



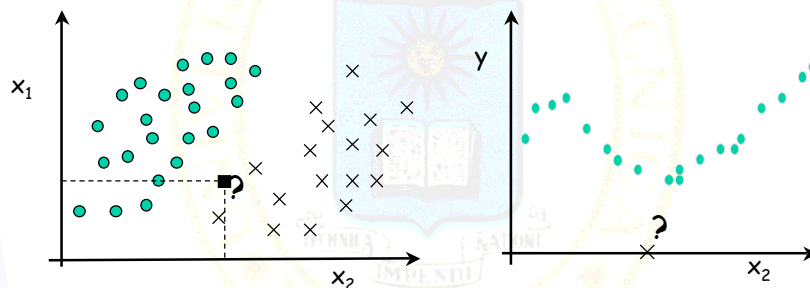
## índice

- **Técnicas clásicas de Reconocimiento de Patrones**
  - Planteamiento del problema
  - Preprocesamiento de características
  - Reconocedores



## Aprendizaje: Planteamiento

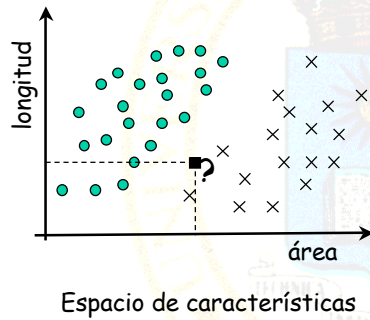
Encontrar un modelo que predice la salida correcta a partir de salidas anteriores



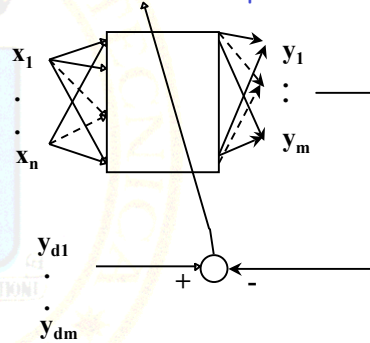


## Esquema de funcionamiento: aprendizaje supervisado

Concepto aprendizaje supervisado



Estructura de funcionamiento supervisado

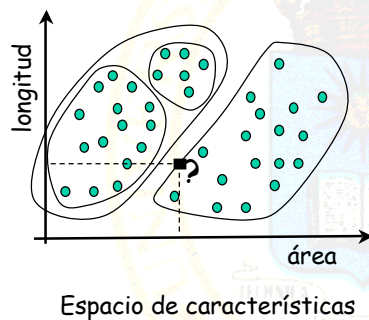


Generalización de funciones de  $R^n \Rightarrow R^m$

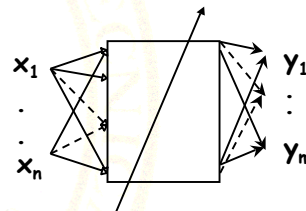


## Esquema de funcionamiento: aprendizaje no supervisado

Concepto aprendizaje no supervisado



Estructura de funcionamiento no supervisada



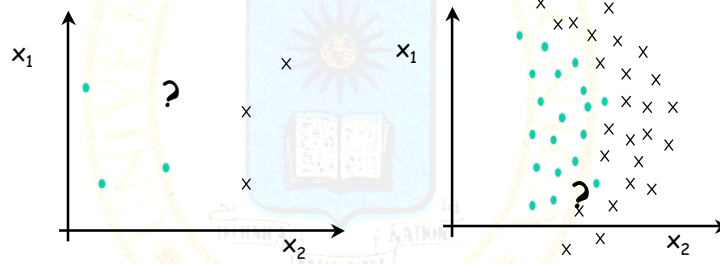
Agrupamiento en clases ("clustering")



## Dificultades en el aprendizaje

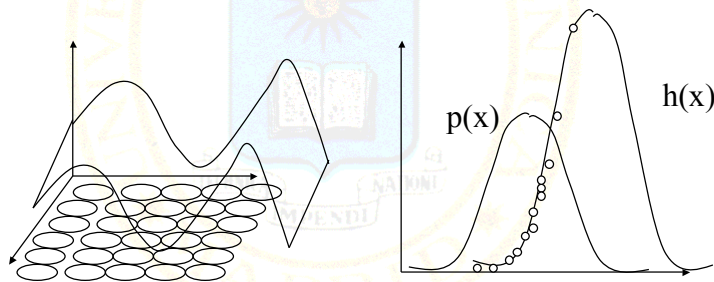
escasez de muestras

presencia de ruido



## Limitaciones del aprendizaje

- **Aprendizaje basado en la f.d.d. de las muestras  $\Rightarrow$  posible incorrelación entre la cantidad de muestras y su importancia en la salida**





## índice

- **Técnicas clásicas de Reconocimiento de Patrones**
  - Planteamiento del problema
  - Preprocesamiento de características
  - Reconocedores



## Selección de características

### Objetivos:

- **Mejorar los resultados del reconocimiento: disminuir las probabilidades de clasificación errónea o del riesgo.**
- **Simplificar el reconocedor, tanto en la fase de aprendizaje como en la de ejecución**

### Formulación:

**Dado un conjunto de características de entrada al clasificador  $x$  (dim  $n$ ), encontrar una función  $y=F(x)$  de  $R^n \Rightarrow R^m$ , tal que optimice un criterio de calidad  $Q(y)$  definido sobre  $y$**

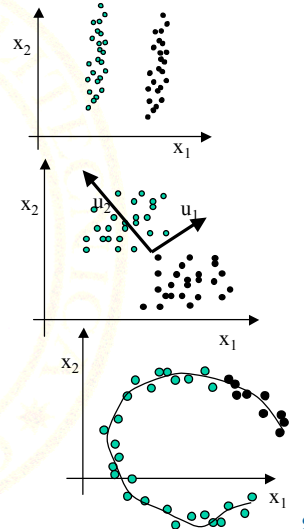


## Reducción de dimensionalidad: Selección de características

Supresión de las características  
menos relevantes

Cálculo de nuevas características  
mediante combinación lineal

Cálculo de nuevas entradas  
mediante procesamiento no lineal

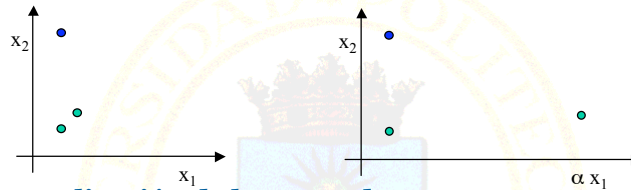


## Selección de características: Criterios de calidad (1/4)

- **Directos: minimización de la probabilidad de clasificación errónea o riesgo esperador**
  - óptimo: tiene en cuenta que un mayor número de características no necesariamente mejora la clasificación
  - dependientes del clasificador o suposición de un clasificador óptimo
  - resolución muy compleja
- **Indirectos: aumento de la separabilidad entre las muestras pertenecientes a cada clase**
  - ligado al concepto de distancia
  - la reducción de la dimensión de las características siempre empeora el criterio
  - resolución más sencilla



## Selección de características: Criterios de calidad (2/4)



- **Normalización de las entradas**

- Cada componente por separado

- $x_i^n = (x_i^n - \bar{x}_i) / \sigma_i$

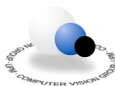
consegue media 0 y  $\sigma=1$

Conjuntamente

$$x' = \Lambda^{-1/2} U^T (x^n - \bar{x})$$

siendo  $\Lambda = \text{diag}\{\lambda_1 \dots \lambda_D\}$ ;  $U = [u_1 \dots u_D]$

donde  $\lambda_i$   $u_i$  son los valores propios y  $U^T$  los vectores propios de la matriz de C de covarianzas consigue media 0 y  $C=I$



## Selección de características: Criterios de calidad (3/4)

- **Distancias en el espacio de entradas**

- Distancia de Manhattan

- $d = \sum_i |x_i - m_i|$

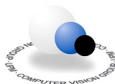
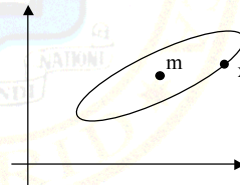
- Distancia euclídea

- $d = \text{sqrt}(\sum_i (x_i - m_i)^2) = \text{sqrt}((\mathbf{x} - \mathbf{m})^T (\mathbf{x} - \mathbf{m}))$

- Distancia de Mahalanovich

- $d = \text{sqrt}((\mathbf{x} - \mathbf{m})^T \mathbf{C}^{-1} (\mathbf{x} - \mathbf{m}))$

- siendo  $\mathbf{C} = \sum_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$  la matriz de covarianzas





## Selección de características: Criterios de calidad (4/4)

### ▪ Separabilidad de características

- Matriz de dispersión de un grupo
  - $S_i = \sum_j (x_j - m_i) (x_j - m_i)^T$
- Matriz de dispersión intra-grupos
  - $S_W = \sum_i S_i$
- Matriz de dispersión inter-grupos
  - $S_B = \sum_i n_i (m_i - m) (m_i - m)^T$
- Matriz de dispersión total
  - $S_T = \sum_j (x_j - m) (x_j - m)^T = S_W + S_B$



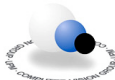
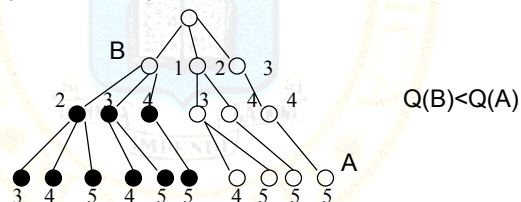
## Selección de características: Supresión de características (1/2)

### ▪ Búsqueda exhaustiva: $O(2^n)$ ó $O(n!/(n-m)!m!)$

Brach and bound: Algoritmos de Dijkstra, Floyd, A\*

- óptimo
- ahorro computacional basado en el cálculo del criterio para conjuntos mayores, teniendo en cuenta el decrecimiento monótono de dicho criterio de calidad

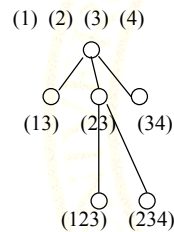
ejemplo: 2 mejores características de 5



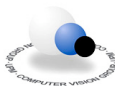
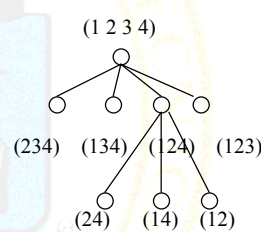
## Selección de características: Supresión de características (1/2)

- **Búsqueda por técnicas heurísticas subóptimas:**

adición de características



eliminación de características



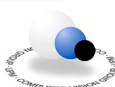
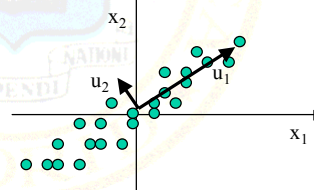
## Selección de características: Combinación lineal no supervisada

- **Análisis de componentes principales:**

- Dado un conjunto de  $N$  vectores  $n$ -dimensionales  $x_i$ , donde  $x_i = \sum_{j=1}^n a_{ij} u_j = \sum_{j=1}^m a_{ij} u_j + \sum_{j=m+1}^n a_{ij} u_j$  hallar una base ortonormal  $u_j$  tal que al seleccionar  $M$  vectores de base  $x_i' = \sum_{j=1}^m a_{ij} u_j + \sum_{j=m+1}^n b_j u_j$  minimice  $E = \sum_{i=1}^N (x_i - x_i')^2$

Solución:  $u_j$  son los vectores propios correspondientes a los mayores valores propios de la matriz de covarianzas y

$$E = \frac{1}{2} \sum_{i=m+1}^n \lambda_i$$





CVG-UPM

COMPUTER VISION

*Ejemplo PCA*

original

PCA 25

PCA 5

PCA 1

*P. Campoy*

Neural Networks and Pattern Recognition

17

CVG-UPM

COMPUTER VISION

*Ejemplo PCA*

Imágenes sintéticas creadas sobre el subespacio de dimensión 1

*P. Campoy*

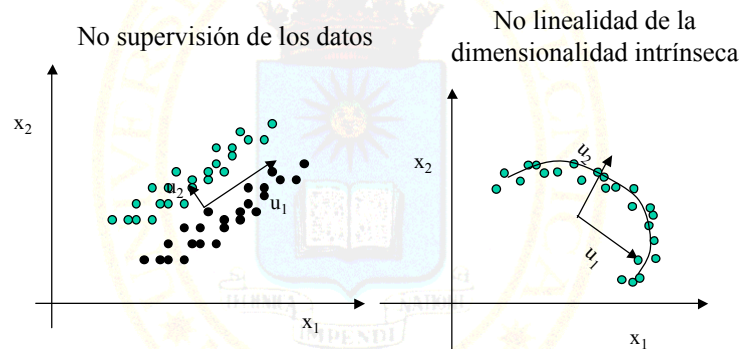
Neural Networks and Pattern Recognition

18



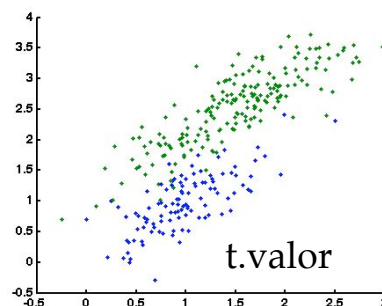
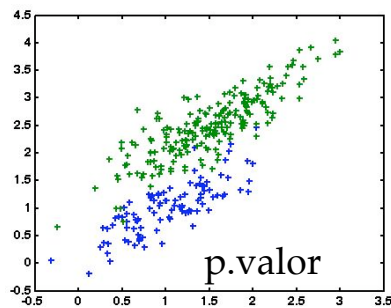
## Selección de características: Combinación lineal no supervisada

### Limitaciones del Análisis de Componentes Principales



## Ejemplo 1

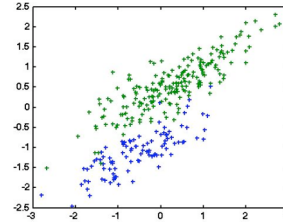
```
>> load datos_D2_C2.mat
>> plot(p.valor(1,1:100),p.valor(2,1:100),'+'); hold all;
>> plot(p.valor(1,101:300),p.valor(2,101:300),'+');
```





## Ejemplo 1: normalización

```
[D,N]=size(p.valor); [D,Nt]=size(t.valor);
clear pn; clear meanp; clear stdp;
meanp=mean(p.valor'); stdp=std(p.valor)';
for i=1:N
    pn(:,i)=(p.valor(:,i)-meanp)./stdp;
end
for i=1:Nt
    tn(:,i)=(t.valor(:,i)-meanp)./stdp;
end
```



alternativas:

```
[pn,meanp,stdp,tn,meant,stdt]=prestd(p.valor,t.valor);
[pn,meanp,stdp]=zscore(p.valor'); pn=pn';meanp=meanp';
std=std'
```



## Ejemplo 1: PCA

```
[transMatc,Diag]=eig(cov(pn')); ncompca=1;
for i=1:ncompca
    transMat(i,:)=transMatc(:,D+1-i)';
end
error=Diag(1,1)/2;
```

alternativas:

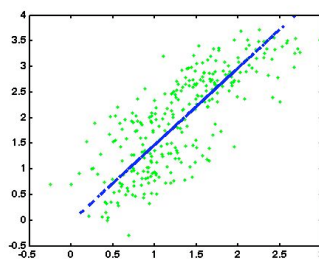
```
[ptrans,transMat]=prepca(pn,0.2);
[transMatc, ptransc] = princomp(pn'); transMatc=transMatc';
ptrans=ptransc';
[residual,preconstructed]=pcares(pn',ncompca);
```



## Ejercicio 1.1

>> load datos\_D2\_C2

1. Calcular los vectores de coordenadas reducidas de los puntos de test sobre el subespacio definido por el PCA (3 puntos)
2. Calcular los vectores de dimensión completa del apartado anterior y dibujarlos junto con los puntos de test. (4 puntos)
3. Calcular el ECM cometido mediante la aproximación PCA (3 puntos)



P. Campoy

Neural Networks and Pattern Recognition

23

## Selección de características: Combinación lineal supervisada (1/2)

### Discriminante bidimensional

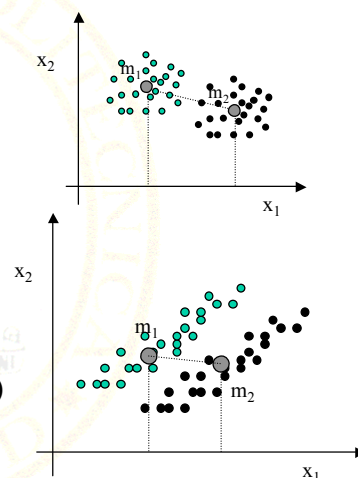
- $y = \mathbf{w}^T \mathbf{x}$
- idea inicial: maximizar  $m_2 - m_1 = \mathbf{w}^T(\mathbf{m}_2 - \mathbf{m}_1) \Rightarrow \mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$   
no valida para distribuciones no isotrópicas

discriminante de Fisher:

$$J(\mathbf{w}) = (\mathbf{m}_2 - \mathbf{m}_1)^2 / (s_1^2 + s_2^2) = (\mathbf{w}^T \mathbf{S}_B \mathbf{w}) / (\mathbf{w}^T \mathbf{S}_W \mathbf{w})$$

maximización:  $\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1)$

isotrópico:  $\mathbf{S}_W^{-1} = \mathbf{I}$



P. Campoy

Neural Networks and Pattern Recognition

25



## Selección de características: Combinación lineal supervisada (2/2)

- **Discriminante de Fisher multidimensional:**

- transformación:  $\mathbf{y} = \mathbf{W}^T \mathbf{x}$

- discriminante:

$$J(\mathbf{w}) = \text{Traza}\{\mathbf{S}_W^{-1}\mathbf{S}_B\} = \text{Traza}\{(\mathbf{W}^T\mathbf{S}_W\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{S}_B\mathbf{W})\}$$

maximización:

$\mathbf{W}$  vectores propios de  $\mathbf{S}_W^{-1}\mathbf{S}_B$

$\mathbf{W}$  tiene una dimensión máxima de  $c-1$ ,  
siendo  $c$  el número de clases

isotrópico:  $\mathbf{S}_W^{-1} = \mathbf{I}$



## índice

- **Técnicas clásicas de Reconocimiento de Patrones**

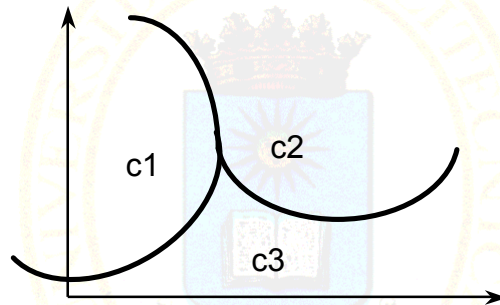
- Planteamiento del problema

- Preprocesamiento de características

- Reconocedores



## Objetivo del Reconocedor:



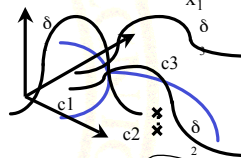
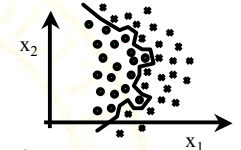
División el espacio de entrada en regiones asociadas a los patrones



## Clasificadores no neuronales

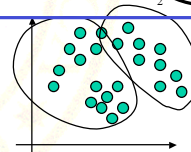
### Supervisados:

- Vecino más cercano
- Bayesianos (ajuste de f.d.d.)



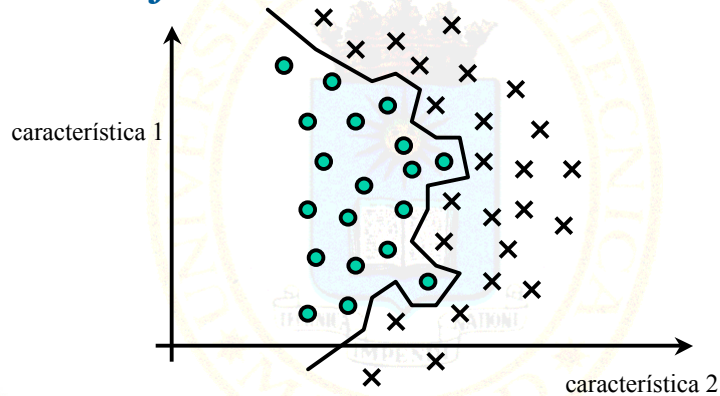
### No supervisados

- Agrupamiento basado en distancias



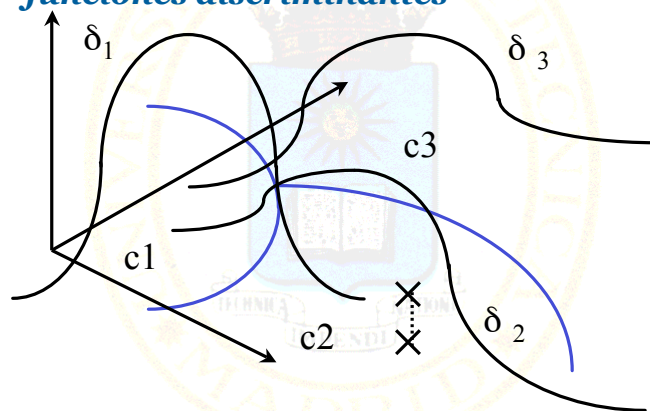
## Reconocedores supervisados clásicos: Reconocedores no paramétricos

### ▪ Clasificador vecino más cercano



## Reconocedores supervisados clásicos: Reconocedores paramétricos

### ▪ Minimización del error o riesgo $\Rightarrow$ funciones discriminantes

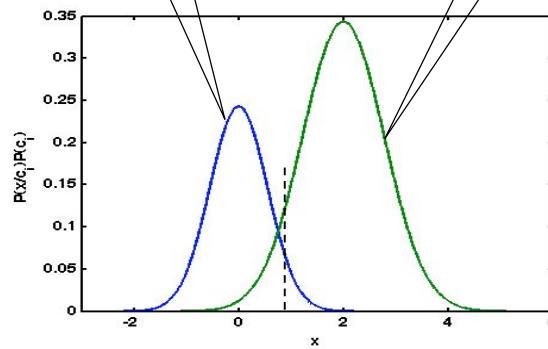




## Reconocedor Bayesiano

- funciones discriminantes de mayor probabilidad

$$P(c_1/x) = \frac{P(x/c_1)P(c_1)}{P(x)} \quad P(c_2/x) = \frac{P(x/c_2)P(c_2)}{P(x)}$$



P. Campoy

Neural Networks and Pattern Recognition

32

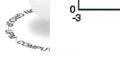
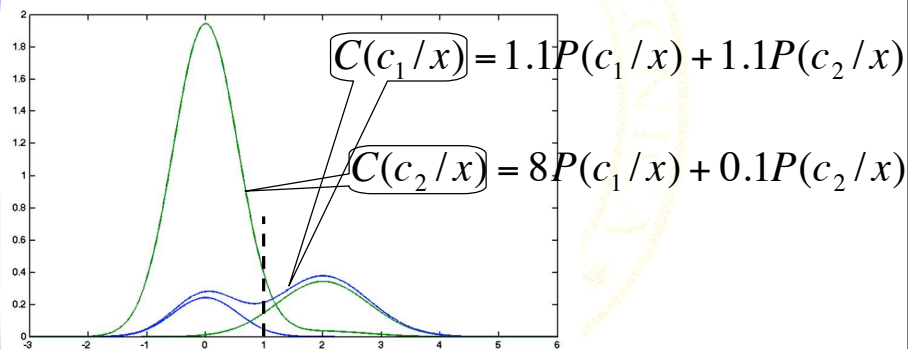


## Reconocedor Bayesiano

- funciones discriminantes de menor costo

$$C(c_1/x) = C(c_1/c_1)P(c_1/x) + C(c_1/c_2)P(c_2/x)$$

$$C(c_2/x) = C(c_2/c_1)P(c_1/x) + C(c_2/c_2)P(c_2/x)$$



P. Campoy

Neural Networks and Pattern Recognition

33





## Comandos Matlab de clasificación

```
>> nnclass=knnclassify(t.valor', p.valor', p.clase,k);
>> [bayclass, err, posterior] = classify(t.valor', p.valor', ...
    p.clase, 'linear',prior);

>> no_erroneas_nn=size(find(nnclase'~=t.clase),2)
>> no_erroneas_bay=size(find(bayclase'~=t.clase),2)
```



## Ejercicio 2.2

- Dados los puntos del ejemplo anterior
  - >> load datos\_D1\_C2.mat
  - a) Calcular la clase del dato t.valor=1 de menor error en la clasificación, suponiendo las iguales las probabilidades a priori  $P(\text{clase}=1)=P(\text{clase}=2)$  (2 puntos)
  - b) Idem, suponiendo que las probabilidades a priori son proporcionales al numero de muestras de cada clase en los datos de entrenamiento (4 puntos)
  - c) Calcular la clase del dato t.valor=1 de menor coste, suponiendo que la matriz de costes es: (4 puntos)

$$C(c1/c1) = 1.1 \quad C(c1/c2) = 1.1$$

$$C(c2/c1) = 8 \quad C(c2/c2) = 0.1$$





## Ejercicio 2.3

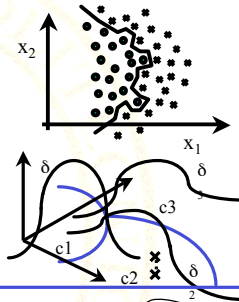
- Dados los datos del ejemplo 2.1  
`>> load datos_D2_C2.mat`
  - a) Calcular y analizar los errores de clasificación de p.valor y también de t.valor, cuando se usan los datos de p.valor como datos de entrenamiento y el clasificador más cercano (2,5 puntos)
  - b) idem usando el clasificador bayesiano (2,5 puntos)
  - c) Calcular y analizar los errores de clasificación de t.valor usando los dos tipos de clasificadores mencionados (2,5 puntos)
  - d) Calcular y analizar los errores de clasificación de t.valor cuando se usa el clasificador bayesiano con los siguientes valores de probabilidades a priori: [1 0.1],[1 2] y [1 10] (2,5 puntos)



## Clasificadores no neuronales

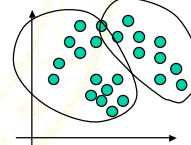
### ▪ **Supervisados:**

- Vecino más cercano
- Bayesianos (ajuste de f.d.d.)



### ▪ **No supervisados**

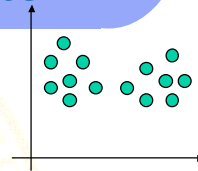
- Agrupamiento basado en distancias



## Reconocedores no supervisados: Algoritmos de agrupamiento

Objetivo:

“minimizar”  $S_W$  y “maximizar”  $S_B$



- **Agrupamiento dinámico (ISODATA o k-means)**
  - Asignar cada vector de muestra al grupo de media más cercana, hasta que las nuevas medias recalculadas no varíen
  - > `class=kmeans(p',n_classes)`
- **Agrupamiento jerárquico**
  - Asignar un grupo para cada muestra y unir los dos grupos más cercanos hasta que se obtenga el número de grupos deseados



## Clasificadores no supervisados

- **Grupos redondeados en el espacio de características**

