Reconocimiento de Patrones

Autores:
José Luis Alba - Universidad de Vigo
Jesús Cid - Universidad Carlos III de Madrid

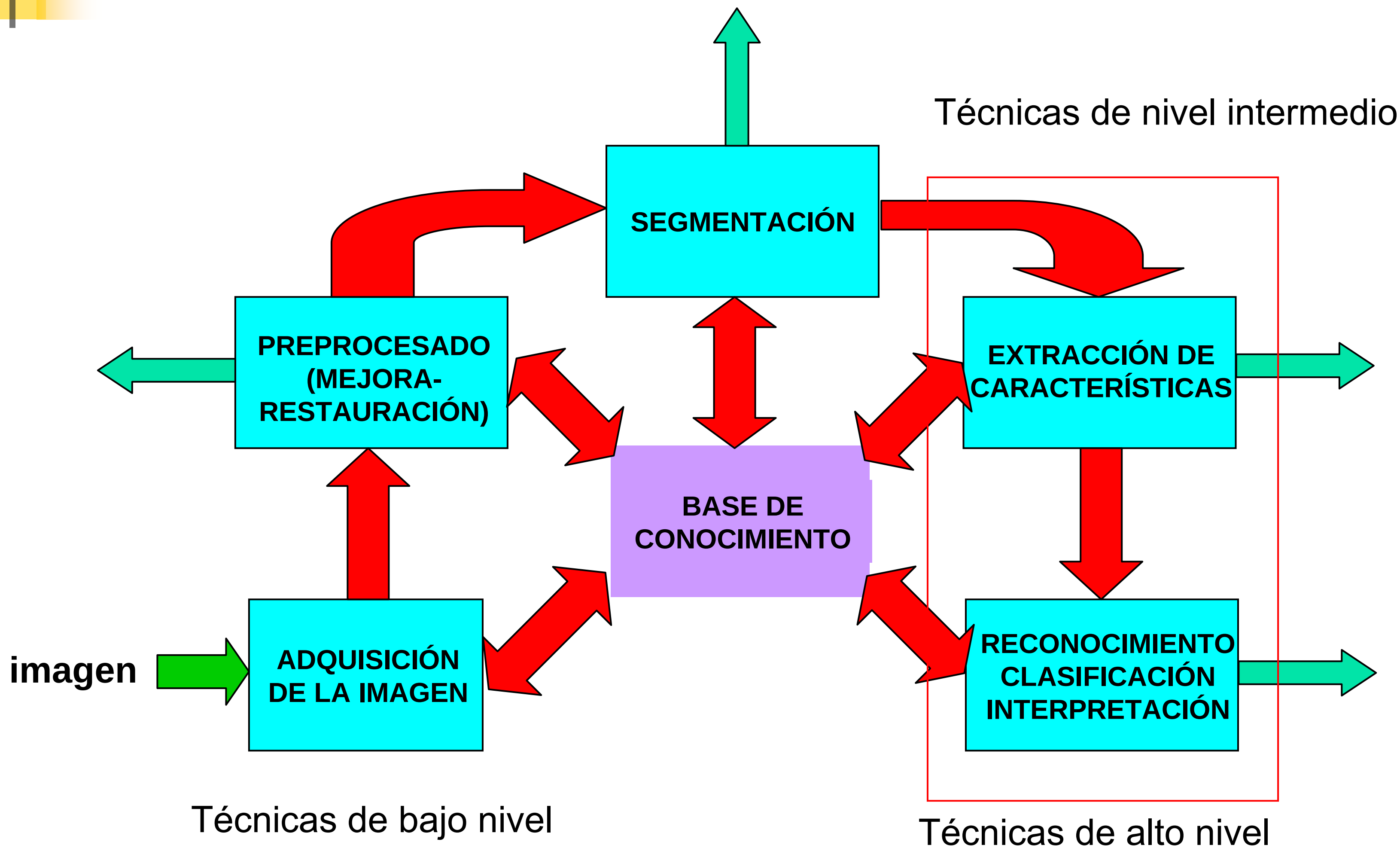
Ultima revisión: mayo de 2006

Índice

- Introducción
  - Esquema general del análisis de imágenes
  - Elementos del reconocimiento de patrones
- Patrones
  - Patrones vectoriales
  - Patrones estructurados
- Reconocimiento de patrones mediante funciones discriminantes
  - Elementos
  - Mínima distancia y Adaptación (Pattern Matching)
  - Clasificadores estadísticamente óptimos
  - Redes Neuronales Artificiales

Introducción

- Esquema general del análisis de imágenes

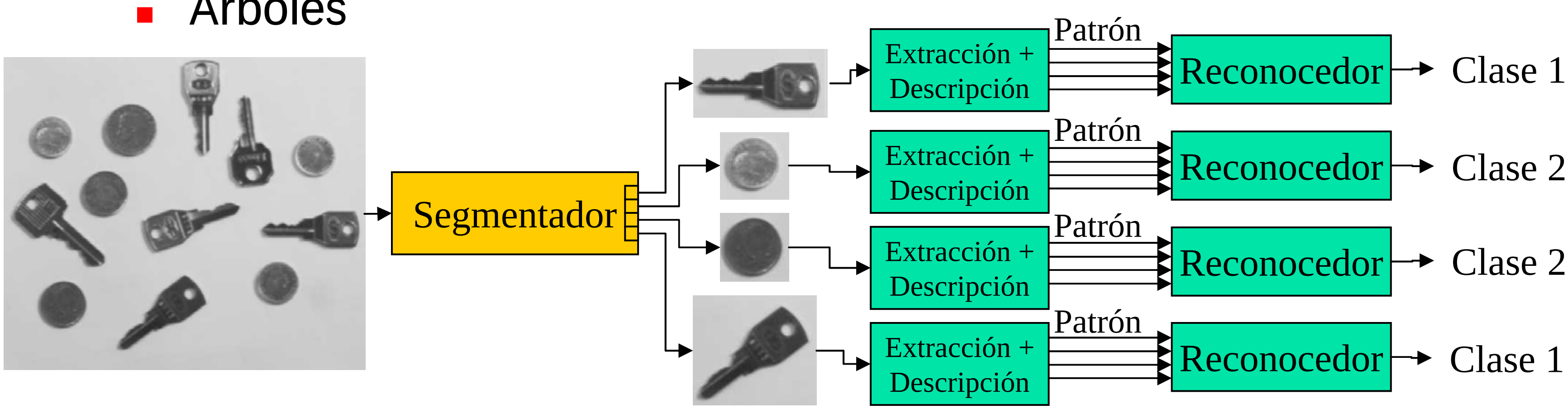


Elementos del reconocimiento de patrones

- **Patrones**
  - Tras los procesos de segmentación, extracción de características y descripción, cada objeto queda representado por una colección (posiblemente ordenada y estructurada) de descriptores, denominada **patrón**.
  - En los problemas de reconocimiento, cada patrón se supone perteneciente a una categoría o clase, c_i
  - El sistema de reconocimiento debe asignar cada objeto (de interés) a su categoría.
- **Reconocimiento o clasificación:** proceso por el que se asigna una "etiqueta", que representa una clase, a un patrón concreto.
- **Clase**: conjunto de entidades que comparten alguna característica que las diferencia de otras.
- **Clase de rechazo**: conjunto de entidades que no se pueden etiquetar como ninguna de las clases del problema
- **Extractor de características**: subsistema que extrae información relevante para la clasificación a partir de las entidades cuantificables
- **Clasificador**: subsistema que utiliza un vector de características de la entidad cuantificable y lo asigna a una de M clases
- **Evaluación del error de clasificación**: "error de clasificación", "tasa de error empírica", "tasa de rechazo empírica", "conjunto de datos independientes".
- **Falso rechazo (falso negativo) y falsa aceptación (faso positivo):** para problemas de 2 clases estas definiciones reflejan la importancia de una decisión contra la opuesta. El sistema de clasificación se puede "sintonizar" para que trabaje ponderando un tipo de error sobre el otro

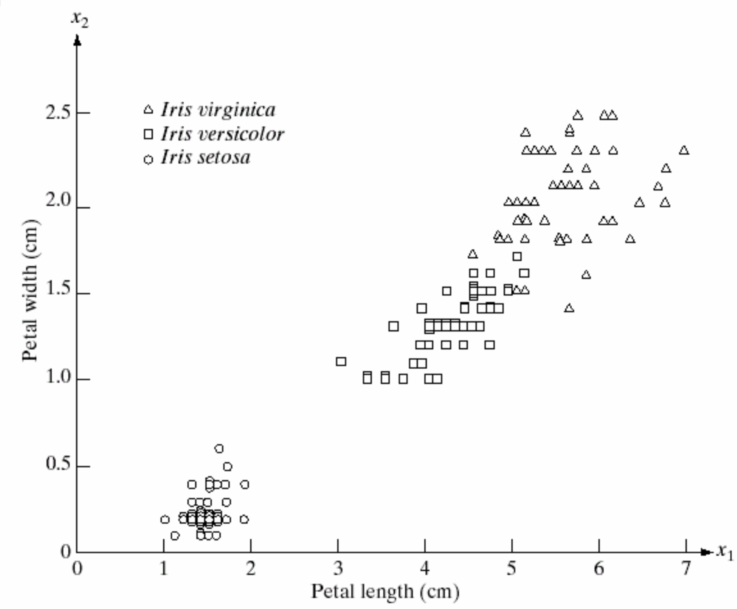
Patrones

- Para el reconocimiento automático, es importante que
  - Patrones que describen objetos de una misma clase, presenten características similares.
  - Patrones que describen objetos de diferentes clases presenten características diferenciadas.
- Tipos de patrones:
  - Vectores: $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$
  - Cadenas
  - Árboles



Patrones vectoriales

- Ejemplo: Clasificación de tipos de Iris (flores).
  - Tres categorías
  - Patrones bidimensionales
    - Longitud del pétalo
    - Anchura del pétalo
- Los descriptores utilizados sirven para discriminar *iris setosa* de las otras dos, pero no para discriminar entre *iris virginica* e *iris versicolor*



Iris Setosa

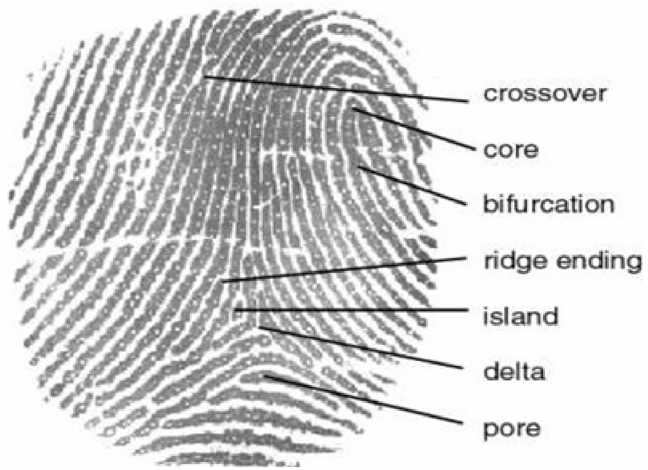
Iris Versicolor

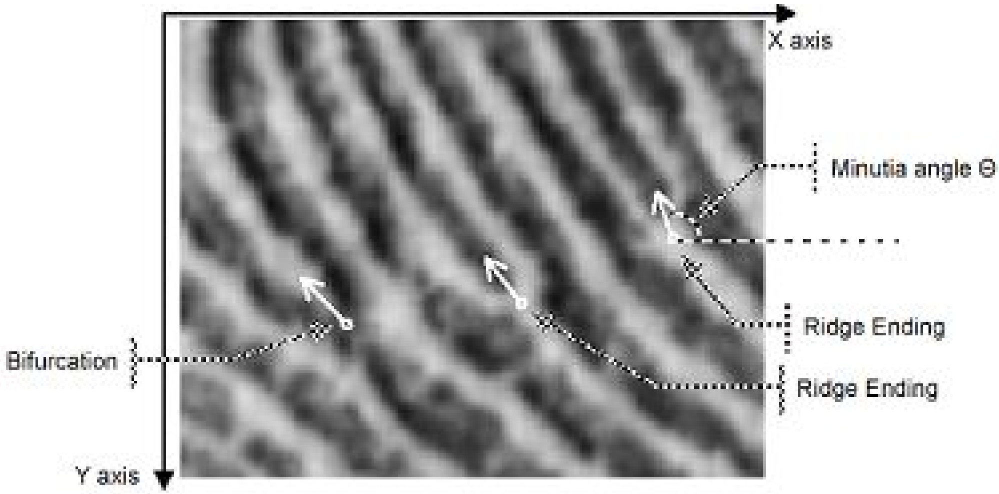
Iris Virginica

http://www.et.ethz.ch/eTutorials/evim/dateien/u3/irisbilder.htm

Patrones estructurados

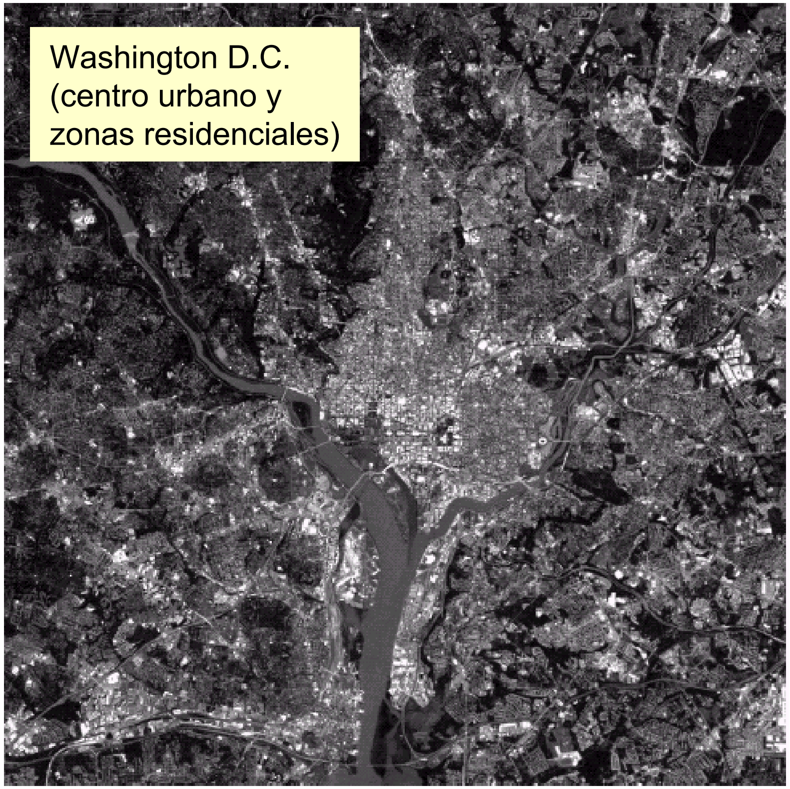
- Codifican relaciones (espaciales o de otro tipo) entre componentes del objeto o descriptores.
- Ejemplo:
  - Reconocimiento de huellas dactilares
    - Los algoritmos de reconocimiento suelen basarse en la detección de las *minucias* (minutiae), las cadenas (ridges) que forman, y su relación entre ellas

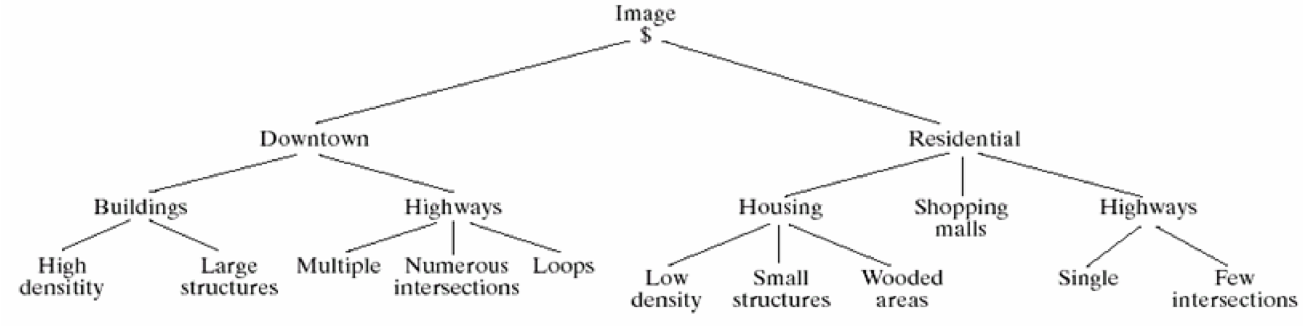




Imágenes tomadas de http://perso.wanadoo.fr/fingerchip/index.htm

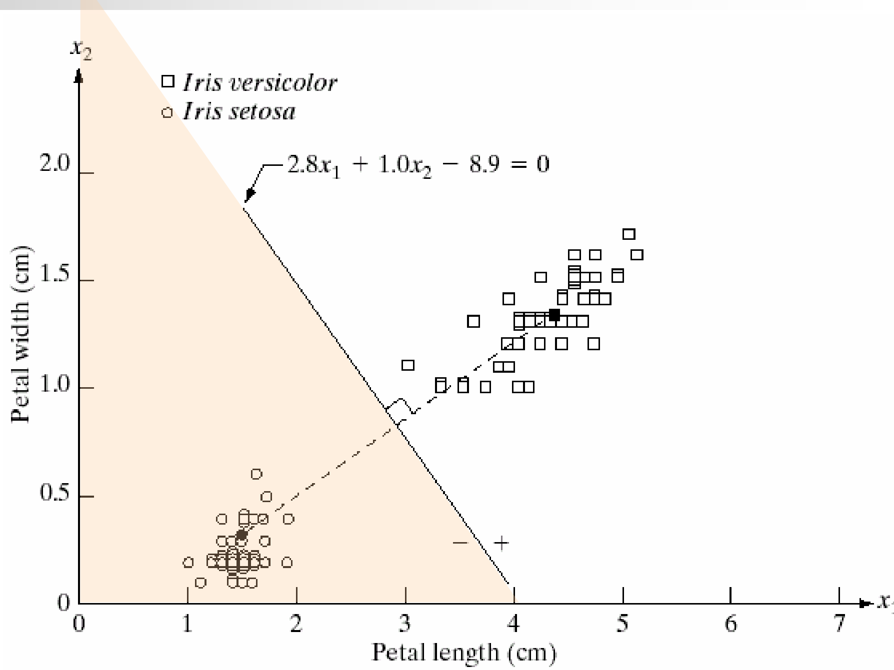
- Ejemplo:
  - Imagen de satélite
  - Descripción en árbol
    - Cada rama codifica una relación “compuesto de”

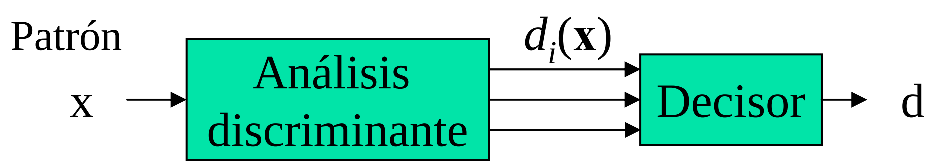




Reconocimiento mediante funciones discriminantes

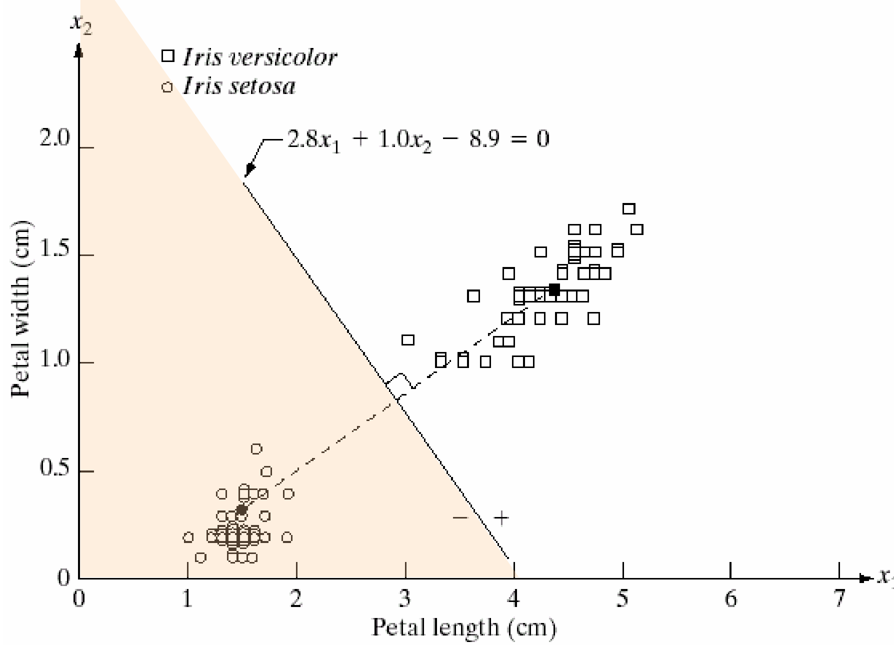
- Elementos:
  - Función discriminante, $d_i(\mathbf{x})$:
    - Mide la relevancia de la clase c_i para el patrón **x**.
  - Región de decisión, $\mathcal{R}_i$:
    - El conjunto de todos los puntos del espacio que el reconocedor asigna a la clase c_i
  - Frontera de decisión:
    - Separa regiones de decisión.
  - Decisor:
    - Típicamente (aunque no siempre), seleccionan la clase de mayor (o menor) valor de la función discriminante $d_i(\mathbf{x})$.

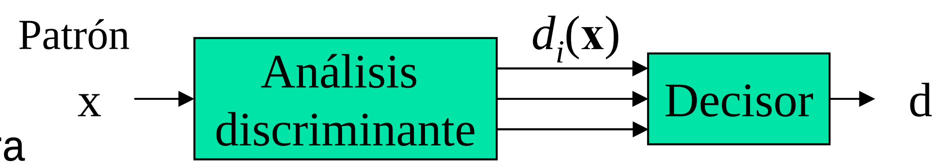




- Aprendizaje:
  - Conjunto de entrenamiento:
    - $\{\mathbf{x}^{(k)}, c^{(k)}, k=1,\ldots,K\}$
    - Conjunto de patrones *etiquetados* (cuya clase es conocida)
  - Algoritmo de entrenamiento:
    - Es un conjunto de reglas de ajuste de los parámetros de las funciones discriminantes

      $d_i(\mathbf{x},\mathbf{w}_i)$
  - Conjunto de prueba (test)
    - Conjunto de patrones etiquetados NO utilizados durante el entrenamiento.
    - Sirven para evaluar el rendimiento del clasificador.
    - Generalización: capacidad para clasificar correctamente patrones no utilizados durante el entrenamiento.





- Métodos (lista no exahustiva):
  - Adaptación (*Pattern Matching*)
    - Representan cada clase mediante un patrón prototipo.
      - Clasificador de mínima distancia
      - Adaptación por correlación
  - Clasificadores estadísticamente óptimos
    - Se fundamentan en la Teoría de la Decisión Estadística
      - Clasificador bayesiano para clases gausianas
  - Redes neuronales
    - Se fundamentan en la teoría del aprendizaje estadístico
      - Perceptrón para dos clases
      - Perceptrón multicapa

Clasificador de mínima distancia

- Se caracteriza por las funciones discriminantes $d_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_i\|^2$

  siendo $\mathbf{m}_i$ un patrón prototipo de la clase *i*.
- Asignan la muestra a la clase "más próxima" $d = \arg\min_i \{d_i(\mathbf{x})\}$
- Dado que $d_i(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_i)^T(\mathbf{x} - \mathbf{m}_i) = \mathbf{x}^T\mathbf{x} - 2\mathbf{m}_i^T\mathbf{x} + \mathbf{m}_i^T\mathbf{m}_i$

  se puede prescindir del primer término, que no depende de la clase, de modo que las funciones discriminantes

  $$d_i(\mathbf{x}) = -2\mathbf{m}_i^T\mathbf{x} + \mathbf{m}_i^T\mathbf{m}_i$$

  son equivalentes a las anteriores.
- Las fronteras de decisión del clasificador de mínima distancia son de la forma $d_i(\mathbf{x}) = d_j(\mathbf{x})$

  es decir, $-2(\mathbf{m}_j - \mathbf{m}_i)^T\mathbf{x} + \mathbf{m}_i^T\mathbf{m}_i - \mathbf{m}_j^T\mathbf{m}_j = 0$

  que son *hiperplanos*

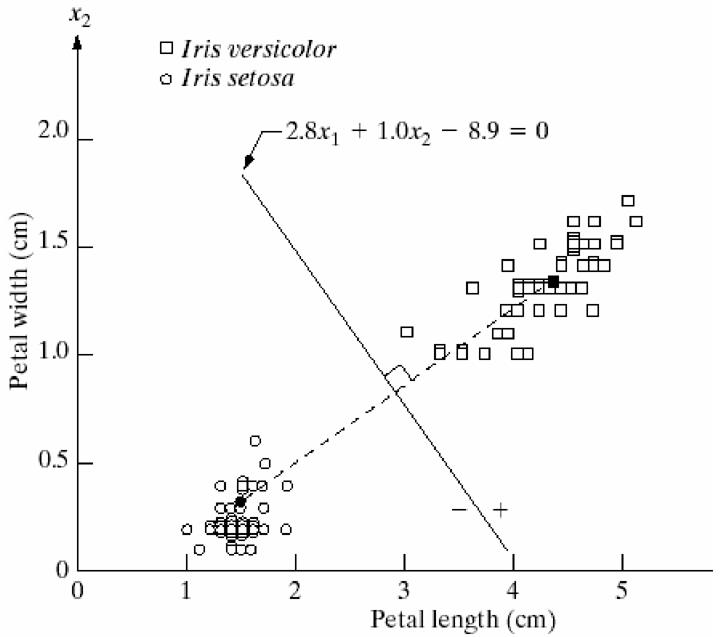
- Prototipo
  - El prototipo m_i que caracteriza a cada clase puede obtenerse mediante extracción de características sobre una imagen previamente etiquetada.
  - También puede obtenerse a partir de una colección de patrones etiquetados:

$$\mathbf{m}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\mathbf{x}_{i,j}$$

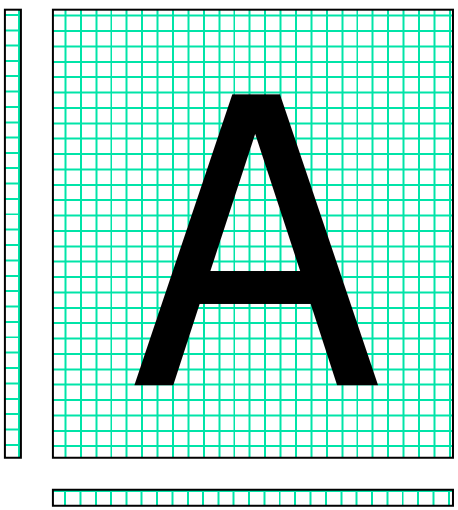
siendo $\{\mathbf{x}_{i,j}\}$ los patrones de la clase i del *conjunto de entrenamiento.*

Iris Setosa Iris Versicolor

http://www.et.ethz.ch/eTutorials/evim/dateien/u3/irisbilder.htm



- Ejemplo: Reconocimiento de caracteres.
  - Clases: C_k:{A,B,...,a,b,...0,1,2,...*a,b,...* a,b,...a,b,..a,b,...a,b,...}
  - Clase de rechazo: Cr:{!,”,$,%,&,/,(,),=,?,¿,~,×...}
  - Extractor de características:



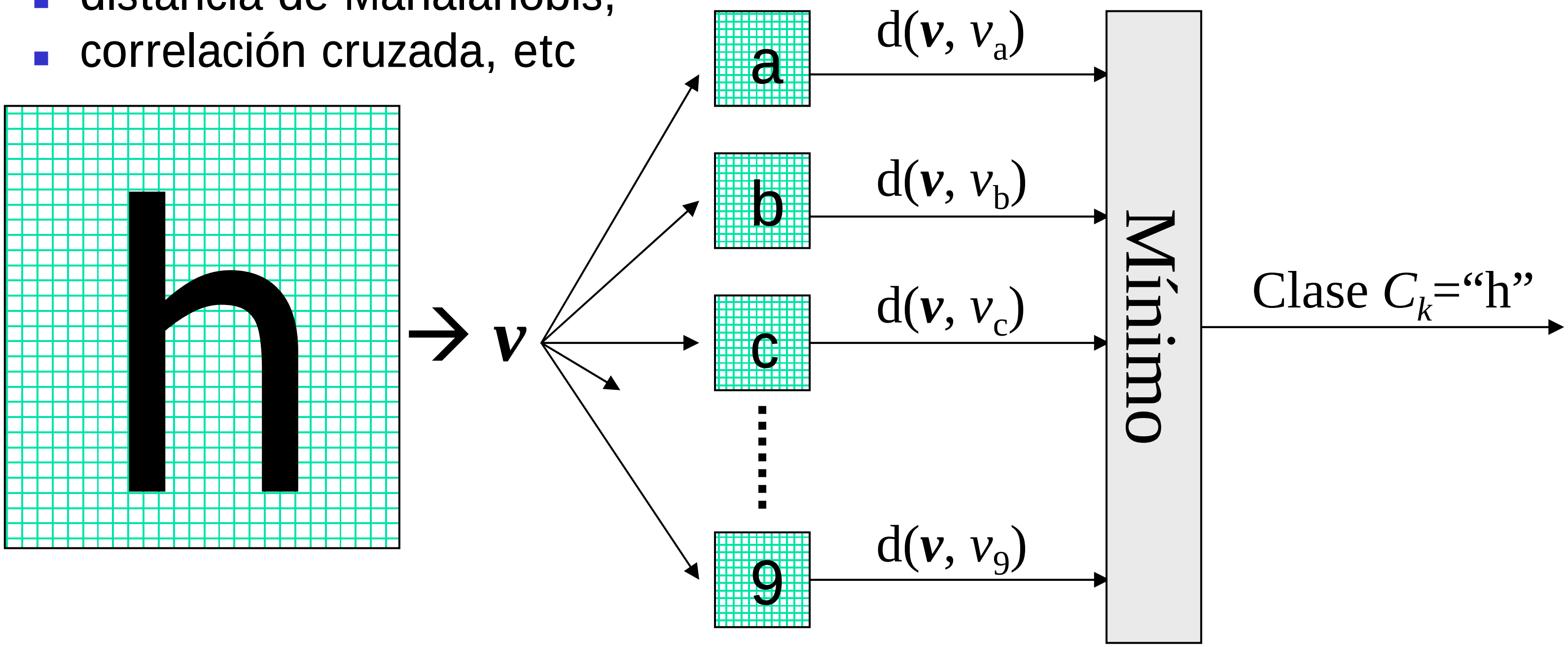
$$z \equiv (20\times 18) \text{ ó } (360\times 1) \text{ imagen o vector binario}$$

$$\begin{cases} P_x = \sum_{n=1}^{20} z(n,:) \text{ (proyección eje x} \rightarrow \text{suma de las filas)} \\ P_y = \sum_{n=1}^{18} z(:,n) \text{ (proyección eje y} \rightarrow \text{suma de las columnas)} \end{cases}$$

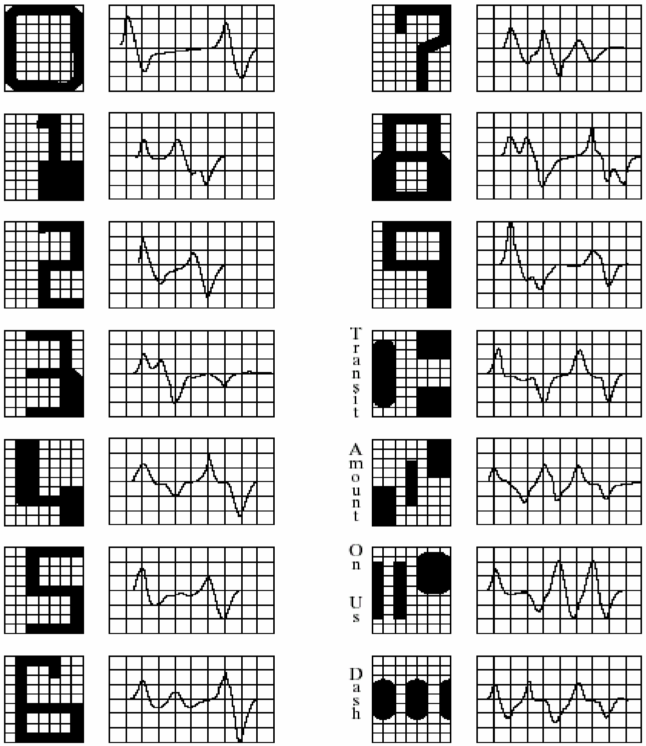
$$\begin{cases} S_x = \sum_{n=2}^{20} | z(n,:) - z(\text{n -1},:) | \text{ (suma de saltos eje x)} \\ S_y = \sum_{n=2}^{18} | z(:,n) - z(n-1,:) | \text{ (suma de saltos eje y)} \end{cases}$$

$$\left.\begin{cases} \text{perímetro} \\ \text{n}^{\text{o}} \text{ de huecos (contornos cerrados)} \\ \text{centro de masas} \end{cases}\right\} \equiv v \text{ (vector de dimensión 3)}$$

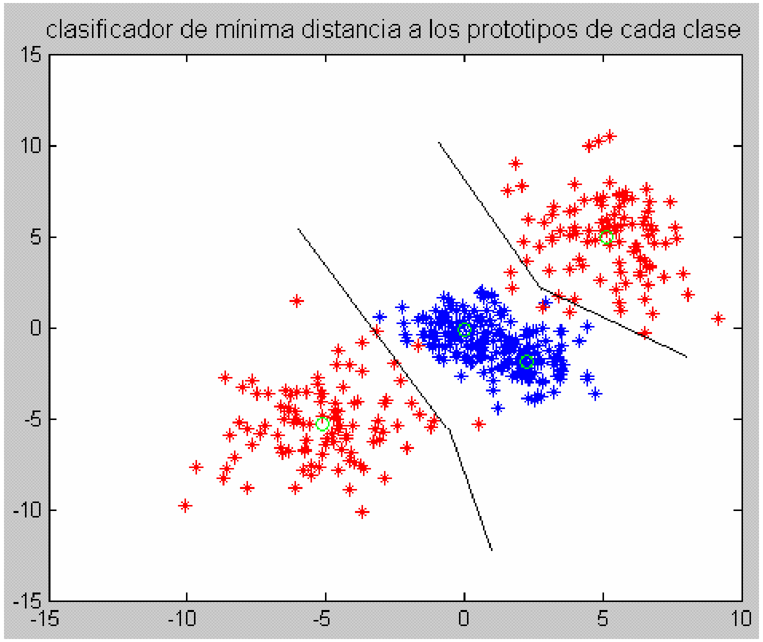
- El reconocedor de caracteres asigna el vector "$\boldsymbol{v}$" a la clase que más se "parece".
- Medidas de similitud:
  - distancia euclidea
  - distancia de Hamming,
  - distancia de Mahalanobis,
  - correlación cruzada, etc



- Ventajas del clasificador de mínima distancia
  - Simplicidad
  - Facilidad de ajuste
- Inconvenientes:
  - Sólo funciona cuando las clases forman nubes poco dispersas y bien separadas
    - Ejemplo: fuente de caracteres E-13B de la *American Bankers Association*, especialmente diseñada para facilitar el reconocimiento automático.
    - La signatura refleja la *derivada* de la cantidad de negro en dirección vertical, al mover un *scanner* de izquierda a derecha
    - El muestreo en los puntos de la cuadrícula proporciona información con suficiente capacidad discriminante



- Extensiones del clasificador de mínima distancia:
  - Permiten obtener fronteras de clasificación más complejas, y modelar categorías que no quedan adecuadamente representadas por su media.
  - *k*-NN (*Nearest Neighbour*):
    - Cada clase, C_j, se caracteriza por una colección de prototipos, $\mathbf{m}_{ij}$
    - Cada patrón, **x**, se asigna a la clase mayoritaria de los *k* prototipos más próximos
    - Selección de prototipos:
      - Cada muestra de entrenamiento, un prototipo.
      - Prototipos obtenidos mediante un algoritmo de agrupamiento (como el k-means –ver presentación de Análisis de imágenes-).



1-NN, 2 prototipos por clase

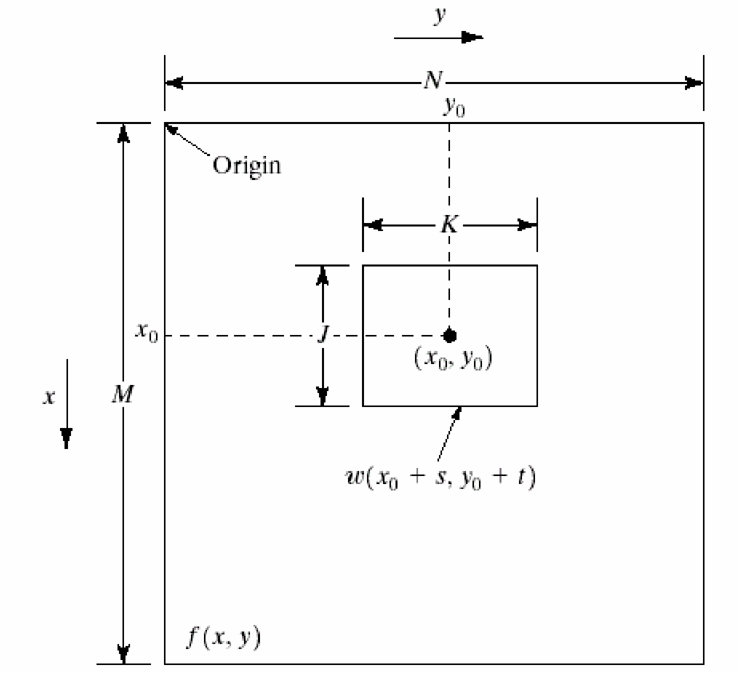
Adaptación por correlación

- Los métodos de adaptación por correlación también se basan en la comparación de la imagen a clasificar con una o varias imágenes patrón que caracterizan a cada clase.
- Es equivalente al de mínima distancia si los patrones están normalizados:

$$d_i(\mathbf{x}) = -2\mathbf{m}_i^T\mathbf{x} + \mathbf{m}_i^T\mathbf{m}_i \Rightarrow d_i'(\mathbf{x}) = \mathbf{m}_i^T\mathbf{x}$$

- Utilizan medidas de similitud basadas en correlaciones:
  - Correlación:

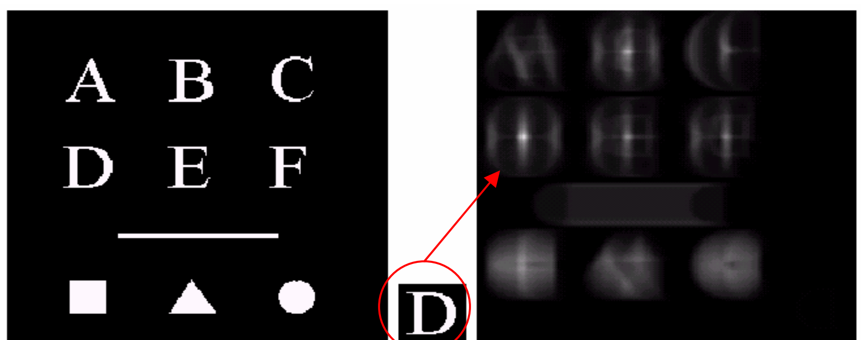
$$c(x,y) = \sum_s \sum_t f(s,t)\,w(x+s, y+t)$$



- Coeficiente de correlación:
  - Proporciona una medida invariante frente a cambios en la amplitud.

$$c(x,y)=\frac{\sum_s\sum_t\left(f(s,t)-\bar{f}(s,t)\right)\left(w(x+s,y+t)-\overline{w}\right)}{\left[\sum_s\sum_t\left(f(s,t)-\bar{f}(s,t)\right)^2\right]^{1/2}\left[\sum_s\sum_t\left(w(x+s,y+t)-\overline{w}\right)^2\right]^{1/2}}$$

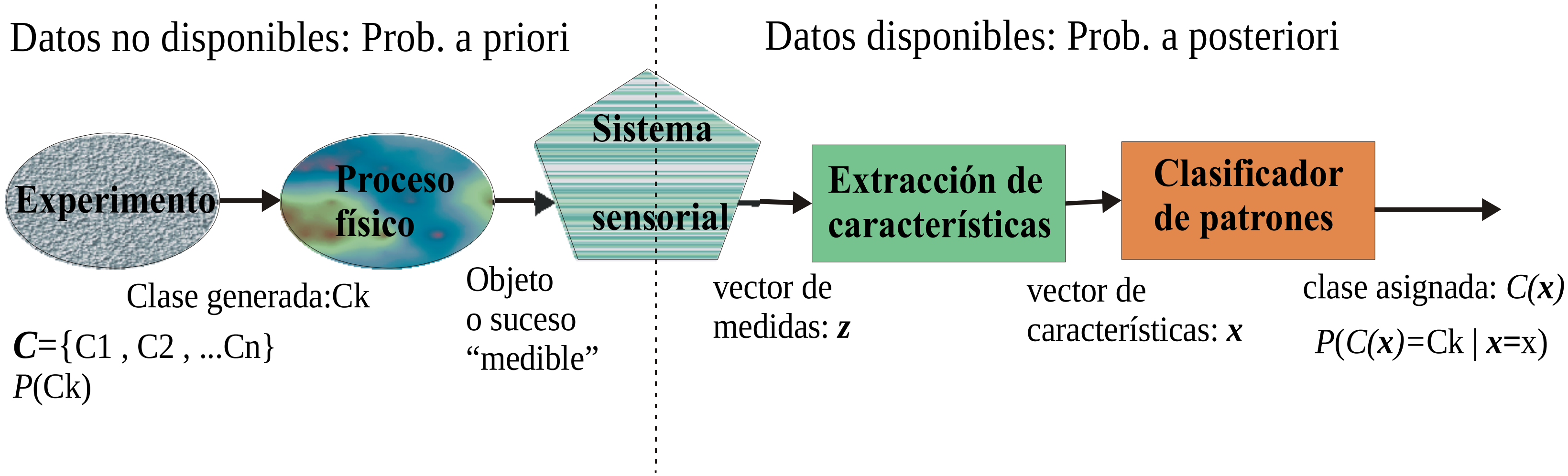
  donde los promedios de *w* se calculan sobre todo el patrón (habitualmente más pequeño que la imagen), mientras que los de *f* se calculan sobre la región en torno al píxel del tamaño del patrón.



  - Otras normalizaciones (para obtener invarianzas respecto a cambios de escala u orientación) son posibles, pero a mayor complejidad

Clasificadores estadísticamente óptimos

- Introducción:
  - El mecanismo de generación de patrones se puede representar de forma probabilística:



Clasificadores estadísticamente óptimos

- Caracterización estadística
  - La teoría de la decisión parte del supuesto de que los patrones son realizaciones independientes de un modelo probabilístico $p(c,\mathbf{x})=P(c\,|\mathbf{x})\;p(\mathbf{x})$
    - En general, en los problemas de reconocimiento de patrones tenemos clases mutuamente exclusivas y colectivamente exhaustivas, es decir:

$$\sum_{k=1}^{C} P(c_k) = 1$$

siendo C el número de clases.
  - Elementos:
    - Los **costes**, L_{ij}: coste de asignar a c_j un patrón de la clase c_i.
    - El **riesgo medio condicional** de asignar un patrón a clase j:

$$r_j(\mathbf{x}) = \sum_{k=1}^{C} L_{kj} P(c_k \mid \mathbf{x})$$

- <u>Clasificador de mínimo riesgo</u>
  - Será aquél basado en las reglas de decisión

$$i = \arg\min_j r_j(\mathbf{x}) = \arg\min_j \sum_{k=1}^{C} L_{kj} P(c_k \mid \mathbf{x})$$

aplicando la regla de Bayes

$$P(c_k \mid \mathbf{x}) = \frac{P(c_k)\, p_{x|c_k}(\mathbf{x} \mid c_k)}{p_x(\mathbf{x})}$$

se obtiene la expresión alternativa

$$i = \arg\min_j \sum_{k=1}^{C} L_{kj}\, p(\mathbf{x} \mid c_k) P(c_k)$$

- <u>Clasificador de mínima probabilidad de error</u>
  - Se obtiene dando idéntico valor a todos los errores: $L_{ij} = 1-\delta_{ij}$ ➔

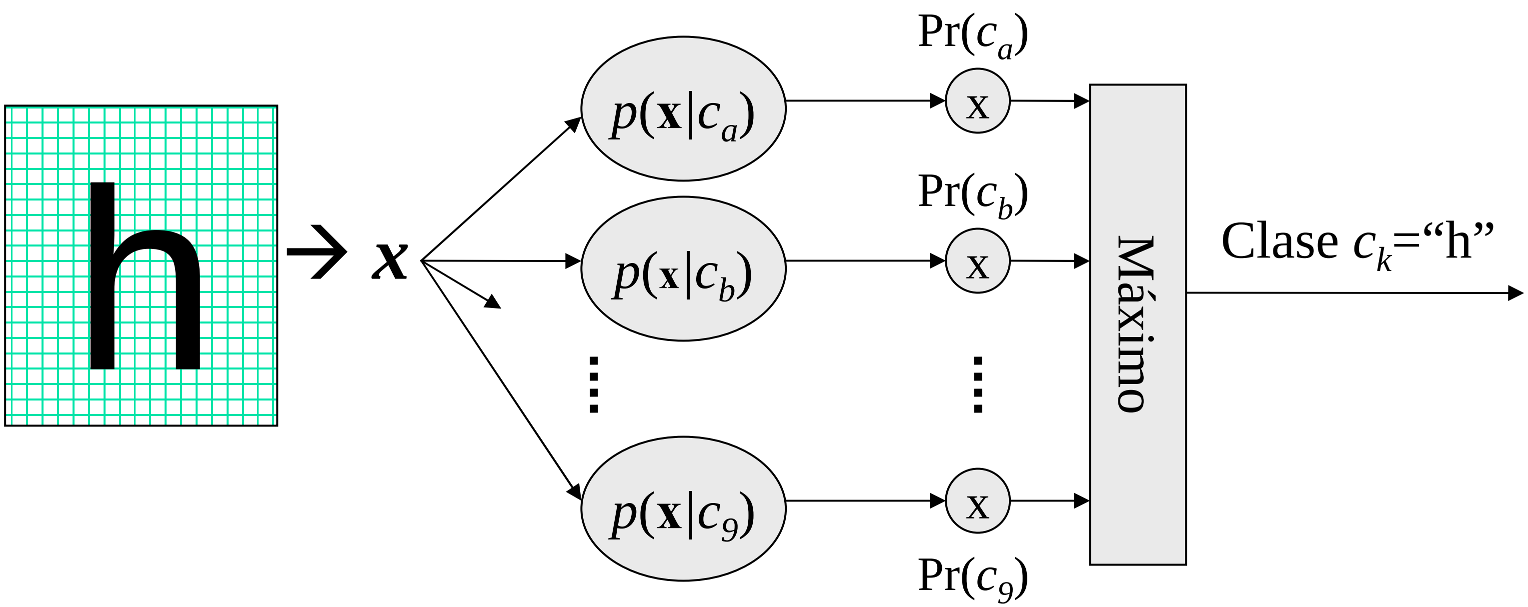
➔ $$r_j(\mathbf{x}) = \sum_{k=1}^{C} P(c_k \mid \mathbf{x}) - \sum_{k=1}^{C} \delta_{kj} P(c_k \mid \mathbf{x}) = 1 - P(c_j \mid \mathbf{x})$$

(que es la probabilidad de error)

➔ $$i = \arg\min_j \{1 - P(c_j \mid \mathbf{x})\} = \arg\max_j \{P(c_j \mid \mathbf{x})\}$$

(decisión MAP (*máximo a posteriori*), que selecciona la categoría más probable, dada la observación)

- Ejemplo: Reconocimiento de caracteres mediante un decisor MAP



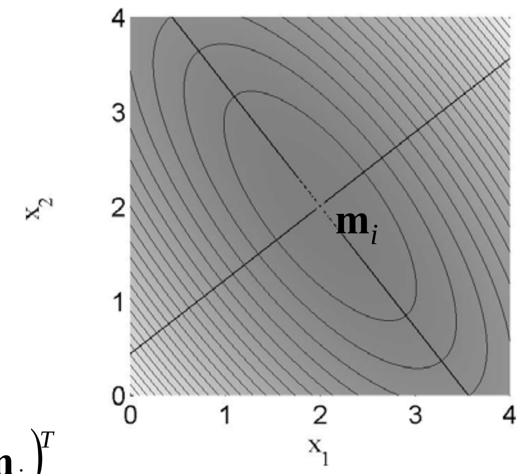
Diseño del clasificador Bayesiano

- En la mayoría de las aplicaciones prácticas, no se conocen $P(c_k)$, $p(\mathbf{x}|c_k)$ ni $P(c_k|\mathbf{x})$.
- En tal caso, deben estimarse a partir de un conjunto de entrenamiento.
- Se presentan dos alternativas:
  - Estimar $P(c_k|\mathbf{x})$, o bien
  - Estimar $P(c_k)$ y $p(\mathbf{x}|c_k)$
    - $P(c_k)$: puede suponerse constante (categorías equiprobables) o estimarse como la proporción de observaciones etiquetadas y no etiquetadas.
    - $p(\mathbf{x}|c_k)$: hay muchos métodos. Los más sencillos suponen que las distribuciones son gausianas. La estimación se reduce a los parámetros de media y covarianzas.

Clasificador bayesiano para clases gausianas

- Se caracteriza por

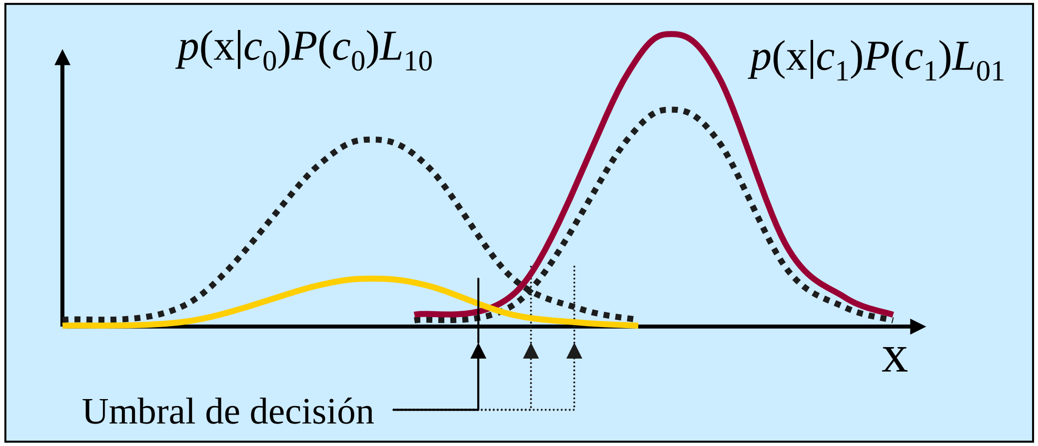
$$p(\mathbf{x} \mid c_\mathrm{i}) = \frac{1}{(2\pi)^{\mathrm{N}/2} \left|\mathbf{V}_i\right|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T \mathbf{V}_i^{-1} (\mathbf{x}-\mathbf{m}_i)\right]$$



siendo

$$\mathbf{m}_i = E\{\mathbf{x} \mid c_\mathrm{i}\} \approx \frac{1}{N_i} \sum_{d^{(k)}=c_i} \mathbf{x}^{(k)}$$

$$\mathbf{V}_i = E\{(\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T \mid c_\mathrm{i}\} \approx \frac{1}{N_i} \sum_{d^{(k)}=c_i} (\mathbf{x}^{(k)}-\mathbf{m}_i)(\mathbf{x}^{(k)}-\mathbf{m}_i)^T$$



- Función discriminante: suele utilizarse el logaritmo:

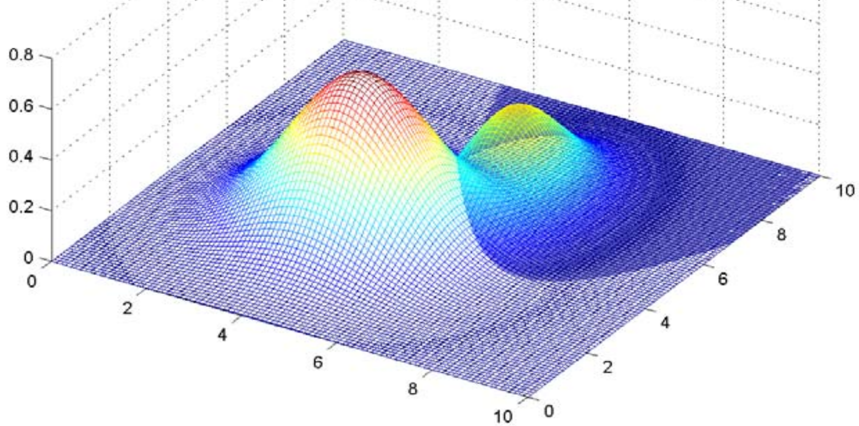
$$d_i(\mathbf{x}) = \ln\left(p(\mathbf{x}\,|\,c_i)P(c_i)\right) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{V}_i|^{1/2} - \frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T\mathbf{V}_i^{-1}(\mathbf{x}-\mathbf{m}_i) + \ln\left(P(c_i)\right)$$

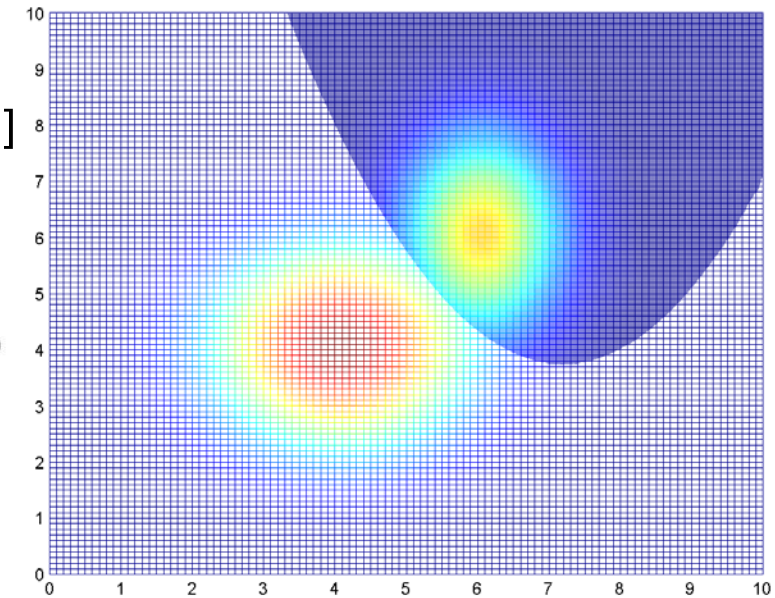
$$\equiv -\frac{1}{2}\ln|\mathbf{V}_i|^{1/2} - \frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T\mathbf{V}_i^{-1}(\mathbf{x}-\mathbf{m}_i) + \ln\left(P(c_i)\right)$$

→ Es una forma cuadrática para cada clase, con lo que la frontera de decisión será una cuadrática también.

Ej: $P(c_1)=0.6$; $P(c_2)=0.4$;

$\mathbf{V}_1$ =[0.77 0;0 0.77]; $\mathbf{V}_2$ =[0.77 0;0 1.26]





- Cuando las matrices de covarianzas de todas las clases coinciden (**V**$_i$=**V**) , el discriminante se reduce a:

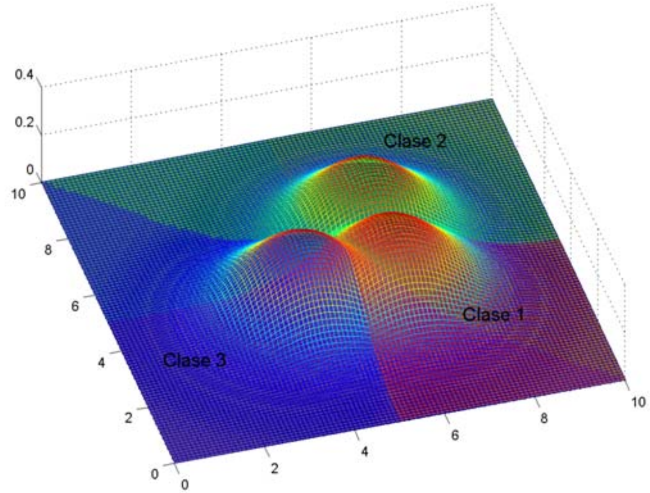
$$d_i(\mathbf{x}) \equiv \mathbf{m}_i^T \mathbf{V}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{m}_i^T \mathbf{V}^{-1}\mathbf{m}_i + \ln(P(c_i))$$

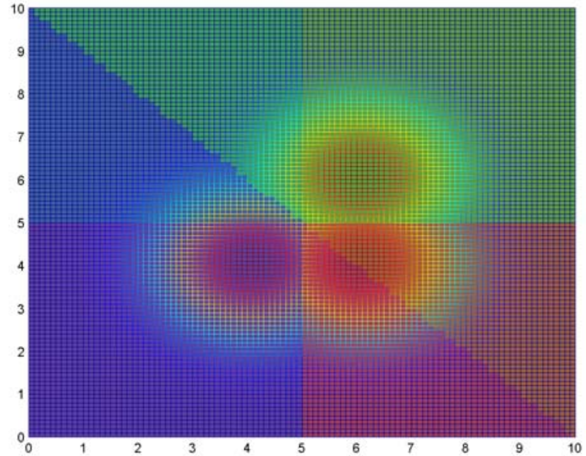
  y, por tanto, es una función discriminante lineal

- Si, finalmente, las matrices son proporcionales a la identidad, resulta

$$d_i(\mathbf{x}) \equiv \mathbf{m}_i^T \mathbf{x} - \frac{1}{2}\mathbf{m}_i^T \mathbf{m}_i$$

  que coincide con la función discriminante del clasificador de mínima distancia.





Ejemplo: clasificación de terrenos en imágenes multiespectrales.

- Las Imágenes multiespectrales codifican información en varias bandas, sensibles a diferentes longitudes de onda del espectro electromagnético.

- Los clasificadores bayesianos se han mostrado efectivos para clasificación de terrenos en muchos tipos de imágenes

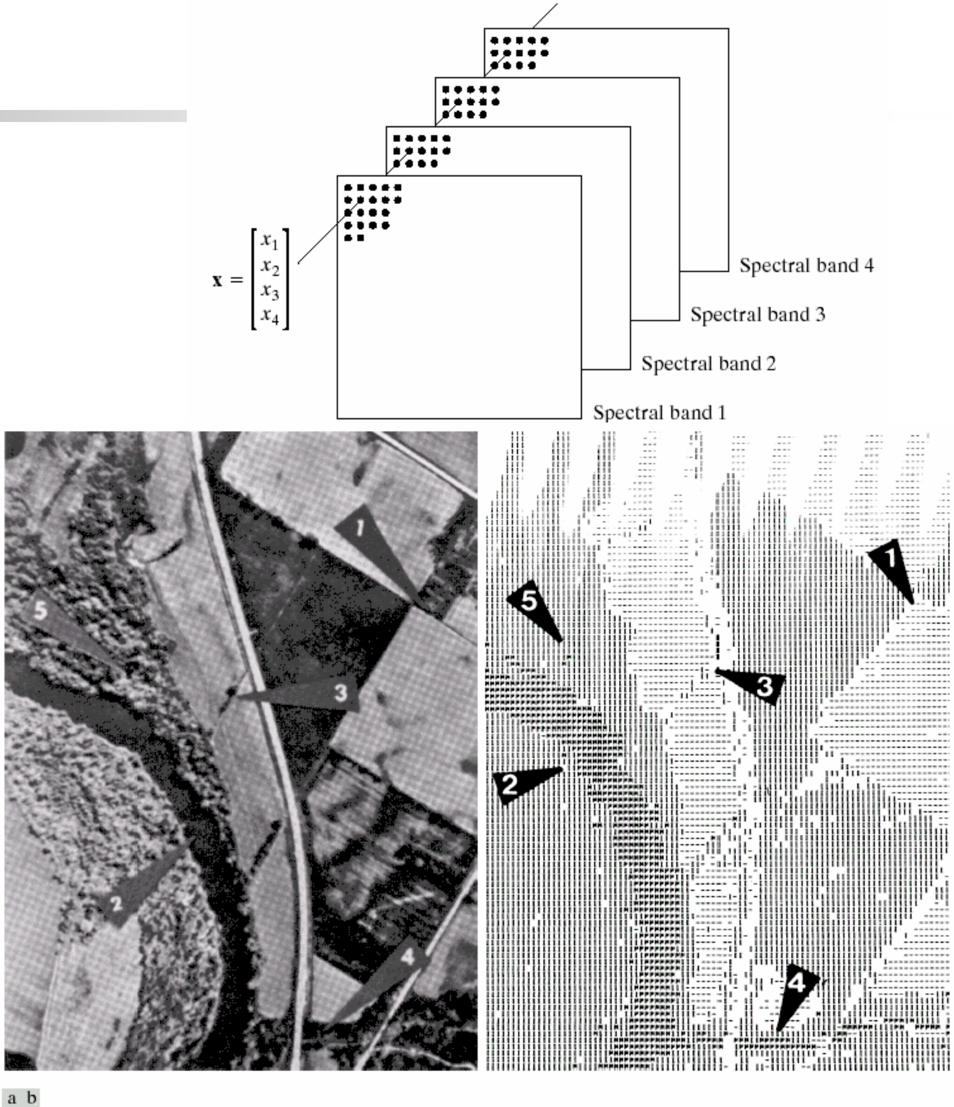
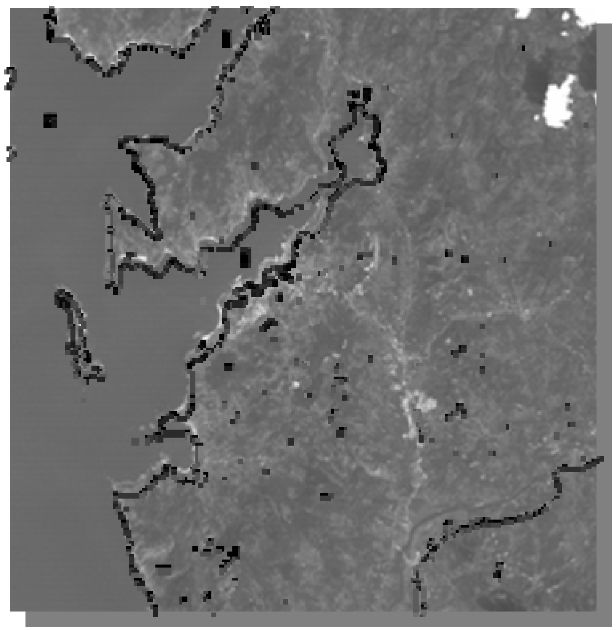
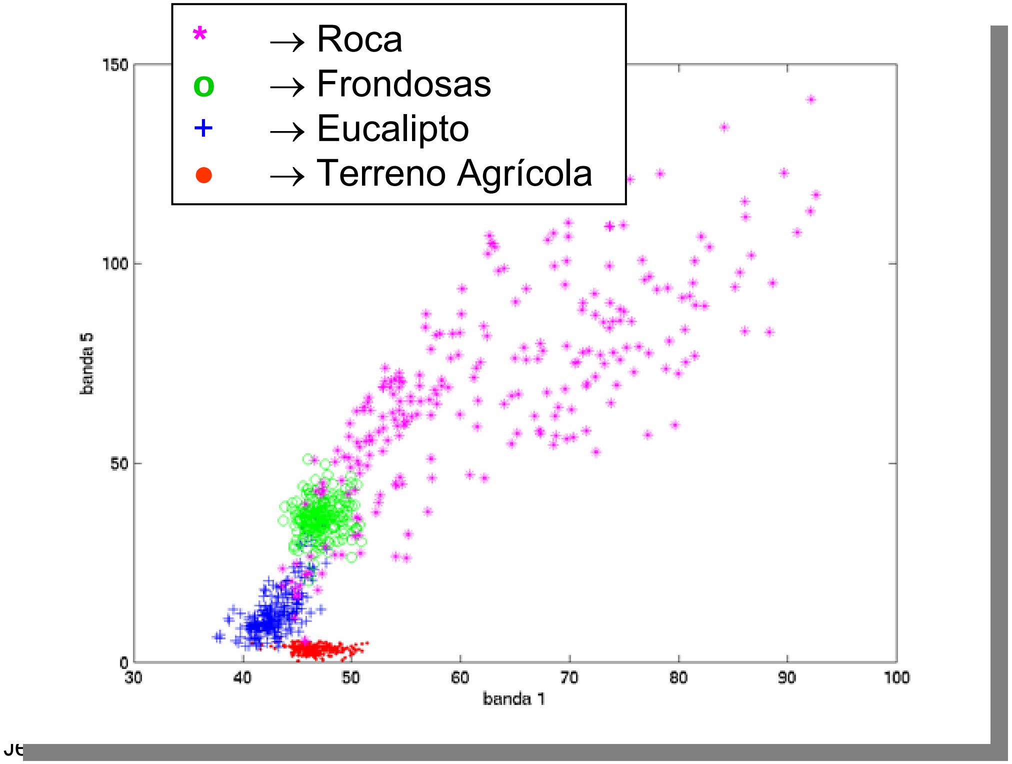


FIGURE 12.13 (a) Multispectral image. (b) Printout of machine classification results using a Bayes classifier. (Courtesy of the Laboratory for Applications of Remote Sensing, Purdue University.)

Inconvenientes del modelo gausiano:

- Con frecuencia, no es realista (los datos no tienen una distribución gausiana)
  - Ejemplo: Diagrama de dispersión de las bandas 1 (azul) y 5 (infrarrojo cercano) de un conjunto de datos pertenecientes a una imagen *Landsat* (7 bandas) de las Ría de Vigo.





Clasificador bayesiano para mezclas de gausianas

- Modelo más general: $\boldsymbol{x} \in C_k$ se asume con fdp compuesta por una mezcla de gausianas de medias μ_{kl} , covarianza $\mathbf{V}_{kl}$ y factores de mezcla $\mathbf{w}_l$ es decir:

$$p(\mathbf{x}\,|\,c_k) = \sum_{l=1}^{L} w_{kl}\, p(\mathbf{x}\,|\,l, c_k) = \sum_{l=1}^{L} w_{kl} \frac{1}{(2\pi)^{p/2}\,|\,\mathbf{V}_{kl}\,|^{1/2}} \exp\left[-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{kl})^T \mathbf{V}_{kl}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{kl})\right]$$

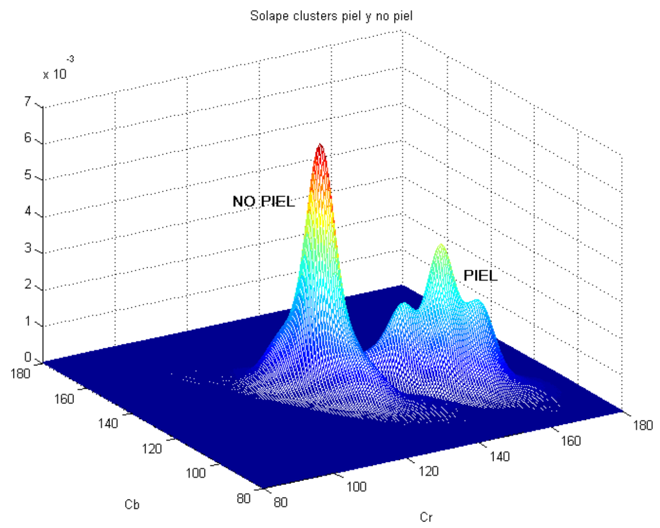
- La estimación ML de los parámetros $\mathbf{w}_l$, μ_{kl} y Σ_{kl} para cada clase se realiza de forma iterativa mediante un algoritmo conocido como Esperanza-Maximización (EM).
- El clasificador MAP responde a la expresión:

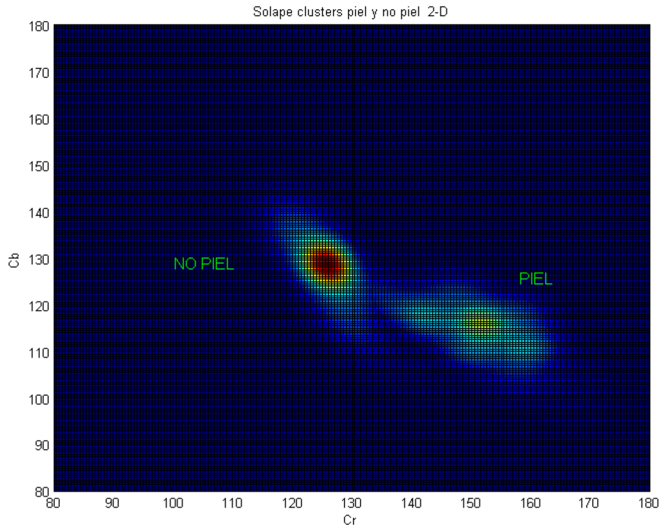
$$C(x) = \arg\max_{k} \{P(C_k)\, p(x\,|\,C_k)\}$$

- Las fronteras de decisión son superficies hipercuádricas a tramos

Ejemplo:

- Clasificación de piel/no piel en un espacio de color bi-dimensional (sin información de luminancia), modelando cada clase como una mezcla de gausianas y estimando parámetros con el algoritmo EM:





Redes Neuronales Artificiales

- Background
- Perceptrón para dos clases
  - Algoritmos de entrenamiento
    - Clases separables linealmente
    - Clases no separables linealmente

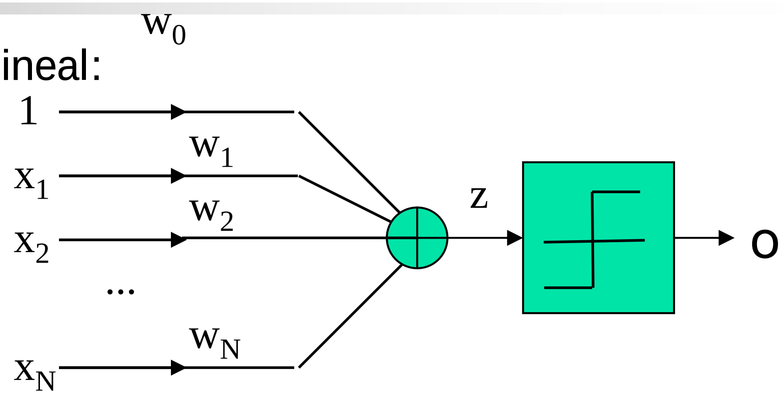
Motivación:

- El análisis previo pone de manifiesto que, para diseñar un clasificador estadísticamente óptimo a partir de un conjunto de datos de entrenamiento, pueden adoptarse tres estrategias:
  - Estimar $P(c_k|\mathbf{x})$
  - Estimar $P(\mathbf{x}|c_k)$ y $P(c_k)$
  - Determinar una función discriminante que proporcione la frontera de decisión óptima.
- El término "redes neuronales" engloba a un conjunto de técnicas que proporcionan soluciones flexibles (adaptables a cada problema) para cada una de estas estrategias.

Perceptron para dos clases

- Implementa la función discriminante lineal:

$$z = d(x) = w_0 + \sum_{i=1}^{n} w_i x_i = \mathbf{w}_e^T \mathbf{x}_e$$



- Para un conjunto finito de muestras podemos hallar un vector de pesos apropiado buscando una solución de estas inecuaciones (para dos clases):

$$+\mathbf{w}_e{}^t \mathbf{x}_{1j} > 0, \quad 1 \le j \le M_1$$

$$-\mathbf{w}_e{}^t \mathbf{x}_{2j} > 0, \quad 1 \le j \le M_2$$

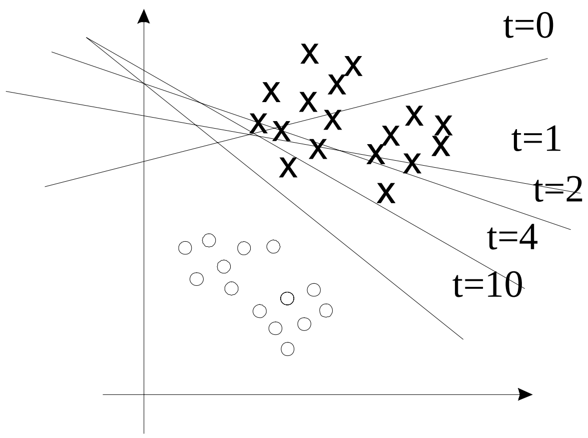
- ¿Cómo determinar los pesos?
  → Algoritmo de entrenamiento: procedimiento recursivo que modifica el vector de pesos hasta que se cumplen las desigualdades:
  - Iniciar aleatoriamente los pesos
  - Presentar cada muestra al sistema de inecuaciones
    - Si bien clasificada no alterar pesos
    - Si mal clasificada alterar los pesos de forma conveniente

Algoritmos de entrenamiento

- Clases separables linealmente
  - **Regla del Perceptrón**:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + \frac{\alpha}{2}\left[d^{(k)} - o^{(k)}\right]\mathbf{x}^{(k)} \quad (0 < \alpha < 1); \qquad con \quad d^{(k)}, o^{(k)} = \{-1;1\}$$

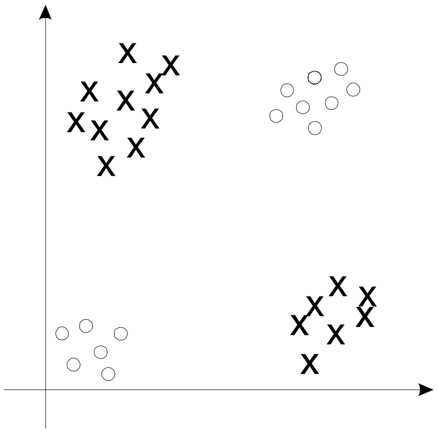
    - Si no hay error, no hay cambio
    - Si hay error, $\mathbf{w}^{(k+1)\mathrm{T}}\,\mathbf{x}^{(k)}$ cambia en el sentido de corregir el error.
  - Teorema de entrenamiento del perceptrón:
    - Si las clases son linealmente separables, la regla del perceptrón converge a una solución de cero errores en un número finito de pasos.



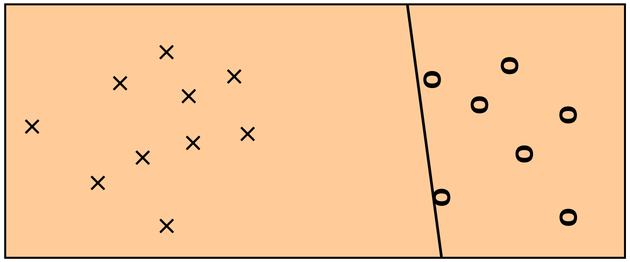
- Clases no separables linealmente
  - En este caso, la regla del perceptrón no se detiene.
  - Soluciones:
    - **Discriminantes no lineales**, pero lineales en los parámetros:

$$z = d(x) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} w_i x_i^2$$

      permiten obtener fronteras no lineales que obtengan soluciones de cero errores → **perceptrón multicapa**
    - **Variantes heurísticas**, como el *algoritmo del Bolsillo*, que conserva en una memoria aparte (*bolsillo*) los pesos que han dado lugar a la secuencia de pasos de entrenamiento libre de errores más larga



- Inconvenientes de la Regla del Perceptrón:
  - Mala generalización:
    - aunque el problema sea intrínsecamente separable (para el clasificador usado), el algoritmo puede colocar la frontera en una posición inadecuada



- El ADALINE ("*ADAptive LInear NEuron*")
  - Tiene la misma arquitectura del Perceptrón, pero se entrena mediante un algoritmo de gradiente:

$$\mathbf{w}_e^{(k+1)} = \mathbf{w}_e^{(k)} - \alpha \nabla_{\mathbf{w}} J\left(\mathbf{w}_e^{(k)}\right)$$

    siendo $J(\mathbf{w_e})$ una función de coste. La más habitual es el error cuadrático:

$$J(\mathbf{w}) = \frac{1}{2N}\sum_{k=1}^{N}\left(d^{(k)} - \mathbf{w}_e^{(k)^T}\mathbf{x}_e^{(k)}\right)^2 \approx \frac{1}{2}E\left\{\left(d^{(k)} - \mathbf{w}_e^{(k)^T}\mathbf{x}_e^{(k)}\right)^2\right\}$$

  - En la versión estocástica (el algoritmo de Widrow-Hoff) se aplica el gradiente sobre una sola muestra, resultando

$$\mathbf{w}_e^{(k+1)} = \mathbf{w}_e^{(k)^T} + \alpha\left[d^{(k)} - z^{(k)}\right]\mathbf{x}_e^{(k)} \quad \left(0 < \alpha < 1\right)$$

  - Puede demostrarse que, para α suficientemente pequeño, la regla del ADALINE converge (aunque no necesariamente a una solución de mínimo numero de errores).

Perceptrón multicapa

- Arquitectura básica

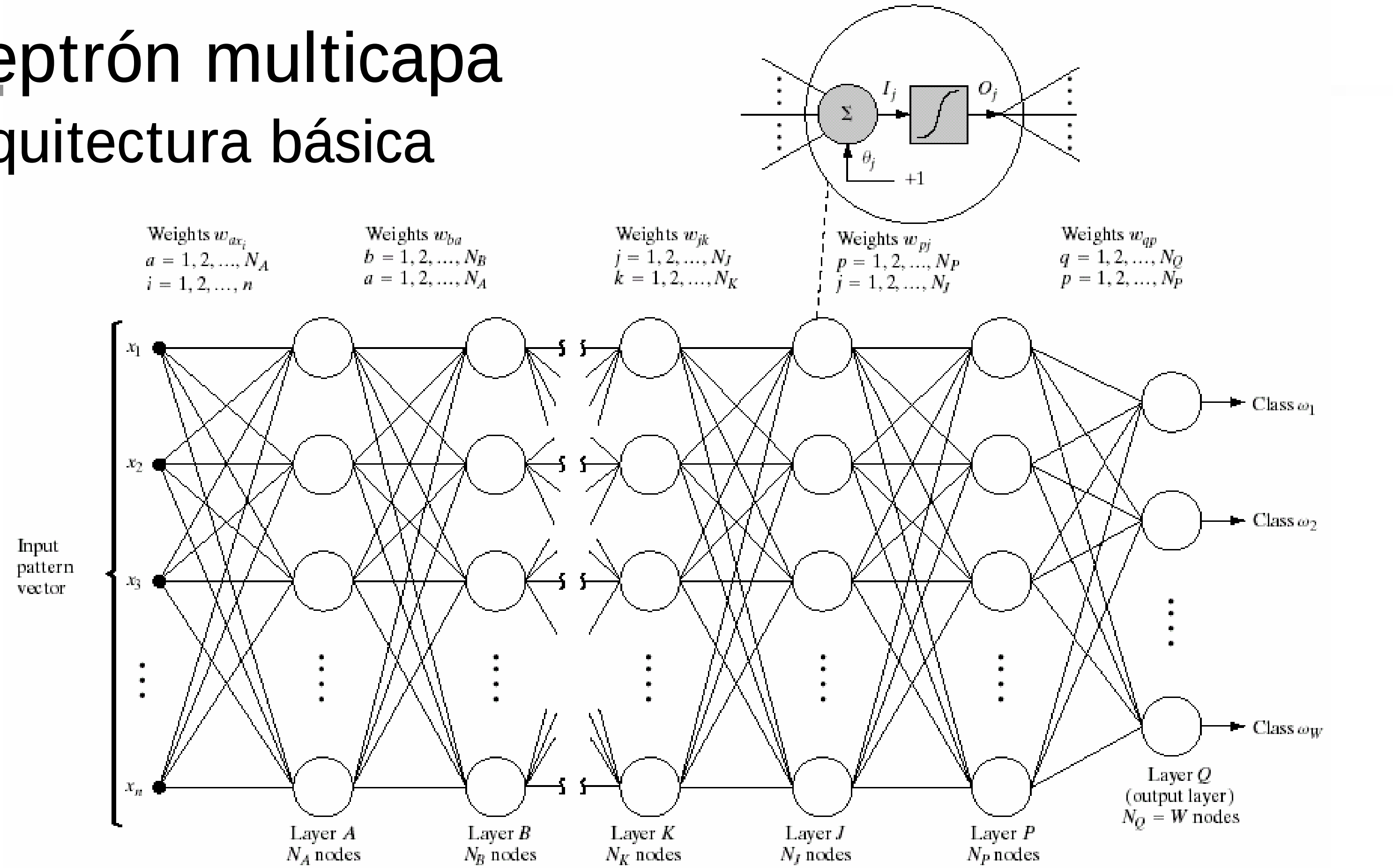


FIGURE 12.16 Multilayer feedforward neural network model. The blowup shows the basic structure of each neuron element throughout the network. The offset, θ_j, is treated as just another weight.

- El perceptrón multicapa es un **aproximador universal**.