

Tema 2: Clasificación supervisada



escola politècnica superior ■



Universitat de les
Illes Balears
Departament de Ciències
Matemàtiques i Informàtica

**10529 Informàtica Encastada
i Aplicacions**
Máster en Tecnologías de la
Información y las Comunicaciones

Alberto ORTIZ RODRÍGUEZ

Contenido

- Introducción
 - planteamiento del problema
- Clasificación Bayesiana
 - introducción
 - minimización de la probabilidad de error
 - clasificadores para distribuciones normales y de distancia mínima
- Funciones de discriminación lineales y el algoritmo del perceptrón
 - algoritmo básico del perceptrón
 - variantes

- **Introducción**
- Clasificación Bayesiana
- Estimación de funciones de densidad de probabilidad
- Funciones de discriminación lineales y el algoritmo del perceptrón

- **Clasificación supervisada:**
 - Se trata de **clasificar un patrón nuevo en la clase correcta**, habiendo inicialmente diseñado un clasificador a partir de la información proveniente de un **conjunto de entrenamiento**, en el que, en particular, los ejemplares están **etiquetados** con la clase a la que pertenecen
 - El **algoritmo de diseño del clasificador** hace uso de las **etiquetas** para generar los parámetros del clasificador

- Introducción
- **Clasificación Bayesiana**
- Estimación de funciones de densidad de probabilidad
- Funciones de discriminación lineales y el algoritmo del perceptrón

Clasificación Bayesiana

- **Objetivo:** clasificar un patrón nuevo en la **clase más probable**
 - Dada una tarea de clasificación en M clases, $\omega_1, \omega_2, \dots, \omega_M$, y un ejemplar representado por un vector de características \mathbf{x} , se trata de determinar:
$$p(\omega_i|\mathbf{x}), i = 1, 2, \dots, M \text{ (probabilidades } a \text{ posteriori)}$$
 - Dado \mathbf{x} , cuál es la probabilidad de que su clase de origen sea ω_i
 - El clasificador decide la clase más probable en base al **máximo** de las probabilidades *a posteriori*:
 - **Regla de clasificación Bayesiana**
si $p(\omega_i|\mathbf{x}) > p(\omega_j|\mathbf{x}), \forall j \neq i$, entonces \mathbf{x} es asignado a la clase ω_i

Clasificación Bayesiana

- Repaso de probabilidades:

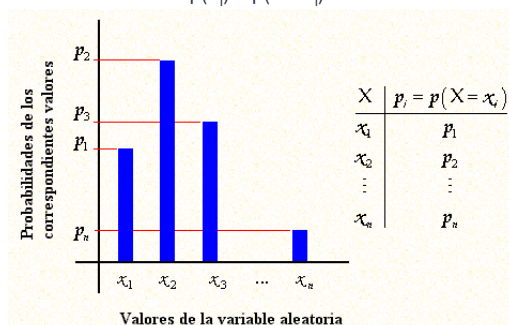
- **Función de probabilidad:** $p: \Omega \rightarrow [0, 1]$

$$v \rightarrow p(v)$$

- asigna un valor a cada evento 'v' posible en función de la frecuencia con que se presenta dicho evento
 - Ω puede plantearse como el conjunto de eventos correspondientes a que cierta variable aleatoria discreta X tome ciertos valores: $p(A_i) = p(X = x_i)$

- en particular:

$$p(\Omega) = \sum_{i=1}^n p(A_i) = 1$$



Alberto Ortiz / EPS (última revisión 25/06/2010)

7

Clasificación Bayesiana

- Repaso de probabilidades:

- **Probabilidad condicionada:** $p(B|A_i) = \frac{p(B \cap A_i)}{p(A_i)}$

- **Ley de probabilidades totales:**

Dados M eventos $A_i, i = 1, \dots, M$, tales que $\sum_{i=1}^M p(A_i) = 1$ para cualquier evento arbitrario B :

$$p(B) = \sum_{i=1}^M p(B|A_i)p(A_i)$$

- **Regla de Bayes**

A partir de la definición de probabilidad condicionada, dados dos eventos A y B :

$$p(B|A)p(A) = p(A|B)p(B)$$

✱ *Todo esto se verifica exactamente en las mismas condiciones sustituyendo probabilidades por **funciones de densidad de probabilidad***

Alberto Ortiz / EPS (última revisión 25/06/2010)

8

Clasificación Bayesiana

- Clasificación Bayesiana: **caso de 2 clases** (ω_1, ω_2)

- Sean las **probabilidades a priori** de las clases $p(\omega_1)$ y $p(\omega_2)$

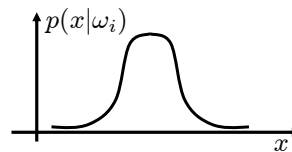
- Si se desconocen, se estiman a través de las muestras de entrenamiento:

$$p(\omega_1) \approx \frac{n_1}{n_1 + n_2}, \quad p(\omega_2) \approx \frac{n_2}{n_1 + n_2}$$

- También se suponen conocidas las f.d.p. (**funciones de densidad de probabilidad**) de las clases

$$p(x|\omega_i), i = 1, 2$$

- Si se desconocen, se han de estimar a partir de los datos de entrenamiento disponibles (existen técnicas que permiten hacerlo)



- Empleando la regla de Bayes, se deduce que:

$$p(\omega_i|x) = \frac{p(x|\omega_i)p(\omega_i)}{p(x)} = \frac{p(x|\omega_i)p(\omega_i)}{\sum_{i=1}^2 p(x|\omega_i)p(\omega_i)}$$

↑
probabilidades totales

Alberto Ortiz / EPS (última revisión 25/06/2010)

9

Clasificación Bayesiana

- Clasificación Bayesiana: **caso de 2 clases** (ω_1, ω_2)

$$\left. \begin{aligned} p(\omega_i|x) &= \frac{p(x|\omega_i)p(\omega_i)}{p(x)} \\ p(\omega_i|x) &> p(\omega_j|x), \forall j \neq i \end{aligned} \right\} \Rightarrow p(x|\omega_i)p(\omega_i) > p(x|\omega_j)p(\omega_j), \forall j \neq i$$

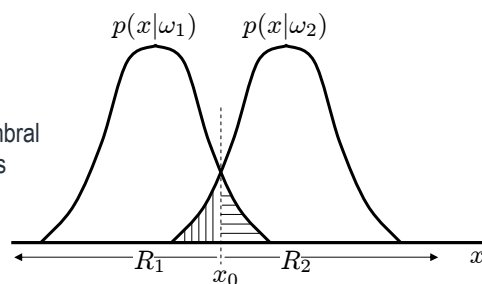
- Si las **probabilidades a priori son iguales** ($p(\omega_i) = 1/M = 0.5$), entonces la regla de clasificación pasa a depender sólo de las f.d.p. de las clases:

$$p(x|\omega_i) > p(x|\omega_j), \forall j \neq i$$

- En el caso unidimensional ($\equiv 1$ característica), x_0 es un umbral que particiona el espacio en dos regiones, R_1 y R_2

- Es obvio que los **errores de clasificación** son inevitables:

- x puede encontrarse en la región R_2 y pertenecer a la clase ω_1 (lo mismo para R_1 y ω_2)



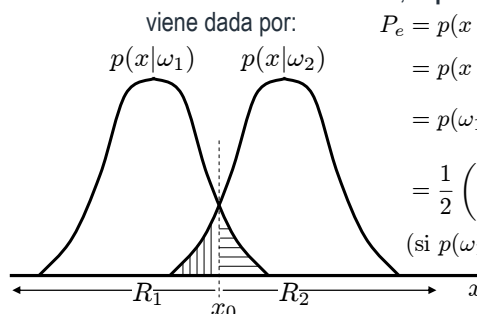
Alberto Ortiz / EPS (última revisión 25/06/2010)

10

Clasificación Bayesiana

• Clasificación Bayesiana: caso de 2 clases (ω_1, ω_2)

– En el caso unidimensional, la **probabilidad de cometer un error de clasificación** viene dada por:



$$\begin{aligned}
 P_e &= p(x \in R_2 \cap x \text{ es de } \omega_1) + p(x \in R_1 \cap x \text{ es de } \omega_2) \\
 &= p(x \in R_2 | \omega_1) p(\omega_1) + p(x \in R_1 | \omega_2) p(\omega_2) \\
 &= p(\omega_1) \int_{R_2} p(x | \omega_1) dx + p(\omega_2) \int_{R_1} p(x | \omega_2) dx \\
 &= \frac{1}{2} \left(\int_{-\infty}^{x_0} p(x | \omega_2) dx + \int_{x_0}^{+\infty} p(x | \omega_1) dx \right) \\
 &\quad (\text{si } p(\omega_1) = p(\omega_2) = 0.5)
 \end{aligned}$$

TEOREMA

El clasificador Bayesiano minimiza la probabilidad del error de clasificación

Es decir: si desplazamos x_0 a izquierda o derecha incrementamos P_e

Clasificación Bayesiana

• Clasificación Bayesiana: caso de 2 clases (ω_1, ω_2)

DEMOSTRACIÓN (de la optimalidad del clasificador Bayesiano)

Por un lado:

$$\begin{aligned}
 P_e &= p(x \in R_2 \cap \omega_1) + p(x \in R_1 \cap \omega_2) = p(x \in R_2 | \omega_1) p(\omega_1) + p(x \in R_1 | \omega_2) p(\omega_2) \\
 &= p(\omega_1) \int_{R_2} p(x | \omega_1) dx + p(\omega_2) \int_{R_1} p(x | \omega_2) dx \\
 &= \int_{R_2} p(\omega_1 | x) p(x) dx + \int_{R_1} p(\omega_2 | x) p(x) dx
 \end{aligned}$$

Por otro lado: $\int_{\Lambda} p(x | \omega_1) dx = 1 \Rightarrow \int_{\Lambda} \frac{p(\omega_1 | x) p(x)}{p(\omega_1)} dx = 1$

$$\Rightarrow \int_{R_1} p(\omega_1 | x) p(x) dx + \int_{R_2} p(\omega_1 | x) p(x) dx = p(\omega_1)$$

Por tanto:

$$P_e = p(\omega_1) - \int_{R_1} (p(\omega_1 | x) - p(\omega_2 | x)) p(x) dx$$

$\Rightarrow P_e$ es mínimo si R_1 se escoge de forma que a lo largo de R_1 $p(\omega_1 | x) > p(\omega_2 | x)$

Clasificación Bayesiana

• Clasificación Bayesiana

- **Ejemplo:** sea un problema de 2 clases equiprobables ($p(\omega_1) = p(\omega_2) = 0.5$) tales que (las f.d.p.s son Gaussianas de varianza 0.5 y medias 0 y 1 respectivamente):

$$p(x|\omega_1) = \frac{1}{\sqrt{\pi}} e^{-x^2}, \quad p(x|\omega_2) = \frac{1}{\sqrt{\pi}} e^{-(x-1)^2}$$

Calcular el umbral óptimo x_0 para probabilidad de error mínima :

$$x_0 : p(\omega_1|x_0) = p(\omega_2|x_0)$$

$$x_0 : p(x_0|\omega_1)p(\omega_1) = p(x_0|\omega_2)p(\omega_2)$$

$$x_0 : e^{-x_0^2} = e^{-(x_0-1)^2} \Rightarrow x_0^2 = (x_0 - 1)^2 = x_0^2 - 2x_0 + 1 \Rightarrow x_0 = 0.5$$

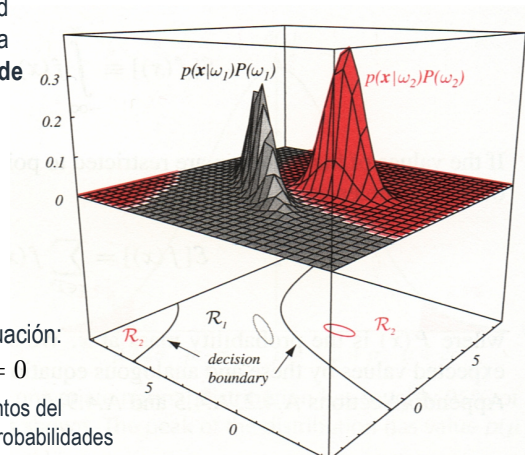
Clasificación Bayesiana

• Clasificación Bayesiana: caso de 2 clases (ω_1, ω_2) y 2 características

- Minimizar la probabilidad de error es equivalente a **particionar el espacio de características en M regiones** (tantas como clases)
- Si las regiones R_i y R_j son contiguas, están separadas por una **curva de decisión** que viene descrita por la ecuación:

$$p(\omega_i|x) - p(\omega_j|x) = 0$$

- Corresponde a los puntos del plano en los que las probabilidades a posteriori coinciden



Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Asumimos que las f.d.p. de las clases obedecen a una **distribución Gaussiana**

L-dimensional:

$$p(x|\omega_i) = \frac{1}{\sqrt{(2\pi)^L |\Sigma_i|}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}, \quad i = 1, \dots, M$$

$$\mu = E[(x_1, x_2, \dots, x_L)]^T = (\mu_1, \mu_2, \dots, \mu_L)^T$$

$$\Sigma = E[(x - \mu)(x - \mu)^T] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1L} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2L} \\ \vdots & \vdots & & \vdots \\ \sigma_{L1} & \sigma_{L2} & \dots & \sigma_L^2 \end{bmatrix}$$

– Modeliza adecuadamente muchos casos y es tratable matemática y computacionalmente

$$\mu_s = \frac{1}{N} \sum_{k=1}^N x_{sk}$$

– En el **caso unidimensional:**

$$p(x|\omega_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\sigma_{st} = \frac{1}{N} \sum_{k=1}^N (x_{sk} - \mu_s)(x_{tk} - \mu_t)$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

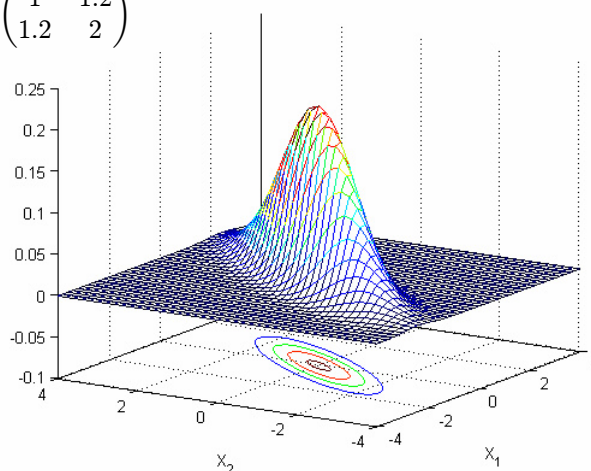
15

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Ejemplo:

$$\mu = (0, 0) \quad \Sigma = \begin{pmatrix} 1 & 1.2 \\ 1.2 & 2 \end{pmatrix}$$



Alberto Ortiz / EPS (última revisión 25/06/2010)

16

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– **Objetivo:** Derivar el clasificador Bayesiano para el caso

$$p(x|\omega_i) = \frac{1}{\sqrt{(2\pi)^L |\Sigma_i|}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}, \quad i = 1, \dots, M$$

- Debido a la forma exponencial de las f.d.p.s, es preferible trabajar con las **funciones de discriminación** $g_i(x)$ siguientes, las cuales involucran a la **función monótona** $\ln(\cdot)$:

$$g_i(x) = \ln(p(x|\omega_i)p(\omega_i)) = \ln p(x|\omega_i) + \ln p(\omega_i)$$

$$g_i(x) = c_i - \frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) + \ln p(\omega_i)$$

$$c_i = -\frac{L}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i|$$

- Finalmente:

$$g_i(x) = -\frac{1}{2}x^T \Sigma_i^{-1} x + \frac{1}{2}x^T \Sigma_i^{-1} \mu_i + \frac{1}{2}\mu_i^T \Sigma_i^{-1} x - \frac{1}{2}\mu_i^T \Sigma_i^{-1} \mu_i + \ln p(\omega_i) + c_i$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

17

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– **Caso de 2 características no correlacionadas**

$$L = 2, \quad \Sigma_i = \begin{pmatrix} \sigma_{i1}^2 & 0 \\ 0 & \sigma_{i2}^2 \end{pmatrix}$$

$$g_i(x) = -\frac{1}{2}x^T \Sigma_i^{-1} x + \frac{1}{2}x^T \Sigma_i^{-1} \mu_i + \frac{1}{2}\mu_i^T \Sigma_i^{-1} x - \frac{1}{2}\mu_i^T \Sigma_i^{-1} \mu_i + \ln p(\omega_i) + c_i \Rightarrow$$

$$\Rightarrow g_i(x) = -\frac{1}{2} \left(\frac{x_1^2}{\sigma_{i1}^2} + \frac{x_2^2}{\sigma_{i2}^2} \right) + \left(\frac{\mu_{i1}x_1}{\sigma_{i1}^2} + \frac{\mu_{i2}x_2}{\sigma_{i2}^2} \right) - \frac{1}{2} \left(\frac{\mu_{i1}^2}{\sigma_{i1}^2} + \frac{\mu_{i2}^2}{\sigma_{i2}^2} \right) + \ln p(\omega_i) + c_i$$

- Las **reglas de decisión** vienen ahora dadas por las ecuaciones $g_i(x) - g_j(x) = 0$

**CLASIFICADOR
CUADRÁTICO**

- $L = 2$: elipses, parábolas, hipérbolas, etc. – **cónicas**, regla = curva 2D
- $L = 3$: elipsoides, paraboloides, hiperboloides, etc. – **cuádricas**, regla = superficie 3D
- $L > 3$: hiperkuádricas

Alberto Ortiz / EPS (última revisión 25/06/2010)

18

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

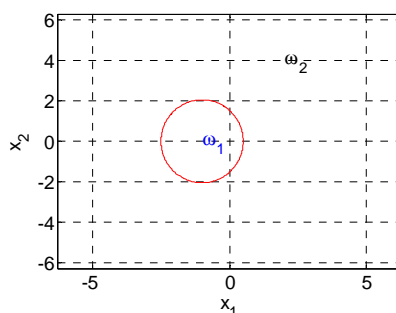
– Ejemplo: (clases equiprobables)

$$\mu_1 = (0, 0)^T, \mu_2 = (1, 0)^T, \Sigma_1 = \begin{pmatrix} 0.10 & 0.00 \\ 0.00 & 0.15 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 0.20 & 0.00 \\ 0.00 & 0.25 \end{pmatrix}$$

$$g_1(x) = -5.0 x_1^2 - 3.3 x_2^2 + 0.2620$$

$$g_2(x) = -2.5 x_1^2 - 2.0 x_2^2 + 5.0 x_1 - 2.840$$

$$g_1(x) - g_2(x) = -2.500 x_1^2 - 1.333 x_2^2 + 3.102 - 5.0 x_1 = 0$$



Alberto Ortiz / EPS (última revisión 25/06/2010)

19

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

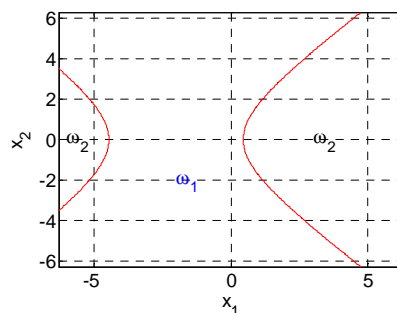
– Ejemplo: (clases equiprobables)

$$\mu_1 = (0, 0)^T, \mu_2 = (1, 0)^T, \Sigma_1 = \begin{pmatrix} 0.10 & 0.00 \\ 0.00 & 0.15 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 0.15 & 0.00 \\ 0.00 & 0.10 \end{pmatrix}$$

$$g_1(x) = -5.0 x_1^2 - 3.333 x_2^2 + 0.2620$$

$$g_2(x) = -3.333 x_1^2 - 5.0 x_2^2 + 6.667 x_1 - 3.071$$

$$g_1(x) - g_2(x) = -1.667 x_1^2 + 1.667 x_2^2 + 3.333 - 6.667 x_1 = 0$$



Alberto Ortiz / EPS (última revisión 25/06/2010)

20

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Clases con la misma matriz de covarianza: hiperplanos de decisión

- Si las clases tienen la misma matriz de covarianza ($\Sigma_i = \Sigma$) entonces el término cuadrático y parte del constante coinciden en todas las funciones de discriminación:

$$g_i(x) = -\frac{1}{2}x^T \Sigma^{-1} x + \frac{1}{2}x^T \Sigma^{-1} \mu_i + \frac{1}{2}\mu_i^T \Sigma^{-1} x - \frac{1}{2}\mu_i^T \Sigma^{-1} \mu_i + \ln p(\omega_i) + c$$

- Por tanto, desaparece de las ecuaciones $g_i(x) - g_j(x) = 0$. Esto permite definir unas funciones de discriminación más útiles:

$$g_i(x) = w_i^T x + w_{i0}$$

$$w_i^T = \mu_i^T \Sigma^{-1}, \quad w_{i0} = \ln p(\omega_i) - \frac{1}{2}\mu_i^T \Sigma^{-1} \mu_i$$

**CLASIFICADOR
LINEAL**

- De esta forma, las funciones de discriminación son lineales (y no cuadráticas) y las reglas de decisión pasan a ser **hiperplanos de decisión**: (2D) rectas, (3D) planos, ...
- Veamos dos casos para la matriz de covarianza: (1) $\Sigma = \sigma^2 I$ y (2) cualquier Σ

Alberto Ortiz / EPS (última revisión 25/06/2010)

21

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Clases con la misma matriz de covarianza: $\Sigma = \sigma^2 I$

- Entonces, las funciones de discriminación pasan a tener las siguientes expresiones:

$$g_i(x) = \frac{1}{\sigma^2} \mu_i^T x + w_{i0} = \frac{1}{\sigma^2} \mu_i^T x + \ln p(\omega_i) - \frac{1}{2\sigma^2} \mu_i^T \mu_i$$

de forma que las reglas de decisión se pueden escribir como:

$$g_{ij}(x) \equiv g_i(x) - g_j(x) = 0$$

$$\Rightarrow \frac{1}{\sigma^2} (\mu_i^T - \mu_j^T) x + \ln p(\omega_i) - \ln p(\omega_j) - \frac{1}{2\sigma^2} (\mu_i^T \mu_i - \mu_j^T \mu_j) = 0$$

$$\Rightarrow (\mu_i - \mu_j)^T x + \sigma^2 \ln \frac{p(\omega_i)}{p(\omega_j)} - \frac{1}{2} (\mu_i - \mu_j)^T (\mu_i + \mu_j) = 0$$

$$\Rightarrow (\mu_i - \mu_j)^T \left[x + \sigma^2 \ln \left(\frac{p(\omega_i)}{p(\omega_j)} \right) \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2} - \frac{1}{2} (\mu_i + \mu_j) \right] = 0$$

$$\Rightarrow w^T (x - x_0) = 0$$

$$w = \mu_i - \mu_j, \quad x_0 = \frac{1}{2} (\mu_i + \mu_j) - \sigma^2 \ln \left(\frac{p(\omega_i)}{p(\omega_j)} \right) \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2}$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

22

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Clases con la misma matriz de covarianza: $\Sigma = \sigma^2 I$

• reglas de decisión:

$$g_{ij}(x) : w^T(x - x_0) = 0$$

$$w = \mu_i - \mu_j, \quad x_0 = \frac{1}{2}(\mu_i + \mu_j) - \sigma^2 \ln \left(\frac{p(\omega_i)}{p(\omega_j)} \right) \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2}$$

• caso de 2 características:

$$g_{ij}(x) : w^T(x - x_0) = 0$$

$$\Rightarrow \Delta\mu_1 x_1 + \Delta\mu_2 x_2 - \Delta\mu_1 x_{01} - \Delta\mu_2 x_{02} = 0$$

$$\Rightarrow Ax_1 + Bx_2 + C = 0$$

¿CUÁL es esta recta?

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Clases con la misma matriz de covarianza: $\Sigma = \sigma^2 I$

• reglas de decisión: **caso de 2 características**

$$g_{ij}(x) : w^T(x - x_0) = 0$$

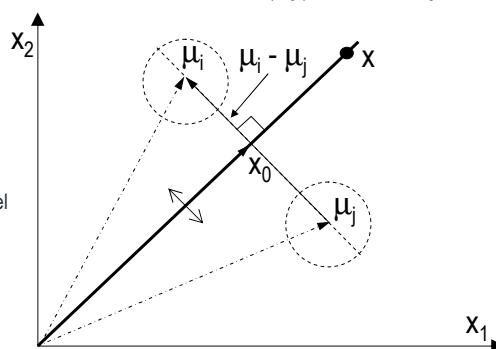
$$w = \mu_i - \mu_j, \quad x_0 = \frac{1}{2}(\mu_i + \mu_j) - \sigma^2 \ln \left(\frac{p(\omega_i)}{p(\omega_j)} \right) \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|^2}$$

• cualquier punto x

tal que $x - x_0$ sea
ortogonal a $\mu_i - \mu_j$
pertenece a la recta

• x_0 siempre está sobre
el vector $\mu_i - \mu_j$

- si $p(\omega_i) = p(\omega_j)$, x_0 es el
promedio de μ_i y μ_j
- si $p(\omega_i) < p(\omega_j)$, x_0
se desplaza hacia μ_i
sobre $\mu_i - \mu_j$

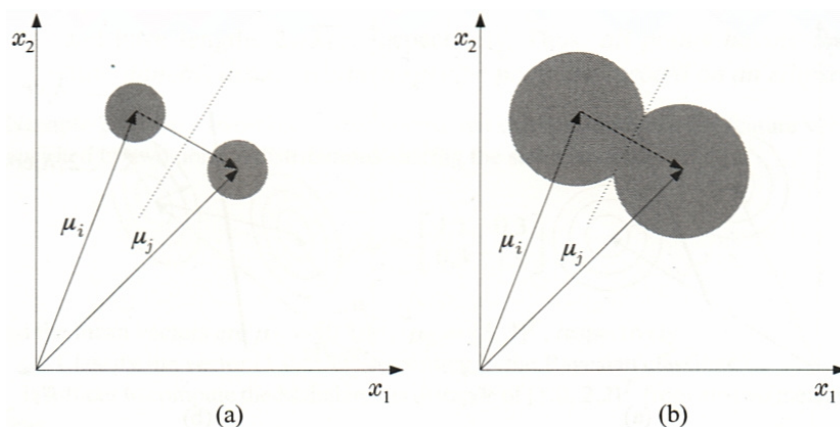


Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Clases con la **misma matriz de covarianza**: $\Sigma = \sigma^2 I$

- los círculos corresponden a $3\sigma \equiv 98\%$



Alberto Ortiz / EPS (última revisión 25/06/2010)

25

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Clases con la **misma matriz de covarianza**: cualquier Σ

- Recuperamos las funciones de discriminación lineales originales:

$$g_i(x) = w_i^T x + w_{i0}$$

$$w_i^T = \mu_i^T \Sigma^{-1}, \quad w_{i0} = \ln p(\omega_i) - \frac{1}{2} \mu_i^T \Sigma^{-1} \mu_i$$

- Entonces:

$$g_{ij}(x) \equiv g_i(x) - g_j(x) = 0$$

$$\Rightarrow (\mu_i - \mu_j)^T \Sigma^{-1} x + \ln \left(\frac{p(\omega_i)}{p(\omega_j)} \right) - \frac{1}{2} (\mu_i - \mu_j)^T \Sigma^{-1} (\mu_i + \mu_j) = 0$$

$$\Rightarrow (\mu_i - \mu_j)^T \Sigma^{-1} \left[x + \ln \left(\frac{p(\omega_i)}{p(\omega_j)} \right) \frac{\mu_i - \mu_j}{(\mu_i - \mu_j)^T \Sigma^{-1} (\mu_i - \mu_j)} - \frac{1}{2} (\mu_i + \mu_j) \right] = 0$$

$$\Rightarrow w^T (x - x_0) = 0$$

$$w = \Sigma^{-1} (\mu_i - \mu_j), \quad x_0 = \frac{1}{2} (\mu_i + \mu_j) - \ln \left(\frac{p(\omega_i)}{p(\omega_j)} \right) \frac{\mu_i - \mu_j}{(\mu_i - \mu_j)^T \Sigma^{-1} (\mu_i - \mu_j)}$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

26

Clasificación Bayesiana

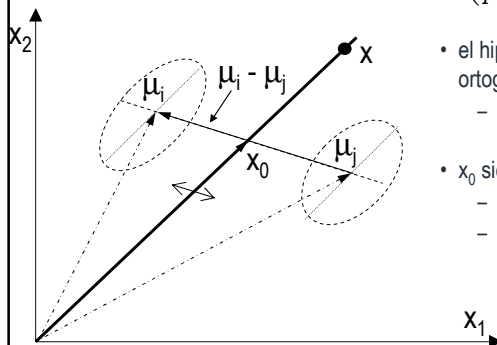
• Clasificación Bayesiana para distribuciones normales

– Clases con la **misma matriz de covarianza**: cualquier Σ

$$g_{ij} : w^T(x - x_0) = 0$$

$$w = \Sigma^{-1}(\mu_i - \mu_j)$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \ln\left(\frac{p(\omega_i)}{p(\omega_j)}\right) \frac{\mu_i - \mu_j}{(\mu_i - \mu_j)^T \Sigma^{-1}(\mu_i - \mu_j)}$$



- el hiperplano de decisión ya no es necesariamente ortogonal a $\mu_i - \mu_j$ sino a $\Sigma^{-1}(\mu_i - \mu_j)$
 - $\Sigma^{-1}(\mu_i - \mu_j)$ es el resultado de transformar $(\mu_i - \mu_j)$ a través de la matriz Σ^{-1}
- x_0 siempre está sobre el vector $\mu_i - \mu_j$
 - si $p(\omega_i) = p(\omega_j)$, x_0 es el promedio de μ_i y μ_j
 - si $p(\omega_i) < p(\omega_j)$, x_0 se desplaza hacia μ_i sobre $\mu_i - \mu_j$

Alberto Ortiz / EPS (última revisión 25/06/2010)

27

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– **Ejemplo**: en un problema de clasificación bidimensional en dos clases equiprobables, las clases presentan dos distribuciones normales con los siguientes parámetros:

$$\mu_1 = (0, 0)^T, \mu_2 = (3, 3)^T, \Sigma = \Sigma_1 = \Sigma_2 = \begin{pmatrix} 1.1 & 0.3 \\ 0.3 & 1.9 \end{pmatrix}$$

Clasificar el vector $\mathbf{x} = (1.0, 2.2)^T$ utilizando un clasificador Bayesiano.

$$g_{12} = w^T(x - x_0)$$

$$w = \Sigma^{-1}(\mu_1 - \mu_2)$$

$$x_0 = \frac{1}{2}(\mu_1 + \mu_2) - \ln\left(\frac{p(\omega_1)}{p(\omega_2)}\right) \frac{\mu_1 - \mu_2}{(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2)}$$

$$\begin{aligned} g_{12} &= (-3, -3) \begin{pmatrix} 1.1 & 0.3 \\ 0.3 & 1.9 \end{pmatrix}^{-1} ((1, 2.2)^T - (1.5, 1.5)^T) \\ &= 0.36 > 0 \end{aligned}$$

- Por tanto, $\mathbf{x} \rightarrow \omega_1$.

Alberto Ortiz / EPS (última revisión 25/06/2010)

28

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

- Hasta ahora hemos considerado **pares de clases** y hemos derivado las fronteras entre pares de clases a través de las funciones de discriminación:

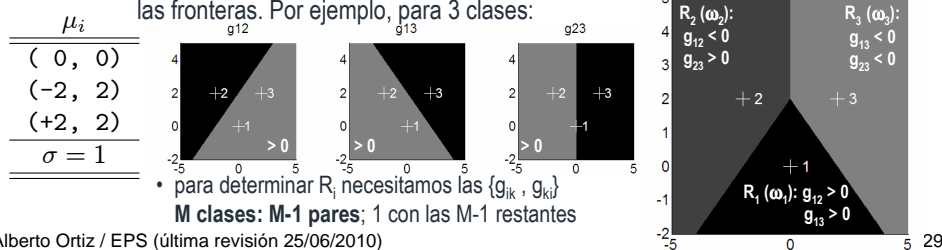
$$g_{ij}(x) = g_i(x) - g_j(x)$$

- En el caso de un problema de **2 clases**:

si $g_{12} > 0$, entonces $x \rightarrow \omega_1$

si $g_{12} < 0$, entonces $x \rightarrow \omega_2$

- Si hay **más clases**, hay que determinar $g_i | g_i > g_j, \forall j \neq i$, por lo que hay que emplear varias g_{ij} para decidir, ya que las g_{ij} aisladas sólo nos sirven para trazar las fronteras. Por ejemplo, para 3 clases:



Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Clasificadores de distancia mínima

- Vamos a ver lo anterior desde otro punto de vista
- Asumimos **clases equiprobables con la misma matriz de covarianza**. Entonces:

$$g_i(x) = c_i - \frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1}(x - \mu_i) + \ln p(\omega_i) \left\} \rightarrow c_i = -\frac{L}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i|$$

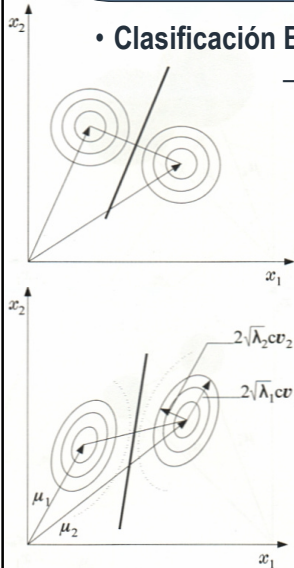
$$\rightarrow g_i(x) = -\frac{1}{2}(x - \mu_i)^T \Sigma^{-1}(x - \mu_i)$$

- Asignamos x a la clase para la que la probabilidad es mayor $\Rightarrow g_i(x) > g_j(x) \forall j \neq i$
 - (1) Si $\Sigma = \sigma^2 I$, $g_i(x)$ es mayor cuanto más cerca está x de μ_i
 \Rightarrow asignar x a la clase cuyo centro μ_i está más próximo (**distancia euclídea**)
 - (2) Para Σ **genérico**, hay que asignar x a la clase para la que la expresión siguiente es menor:
 $d_m^2 = (x - \mu_i)^T \Sigma^{-1}(x - \mu_i)$
 - $d_m \equiv$ **distancia de Mahalanobis**

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

– Clasificadores de distancia mínima



- En el caso (1), los puntos a la misma distancia (euclídea) se encuentran sobre **circunferencias** (hiperesferas en el caso general):

$$d_e = \sqrt{\|x - \mu_i\|^2} = c$$

$$(x_1 - \mu_{i1})^2 + (x_2 - \mu_{i2})^2 = c^2$$

- En el caso (2), los puntos a la misma distancia (de mahalanobis) se encuentran sobre **elipses** (hiperelipsoides en el caso general):

$$d_m^2 = (x - \mu_i)^T \Sigma^{-1} (x - \mu_i) = c^2$$

$$\Sigma = \Phi \Lambda \Phi^T \Rightarrow \Sigma^{-1} = \Phi^{-T} \Lambda^{-1} \Phi^{-1} = \Phi \Lambda^{-1} \Phi^T$$

$$(x - \mu_i)^T \Phi \Lambda^{-1} \Phi^T (x - \mu_i) = c^2$$

$$(x' - \mu'_i)^T \Lambda^{-1} (x' - \mu'_i) = c^2$$

$$\frac{(x_1 - \mu_{i1})^2}{\lambda_1} + \frac{(x_2 - \mu_{i2})^2}{\lambda_2} = c^2$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

31

Clasificación Bayesiana

• Clasificación Bayesiana para distribuciones normales

- **Ejemplo:** en un problema de clasificación bidimensional en dos clases equiprobables, las clases presentan dos distribuciones normales con los siguientes parámetros:

$$\mu_1 = (0, 0)^T, \mu_2 = (3, 3)^T, \Sigma = \Sigma_1 = \Sigma_2 = \begin{pmatrix} 1.1 & 0.3 \\ 0.3 & 1.9 \end{pmatrix}$$

Clasificar el vector $\mathbf{x} = (1.0, 2.2)^T$ utilizando un clasificador Bayesiano.

$$d_m^2(x, \mu_1) = (x - \mu_1)^T \Sigma^{-1} (x - \mu_1)$$

$$= (1.0, 2.2) \begin{pmatrix} 0.95 & -0.15 \\ -0.15 & 0.55 \end{pmatrix} \begin{pmatrix} 1.0 \\ 2.2 \end{pmatrix} = 2.952$$

$$d_m^2(x, \mu_2) = (x - \mu_2)^T \Sigma^{-1} (x - \mu_2)$$

$$= (-2.0, -0.8) \begin{pmatrix} 0.95 & -0.15 \\ -0.15 & 0.55 \end{pmatrix} \begin{pmatrix} -2.0 \\ -0.8 \end{pmatrix} = 3.672$$

- $d_m(x, \mu_1) < d_m(x, \mu_2) \Rightarrow \mathbf{x} \rightarrow \omega_1$.
- NOTA: las distancias euclídeas serían $d_e(x, \mu_1) = 2.417$ y $d_e(x, \mu_2) = 2.154$, por lo que, si las utilizáramos, haríamos $\mathbf{x} \rightarrow \omega_2$.

Alberto Ortiz / EPS (última revisión 25/06/2010)

32

- Introducción
- Clasificación Bayesiana
- Estimación de funciones de densidad de probabilidad
- Funciones de discriminación lineales y el algoritmo del perceptrón

Estimación de funciones de densidad de probabilidad

- **Estimación de funciones de densidad de probabilidad**
 - El clasificador Bayesiano asume que disponemos de f.d.p.s de las clases del problema.
 - Hay diferentes métodos para obtener este tipo de información:
 - Se conoce la expresión de la f.d.p. pero se desconocen los parámetros
 - Estimadores de **máxima verosimilitud**
 - otros ...
 - No se conoce la expresión de la f.d.p. → **estimación no paramétrica**
 - Método de las **ventanas de Parzen**
 - Método de los **k vecinos más próximos** (KNN, *K-nearest neighbour*)
 - otros ...

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimadores de máxima verosimilitud

- Sea un problema de clasificación en **M clases** cuyos vectores de características se distribuyen de acuerdo con $p(\mathbf{x}|\omega_i; \theta_i)$, $i = 1, \dots, M$, donde θ_i es el **vector de parámetros** para la clase ω_i
 - Se trata de estimar θ_i a partir de un conjunto de muestras x_1, x_2, \dots, x_N correspondientes a la clase ω_i
- Asumimos que las muestras de una clase no afectan a la estimación de parámetros para las otras clases para poder formular el problema independientemente de la clase
 - ⇒ La estimación se repite para cada clase
- De esta forma, dadas las muestras estadísticamente independientes x_1, x_2, \dots, x_N provenientes de $p(x; \theta)$, calculamos la f.d.p. conjunta siguiente:

$$p(X; \theta) \equiv p(x_1, x_2, \dots, x_N; \theta) = \prod_{k=1}^N p(x_k; \theta)$$

- Entonces la **estimación de θ de máxima verosimilitud** corresponde a:

$$\hat{\theta}_{ML} | p(X; \hat{\theta}_{ML}) = \max\{p(X; \theta)\}$$

- Representa el θ que mejor se adecua a las muestras x_1, x_2, \dots, x_N

Alberto Ortiz / EPS (última revisión 25/06/2010)

35

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimadores de máxima verosimilitud

- Para ello hay que derivar e igualar a 0. Para simplificar los cálculos acudiremos una vez más a la función $\ln(\cdot)$ para definir la **función log-verosimilitud**:

$$L(\theta) \equiv \ln \prod_{k=1}^N p(x_k; \theta) = \sum_{k=1}^N \ln p(x_k; \theta)$$

- Ahora sí derivamos e igualamos a 0:

$$\frac{\partial L(\theta)}{\partial \theta} = \sum_{k=1}^N \frac{\partial \ln p(x_k; \theta)}{\partial \theta} = \sum_{k=1}^N \frac{1}{p(x_k; \theta)} \frac{\partial p(x_k; \theta)}{\partial \theta} = 0$$

- Para valores de N suficientemente elevados, el estimador de máxima verosimilitud es **asintóticamente no sesgado**, responde a una **distribución normal** y presenta la **mínima varianza posible**

Alberto Ortiz / EPS (última revisión 25/06/2010)

36

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimadores de máxima verosimilitud

- Distribución **Gaussiana** L-dimensional y Σ conocida

$$\hat{\mu}_{ML} = \frac{1}{N} \sum_{k=1}^N x_k$$

- Distribución **Gaussiana** L-dimensional, μ y Σ desconocidas

$$\hat{\mu}_{ML} = \frac{1}{N} \sum_{k=1}^N x_k, \quad \hat{\Sigma}_{ML} = \frac{1}{N} \sum_{k=1}^N (x_k - \hat{\mu}_{ML})(x_k - \hat{\mu}_{ML})^T$$

Estimación de funciones de densidad de probabilidad

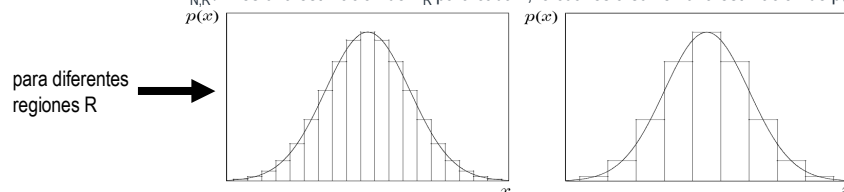
• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica

- Se trata de estimar una cierta f.d.p. $p(\mathbf{x})$ sin estimar previamente sus parámetros. De hecho, ni siquiera se intenta asimilar $p(\mathbf{x})$ a alguna f.d.p. conocida.
- Sea una cierta región R del espacio de características. La probabilidad P_R de que un cierto \mathbf{x} pertenezca a R viene dada por:

$$P_R = p(x \in R) = \int_R p(x') dx'$$

- Supongamos que disponemos de N muestras independientes $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ correspondientes a la f.d.p. que queremos estimar.
 - $k_{N,R}$ = cuántas de las N muestras pertenecen a la región R
 - $k_{N,R}/N$ es una estimación de P_R para cada x , lo cual es a su vez una estimación de $p(x)$



Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica

- Ahora hacemos R suficientemente pequeña como para que $p(x)$ sea aproximadamente constante dentro de R. Entonces:

$$P_R = \int_R p(x') dx' \approx p(x) \int_R dx' = p(x)V$$

... donde V es el (hiper)volumen ocupado por la región R (1D – longitud, 2D – área, 3D – volumen, etc)

- p.e. si R es un (hiper)cubo de dimensión L y longitud de lado h, $V = h^L$

- Por tanto: $p(x)V \approx \frac{k_{N,R}}{N} \Rightarrow p(x) \approx \frac{k_{N,R}/N}{V} = \frac{P_R}{V} = p_{N,R}(x)$

- $p_{N,R}(x) \rightarrow p(x)$ a medida que $N \rightarrow \infty$ si se verifica:

- $V \rightarrow 0$ (regiones pequeñas)
- $k_{N,R} \rightarrow \infty$ (número suficiente de muestras en cada R)
- $k_{N,R}/N \rightarrow 0$ (número total de muestras elevado)

- En resumen:** para cada x, $p(x)$ se aproxima definiendo una región R pequeña alrededor de x y contando cuántos de los x_i “caen” en R ($= k_{N,R}$); $k_{N,R} \gg 1$ y $N \gg 1$, entonces $p_{N,R}(x) \approx p(x)$

Alberto Ortiz / EPS (última revisión 25/06/2010)

39

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **ventanas de Parzen** (Parzen, 1962)

- Supongamos una región R con forma de (hiper)cubo L-dimensional de lado h_N . Entonces:

$$V_N = (h_N)^L$$

- Sea la siguiente función (**función de ventana o kernel**):

$$\varphi(u) = \begin{cases} 1 & |u_j| \leq \frac{1}{2}, j = 1, \dots, L \\ 0 & \text{en caso contrario} \end{cases}$$

- Entonces, para un cierto \mathbf{x} :

$$\varphi\left(\frac{x - x_i}{h_N}\right) = 1 \text{ si } x_i \in \text{(hiper)cubo de volumen } V_N \text{ centrado en } x$$

- Entonces, el número de muestras que se encuentran dentro del (hiper)cubo centrado en \mathbf{x} es:

$$k_N(x) = \sum_{i=1}^N \varphi\left(\frac{x - x_i}{h_N}\right)$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

40

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **ventanas de Parzen**

• Finalmente:

$$p_N(x) = \frac{k_N(x)/N}{V_N} = \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N^L} \varphi\left(\frac{x - x_i}{h_N}\right)$$

• Por tanto, dado cierto \mathbf{x} , para obtener la estimación de $\mathbf{p}(\mathbf{x})$: (caso 1D)

```
function p = parzen_cubo_1D(x,X,h)
% x = punto a evaluar
% X = vector de muestras
% h = longitud del lado del (hiper)cubo

N = length(X); kn = 0;
for i=1:N
    if abs((x-X(i))/h) <= 0.5, kn = kn+1; end
end
p = (kn/N)/h;
```

Alberto Ortiz / EPS (última revisión 25/06/2010)

41

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **ventanas de Parzen**

• Comentarios sobre p_N :

– p_N es una f.d.p. legítima

1. $p_N(x) \geq 0, \forall x$ (obvio, es una suma de 1's)

2. $\int p_N(x) dx = 1$

$$\begin{aligned} \int \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N^L} \varphi\left(\frac{x - x_i}{h_N}\right) dx &= \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N^L} \int \varphi\left(\frac{x - x_i}{h_N}\right) dx = \\ &= \{u = x - x_i\} = \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N^L} \int \varphi\left(\frac{u}{h_N}\right) du = \\ &= \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N^L} \int_{(-h_N/2, \dots, -h_N/2)}^{(+h_N/2, \dots, +h_N/2)} du = \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N^L} h_N^L = 1 \end{aligned}$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

42

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **ventanas de Parzen**

• Comentarios sobre p_N :

– p_N es el promedio de N funciones centradas en las muestras x_i

$$p_N(x) = \frac{1}{N} \sum_{i=1}^N \frac{1}{h_N^L} \varphi\left(\frac{x - x_i}{h_N}\right)$$

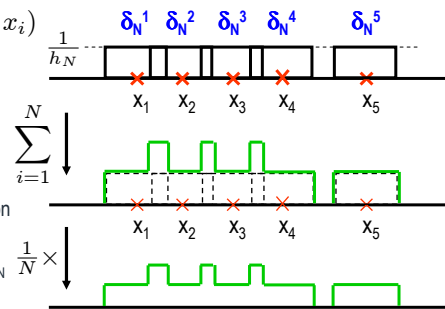
$$p_N(x) = \frac{1}{N} \sum_{i=1}^N \delta_N(x - x_i)$$

$$\delta_N(u) = \frac{1}{h_N^L} \varphi\left(\frac{u}{h_N}\right)$$

– Si h_N es grande, la amplitud de δ_N es pequeña y p_N es la superposición de N funciones "anchas"

– Si h_N es pequeño, la amplitud de δ_N es grande y p_N es la superposición de N pulsos "agudos"

– $h_N \rightarrow 0 \Rightarrow \delta_N \rightarrow \delta$



Alberto Ortiz / EPS (última revisión 25/06/2010)

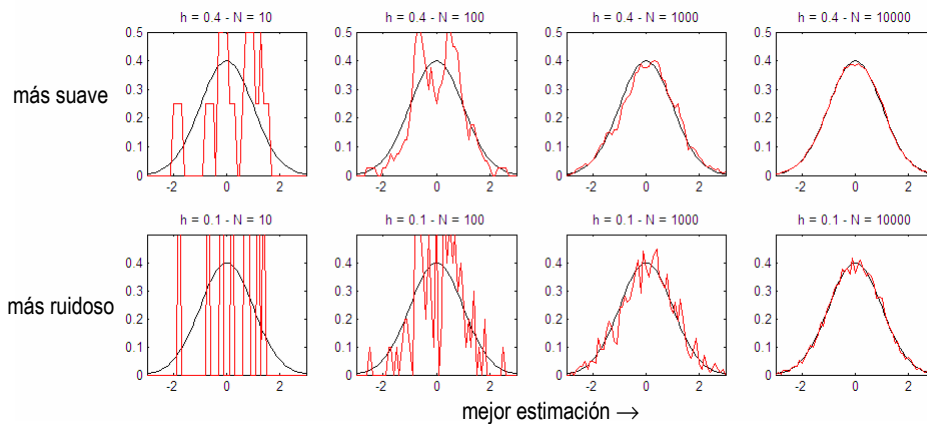
43

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **ventanas de Parzen**

• **Ejemplo:** N valores aleatorios extraídos de una distribución $N(0,1)$ – *kernel rectangular*



Alberto Ortiz / EPS (última revisión 25/06/2010)

44

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **ventanas de Parzen**

- Como ya hemos visto en el ejemplo anterior, al aproximar funciones continuas $p(\cdot)$ por funciones escalón discontinuas $[p(\cdot)]$, la estimación resultante presenta también **discontinuidades**

- Para evitarlo, el propio Parzen sugirió utilizar **kernels $\varphi(\cdot)$ continuos**

- Se puede demostrar que la estimación $p_N(x)$ resultante es una f.d.p. legítima si:

$$\varphi(u) \geq 0 \text{ y } \int \varphi(u) du = 1$$

- Uno de los más utilizados es el **kernel Gaussiano** de media 0 y varianza 1:

$$\varphi(u) = \frac{1}{\sqrt{(2\pi)^L}} e^{-\frac{1}{2}u^T u}$$

```
function p = parzen_gauss_1D(x,X,h)
% x = punto a evaluar
% X = vector de muestras
% h = ancho de banda (desviacion tipica)
N = length(X); kn = 0;
for i=1:N
    kn = kn + 1/(sqrt(2*pi))*exp(-0.5*((x-X(i))/h)^2);
end
p = (kn/N)/h;
```

Alberto Ortiz / EPS (última revisión 25/06/2010)

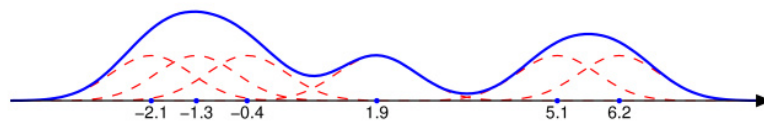
45

Estimación de funciones de densidad de probabilidad

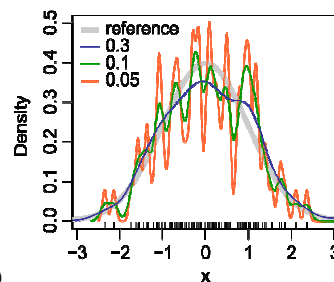
• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **ventanas de Parzen**

- Ahora $p_N(x)$ se obtiene como el **promedio de N Gaussianas** centradas en las muestras x_i



- Efecto de variar el ancho de banda h_N



Alberto Ortiz / EPS (última revisión 25/06/2010)

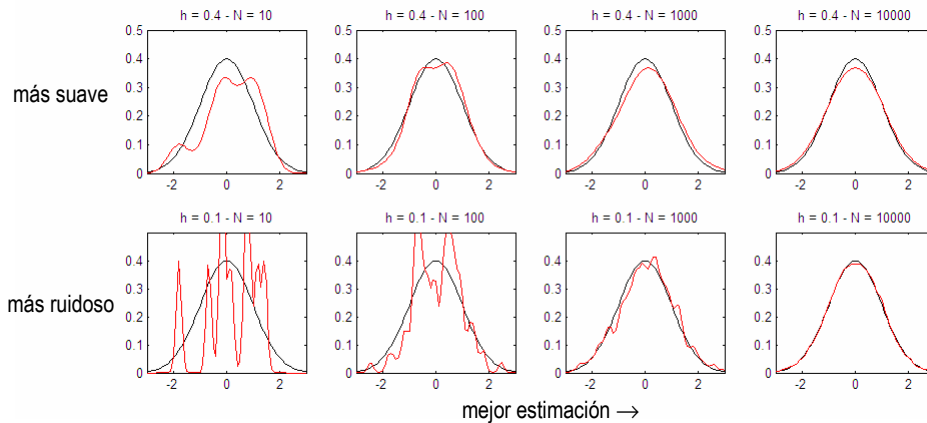
46

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **ventanas de Parzen**

- **Ejemplo:** N valores aleatorios extraídos de una distribución $N(0,1)$ – *kernel* Gaussiano



Alberto Ortiz / EPS (última revisión 25/06/2010)

47

Estimación de funciones de densidad de probabilidad

• Estimación de funciones de densidad de probabilidad

– Estimación no paramétrica: **k vecinos más próximos**

- Dado x y una colección de muestras x_1, x_2, \dots, x_N , provenientes de una cierta f.d.p. $p(x)$, para estimar $p(x)$:

– **Método de las ventanas de Parzen** – se fija un volumen de búsqueda alrededor de x , V_N , y se determina el número k_N de muestras pertenecientes a dicho volumen

$$p_N(x) = \frac{k_N/N}{V_N}$$

– **Método de los k vecinos más próximos** – se buscan las k_N muestras más próximas a x y se determina el volumen que las contiene V_{kN}

$$p_N(x) = \frac{k_N/N}{V_{kN}}$$

```
function p = knn_1D(x,X,k)
% x = punto a evaluar
% X = vector de muestras
% k = número de vecinos
N = length(X);
d = abs(X - x);
ds = sort(d);
V = 2*ds(min(N,k));
p = (k/N)/V;
```

Alberto Ortiz / EPS (última revisión 25/06/2010)

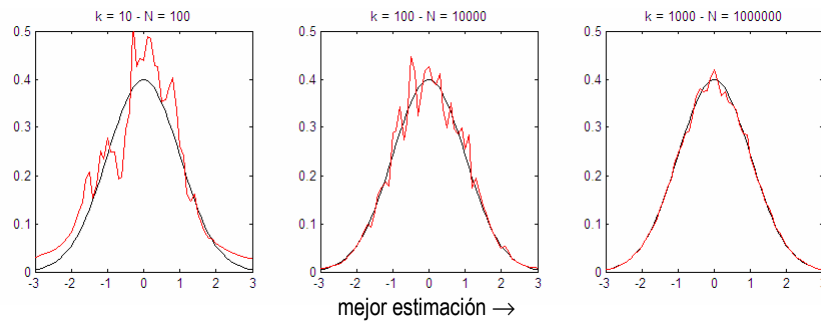
48

Estimación de funciones de densidad de probabilidad

- **Estimación de funciones de densidad de probabilidad**

- Estimación no paramétrica: **k vecinos más próximos**

- **Ejemplo:** N valores aleatorios extraídos de una distribución $N(0,1)$



- En este caso, se ha empleado: $k_N = \sqrt{N}$
 - En el caso de las ventanas de Parzen, se sugiere: $h_N = \frac{h_1}{\sqrt{N}}$

Índice

- Introducción
- Clasificación Bayesiana
- Estimación de funciones de densidad de probabilidad
- **Funciones de discriminación lineales y el algoritmo del perceptrón**

Funciones de discriminación lineales y el algoritmo del perceptrón

- Ya hemos visto que, dependiendo de las f.d.p.s de las clases (caso Gaussiano), un clasificador Bayesiano puede derivar en un conjunto de **funciones de discriminación lineales**. Por ejemplo, para 2 clases:

$$g_{12}(x) = w^T(x - x_0), \quad g_{12}(x) \begin{cases} > 0 & x \in \omega_1 \\ < 0 & x \in \omega_2 \end{cases}$$

– clasificador simple y computacionalmente muy interesante

- En esta sección del tema, nos volvemos a concentrar en funciones de discriminación lineales, pero desde una perspectiva diferente: **no asumimos una f.d.p. para las clases**

– Por tanto, independientemente de la f.d.p. de las clases, esperamos que éstas sean separables mediante (hiper)planos (1D – punto, 2D – recta, 3D – plano, etc.)

• En este caso se dice que **las clases son linealmente separables**

– Veremos cómo se puede encontrar un (hiper)plano que separe las clases entre sí (**algoritmo del perceptrón**)

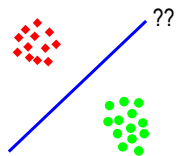
Alberto Ortiz / EPS (última revisión 25/06/2010)

51

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales

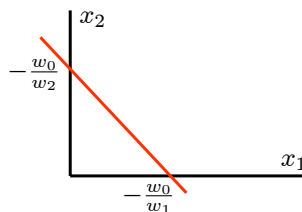
– **Objetivo:** encontrar un (hiper)plano que permita separar las muestras de entrenamiento de 2 clases



$$g_{12}(x) = w^T(x - x_0), \quad g_{12}(x) \begin{cases} > 0 & x \in \omega_1 \\ < 0 & x \in \omega_2 \end{cases}$$

$$\text{(hiper)plano: } x \mid g_{12}(x) = w^T(x - x_0) = 0$$

– Por ejemplo, $L = 2$ características:



$$(w_1, w_2) \left[\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} x_{01} \\ x_{02} \end{pmatrix} \right] = 0$$

$$w_1 x_1 + w_2 x_2 - (w_1 x_{01} + w_2 x_{02}) = 0$$

$$w_1 x_1 + w_2 x_2 + w_0 = 0$$

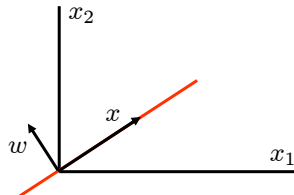
Alberto Ortiz / EPS (última revisión 25/06/2010)

52

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales

– Por el momento, consideramos (hiper)planos que **pasan por el origen**:



$$w_0 = 0$$

↓

$$w_1 x_1 + w_2 x_2 = 0$$

↓

$$w^T x = 0$$

• $x_0 = 0$ es equivalente a hacer $w_0 = 0$ en el caso general

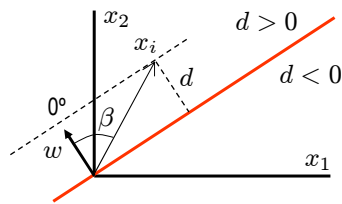
• w es un vector ortogonal a la recta

$$w^T x_i = \|w\| \|x_i\| \cos \beta \neq 0$$

$$\Downarrow \quad \|w\| = 1$$

$$\|x_i\| \cos \beta = d$$

$$\begin{cases} \beta \in [-\frac{\pi}{2}, +\frac{\pi}{2}] & w^T x_i = d > 0 \\ |\beta| > \frac{\pi}{2} & w^T x_i = d < 0 \end{cases}$$



• $w^T x_i$ indica cómo de lejos está la muestra del (hiper)plano de discriminación

Alberto Ortiz / EPS (última revisión 25/06/2010)

53

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **planteamiento del problema**

– Asumimos que las clases ω_1 y ω_2 son linealmente separables y, por tanto, que existe un (hiper)plano definido por $w^T x = 0$ tal que:

$$w^T x_i > 0, \quad \forall x_i \in \omega_1$$

$$w^T x_i < 0, \quad \forall x_i \in \omega_2$$

– Para encontrar el (hiper)plano, planteamos la siguiente función (**coste del perceptrón**):

$$J(w) = \sum_{x_i \in \mathcal{Y}} (\delta_{x_i} w^T x_i)$$

- donde:
- \mathcal{Y} es el conjunto de muestras x_i mal clasificadas por w
 - $\delta_{x_i} = -1$ si $x_i \in \omega_1$ y $\delta_{x_i} = +1$ si $x_i \in \omega_2$
 - $J(w) > 0, \forall w$
($x_i \in \omega_1$ pero $x_i \rightarrow \omega_2$, entonces $\delta_{x_i} w^T x_i = (-1)(< 0) > 0$)
 - $J(w) = 0$ si las muestras están bien clasificadas ($\mathcal{Y} = \emptyset$)
 - $J(w)$ es continua y lineal a trozos (si variamos w , J varía linealmente hasta que \mathcal{Y} cambia) \Rightarrow minimización no trivial

Alberto Ortiz / EPS (última revisión 25/06/2010)

54

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

- El siguiente esquema iterativo (**algoritmo del perceptrón**, Rosenblatt 1950s) encuentra un hiperplano que separa las 2 clases si éstas son **linealmente separables**:

$$w(t+1) = w(t) - \rho_t \sum_{x_i \in \mathcal{Y}} \delta_{x_i} x_i$$

$$w(0) = \text{cualquier vector de } \mathbb{R}^L$$

- La convergencia se produce si las clases son **linealmente separables** y si la **secuencia de valores ρ_t** cumple ciertas condiciones:

$$\lim_{t \rightarrow \infty} \sum_{k=0}^t \rho_k = \infty, \quad \lim_{t \rightarrow \infty} \sum_{k=0}^t \rho_k^2 < \infty$$

- por ejemplo, $\rho_t = c/t$ y $\rho_t = \rho$ (ρ acotado) cumplen con las condiciones
- la secuencia ρ_t determina la velocidad de convergencia

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

- **Variantes del algoritmo básico:**

- (1a) Para tratar clases que necesitan un (hiper)plano que no pase por el origen, los vectores de características pasan a ser $x_i^* = (x_i, 1)^T$ y se plantea el (hiper)plano:

$$w^T x + w_0 = 0 \quad \equiv \quad \underbrace{(w^*)^T}_{w^T} \begin{pmatrix} x_{\#1} \\ x_{\#2} \\ \vdots \\ x_{\#L} \\ 1 \end{pmatrix} = (w^*)^T x^* = 0$$

► Para simplificar, emplearemos w^T en vez de $(w^*)^T$ y x en vez de x^*

- (1b) Por otro lado, el cálculo de la modificación de w en cada iteración se puede ver como:

$$w(t+1) = w(t) - \rho_t \sum_{x_i \in \mathcal{Y}} \delta_{x_i} x_i \quad \equiv \quad \begin{array}{l} S = \overbrace{(0, 0, \dots, 0)}^{L+1} \\ \text{para } i = 1 \text{ hasta } n_{\text{total_muestras}} \\ \quad \text{si } x_i \in \omega_1 \text{ y } w(t)^T x_i < 0, \text{ entonces } S = S + \rho_t x_i \\ \quad \text{sino} \\ \quad \text{si } x_i \in \omega_2 \text{ y } w(t)^T x_i > 0, \text{ entonces } S = S - \rho_t x_i \\ \text{fin para} \\ w(t+1) = w(t) + S \end{array}$$

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

– Variantes del algoritmo básico:

(1) **Ejemplo:**

– La línea discontinua corresponde a:

$$(1, 1, -0.5)x = x_1 + x_2 - 0.5 = 0$$

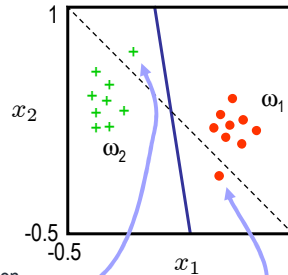
donde el vector $w = (1, 1, -0.5)^T$ es el resultado del paso anterior del algoritmo del perceptrón original con $\rho_t = \rho = 0.7$

– Las muestras clasificadas incorrectamente son $(0.40, 0.05)^T$ y $(-0.20, 0.75)^T$

– Aplicando un nuevo paso del algoritmo:

$$w(t+1) = w(t) - \rho_t \sum_{x_i \in \mathcal{Y}} \delta_{x_i} x_i \quad w(t+1) = \begin{pmatrix} 1 \\ 1 \\ -0.5 \end{pmatrix} - 0.7(-1) \begin{pmatrix} 0.40 \\ 0.05 \\ 1 \end{pmatrix} - 0.7(+1) \begin{pmatrix} -0.20 \\ 0.75 \\ 1 \end{pmatrix} = \begin{pmatrix} 1.42 \\ 0.51 \\ -0.5 \end{pmatrix}$$

– El resultado clasifica correctamente todas las muestras y el algoritmo termina con el (hiper)plano: $1.42x_1 + 0.51x_2 - 0.5 = 0$



Alberto Ortiz / EPS (última revisión 25/06/2010)

57

Funciones de discriminación lineales y el algoritmo del perceptrón

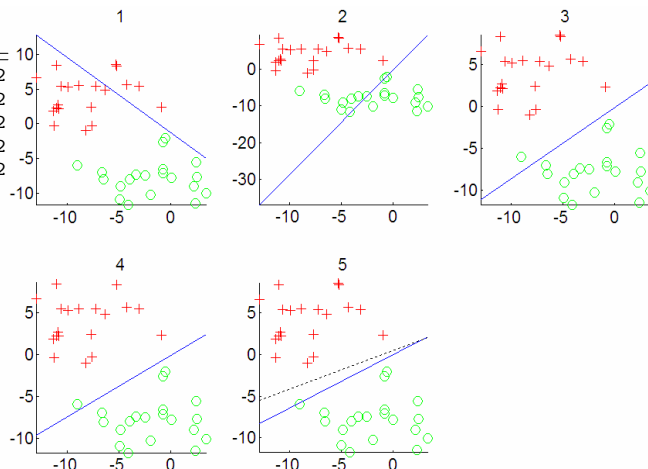
• Funciones de discriminación lineales: **algoritmo del perceptrón**

– Variantes del algoritmo básico:

(1) **Ejemplo:**

$\rho = 1.5$

$w(0) =$	+0.74	+0.67	+0.92
$w(1) =$	-218.29	+77.27	+23.42
$w(2) =$	-152.98	+179.38	+11.42
$w(3) =$	-139.43	+188.33	+9.92
$w(4) =$	-125.88	+197.29	+8.42



Alberto Ortiz / EPS (última revisión 25/06/2010)

58

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

– Variantes del algoritmo básico:

(1) Implementación:

```
function ww = perceptron_2D(data,rho,nit)

n = size(data,1)/2; % n° de elementos de cada clase
w0 = rand; w1 = rand; w2 = sqrt(1-w1^2); % inicialización aleatoria de w
w = [w1 w2 w0]';
x = [data(:,1:2)'; ones(1,2*n)]; % paso de x a x*

for t = 1:nit
    sumato = zeros(3,1); ic = 0; % inicialización de la iteración
    for k = 1:2*n % sumatorio
        xi = x(:,k);
        if w'*xi < 0 & data(k,3) == 1, sumato = sumato + rho*xi; ic = ic + 1;
        elseif w'*xi > 0 & data(k,3) == 2, sumato = sumato - rho*xi; ic = ic + 1;
        end
    end
    w = w + sumato; % actualización de w
    if ic == 0, break; end % incorrectamente clasificados = 0?
end
```

Alberto Ortiz / EPS (última revisión 25/06/2010)

59

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

– Variantes del algoritmo básico:

(2) Algoritmo del bolsillo

- Se detiene al cabo de T iteraciones y proporciona el mejor (hiper)plano que ha encontrado
- Resuelve parcialmente el problema de convergencia del algoritmo del perceptrón cuando las clases no son linealmente separables

(1) Inicializar $w(0)$ aleatoriamente

(2) $w_s \leftarrow w(0)$,

$h_s \leftarrow$ no. de patrones clasificados correctamente por $w(0)$

(3) **para** $t = 0$ **hasta** T

(3.1) $w(t+1) = w(t) - \rho_t \sum_{x_i \in \mathcal{Y}} \delta_{x_i} x_i$

(3.2) $h =$ no. de patrones clasificados correctamente por $w(t+1)$

(3.3) **si** $h > h_s$

entonces $w_s \leftarrow w(t+1)$, $h_s \leftarrow h$

fin para

Alberto Ortiz / EPS (última revisión 25/06/2010)

60

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

– Variantes del algoritmo básico:

(3) Construcción de Kesler

- Permite **tratar M > 2** clases
- Para cada muestra de la clase ω_i , x_{ik} , se definen M-1 vectores de dimensión $(L+1)M \times 1$, de forma que en la posición de bloque i se coloca x_{ik} (posición de ω_i) y en la posición de bloque j coloca $-x_{ik}$ (posición de ω_j)
- Los (M-1)N vectores resultantes son luego clasificados mediante el algoritmo del perceptrón forzando a que se verifique (\equiv el algoritmo se detiene cuando ...):

$$\tilde{w}^T \tilde{x}_k = (w_1^T, w_2^T, \dots, w_M^T) \tilde{x}_k > 0, \forall k$$

- El vector resultante contiene en la posición de bloque i el vector w_i para la clase ω_i , de forma que x es asignado a la clase ω_i si:

$$w_i^T x > w_j^T x, \forall j \neq i$$

$$y: g_{ij} = (w_i - w_j)^T x^*$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

61

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

– Variantes del algoritmo básico:

(3) Ejemplo: sean las muestras de un problema de clasificación en 3 clases:

$$\omega_1 : (1, 1)^T \rightarrow \begin{pmatrix} \overbrace{1, 1, 1}^{x[\omega_1]} \overbrace{-1, -1, -1}^{-x[\omega_2]} \overbrace{0, 0, 0}^0 \\ \overbrace{1, 1, 1}^{x[\omega_1]} \overbrace{0, 0, 0}^0 \overbrace{-1, -1, -1}^{-x[\omega_3]} \end{pmatrix}^T$$

$\omega_1 : \dots$

$$\omega_2 : (1, -2)^T \rightarrow \begin{pmatrix} \overbrace{-1, 2, -1}^{-x[\omega_1]} \overbrace{1, -2, 1}^{x[\omega_2]} \overbrace{0, 0, 0}^0 \\ \overbrace{0, 0, 0}^0 \overbrace{1, -2, 1}^{x[\omega_2]} \overbrace{-1, 2, -1}^{-x[\omega_3]} \end{pmatrix}^T$$

$\omega_2 : \dots$

$$\omega_3 : (-2, 1)^T \rightarrow \begin{pmatrix} \overbrace{2, -1, -1}^{-x[\omega_1]} \overbrace{0, 0, 0}^0 \overbrace{-2, 1, 1}^{x[\omega_3]} \\ \overbrace{0, 0, 0}^0 \overbrace{-2, 1, 1}^{-x[\omega_2]} \overbrace{-2, 1, 1}^{x[\omega_3]} \end{pmatrix}^T$$

$\omega_3 : \dots$

ω_1	ω_2	ω_3
$(1, 1)^T$	$(1, -1)^T$	$(-1, 1)^T$
$(2, 2)^T$	$(1, -2)^T$	$(0, 1)^T$
$(1, 2)^T$	$(0, -1)^T$	$(-2, 1)^T$
$(2, 1)^T$	$(0, -2)^T$	$(-1, 0)^T$

$$\tilde{w} = (\overbrace{w_{11}, w_{12}, w_{13}}^{w_1^T}, \overbrace{w_{21}, w_{22}, w_{23}}^{w_2^T}, \overbrace{w_{31}, w_{32}, w_{33}}^{w_3^T})^T$$

resolver exigiendo: $\tilde{w}^T \tilde{x} > 0, \forall \tilde{x}$

$$\rho_t = \rho = 0.5 \rightarrow \begin{aligned} w_1 &= (2.47, 1.66, 0.37)^T \\ w_2 &= (1.01, -1.36, -0.18)^T \\ w_3 &= (-2.07, 1.39, 2.23)^T \end{aligned}$$

Alberto Ortiz / EPS (última revisión 25/06/2010)

62

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

– Variantes del algoritmo básico:

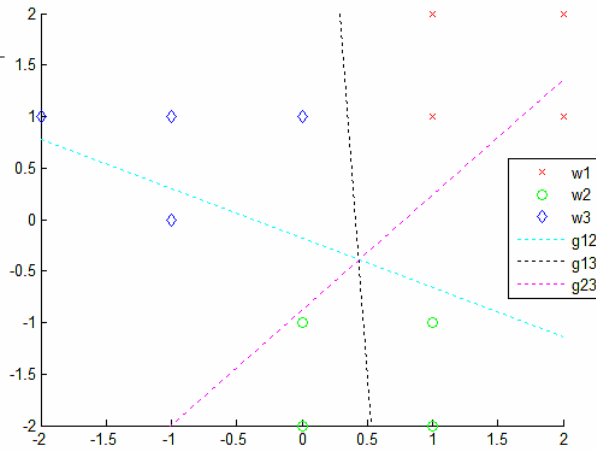
(3) Ejemplo:

ω_1	ω_2	ω_3
$(1, 1)^T$	$(1, -1)^T$	$(-1, 1)^T$
$(2, 2)^T$	$(1, -2)^T$	$(0, 1)^T$
$(1, 2)^T$	$(0, -1)^T$	$(-2, 1)^T$
$(2, 1)^T$	$(0, -2)^T$	$(-1, 0)^T$

$$\begin{aligned} w_1 &= (2.47, 1.66, 0.37)^T \\ w_2 &= (1.01, -1.36, -0.18)^T \\ w_3 &= (-2.07, 1.39, 2.23)^T \end{aligned}$$

↓

$$\begin{aligned} g_{12} &= 1.46x_1 + 3.02x_2 + 0.55 \\ g_{13} &= 4.54x_1 + 0.3x_2 - 1.86 \\ g_{23} &= 3.08x_1 - 2.75x_2 - 2.41 \end{aligned}$$



Alberto Ortiz / EPS (última revisión 25/06/2010)

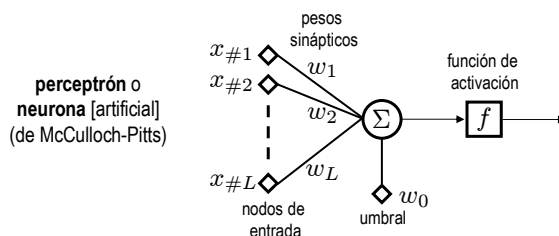
63

Funciones de discriminación lineales y el algoritmo del perceptrón

• Funciones de discriminación lineales: **algoritmo del perceptrón**

– Implementación de la operación de clasificación

- Una vez el algoritmo del perceptrón ha convergido hacia un cierto (hiper)plano caracterizado por $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_L, \mathbf{w}_0)$, la siguiente estructura permite implementar la operación de clasificación:



$$w^T x + w_0 > 0, x \rightarrow \omega_1$$

$$w^T x + w_0 < 0, x \rightarrow \omega_2$$

p.e. limitador estricto

$$f(u) = \begin{cases} -1 & u < 0 \\ +1 & u > 0 \end{cases}$$

- Constituye el ejemplo más simple de **máquina que aprende** (\equiv estructura cuyos parámetros libres son actualizados mediante un algoritmo de aprendizaje para aprender una cierta tarea en base a datos de entrenamiento)

Alberto Ortiz / EPS (última revisión 25/06/2010)

64