en donde el algoritmo conoce la respuesta correcta a los patrones de entrada y encuentra una regla de clasificación (hipótesis) que asigna una etiqueta a cada elemento en el espacio de entrada. El objetivo de la hipótesis no es responder bien a los datos de entrenamiento, sino a nuevos datos, por consiguiente, es necesario realizar validación cruzada. El desempeño es evaluado por medio de los errores de entrenamiento y generalización. En el primero se evalúan los datos nuevos (que no han sido pasados por el algoritmo) y en el segundo los datos

con los que se aprende. La validación cruzada consiste entonces, en un compromiso entre estos errores, ya que si el error de entrenamiento se hace muy pequeño puede existir sobre ajuste a los datos y no generalizar.

Los algoritmos de *boosting* toman un algoritmo débil (algoritmo que retorna una hipótesis para cualquier distribución con un error arbitrariamente cercano a la probabilidad de etiquetar bien un dato adivinando: 50%) que busca minimizar el error empírico y a partir de llamadas iterativas a éste, generan una hipótesis fuerte cuyo error es arbitrariamente pequeño.

*Adaboost* [8] es un algoritmo de *boosting* adaptativo y consiste en la creación de una hipótesis fuerte a partir de la combinación lineal de hipótesis débiles. Los pesos de esta combinación lineal son modificados en cada iteración de acuerdo al error de entrenamiento de la hipótesis retornada por el aprendiz débil, de tal forma que en la siguiente iteración del algoritmo, el objetivo es aprender los datos en los que se equivoca la hipótesis anterior. En este caso los errores se hallan respecto a la distribución  $D_t$  de la iteración  $t$  y están dados por:

$$\epsilon_t = P_{r_i \sim D} [H_t(x_i) \neq y_i] = \sum_{i: h_t(x_i) \neq y_i} D_t(i) \quad (1)$$

### III. DISEÑO E IMPLEMENTACIÓN

#### A. Adquisición señales EOG

La arquitectura del sistema de adquisición consta esencialmente de dos partes:

1. Acondicionamiento de la señal: Las señales EOG horizontal y vertical se toman por medio de electrodos superficiales pediátricos de Ag/AgCL y son amplificadas 500 veces con amplificadores de instrumentación INA128. Sigue una etapa de aislamiento implementada con un ISO124 que independiza la alimentación de los elementos conectados al usuario de la alimentación del resto del circuito. Se implementa un filtro pasa banda entre el ancho de banda de señales EOG [10mHz-35Hz], conformado por un filtro pasa bajas (2º orden) y un filtro pasa altas (4º orden), con frecuencias de corte en los límites de la banda. Este filtraje elimina ruidos producidos por el movimiento de músculos cercanos, la red eléctrica y la componente continua del EOG. Finalmente se realiza un acondicionamiento para poner la señal en el rango del ADC [0V,5V].
2. Digitalización y transmisión al computador: Esta compuesta por la conversión análoga-digital y la transmisión serial al computador. Para esto se usa el PSOC CY8C27443-24PXI que cuenta con un ADC dual configurado a 105Hz para cada canal y un módulo de transmisión serial.

#### B. Construcción de la base de datos

Para conformar la base de datos necesaria en el entrenamiento, se implementó una aplicación en la que se programaron 10 protocolos con el objetivo de registrar diferentes movimientos sacádicos del ojo, variando las direcciones y la magnitud de los desplazamientos. La prueba tiene una duración de 5 minutos aprox. y consiste en que el usuario manteniendo la cabeza en el centro de la pantalla y aproximadamente a 30cm de la misma, siga el movimiento de un indicador y en determinados momentos realice parpadeos voluntarios.

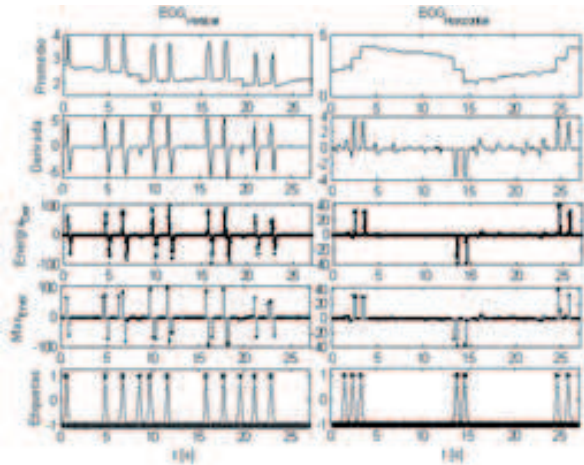


Figura 2. Preprocesamiento de los datos de un protocolo. Las flechas en la gráfica indican un 1 en la columna de codificación de la dirección que indican y la estrella un 1 en la columna de click

La toma de datos se realizó con 20 personas entre 21 y 58 años, sin discapacidad motora. Esto no representa ningún problema ya que como se mencionó anteriormente, la forma del EOG es similar en todas las personas y en el caso que los usuarios discapacitados presenten algún tipo de variación considerable, el sistema puede ser fácilmente adaptado tomando una nueva base de datos con ellos y corriendo el algoritmo de entrenamiento con ella.

#### C. Preprocesamiento de los datos

Antes de utilizar los datos en cualquier algoritmo de aprendizaje, es necesario extraer de ellos determinadas características sobre las cuales se va a crear la hipótesis. En este caso estamos interesados en detectar los movimientos sacádicos producidos por el ojo y según [6] una forma de hacerlo es evaluando la derivada de la señal. El preprocesamiento realizado fue el siguiente:

- Para eliminar ruidos se promedia cada señal con intervalos de 47.61ms (cada 5 datos).
- Se halla una aproximación a la derivada en cada dato promediado por medio de la ecuación 2, en donde tomamos  $\Delta t = 333,33ms$ , que es el tiempo transcurrido entre 7 datos promediados.

$$\frac{\delta \bar{X}}{\delta t} = \frac{\Delta \bar{X}}{\Delta t} = \frac{\bar{X}_i - \bar{X}_{i+7}}{t_i - t_{i+7}} \quad (2)$$

- Se mide la energía de cada 3 datos de la derivada conservando el signo para hacer más evidentes los cambios y por último se escoge la energía de máxima magnitud entre 5 datos consecutivos.

En la figura 2 se muestra el preprocesamiento para uno de los protocolos implementados, junto con el etiquetamiento realizado gracias a códigos guardados durante la ejecución del programa de adquisición. Después de etiquetar los datos, se dividieron por clases y se repartieron aleatoriamente en dos grupos, uno para entrenamiento y el otro para evaluación. Cada grupo quedó conformado con aproximadamente 2500 datos.

**Algoritmo 1** Algoritmo débil usado.

```

Entrada:  $S = \{x_{ord}, y_{ord}\}_i^m, D$ 
- Inicializar  $e_{min} = mx_{ant} = x_{ord}(1) - 0,5$ 
for all  $t = 1 : m$  do
   $h_t = \frac{x_{ord}(t) - a_{ant}}{2}$  [ If  $x_i \geq h_t \rightarrow y_i = 1$ ; else  $y_i = -1$  ]
  - Evaluar el error pesado  $e_t$  de  $h_{ip_t}$  (respecto a  $D$ ).
  if  $e_t < 0,5$  then
     $e_t = (1 - e_t)$  - Invertir la polaridad de la hipótesis
  end if
  if  $e_t < e_{min}$  then
     $e_{min} = e_t$ ;  $h_{out} = h_t$ 
  end if
   $x_{ant} = x_{ord}(t)$ 
end for
Salida:  $h_{out}$ 

```

**D. Adaboost Multivariable**

Originalmente *Adaboost* es un algoritmo diseñado para problemas binarios, sin embargo, existen varios métodos para extenderlo a problemas multivariable. Se han propuesto varias aproximaciones como AdaBoost.M1 [9] que usa un aprendiz débil multivariable y AdaBoost.MH [8] o AdaBoost.M2 [10] que crean problemas binarios al comparar las clases entre sí. También hay acercamientos usando métodos de codificación de salida, como AdaBoost.MO [11] que usa una matriz de codificación (código por clase) y modifica la salida de la hipótesis final midiendo su distancia a los códigos y AdaBoost.OC [10] que usa codificación EOCO.

El objetivo de la implementación de algún algoritmo de *boosting* multivariable en la clasificación de señales EOG es tomar los datos bidimensionales entregados por el preprocesamiento y retornar la dirección del movimiento realizado. Para esto se definen 5 movimientos básicos: arriba, abajo, derecha, izquierda y clic (parpadeo voluntario) y 4 movimientos diagonales, que corresponden a combinaciones de los anteriores. La codificación usada se presenta en la figura 3

Movimiento	Arriba	Abajo	Derecha	Izquierda	Clic
Arriba	1	0	0	0	0
Abajo	0	1	0	0	0
Derecha	0	0	1	0	0
Izquierda	0	0	0	1	0
Clic	0	0	0	0	1
Arriba-Derecha	1	0	1	0	0
Arriba-Izquierda	1	0	0	1	0
Abajo-Derecha	0	1	1	0	0
Abajo-Izquierda	0	1	0	1	0
Clic-Derecha	0	0	1	0	1
Clic-Izquierda	0	0	0	1	1

Figura 3. Codificación movimientos. A: Mov. básicos, B: Mov. diagonales.

Debido a la forma de los datos y a su proporcionalidad con el movimiento del ojo, se usa como clasificador débil un separador lineal. La implementación de este clasificador corresponde al algoritmo 1 y recibe como entradas un vector con los datos ordenados de la dimensión  $l$  y las respectivas etiquetas ordenadas. El algoritmo recorre los datos generando la hipótesis  $i$  y evalúa su error pesado (ecuación 1) en todos los datos. Luego, si este error es mayor a 0.5 invierte la polaridad de la hipótesis y halla el respectivo error  $(1 - e_i)$ . Después se compara con el mínimo error hasta el momento y si es menor, se guarda la hipótesis  $i$ . Las hipótesis  $i$  recorren los puntos medios de los datos, es decir, es el valor medio entre un dato  $i$  y el dato inmediatamente anterior a él.

Como el clasificador débil es binario se plantea la solución creando varios problemas binarios a partir del problema multivariable. La exigencia en la precisión del algoritmo débil es mucho más alta en los problemas multivariable, ya que la probabilidad

**Algoritmo 2** Algoritmo de *boosting* multivariable usado.

```

Entrada:  $S = x, y, y_{i=1}^m$  donde  $x_i \in X, Y_i \subseteq \gamma$ 
for all  $i = 1 : T$  do
  for all  $l = 1 : L$  do
    - Correr alg. débil usando la distribución  $D$  y obtener la hip. débil  $h(x)_t^l$  con error:  $\epsilon_t(l) = P_{r_i \sim D}[h_t(x_i) \neq y_i]$  - Escoger:

```

$$\alpha_t(l) = \frac{1}{2} \ln \frac{1 - \epsilon_t(l)}{\epsilon_t(l)} \quad (3)$$

- Actualizar:

$$D_{t+1} = \frac{D_t(i, l) \exp(-\alpha_t Y(i, l) h_t(X_i, l))}{Z_t} \quad (4)$$

donde  $Z_t$  es un factor de normalización para que  $D_{t+1}$  sea una distribución.

$$Z_t = 2\sqrt{\epsilon_t(l)(1 - \epsilon_t(l))}$$

**end for**  
**end for**  
Salida:

$$H(x, l) = \text{sign}\left[\sum_{t=1}^T \frac{\alpha_t}{\sum_t \alpha_t} h_t(x, l)\right] \quad (5)$$

de que al adivinar la etiqueta, ésta sea correcta, es menor al 50%. Al evaluar la hipótesis débil con AdaBoost.MH, se encontró que no tiene el nivel de precisión requerido y por consiguiente se realiza una adaptación del algoritmo *Adaboost* (cuya exigencia a la hipótesis débil es menor), a la lógica usada por AdaBoost.MH. Para esto se crea una distribución  $D$  a cada clase de datos y se actualizan los valores de  $\alpha_t$  y  $Z_t$  de acuerdo a *Adaboost*. El resultado es el pseudo-código presentado en la figura 2, el cuál es el algoritmo implementado en la clasificación de las señales EOG.

**E. Mouse Electrooculográfico basado en Boosting**

La solución final del mouse electrooculográfico consta de dos etapas. La primera es la tarjeta de adquisición de señales descrita en el apartado A y la segunda es la aplicación EOG mouse.

EOG mouse se desarrollo en Visual Studio.NET 2005 con VB.net como lenguaje y está compuesto de los siguientes módulos:

1. RX Serial: Se encarga de la recepción de los datos de la tarjeta de adquisición. Organiza los datos en parejas (Med. Vertical, Med. Horizontal).
2. Preprocesamiento de los datos: Consiste en el algoritmo descrito en el apartado C pero en vez de recorrer un vector, procesa secuencialmente los datos entregados por el módulo 1.
3. Hipótesis *Boosting*: Es la hipótesis combinada obtenida del algoritmo de *boosting* implementado. La hipótesis se carga al iniciar la aplicación desde archivos .txt a vectores. Se implementó una rutina para evaluar la hipótesis con cada par de datos provenientes del preprocesamiento, dando como resultado un vector con 5 posiciones que representa el movimiento detectado.
4. Acciones mouse: En esta etapa se identifican los movimientos realizados usando la codificación descrita y de acuerdo a ellos se utilizan rutinas disponibles en Visual Studio y en internet para mover el cursor y simular el click.

TABLE I

COMPARACIÓN DE PORCENTAJE DE ACIERTOS DE ENTRENAMIENTO Y GENERALIZACIÓN

Clases	1	2	3	4	5	Total
Ent. [%]	98.56	99.15	98.92	98.74	99.77	96.55
Gen. [%]	98.08	99.10	98.68	91.61	99.76	95.41

## IV. RESULTADOS

En la tabla I se muestran los resultados obtenidos al correr el algoritmo 100 veces. Podemos ver que en general no hay ninguna clase que sea más difícil de aprender que la otra, ya que todas llegan a errores bajos. La clase con más precisión en generalización es la 5, lo cual tiene sentido ya que sus valores están en un rango diferente al del resto. Dado que 100 es un número bajo de iteraciones, comparado al número de entrenamientos que suelen hacerse en *boosting* (500-1000), podemos decir que los resultados obtenidos son muy buenos. Esto evidencia la eficacia del algoritmo débil que es potencializada por la aplicación del *boosting*.

En la Figura 4 se presentan los resultados para 500 iteraciones. Usando validación cruzada, podemos ver que el punto mínimo de ambos errores se da en 271 iteraciones (4.2% para generalización y 2.96% para entrenamiento). A partir de este valor, aunque el error de entrenamiento sigue decreciendo, el de generalización empieza a aumentar, lo cual indica que las hipótesis se sobre ajustan a los datos. De acuerdo a esto, se escoge como hipótesis combinada final la lograda en la iteración 271.

Aunque hay varias aplicaciones de este tipo y diferentes estudios en torno al tema, muy pocos dan claridad acerca de los métodos de clasificación usados y más aún, de la efectividad porcentual de los mismos en una base de datos considerable. Al comparar los resultados obtenidos, con la implementación de un mouse que usa señales EMG y redes neuronales [12] que reporta errores inferiores al 5%, se observa que el sistema de clasificación implementado es altamente competitivo.

## V. CONCLUSIONES

En este trabajo se desarrolló una forma alternativa de acceder a diferentes tecnologías para usuarios discapacitados, buscando mejorar su habilidad para desenvolverse libremente en el mundo actual.

En el desarrollo del sistema de control propuesto, se analizaron características importantes de las señales EOG y los requerimientos necesarios para desarrollar una adecuada instrumentación para su digitalización. Además de esto, se realizó un nuevo acercamiento al análisis EOG, partiendo de algoritmos de *boosting* multivariable para la clasificación de las señales. El preprocesamiento de los datos logró no solo identificar los movimientos sacádicos sino también condensar su información en muy pocos datos, de tal forma que se pudo asociar cada dato a una clase sin perder mucha información en el proceso.

El algoritmo de *Boosting* Multivariable implementado permitió encontrar una buena solución, que no se sobre ajusta a los datos y generaliza bien en los mismos.

Quedan planteados para futuros trabajos, la implementación

de las funciones disponibles en un mouse que no se obtuvieron

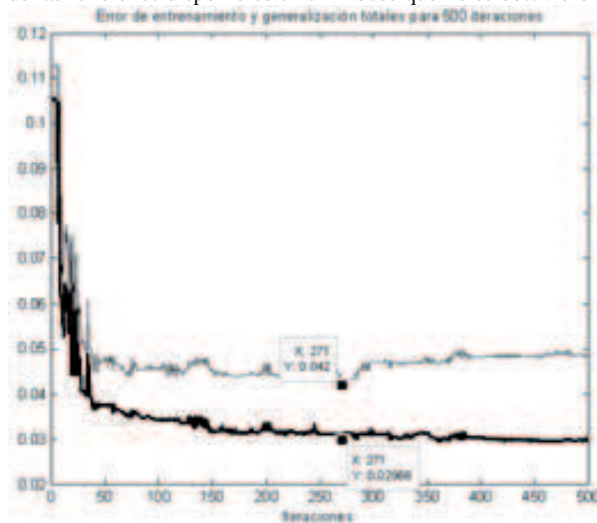


Figura 4. Errores de entrenamiento y generalización (500 iteraciones).

acá; además del diseño de una interfaz con el usuario más amigable. También es necesario realizar pruebas adicionales de validación con un grupo considerable de personas, preferiblemente con algún tipo de discapacidad. Desde el punto de vista del aprendizaje, se puede buscar implementar Ada-Boost.MH u otro algoritmo multivariable, algunos aquí nombrados, con otras hipótesis débiles para así poder comparar su funcionamiento en la clasificación de señales EOG.

## REFERENCIAS

- [1] R. Barea y otros, Diseño de un ratón electrooculográfico para el control de Interfaces Gráficas. España. [Online]. Available: [http://www.depeca.uah.es/personal/barea/electrooculografia/saaei00\\_ratón.pdf/](http://www.depeca.uah.es/personal/barea/electrooculografia/saaei00_ratón.pdf/). Consultada en: 12/2007
- [2] J. Gips y otros *Eagle Eyes Project*, Boston College. [Online]. Available: <http://www.bc.edu/schools/csom/eagleeyes/contact.html>. Consultada en: 12/2007
- [3] J. Gips, M. Betke, *CameraMouse.org.*, Boston College. [Online]. Available: <http://www.cameramouse.org/index.html>. Consultada en: 12/2007
- [4] Technologies, Brain Actuated. *Cyberlink - Brainfingers: Hands-free Computer Access Solution*, [Online]. Available: <http://www.brainfingers.com/>. Consultada en: 12/2007
- [5] F. Rodríguez, E. Arenas *Oftalmología básica*, vol. 2, 1999.
- [6] R. Barea, *Interfaz Usuario-Maquina basado en electrooculografía. Aplicación a la movilidad*, Tesis Doctoral, Universidad de Alcalá. Madrid, España., 2001. [Online]. Available: <http://www.depeca.uah.es/personal/barea/tesis/tesis.htm>. Consultada en: 12/2007.
- [7] L. Geddes *Principles of applied biomedical instrumentation.*, New York : John Wiley & Sons, 1975.
- [8] R. Schapire *A Brief Introduction to Boosting.*, Proceeding of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 1401-1406, 1999.
- [9] Y. Freund, R. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting.*, Journal of Computer and System Sciences., vol. 55, pp. 119-139, 1997.
- [10] Y. Freund, R. Schapire, *Using Output codes to boost multiclass learning problems.*, Machine Learning: Proceedings of the Fourteenth International Conference., pp. 313-312, 1997.
- [11] R. Schapire, Y. Singer *Improved boosting algorithms using confidence-rated predictions.*, Machine Learning, vol. 37, 297-336, 1999.
- [12] Z. Huertas *Sistema de control de mouse mediante señales electromiográficas faciales.*, Tesis de maestría. Universidad de los Andes, 2007.