Tema 04 III: Aprendizaje

◆ Aprendizaje: Bibliografía y recursos

Bibliografía de referencia

Pajares y Cruz (2005) **Inteligencia Artificial e Ingeniería del Conocimiento**, capítulos 13 y 14;

Pajares y Cruz (2007) **Ejercicios Resueltos de Visión por computador**

Pajares y Cruz (2010) **Aprendizaje Automático**

Software de apoyo:

Neural Network Toolbox y en general de MATLAB: herramienta para la implementación de redes neuronales con una amplia capacidad funcional.

◆ **Aprendizaje: modelos para el tratamiento de datos**

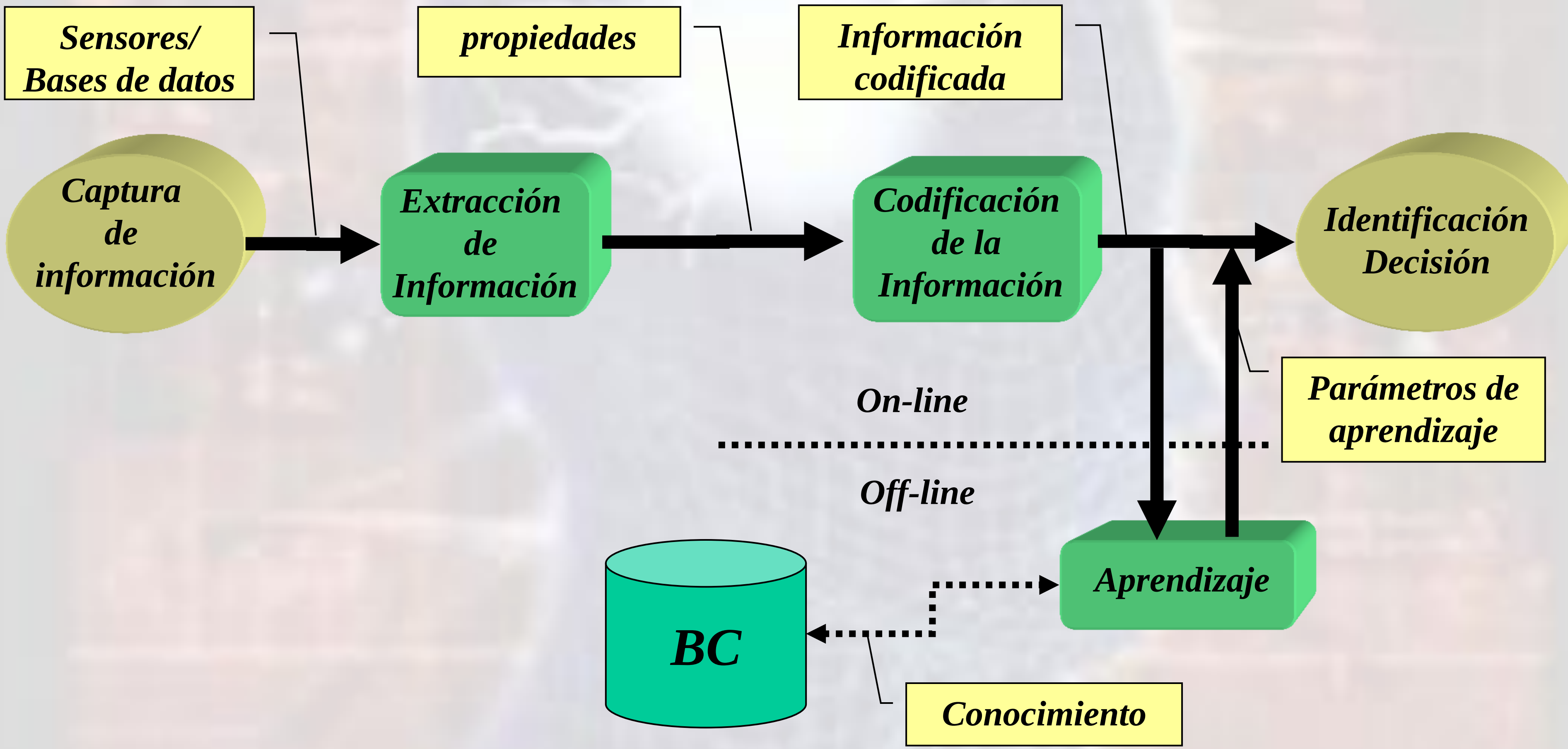
Estimadores estadísticos

- ***Agrupamiento borroso***
- ***Clasificador paramétrico de Bayes***

Redes neuronales

- ***Algoritmo generalizado de Lloyd***
- ***Mapas autoorganizativos (SOM)***
- ***Cuantización vectorial***
- ***Perceptrón y red retropropagación***

◆ Aprendizaje: esquema general



Tema 04 III : Aprendizaje

◆ **Aprendizaje: captura de muestras**

***Clases**: cielo, vegetación y agua*

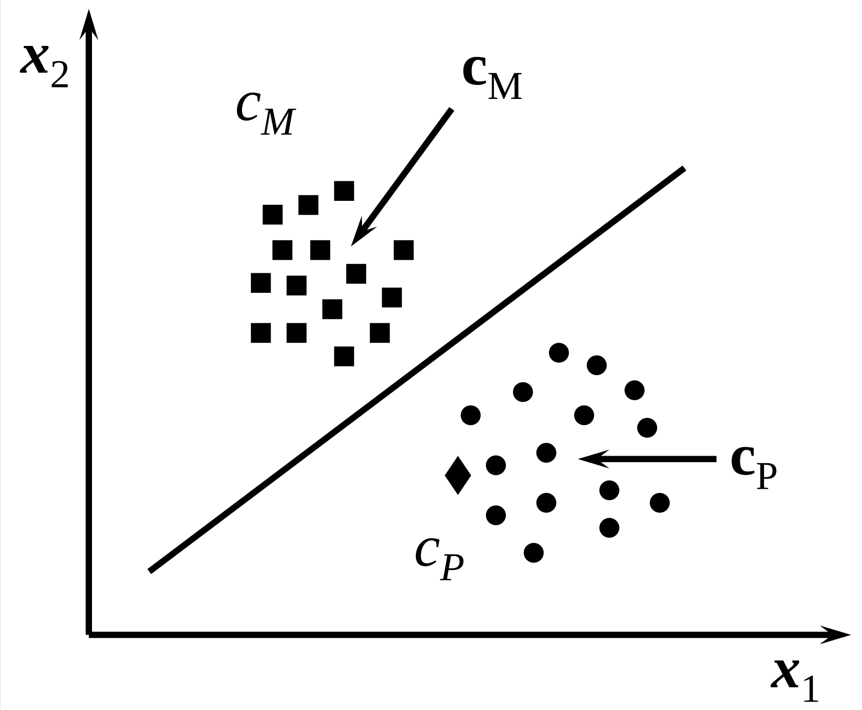


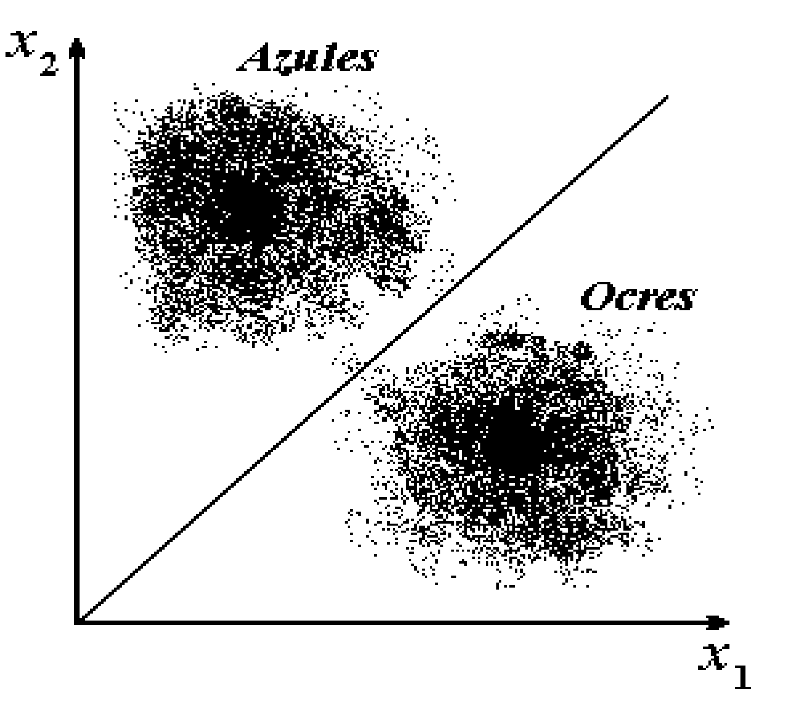
Entrenamiento y aprendizaje

Clasificación



◆ Aprendizaje: Clases





◆ Aprendizaje: agrupamiento borroso

"crisp"

$$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

"fuzzy"

$$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0.3 & 0.5 & 0 \\ 0.7 & 0.5 & 1 \end{bmatrix}$$

2 clases

$$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

- ***Grados de pertenencia***
- ***Probabilidades***

$$U = \begin{bmatrix} x_1 & x_2 & x_3 \\ 0.9 & 0.4 & 0.2 \\ 0.1 & 0.6 & 0.8 \end{bmatrix}$$

Tema 04 III : Aprendizaje

◆ Aprendizaje: agrupamiento K-medias

Algoritmo K-medias

1.- Extraer el número de muestras ***n*** a utilizar y el número de clases **c**.

2.- Inicializar los centros de las clases $\mathbf{v}_i$ y las probabilidades , i =1...c; j =1...n

3.- Normalizar las probabilidades por medio de la ecuación ; i =1...c

4.- Obtener $\mathbf{v}_i$ de acuerdo con la ecuación (1)

5.- Recalcular por medio de la ecuación (2)

6.- Repetir los pasos 3 a 5 hasta que $\mathbf{v}_i$ no cambien o el cambio sea pequeño .

Ecuaciones:

$$v_i = \frac{\sum_{j=1}^{n} \left[P(c_i / \mathbf{x}_j) \right]^b \mathbf{x}_j}{\sum_{j=1}^{n} \left[P(c_i / \mathbf{x}_j) \right]^b} \qquad i = 1,\ldots c \qquad (1)$$

$$P(v_i / \mathbf{x}_j) = \frac{\left(1/d_{ij}\right)^{1/(b-1)}}{\sum_{r=1}^{c} \left(1/d_{rj}\right)^{1/(b-1)}} \quad \text{y} \quad d_{ij} = \left\| \mathbf{x}_j - v_i \right\|^2 \qquad i = 1,\ldots \text{c}; \; j = 1,\ldots \text{n} \qquad (2)$$

$$\Delta = \left\| v_i^{(t+1)} - v_i^{(t)} \right\| < \varepsilon \qquad \textit{criterio finalización}$$

◆ Aprendizaje: agrupamiento K-medias

Ejercicio_10_01.m

Entrenamiento

2 clases

x_1	1	1	1	2	2	6	6	7	7	7
x_2	1	3	5	2	3	3	4	1	3	5

$b = 2$ $\varepsilon = 0.02$

Clasificación $\mathbf{x}_k = (3,4)$

Revisar los valores de este ejemplo

Centros iniciales: $\mathbf{v}_1 = 6.70, \; 3.43$; $\mathbf{v}_2 = 2.39, \; 2.94$

$$U^0 = \begin{matrix} 0.022 & 0.003 & 0.030 & 0.002 & 0.000 & 0.997 & 0.997 & 0.946 & 1.000 & 0.990 \\ 0.978 & 0.997 & 0.970 & 0.998 & 1.000 & 0.003 & 0.003 & 0.054 & 0.000 & 0.010 \end{matrix}$$

Iteración 1: $\mathbf{v}_1 = 6.57, \; 3.24$; $\mathbf{v}_2 = 1.41, \; 2.79$

$$\Delta = \left\| v_i^{(t+1)} - v_i^{(t)} \right\| = 0.0267 > \varepsilon$$

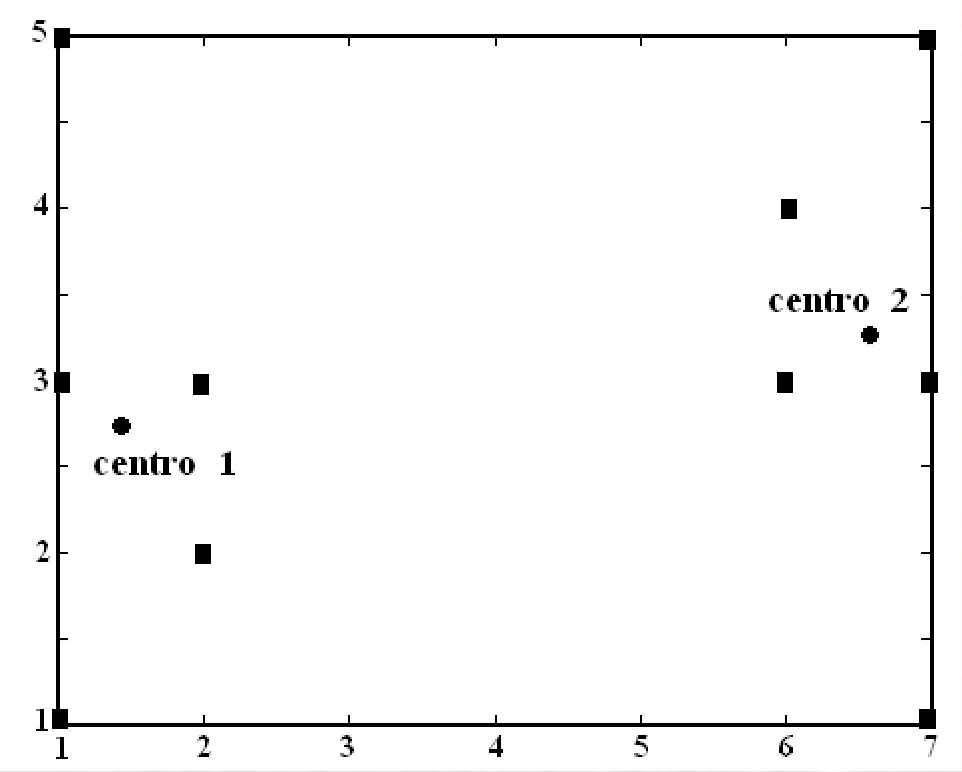
$$U^1 = \begin{matrix} 0.009 & 0.000 & 0.021 & 0.002 & 0.000 & 1.000 & 0.998 & 0.978 & 1.000 & 0.992 \\ 0.991 & 1.000 & 0.979 & 0.998 & 1.000 & 0.000 & 0.002 & 0.022 & 1.000 & 0.008 \end{matrix}$$

◆ Aprendizaje: agrupamiento K-medias

Entrenamiento

Iteración 2: $\mathbf{v}_1 = \left(6.60, \; 3.21\right)$; $\mathbf{v}_2 = \left(1.40, \; 2.79\right)$

$$\Delta = \left\| v_i^{(t+1)} - v_i^{(t)} \right\| = 0.000067 < \varepsilon \longrightarrow \textit{Fin}$$



◆ Aprendizaje: agrupamiento K-medias

Clasificación

pertenencia de $\mathbf{x}_k = (2,3)$

$$\left\| \mathbf{x}_k - \mathbf{v}_1 \right\|_G^2 = (2-6.60)^2 + (3-3.21)^2 = 20.96$$

$$\left\| \mathbf{x}_k - \mathbf{v}_2 \right\|_G^2 = (2-1.40)^2 + (3-2.79)^2 = 0.39$$

$$P(v_1 / \mathbf{x}_k) = 0.1575 \qquad P(v_2 / \mathbf{x}_k) = 0.8425$$

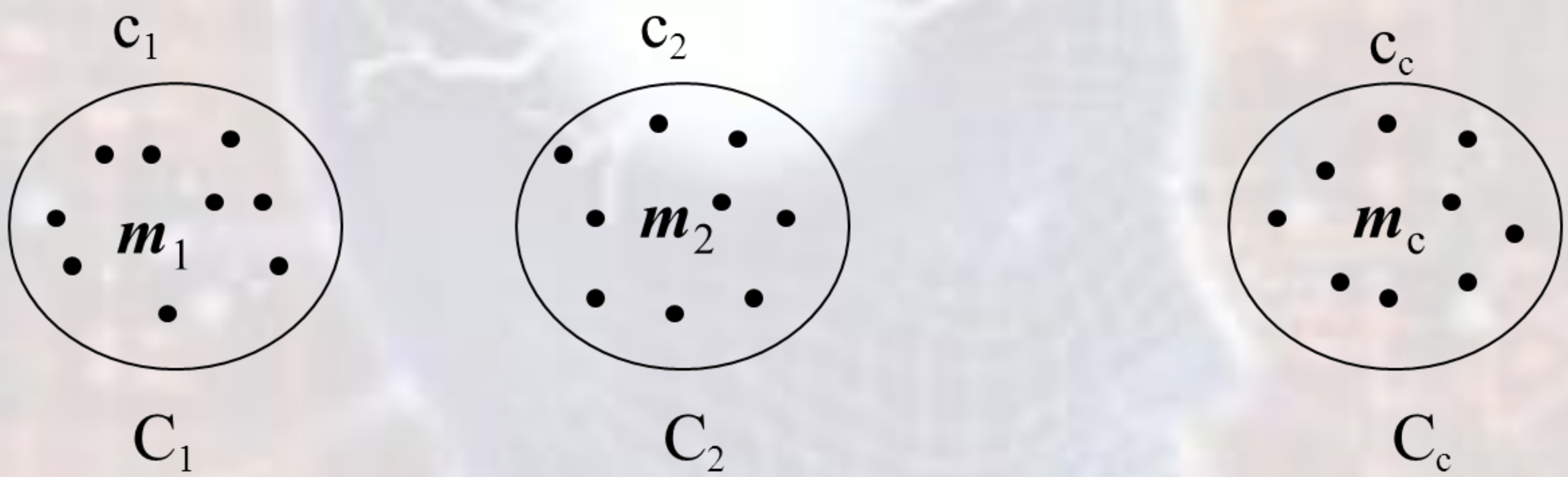
$\mathbf{x}_k = (2,3)$ pertenece a la clase 2 representada por $\mathbf{v}_2$

$\mathbf{v}_2 = (1.40,\ 2.79)$ $\quad \mathbf{v}_1 = (6.60,\ 3.21)$

x_1	1	1	1	2	2	6	6	7	7	7
x_2	1	3	5	2	3	3	4	1	3	5
	clase 2					***clase 1***				

◆ Aprendizaje: estimación paramétrica Bayes

Cada clase sigue una distribución cuya densidad de probabilidad es:



$p(\mathbf{x}_i / \mathrm{C}_1, \mathbf{m}_1)$ $p(\mathbf{x}_i / \mathrm{C}_2, \mathbf{m}_2)$ $p(\mathbf{x}_i / \mathrm{C}_c, \mathbf{m}_c)$

◆ Aprendizaje: estimación paramétrica Bayes

Caso normal multivariable

Sea $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ un conjunto de *n* muestras

Función a estimar por máxima verosimilitud

$$f(\mathbf{x}_i, \mathbf{w}) = p(\mathbf{x}_i / \mathbf{m}, C) = \frac{1}{(2\pi)^{d/2} |C|^{1/2}} \exp\left\{ -\frac{1}{2} (\mathbf{x}_i - \mathbf{m})^{\mathrm{t}} C^{-1} (\mathbf{x}_i - \mathbf{m}) \right\}$$

donde $\mathbf{w} = \{w_1, w_2\}$ es el vector de parámetros a "aprender" (estimar)

Resultado:

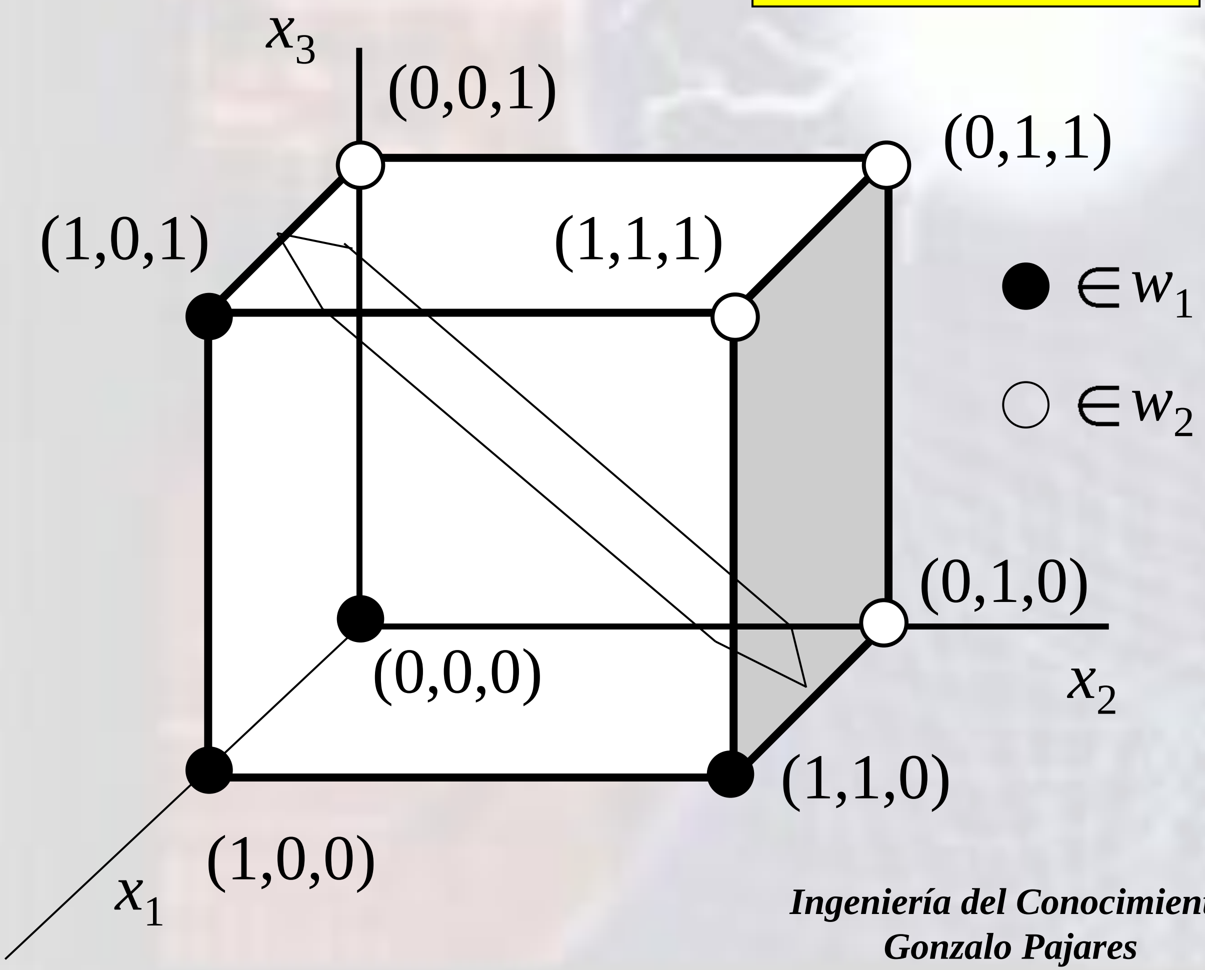
$$\hat{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \qquad C = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_i - \hat{\mathbf{m}})(\mathbf{x}_i - \hat{\mathbf{m}})^{\mathrm{t}}$$

$\mathbf{w} = \{\mathbf{m}, C\} \Rightarrow$ vector de parámetros aprendido

◆ Aprendizaje: estimación paramétrica

Ejemplo 1

Entrenamiento



$$\boldsymbol{m}_1 = \frac{1}{4}\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{m}_2 = \frac{1}{4}\begin{bmatrix} 1 \\ 3 \\ 3 \end{bmatrix}$$

$$\boldsymbol{C}_1 = \boldsymbol{C}_2 = \frac{1}{16}\begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & -1 \\ 1 & -1 & 3 \end{bmatrix}$$

◆ Aprendizaje: estimación paramétrica

Ejemplo 2

Sea X = {$\mathbf{x}_1$,..., $\mathbf{x}_n$} el conjunto de muestras dado en la tabla siguiente, correspondiente a muestras de una tonalidad verde-azulada

muestras	R	G	B
$\mathbf{x}_1$	50	250	200
$\mathbf{x}_2$	10	254	180
$\mathbf{x}_3$	20	240	210
$\mathbf{x}_4$	40	248	190
$\mathbf{x}_5$	56	254	202
m	35.2	249.2	196.4

$$\mathbf{m} = (\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 + \mathbf{x}_4 + \mathbf{x}_5)/5 = (35.2,\ 249.2, 196.4)$$

$$C = ((\mathbf{x}_1 - \mathbf{m})'\ (\mathbf{x}_1 - \mathbf{m}) + (\mathbf{x}_2 - \mathbf{m})'\ (\mathbf{x}_2 - \mathbf{m}) + (\mathbf{x}_3 - \mathbf{m})'\ (\mathbf{x}_3 - \mathbf{m}) + (\mathbf{x}_4 - \mathbf{m})'\ (\mathbf{x}_4 - \mathbf{m}) + (\mathbf{x}_5 - \mathbf{m})'\ (\mathbf{x}_5 - \mathbf{m}))/5$$

$$= \begin{pmatrix} 308.16 & 24.96 & 69.12 \\ 24.96 & 26.56 & -33.28 \\ 69.12 & -33.28 & 107.84 \end{pmatrix}$$

◆ Aprendizaje: estimación estadística, ejercicio propuesto

Ejemplo: estimar las funciones de densidad paramétrica para las dos clases dadas

Vectores patrón:

Clases

c_1:
$\boldsymbol{x}_1 = (1,2)^t$
$\boldsymbol{x}_2 = (2,2)^t$
$\boldsymbol{x}_3 = (3,1)^t$
$\boldsymbol{x}_4 = (2,3)^t$
$\boldsymbol{x}_5 = (3,2)^t$

c_2:
$\boldsymbol{x}_6 = (8,10)^t$
$\boldsymbol{x}_7 = (9,8)^t$
$\boldsymbol{x}_8 = (9,9)^t$
$\boldsymbol{x}_9 = (8,9)^t$
$\boldsymbol{x}_{10} = (7,9)^t$

◆ Aprendizaje: Agrupamiento K-Medias y Bayes

Ejemplos de aplicación:

K-Means imágenes lente ojo de pez (Herrera, Pajares, Guijarro)

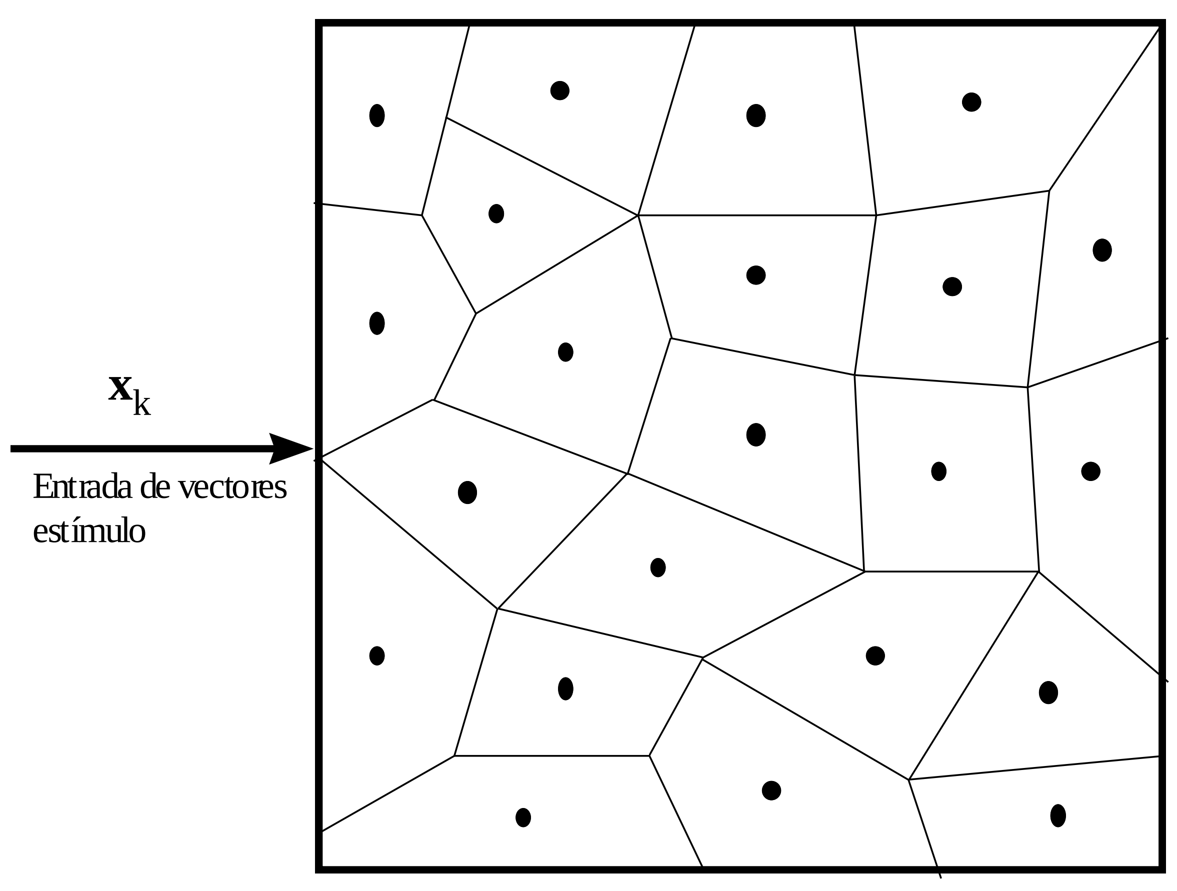
K-Means en agricultura (Romeo, Pajares, Montalvo, Guerrero, Guijarro, de la Cruz)

K-Means y Bayes imágenes aéreas (Guijarro, Pajares, Herrera)

◆ Aprendizaje: cuantización vectorial

Cuantización vectorial:

Un cuantificador vectorial Q es una proyección del espacio euclídeo *d-dimensional* $\Re^d$ en un subconjunto finito C de $\Re^d$



Aprendizaje

Se aprenden los centros $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_m\}$

◆ Aprendizaje: cuantización vectorial

Algoritmo de Lloyd (algoritmo competitivo)

1) **Inicio**: dados los puntos de datos $\mathbf{x}(k)$, $k = 1,2,\ldots,$

y centros de salida iniciales $\mathbf{c}_j(0)$, $j = 1,\ldots,m$.

2) Determinar el centro $\mathbf{c}_j(k)$ más próximo al punto $\mathbf{x}(k)$ (competición)

$$j = \arg\min_j \left\| \mathbf{x}(k) - \boldsymbol{c}_j(k) \right\|$$

3) Actualizar el centro de salida utilizando las ecuaciones,

$$\boldsymbol{c}_j(k+1) = \boldsymbol{c}_j(k) + \gamma(k)\left[\mathbf{x}(k) - \boldsymbol{c}_j(k)\right]$$

$$k = k+1$$

$\gamma(k)$ razón de aprendizaje

◆ Aprendizaje: cuantización vectorial

Entrenamiento

2 clases

x_1	1	1	1	2	2	6	6	7	7	7
x_2	1	3	5	2	3	3	4	1	3	5

2 centros iniciales $c_1 = (1,4)^t$ y $c_2 = (7,2)^t$

razón de aprendizaje $\gamma(k_j) = 0.1$

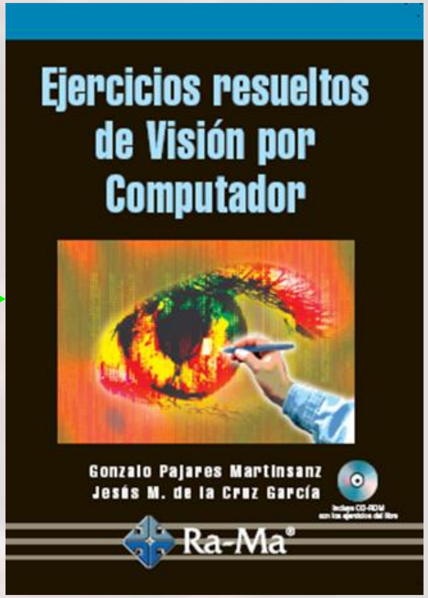
Clasificación

$A = (6,2)^t$

Tema 04 III : Aprendizaje

◆ Aprendizaje: cuantización vectorial

Ejercicio_09_07.m



Iteración 1:

$\boldsymbol{x}^1=(1,1)^t \rightarrow d_1 = 3.00$ y $d_2 = 6.08$, por tanto se actualiza c_1

$$c_1 = \begin{pmatrix} 1 \\ 4 \end{pmatrix} + 0.1\left[\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 4 \end{pmatrix}\right] = \begin{pmatrix} 1.0 \\ 3.7 \end{pmatrix}$$

$\boldsymbol{x}^2=(1,3)^t \rightarrow d_1 = 0.70$ y $d_2 = 6.08$, por tanto se actualiza c_1

$$c_1 = \begin{pmatrix} 1.0 \\ 3.7 \end{pmatrix} + 0.1\left[\begin{pmatrix} 1 \\ 3 \end{pmatrix} - \begin{pmatrix} 1.0 \\ 3.7 \end{pmatrix}\right] = \begin{pmatrix} 1.0 \\ 3.6 \end{pmatrix}$$

$\boldsymbol{x}^3=(1,5)^t \rightarrow d_1 = 1.37$ y $d_2 = 6.71$, por tanto se actualiza c_1

$$c_1 = \begin{pmatrix} 1.0 \\ 3.6 \end{pmatrix} + 0.1\left[\begin{pmatrix} 1 \\ 5 \end{pmatrix} - \begin{pmatrix} 1.0 \\ 3.6 \end{pmatrix}\right] = \begin{pmatrix} 1.0 \\ 3.8 \end{pmatrix}$$

◆ Aprendizaje: cuantización vectorial

Ejercicio_09_07.m

$\boldsymbol{x}^4=(2,2)^t \rightarrow d_1 = 2.03$ y $d_2 = 5.0$, por tanto se actualiza c_1

$$c_1 = \begin{pmatrix} 1.0 \\ 3.8 \end{pmatrix} + 0.1\left[\begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 1.0 \\ 3.8 \end{pmatrix}\right] = \begin{pmatrix} 1.1 \\ 3.6 \end{pmatrix}$$

$\boldsymbol{x}^5=(2,3)^t \rightarrow d_1 = 1.1$ y $d_2 = 5.1$, por tanto se actualiza c_1

$$c_1 = \begin{pmatrix} 1.1 \\ 3.6 \end{pmatrix} + 0.1\left[\begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 1.1 \\ 3.6 \end{pmatrix}\right] = \begin{pmatrix} 1.2 \\ 3.5 \end{pmatrix}$$

$\boldsymbol{x}^6=(6,3)^t \rightarrow d_1 = 4.8$ y $d_2 = 1.4$, por tanto se actualiza c_2

$$c_2 = \begin{pmatrix} 7 \\ 2 \end{pmatrix} + 0.1\left[\begin{pmatrix} 6 \\ 3 \end{pmatrix} - \begin{pmatrix} 7 \\ 2 \end{pmatrix}\right] = \begin{pmatrix} 6.9 \\ 2.1 \end{pmatrix}$$

◆ Aprendizaje: cuantización vectorial

Ejercicio_09_07.m

$\mathbf{x}^7=(6,4)^t \rightarrow d_1 = 4.8$ y $d_2 = 2.1$, por tanto se actualiza c_2

$$c_2 = \begin{pmatrix} 6.9 \\ 2.1 \end{pmatrix} + 0.1\left[\begin{pmatrix} 6 \\ 4 \end{pmatrix} - \begin{pmatrix} 6.9 \\ 2.1 \end{pmatrix}\right] = \begin{pmatrix} 6.8 \\ 2.3 \end{pmatrix}$$

$\mathbf{x}^8=(7,1)^t \rightarrow d_1 = 6.3$ y $d_2 = 1.3$, por tanto se actualiza c_2

$$c_2 = \begin{pmatrix} 6.8 \\ 2.3 \end{pmatrix} + 0.1\left[\begin{pmatrix} 7 \\ 1 \end{pmatrix} - \begin{pmatrix} 6.8 \\ 2.3 \end{pmatrix}\right] = \begin{pmatrix} 6.8 \\ 2.2 \end{pmatrix}$$

$\mathbf{x}^9=(7,3)^t \rightarrow d_1 = 5.8$ y $d_2 = 0.9$, por tanto se actualiza c_2

$$c_2 = \begin{pmatrix} 6.8 \\ 2.2 \end{pmatrix} + 0.1\left[\begin{pmatrix} 7 \\ 3 \end{pmatrix} - \begin{pmatrix} 6.8 \\ 2.2 \end{pmatrix}\right] = \begin{pmatrix} 6.8 \\ 2.2 \end{pmatrix}$$

$\mathbf{x}^{10}=(7,5)^t \rightarrow d_1 = 6.0$ y $d_2 = 2.8$, por tanto se actualiza c_2

$$c_2 = \begin{pmatrix} 6.8 \\ 2.2 \end{pmatrix} + 0.1\left[\begin{pmatrix} 7 \\ 5 \end{pmatrix} - \begin{pmatrix} 6.8 \\ 2.2 \end{pmatrix}\right] = \begin{pmatrix} 6.9 \\ 2.5 \end{pmatrix}$$

◆ Aprendizaje: cuantización vectorial

Ejercicio_09_07.m

Iteración 2:

Realizando el mismo proceso que en la iteración anterior, llegamos al resultado final: $c_1 = (1.3, 3.3)^t$ y $c_2 = (6.8, 2.8)^t$. Observando que se han producido cambios en esta segunda iteración con respecto a los valores obtenidos en la iteración 1.

La clasificación de A = $(6,2)^t$ se realiza computando la distancia del mismo a cada uno de los centros de las clases y eligiendo la clase de pertenencia la de menor distancia:

$$d(A, c_1) = \|A - c_1\| = 4.862 \quad \text{y} \quad d(A, c_2) = \|A - c_2\| = 1.137$$

Clasificación

Por tanto, A se clasifica como perteneciente a la clase c_2

◆ Aprendizaje: cuantización vectorial (SOM)

Self-Organizing Map (algoritmo competitivo)

1) **Inicio**: dados los puntos de datos $\mathbf{x}(k)$, $k = 1,2,\ldots,$

y centros de salida iniciales $\mathbf{c}_j(0)$, $j = 1,\ldots,m$.

2) Actualizar los centros

$$\boldsymbol{c}_j(k+1) = \begin{cases} \boldsymbol{c}_j(k) + \gamma(k)\left[\mathbf{x}(k) - \boldsymbol{c}_j(k)\right] & \text{si } \boldsymbol{c}_j \in K_{\alpha(k)} \\ \boldsymbol{c}_j(k) & \text{en cualquier otro caso} \end{cases}$$

$$k = k+1$$

Consideraciones

Razón de aprendizaje: $\gamma(k)$

Región de vecindad: $K_{\alpha(k)} = (\mathbf{z},\mathbf{z}') = \exp\left(-\dfrac{\left\|\mathbf{z}-\mathbf{z}'\right\|^2}{2\alpha^2(k)}\right)$ con $\alpha(k) = \alpha_{inicial}\left(\dfrac{\alpha_{final}}{\alpha_{inicial}}\right)^{k/k_{\max}}$

donde k es el número de iteración y k_{max} es el número máximo de iteraciones. El ancho inicial de la vecindad $\alpha_{inicial}$ se elige de modo que dicha vecindad cubra todas las unidades y el ancho final de la vecindad α_{final} controla el grado de variación del núcleo entre iteraciones consecutivas.

◆ Aprendizaje: cuantización vectorial (SOM)

Ejercicio_09_09.m

Entrenamiento

2 clases

x_1	1	1	2	6	6	7
x_2	3	5	2	3	4	3

Sabiendo que existen dos únicas clases cuyos centros inicialmente son $c_1 = (1,4)^t$ y $c_2 = (7,2)^t$, ajustar dichos centros a los datos siguiendo el algoritmo SOM con una única iteración. Tomar como referencia los siguientes valores de los parámetros $\alpha_{inicial} = 1.0$, $\alpha_{final} = 0.8$, $k_{\max} = 5.0$. Dadas dos muestras **z** y **z**´ se dice que pertenecen a la región definida por $\alpha(k)$ si $K_{\alpha(k)}(\mathbf{z},\mathbf{z}') > T$, siendo T = 0.2 en este ejemplo. La razón de aprendizaje viene dada por la siguiente expresión $\gamma(k) = 1/(10+k)$.

Clasificación

$A = (6,2)^t$

◆ Aprendizaje: cuantización vectorial (SOM)

Iteración 1 ($k = 1$):

$$\gamma(1) = 1/(10+1) = 0.0909; \quad \alpha(1) = 0.9564$$

$\mathbf{x}^1 = (1,3)^t$

$K_{\alpha(1)}(x^1, c_1) = 0.58 > T$ y $K_{\alpha(1)}(x^1, c_2) = 1.6x10^{-9} < T$ luego sólo se actualiza c_1. En el supuesto de que los dos valores fueran mayores que T se actualizarían los dos centros

$$c_1 = \begin{pmatrix} 1 \\ 4 \end{pmatrix} + 0.0909 * 0.58 \left[\begin{pmatrix} 1 \\ 3 \end{pmatrix} - \begin{pmatrix} 1 \\ 4 \end{pmatrix} \right] = \begin{pmatrix} 1.0000 \\ 3.9996 \end{pmatrix}$$

procediendo de este modo para el resto de patrones $\mathbf{x}^2$ a $\mathbf{x}^6$; se obtienen al final de esta primera iteración los siguientes centros:

$$c_1 = (1.0000, 3.9996)^t \;\text{ y }\; c_2 = (6.9712, 2.0832)^t$$

El proceso se repite para el resto de iteraciones, llegando a obtener los siguientes valores para los centros al cabo de las 5 iteraciones previstas,

$$c_1 = (1.0000, 3.9972)^t \;\text{ y }\; c_2 = (6.8902, 2.3225)^t$$

◆ Aprendizaje: cuantización vectorial (SOM)

Ejercicio_09_09.m

Clasificación

La clasificación de A = $(6, 2)^t$ se realiza computando la distancia del mismo a cada uno de los centros de las clases y eligiendo la clase de pertenencia con la menor distancia:

$$d(A,c_1) = \|A - c_1\| = 5.3841 \text{ y } d(A,c_2) = \|A - c_2\| = 0.9468$$

Por tanto, *A* se clasifica como perteneciente a la clase c_2.

◆ Aprendizaje: cuantización vectorial

Ejercicio_09_10.m

Entrenamiento

	patrones							
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
R	200	90	210	35	215	92	87	41
G	160	130	170	23	172	138	128	22
B	120	60	130	44	133	54	66	37

2 clases

$$T = 20$$

Clasificación

$$A = (93, 120, 70)^t$$

◆ Aprendizaje: cuantización vectorial

Ejercicio_09_10.m

1) Inicialmente se crea una única clase c_1 formada por el primer patrón a procesar

x_1 = {200,160,120} cuyo centro es v_1 = {200,160,120}, $c_1 = \{ x_1\}$

2) segundo patrón x_2 = {90,130,60} calculamos la distancia al único centro disponible $d(x_2, v_1) = \|x_2 - v_1\| = 128.84 > T$, por tanto, creamos una nueva clase con el patrón dado $c_2 = \{ x_2\}$, cuyo centro resulta ser este mismo patrón v_2 = {90,130,60}.

3) tercer patrón x_3 = {210,170,130} calculando las distancias a los dos centros de clase existentes $d(x_3, v_1) = \|x_3 - v_1\| = 17.32$ $d(x_3, v_2) = \|x_3 - v_2\| = 144.57'$, siendo el mínimo $d(x_3, v_1) < T$, por tanto este nuevo patrón se añade a la clase c_1, quedando ésta como sigue $c_1 = \{ x_1, x_3\}$. Se actualiza el centro de la clase c_1, calculando el valor medio entre x_1 y x_3, resultando v_1 = {205,165,125}.

◆ Aprendizaje: cuantización vectorial

Ejercicio_09_10.m

4) cuarto patrón x_4 = {35,23,44} calculando las distancias a los dos centros de clase existentes $d(x_4,v_1) = \|x_4 - v_1\| = 235.85$ $d(x_4,v_2) = \|x_4 - v_2\| = 121.37$, siendo el mínimo $d(x_4,v_2) > T$ por tanto, es necesario crear una nueva clase c_3 = {x_4} cuyo centro es este mismo patrón v_3 = {35,23,44}.

5) quinto patrón x_5 = {215, 172, 133} calculando las distancias a los tres centros de clase existentes $d(x_5,v_1) = 14.59$, $d(x_5,v_2) = 150.72$ y $d(x_5,v_3) = 250.04$ siendo el mínimo $d(x_5,v_1) < T$ por tanto, el nuevo patrón se añade a la clase c_1 = { x_1, x_3, x_5} y se actualiza su centro v_1 = {208.33, 167.33, 127.67}.

6) sexto patrón x_6 = {92, 138, 54} calculando las distancias a los tres centros de clase existentes $d(x_6,v_1) = 140.79$, $d(x_6,v_2) = 10.20$ y $d(x_6,v_3) = 128.74$ siendo el mínimo $d(x_5,v_2) < T$ por tanto, el nuevo patrón se añade a la clase c_2 = {x_2, x_6} y se actualiza su centro v_2 = {91,134,57}.

◆ Aprendizaje: cuantización vectorial

Ejercicio_09_10.m

7) séptimo patrón x_7 = {87, 128, 66} calculando las distancias a los tres centros de clase existentes $d(x_7,v_1)=141.67$, $d(x_7,v_2)=11.53$ y $d(x_7,v_3)=119.22$ siendo el mínimo $d(x_7,v_2)<T$, por tanto, el nuevo patrón se añade a la clase c_2 = {x_2, x_6, x_7} y se actualiza su centro v_2 = {89.67, 132.00, 60.00}.

8) octavo patrón x_8 = {41, 22, 37} calculando las distancias a los tres centros de clase existentes $d(x_8,v_1)=239.46$, $d(x_8,v_2)=122.46$ y $d(x_8,v_3)=9.27$ siendo el mínimo $d(x_8,v_3)<T$, por tanto, el nuevo patrón se añade a la clase c_3 = {x_4, x_8} cuyo centro se actualiza a v_3 = {38.0, 22.5, 40.5}.

Clasificación

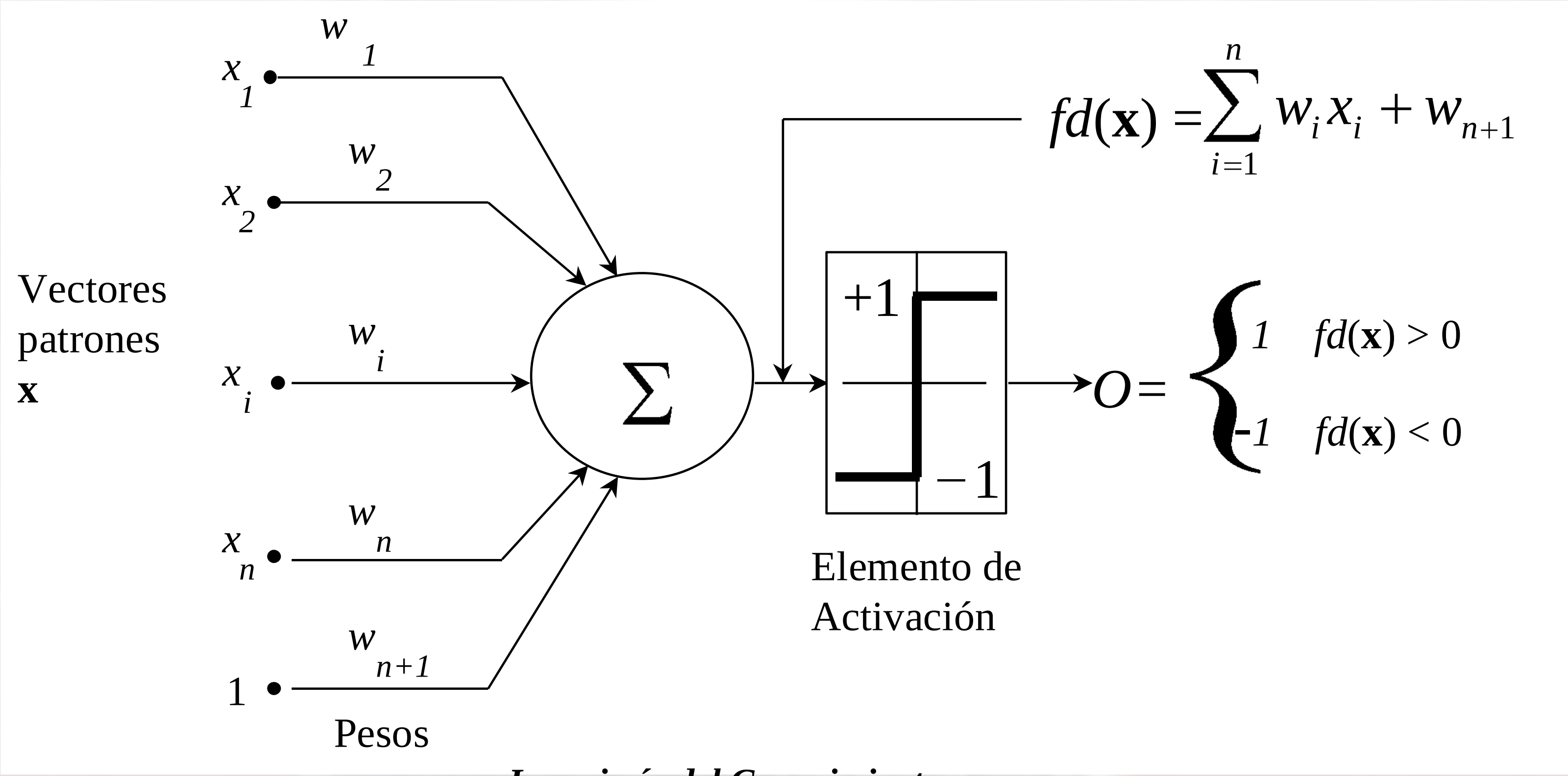
Finalmente, calculamos las distancias del patrón A a cada uno de los centros dados v_1 = {208.33, 167.33, 127.67}, v_2 = {89.67, 132.00, 60.00}, v_3 = {38.0, 22.5, 40.5}, resultando: $d(A,v_1)=137.36$; $d(A,v_2)=15.97<T$; $d(A,v_3)=115.76$

Resultando por tanto, que *A* pertenece a la clase c_2

◆ Aprendizaje: modelos conexionistas, el perceptrón

Objetivo

Aprender una función discriminante: $$fd(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + w_{n+1}$$



◆ Aprendizaje: modelos conexionistas, el perceptrón

Algoritmo

1.- **Inicialización de los pesos y del umbral:** inicialmente se asignan valores aleatorios a cada uno de los pesos w_i *$i = 1,2,..,n$ y al umbral* w_{n+1}

2.- **Presentación de un nuevo par** (**Entrada, Salida esperada**): presentar un nuevo patrón de entrada $\mathbf{x}_p = \{x_1, x_2, ..., x_n\}$ *junto con la salida esperada* $fd_i(k)$

3.- **Cálculo de la salida actual:** $y(k) = f[\mathbf{w}^t\ \mathbf{x}_i + w_{n+1}]$, siendo en este caso *f* la función de transferencia escalón.

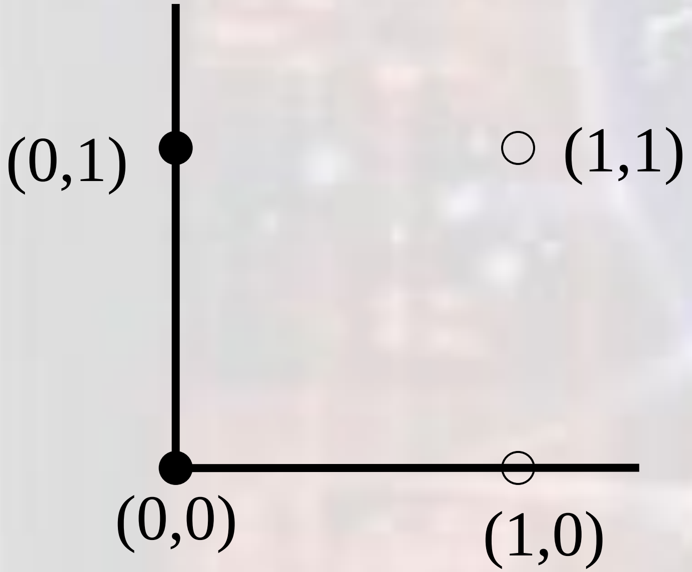
4.- **Adaptación de los pesos**: $w_i(k+1) = w_i(k) + \alpha(k)[fd_i(k) - y(k)]x_i$
La salida esperada fdi(k) es 1 si el patrón pertenece a la clase A y -1 si es de la clase B

5.- Si no hay convergencia (los pesos no cambian) **volver al paso 2**

◆ Aprendizaje: El perceptrón, ejemplo 1

Ejercicio: suponer el ejemplo biclase dado a continuación con $\alpha = 1/2$

Clases ● c_1 ○ c_2



Pesos iniciales: $w_1 = w_2 = w_3 = 0$

Vectores patrón aumentados:

c_1: $\boldsymbol{x}_1 = (0,0,1)^t$, $\boldsymbol{x}_2 = (0,1,1)^t$ $\quad fd_i = 1$

c_2: $\boldsymbol{x}_3 = (1,0,1)^t$, $\boldsymbol{x}_4 = (1,1,1)^t$ $\quad fd_i = -1$

◆ Aprendizaje: El perceptrón, ejemplo 1

1. Inicialización: $\mathbf{w}^t(1)$ = (0,0,0); $\alpha = 1/2$

2.1 Patrón $\mathbf{x} = \{0,0,1\}$: $\mathbf{w}^t\, \mathbf{x} = 0(0)+0(0)+0(1)=0$; O = -1, error = $(fd_i\text{-}O)=1+1= 2$

Pesos modificados: $\mathbf{w}^t(2) = \mathbf{w}^t(1)+ \mathbf{x} = (0,0,1)$

2.2 Patrón $\mathbf{x} = \{0,1,1\}$: $\mathbf{w}^t\, \mathbf{x} = 1$; O = 1, error = $(fd_i\text{-}O)=1 -1= 0 \Rightarrow$

*Pesos **no** modificados:* $\mathbf{w}^t(3) = \mathbf{w}^t(2)$

2.3 Patrón $\mathbf{x} = \{1,0,1\}$: $\mathbf{w}^t\, \mathbf{x} = 1$; O = 1, error = $(fd_i\text{-}O)=-1 -1= -2$

Pesos modificados: $\mathbf{w}^t(4) = \mathbf{w}^t(3)- \mathbf{x} = (-1,0,0)$

2.4 Patrón $\mathbf{x} = \{1,1,1\}$: $\mathbf{w}^t\, \mathbf{x} = -1$; O = -1, error = (fdi -O)=$-1 +1= 0$

*Pesos **no** modificados:* $\mathbf{w}^t(5) = \mathbf{w}^t(4)$

Repetir otra serie de entrenamiento con los mismos patrones:

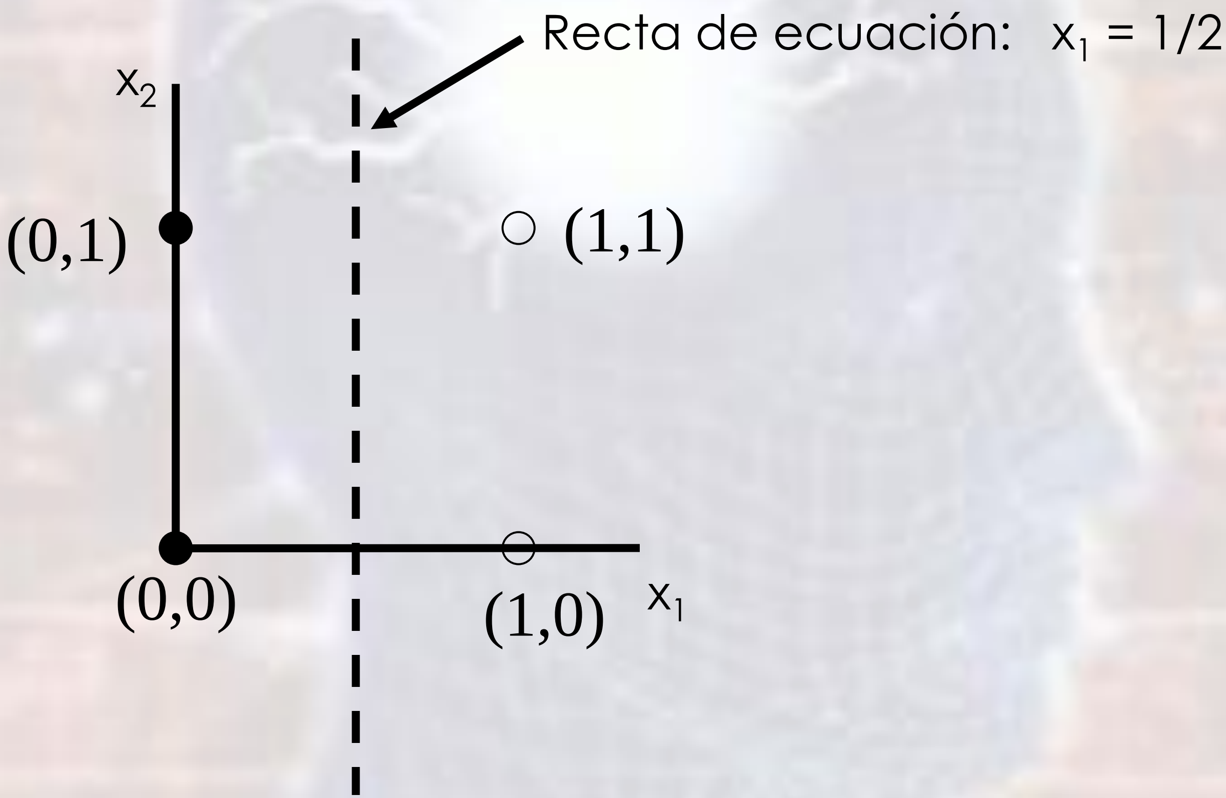
Llegando finalmente en la 10ª iteración a: $\mathbf{w}^t(10) = (-2,0,1)$, que corresponde al hallazgo de una función de decisión en el plano

$$fd_i = -2x_1 + 1$$

que es precisamente la recta $x_1 = 1/2$

◆ Aprendizaje: El perceptrón, ejemplo 1

Función de decisión



◆ Aprendizaje: El perceptrón, ejemplo 2

Ejemplo: Ajuste de pesos para una red que realiza la función **OR** con $\alpha = 1$.

La salida esperada en cada caso es rl resultado de la operación OR

1. Inicialización: $w_1 = 0.5$, $w_2 = 1.5$, $w_3 = 1.5$

2.1 Patrón **x = {0,0}**: $\mathbf{w}^t \mathbf{x}_i = 0(0.5)+0(1.5)+1(1.5)=1.5$; O = 1, error = $(fd_i - O) = 0 - 1 = -1$

Pesos modificados: $w_1 = 0.5 + (-1)0 = 0.5$; $w_2 = 1.5 + (-1)0 = 1.5$; $w_3 = 1.5 + (-1)1 = 0.5$

2.2 Patrón **x = {0,1}**: $\mathbf{w}^t \mathbf{x}_i = 0(0.5)+1(1.5)+1(0.5)=2$; O = 1, error = $(fd_i - O) = 1 - 1 = 0$

no se modifican los pesos. Tampoco se modifican para **x = {1,0}** y **x = {1,1}**

3. Se toman de nuevo los cuatro patrones de entrada

3.1 Patrón **x = {0,0}**: $\mathbf{w}^t \mathbf{x}_i = 0(0.5)+0(1.5)+1(0.5)=0.5$; O = 1, error = $(fd_i - O) = 0 - 1 = -1$

Pesos modificados: $w_1 = 0.5 + (-1)0 = 0.5$; $w_2 = 1.5 + (-1)0 = 1.5$; $w_3 = 0.5 + (-1)1 = -0.5$

3.2 Patrón **x = {0,1}**: $\mathbf{w}^t \mathbf{x}_i = 0(0.5)+1(1.5)+1(-0.5)=1$; O = 1, error = $(fd_i - O) = 1 - 1 = 0$

no se modifican los pesos. Lo mismo para los patrones **x = {1,0}** y **x = {1,1}**

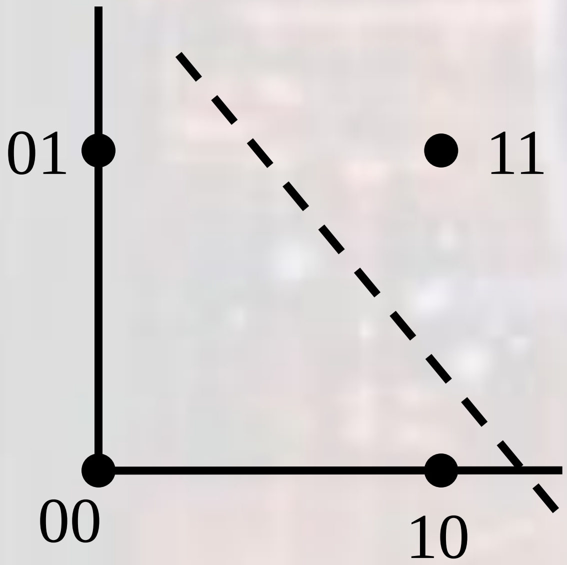
4. Se toman de nuevo los cuatro patrones de entrada

4.1 Patrón **x = {0,0}**: $\mathbf{w}^t \mathbf{x}_i = 0(0.5)+0(1.5)+1(-0.5) = -0.5$; O = 0, error = $(fd_i - O) = 0 - 0 = 0$

Ya no se modifican los pesos

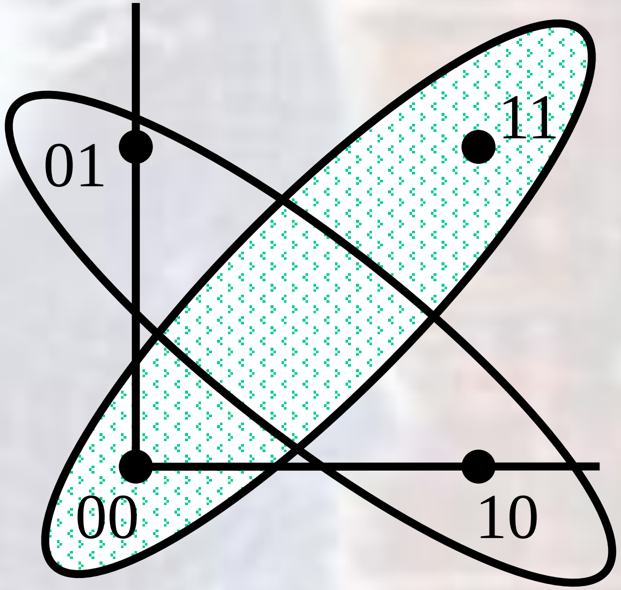
◆ Aprendizaje: El perceptrón, ejemplo 2

Ejemplo: Ajuste de pesos para OR y problema para XOR



OR

Existe recta separando las clases

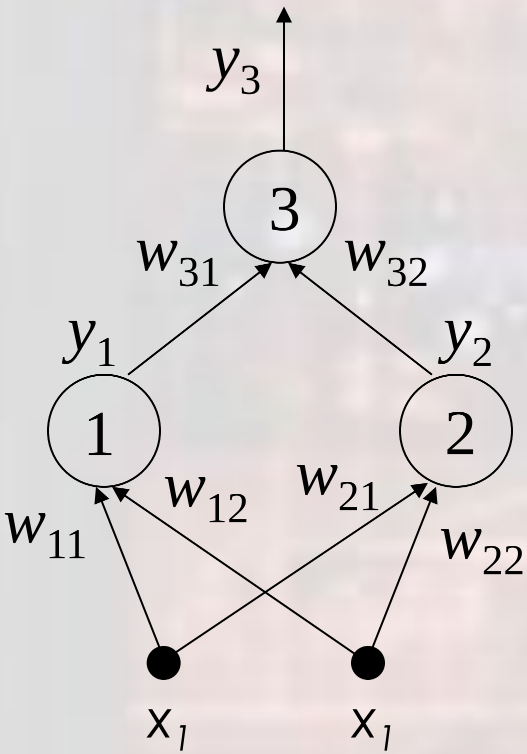


XOR

NO existe recta que separe las clases

◆ Aprendizaje: El perceptrón, multicapa

Ejemplo: solución para XOR



Pesos

$w_{11} = w_{12} = w_{21} = w_{22} = w_{31} = 1; \quad w_{32} = -1.5$

$\theta_1 = 0.5;\ \theta_2 = 1.5;\ \theta_3 = 0.5$

Salidas de las neuronas

$y_1 = f(w_{11}x_1 + w_{12}x_2 - \theta_1) = f(x_1 + x_2 - 0.5)$

$y_2 = f(w_{21}x_1 + w_{22}x_2 - \theta_2) = f(x_1 + x_2 - 1.5)$

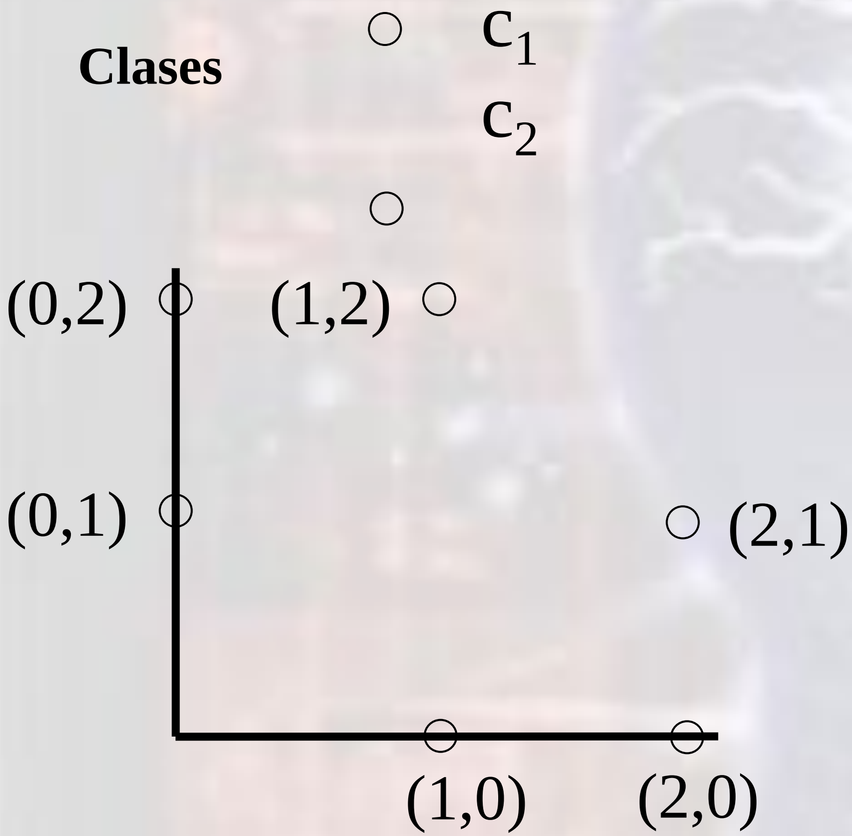
$y_3 = f(w_{31}y_1 + w_{32}y_2 - \theta_3) = f(y_1 + y_2 - 0.5)$

Resultados

x_1	x_2	y_1	y_2	y_3 = XOR
0	0	$f(-0.5) = 0$	$f(-1.5) = 0$	$f(-0.5) = 0$
0	1	$f(0.5) = 1$	$f(-0.5) = 0$	$f(0.5) = 1$
1	0	$f(0.5) = 1$	$f(-0.5) = 0$	$f(0.5) = 1$
1	1	$f(1.5) = 1$	$f(0.5) = 1$	$f(1.5) = 1$

◆ Aprendizaje: El perceptrón, ejercicio propuesto

Ejemplo: suponer el ejemplo biclase dado a continuación con $\alpha = 1/2$



Pesos iniciales: $w_1 = w_2 = w_3 = 0$

Vectores patrón aumentados:

$$c_1: \quad \boldsymbol{x}_1 = (0,1,1)^t, \quad \boldsymbol{x}_2 = (0,2,1)^t, \quad \boldsymbol{x}_3 = (1,2,1)^t \qquad fd_i = 1$$

$$c_2: \quad \boldsymbol{x}_4 = (1,0,1)^t, \quad \boldsymbol{x}_5 = (2,0,1)^t, \quad \boldsymbol{x}_2 = (2,1,1)^t \qquad fd_i = -1$$

Verificar que $\mathbf{w}^t = (-1,1,0)$, siendo la función de decisión: $fd_i = -x_1 + x_2$

◆ Aprendizaje: modelos conexionistas, el perceptrón

Regla Delta de Mínimos Cuadrdos: clases no separables

El **objetivo** es ajustar los pesos para minimizar el error al cuadrado E_{tot} entre la salida deseada fd_i , con , que representa la decisión del maestro y la salida actual o real para el número de muestras de entrenamiento *n*.

$$E_{tot}(\boldsymbol{w},\mathbf{x}) = \sum_{i=1}^{n} \left(\boldsymbol{w}^t \mathbf{x}_i - fd_i\right)^2$$

Actualización de los pesos de conexión:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) \pm \alpha \nabla_w E_{tot}(\boldsymbol{w},\mathbf{x})\Big|_{w(k)}$$

$$\Downarrow$$

$$\nabla_w E_{tot}(\boldsymbol{w},\mathbf{x}) = 2\sum_{i=1}^{n} \left(\boldsymbol{w}^t \mathbf{x}_i - fd_i\right)\mathbf{x}_i \quad \Longrightarrow \quad \boldsymbol{w}(k+1) = \boldsymbol{w}(k) \pm \alpha(k)\sum_{i=1}^{n} \frac{\left(\boldsymbol{w}^t \mathbf{x}_i - fd_i\right)\mathbf{x}_i}{\|\mathbf{x}_i\|}$$

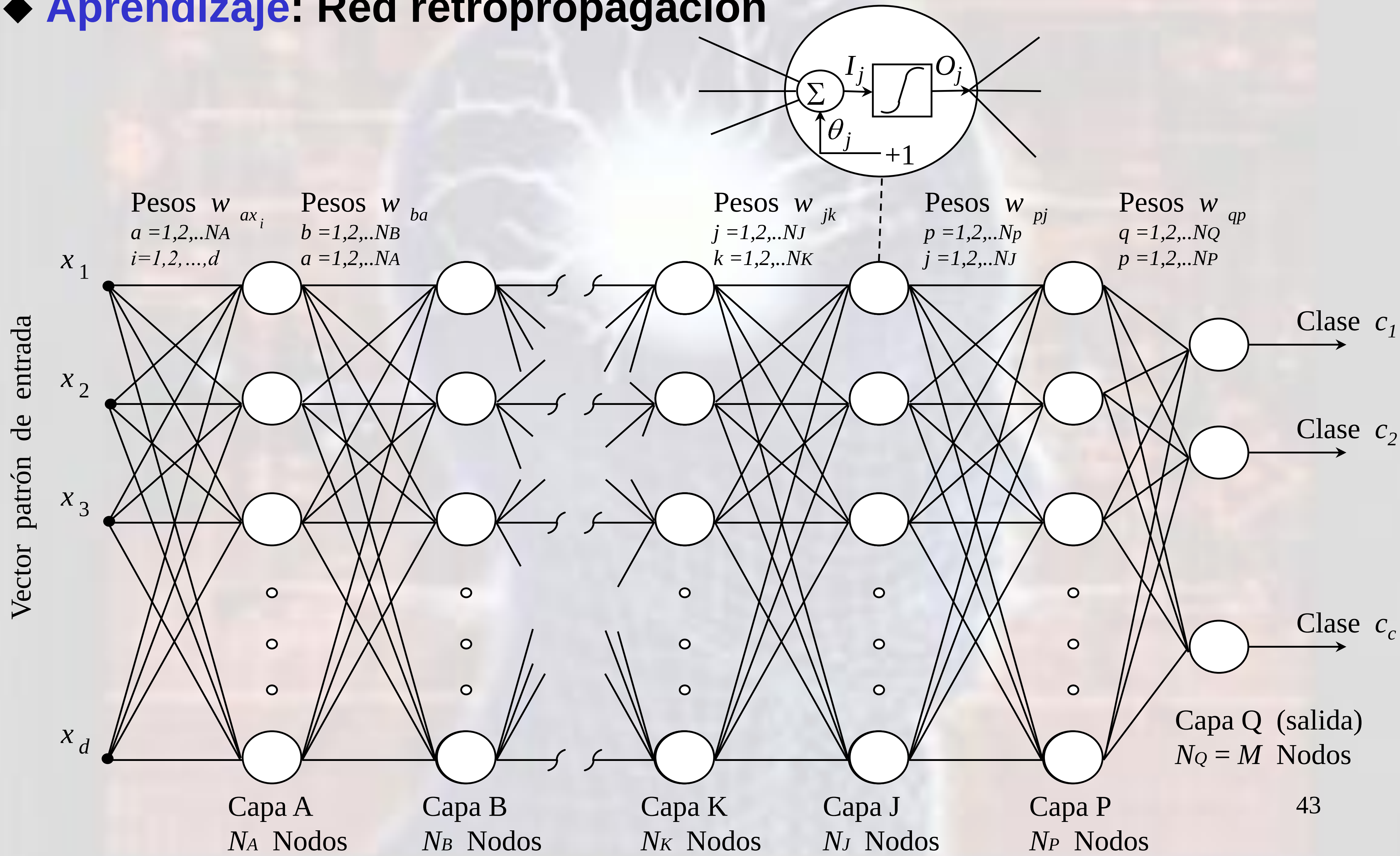
$\alpha(k)$: razón de aprendizaje en [0,1]

◆ **Aprendizaje: red backpropagation**

Reconocimiento de caracteres

Ingeniería del Conocimiento
Gonzalo Pajares

◆ Aprendizaje: Red retropropagación



◆ Aprendizaje: Red retropropagación

Un ejemplo de aplicación:

Clasificación de texturas mediante imágenes multiespectrales

Clasificación de texturas mediante redes neuronales (CEDEX) Pajares y col.