

# Práctica de Técnicas de Inteligencia Artificial.

## Segunda práctica: Visión Artificial

Miguel Cazorla, Vicente Morell y Diego Viejo  
Departamento de Ciencia de la Computación e Inteligencia Artificial  
Universidad de Alicante

Curso 2012 - 2013

### 1. Descripción de la práctica

La práctica de visión artificial para el curso 2012/13 consistirá en el desarrollo de la implementación en Java del algoritmo Adaboost. Dada una interfaz gráfica en donde podéis indicar los ejemplos positivos y negativos clickeando botones distintos en el ratón, se usan esas coordenadas en  $R^2$  como datos de aprendizaje. En la Figura 1 se muestra la interfaz gráfica con los ejemplos de cada clase (rojos y azules) y la salida del Adaboost con los clasificadores débiles seleccionados (los puntos en azul oscuro fueron erróneamente clasificados). El resultado final generado por texto fue: Encontrados 20 clasificadores. Tasa de aciertos: 0.8235294117647058. La interfaz gráfica, o GUI, es la clase principal que dispara el aprendizaje y muestra los resultados.

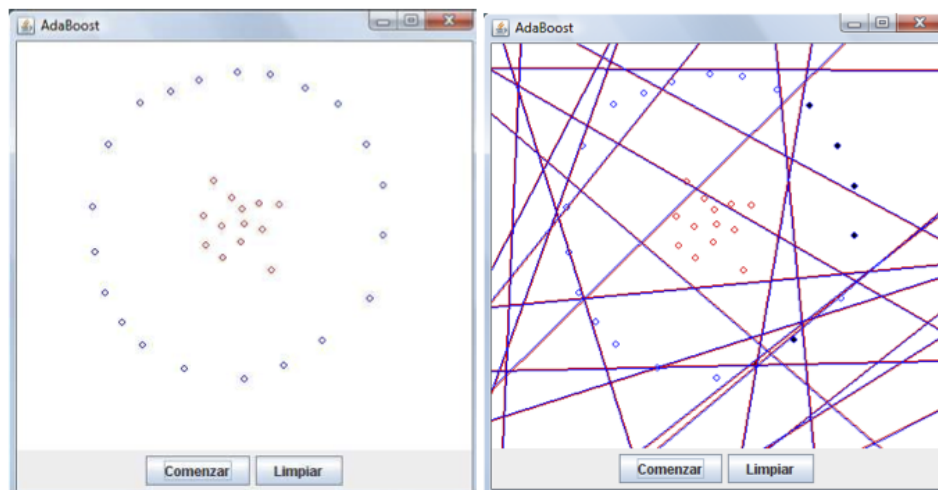


Figura 1: Interfaz gráfica. Izq: Entrada de datos para un problema de clasificación difícil. Der: Salida del clasificador fuerte mostrando los clasificadores débiles.

Con respecto a Adaboost, podéis seguir las transparencias de clase, y de forma orientativa las que hemos puesto en la web sobre *Principles of Boosting and Adaboost*. De cara a la implementación deberéis tener en cuenta las siguientes consideraciones:

- Clasificadores débiles: Para entrenar un clasificador débil se generarán  $K$  líneas aleatoriamente y se seleccionará aquella con menor error (suma de las probabilidades  $D_t(i)$  para cada punto  $i$  mal clasificado). En cuanto a la parametrización de la recta se puede usar el esquema  $w_1x_1 + w_2x_2 + \theta = 0$  o bien  $x_1\cos\theta + x_2\sin\theta = \rho$ .
- Conjunto de entrenamiento: Cada clasificador débil se entrenará con TODOS los ejemplos de entrenamiento suministrados a través de la interfaz gráfica. En términos de Boosting esto significa que no aplicaremos ninguna estrategia de muestreo.
- El valor correcto del peso de cada clasificador es:

$$\alpha_t = \frac{1}{2} \log \left( \frac{1 - \epsilon_t}{\epsilon_t} \right), \quad (1)$$

siendo  $\epsilon_t$  el error del clasificador  $h_t$ .

- El número máximo  $T$  de clasificadores deberá fijarse atendiendo a evitar el *overfitting*, esto es, si con  $T' > T$  se consigue la misma tasa de acierto, es conveniente usar  $T$  clasificadores. Por supuesto deberá parar cuando la tasa de aciertos sea de 1 (100 %).

## 2. Parte experimental

Aparte del proceso de implementación del algoritmo Adaboost, el alumno deberá desarrollar un proceso de experimentación. Habrá que ajustar pues tanto el número de rectas  $K$  necesarias para inferir cada clasificador débil, como el número  $T$  de clasificadores necesarios, siguiendo el principio de evitar el sobreentrenamiento. A continuación, los experimentos realizados por el alumno deberán buscar los límites del Adaboost para, razonadamente, explicar bajo qué condiciones (distribución de los datos de entrada) el error obtenido no puede reducirse más. En caso de que se haya implementado una parte opcional, el alumno deberá demostrar mediante experimentos cómo la parte opcional incorporada mejora las tasas de reconocimiento.

## 3. Documentación

La práctica irá acompañada de la correspondiente documentación. En dicho documento, el alumno explicará y justificará razonadamente los detalles de diseño de la función implementada, experimentación, y resultados.

## **4. Parte optativa**

Como parte optativa de la práctica se deja al alumno la posibilidad de mejorar el clasificador débil usando uno basado en la regla delta vista en clase de teoría.

## **5. Normas de entrega y evaluación de la práctica**

La práctica es individual. Cualquier indicio de plagio supondrá el suspenso de la parte de prácticas y, por consiguiente, de la presente convocatoria de la asignatura. Se avisa al alumno que existe la posibilidad de utilizar sistemas automáticos de detección de copia.

La práctica se valorará de la siguiente manera:

- 25 % por código y funcionamiento. Se probarán varios ejemplos y se valorará el funcionamiento final. Si la práctica no cumple con un mínimo de funcionamiento, se suspenderá por completo sin tener en cuenta los apartados siguientes. En cuanto al código, deberá estar debidamente estructurado e indentado y haciendo un uso adecuado del lenguaje de programación. El código deberá estar debidamente comentado (incluso para la generación de javadoc).
- 25 % por la experimentación.
- 30 % por documentación. La documentación se podrá entregar en cualquier formato. Será responsabilidad del alumno asegurar que la documentación pueda ser leída en cualquier sistema operativo.
- 20 % por la parte optativa.

Fechas de entrega de la práctica:

- El plazo de entrega de la práctica finaliza a las 24 horas del miércoles 23 de Mayo del 2013.
- La entrega se realizará a través del sitio web moodle de la asignatura.