

2022

10 / 11 / 2022

Resumen sesión teoría 10/11

SISTEMAS INTELIGENTES

ADRIAN UBEDA TOUATI 50771466R

Contenido

Boosting y Adaboost	2
¿Es posible combinando clasificadores débiles conseguir clasificadores fuertes?	2
Combinar clasificadores débiles	2
Bagging.....	2
Boosting vs Bagging	3
AdaBoost.....	4
Visión artificial	5
Procesar una imagen	5
Resolución de una imagen.....	5
Convolución.....	6

Boosting y Adaboost

Hay 2 tipos de aprendizaje:

- *Aprendizaje supervisado*: Basado en ejemplos ya clasificados, para aprender y testear
- *Aprendizaje no supervisado*: Parte sin ningún tipo de información, y debe ser capaz de saber que agrupaciones existen.

Hay 2 tipos de clasificadores, el débil (50% aciertos) y fuerte (90% aciertos)

¿Es posible combinando clasificadores débiles conseguir clasificadores fuertes?

Combinar clasificadores débiles

Responde como si fuese el azar

Bagging

Se selecciona n ejemplos del conjunto de la parte de entrenamiento, con remplazo (se pueden coger elementos repetidos)

Generamos el clasificador h_t

Y después combinamos los p clasificadores y hacemos el recuento de votos

Se ejecuta x en cada clasificador débil, y comprueba después el conjunto.

$$H(x) = MAJ(h_1(\mathbf{x}), \dots, h_T(\mathbf{x}))$$

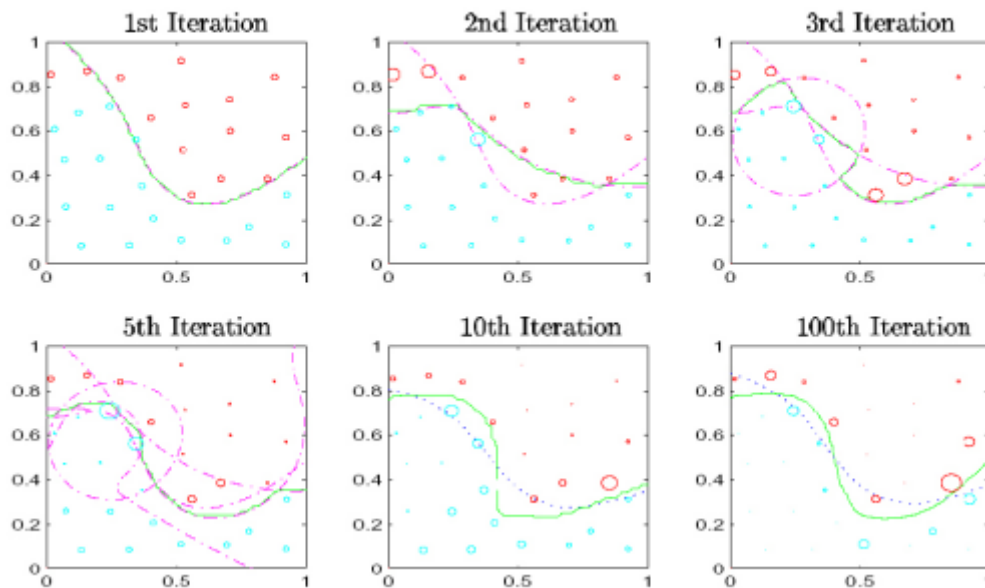
Boosting vs Bagging

Bagging no tiene en cuenta la dificultad de los ejemplos, y que cada clasificador tenga el mismo peso, independientemente de su efectividad no es una buena decisión

Boosting es una versión más avanzada de Bagging, tiene un voto ponderado y la regla tiene en cuenta la dificultad de los ejemplos y la efectividad del clasificador

Ejemplos de entrenamiento de boosting

Boosting: explicación intuitiva



AdaBoost

Que viene de Adaptive Boosting, es el algoritmo encargado de hacer la ponderación

AdaBoost. Adaptive Boosting [Freund, Schapire, 96]

- Initialize distribution over training set $D_1(i) = 1/N$.
- For $t = 1, \dots, T$
 1. Train weak learner using distribution D_t and obtain h_t .
 2. Choose a weight (confidence value) $\alpha_t \in \mathbb{R}$.
 3. Update distribution over training set:

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$$

- Set $H(x) = \text{sign}(f(x)) = \text{sign}\left(\sum_{i=1}^T \alpha_i h_t(x)\right)$

Que tendrá en cuenta la calidad de los clasificadores

El pseudo código seria:

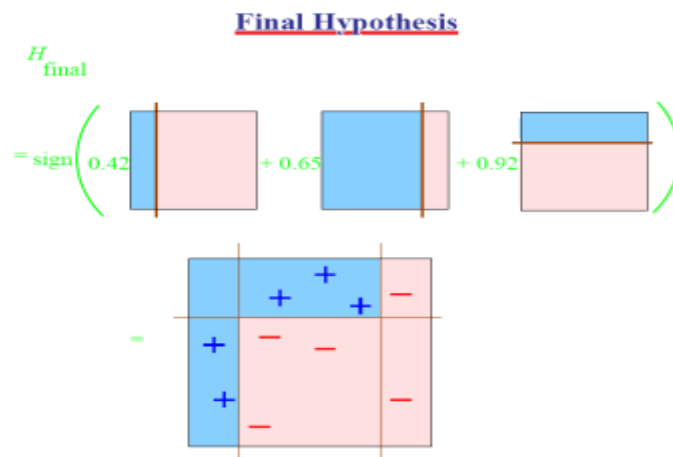
Algorithm 1 Adaboost

```

1: procedure ADABOOST( $X, Y$ )
2:    $D_1(i) = 1/N$       ▷ Indica como de difícil es de clasificar cada punto  $i$ 
3:   for  $t = 1 \rightarrow T$  do      ▷ T es el número de clasificadores débiles a usar
4:     Entrenar  $h_t$  teniendo en cuenta  $D_t$ 
5:     Start
6:       for  $k = 1 \rightarrow A$  do      ▷ A = num. de pruebas aleatorias
7:          $F_p = \text{generaPlanoAlAzar}()$ 
8:          $\epsilon_t = P_{D_t}(h_t(x_i) \neq y_i) \rightarrow \epsilon_{t,k} = \sum_{i=1}^N D_t(i) \cdot (F_k(x) \neq y(x))$ 
9:         return  $\langle F_p | \min(\epsilon_{t,k}) \rangle$ 
10:      End
11:      Del  $h_t$  anterior obtener su valor de confianza  $\alpha_t \in \mathbb{R}$ 
12:      Start
13:         $\alpha_t = 1/2 \log_2 \left( \frac{1-\epsilon_t}{\epsilon_t} \right)$ 
14:      End
15:      Actualizar  $D_{t+1}$ 
16:      Start
17:         $D_{t+1} = \frac{D_t(i) \cdot e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$ 
18:         $Z_t = \sum_i D_t(i)$ 
19:      End
20:   return  $H(x) = \text{sign}(\sum_t \alpha_t \cdot h_t(x))$ 

```

Ejemplo de una hipótesis final



Visión artificial

Sensor de captación, produce una imagen que es una matriz, y si sustituimos cada valor de la matriz por un color, daría la imagen.

El ojo nos engaña por la perspectiva, pero al computador no.

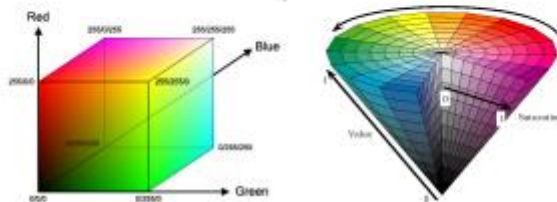
Procesar una imagen

Normalmente se hace referencia a un pixel con una tupla (x,y)

La idea es procesar una imagen $I(x,y)$ y conseguir otra imagen $G(x,y)$, resultado de aplicar algún método o algoritmo.

Modelos de color

- Dependiendo de los datos de la matriz de la imagen, tendremos distintos tipos de imagen
 - Binaria: valores 0 y 1
 - Gris: valores 0 a 255 (usualmente)
 - Color RGB: tres valores entre 0 y 255
- En cuanto al color, existen otros modelos (HSV)



Resolución de una imagen

Es el tamaño de pixeles $M \times N$ que tendrá la imagen

Se puede ampliar la imagen (inventamos pixeles, se repiten pixeles) o se pueden reducir (usar un mecanismo de eliminación de pixeles)

El brillo de una imagen aumenta el valor de cada píxel

Los errores de la imagen se llama ruido, un sistema de filtrados es robusto si al realizar una vez el filtrado es igual que hacerlo 100 veces

Convolución

En el caso continuo sería

$$f(x) * g(x) = h(x) = \int_{-\infty}^{\infty} f(z)g(x-z)dz$$

Y en el caso discreto en dos dimensiones

$$f(x, y) * g(x, y) = h(x, y) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} f(m, n)g(x-m, y-n)$$

Y ser representaría de esta forma

