

# Minería de Datos

## 3. Técnicas de Minería de Datos

*José Hernández Orallo*

jorallo@dsic.upv.es

Máster y Cursos de Postgrado del DSIC  
Universitat Politècnica de València

## Temario

1. Introducción a la Minería de Datos (DM)
  - 1.1. Motivación
  - 1.2. Problemas tipo y aplicaciones
  - 1.3. Relación de DM con otras disciplinas
2. El proceso de KDD
  - 2.1. Las Fases del KDD
  - 2.2. Tipología de Técnicas de Minería de Datos
  - 2.3. Sistemas Comerciales y Herramientas de Minería de Datos
  - 2.4. Preparación y Visualización de datos
3. Técnicas de Minería de Datos
  - 3.1. El Problema de la Extracción Automática de Conocimiento.
  - 3.2. Evaluación de Hipótesis
  - 3.3. Técnicas no supervisadas y descriptivas.
  - 3.4. Técnicas supervisadas y predictivas.
4. Web Mining
  - 4.1. Los Problemas de la Información No Estructurada.
  - 4.2. Extracción de Conocimiento a partir de Documentos HTML y texto.
  - 4.3. Extracción de Información semi-estructurada (XML).
5. Otros Aspectos

2

## Objetivos Tema 3

- Conocer las distintas técnicas de aprendizaje automático y estadísticas utilizadas en minería de datos, su potencial, su coste computacional y sus limitaciones de representación y de inteligibilidad.
- Conocer medidas de evaluación de modelos (p.ej. validación cruzada).
- Utilizar métodos de combinación de técnicas (p.ej. voting) y de reiteración (p.ej. boosting).
- Conocer los métodos descriptivos y no supervisados más importantes.
- Conocer los métodos predictivos y supervisados más importantes.

3

## 3. Técnicas de Minería de Datos

- 3.1. El Problema de la Extracción Automática de Conocimiento.
- 3.2. Evaluación de Hipótesis
- 3.3. Técnicas no supervisadas y descriptivas.
- 3.4. Técnicas supervisadas y predictivas.

4

## El Problema de la Extracción Automática de Conocimiento

La minería de datos no es más que un caso especial de aprendizaje computacional inductivo.

### ¿Qué es aprendizaje?

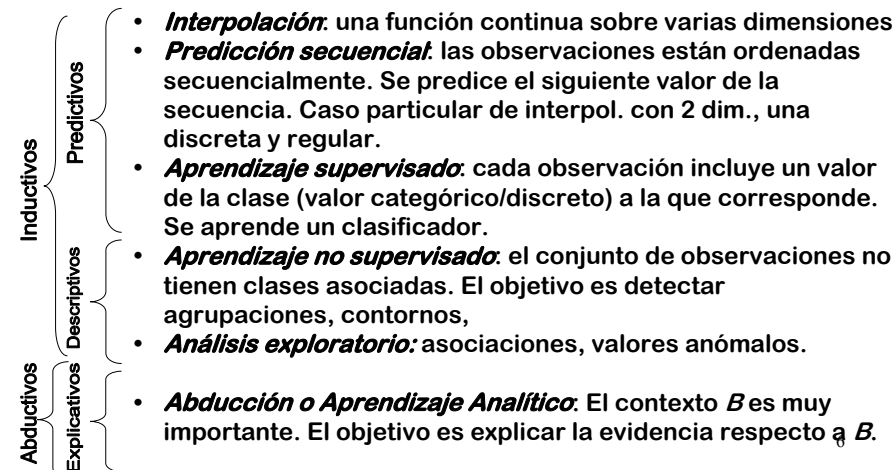
- (visión genérica, Mitchell 1997) es mejorar el comportamiento a partir de la experiencia. Aprendizaje = Inteligencia.
- (visión más estática) es la *identificación de patrones*, de *regularidades*, existentes en la evidencia.
- (visión externa) es la *predicción* de observaciones futuras con *plausibilidad*.
- (visión teórico-informacional, Solomonoff 1966) es *eliminación de redundancia* = *compresión de información*.

Aprendizaje Inductivo: razonamiento hipotético de casos particulares a casos generales.

5

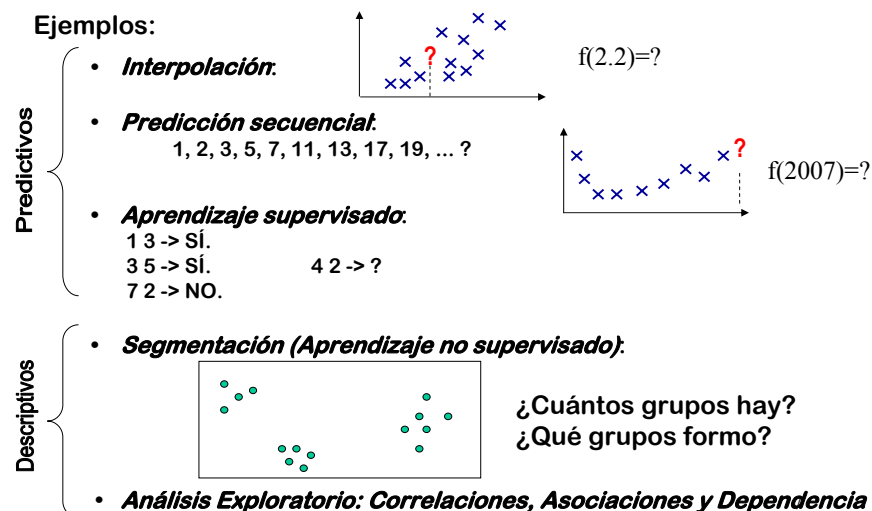
## El Problema de la Extracción Automática de Conocimiento

Clasificación de las técnicas de aprendizaje:



## El Problema de la Extracción Automática de Conocimiento

Ejemplos:



## El Problema de la Extracción Automática de Conocimiento

### **PREDICTIVO: Interpolación y Predicción Secuencial.**

- Se conoce generalmente como “estimación” o “regresión”.

*Ejemplo: estimar el número de hijos de una familia a partir de otros ejemplos de familias.*

*Ejemplo: estimar las ventas del mes que viene a partir de los meses anteriores.*

- **Regresión Lineal:**
  - Regresión lineal global (clásica).
  - Regresión lineal ponderada localmente.
- **Regresión No Lineal:** logarítmica, pick & mix, ...
  - Técnicas no algebraicas.
  - Redes neuronales.
  - Árboles de regresión.

8

## El Problema de la Extracción Automática de Conocimiento

---

### ***PREDICTIVO: Aprendizaje supervisado.***

Dependiendo de si se estima una función o una correspondencia:

- categorización: se estima una correspondencia (las clases pueden solapar).  
*Ejemplo: determinar de un conjunto de temas de qué temas trata una determinada página web (cada página puede tratar de varios temas).*
- clasificación: se estima una función (las clases son disjuntas).  
*Ejemplo: determinar el grupo sanguíneo a partir de los grupos sanguíneos de los padres.*  
*Ejemplo: Determinar si un compuesto químico es cancerígeno.*

9

## El Problema de la Extracción Automática de Conocimiento

---

### ***PREDICTIVO: Aprendizaje supervisado (Clasificación).***

- Técnicas:
  - k-NN (Nearest Neighbor).
  - k-means (competitive learning).
  - Perceptron Learning.
  - Multilayer ANN methods (e.g. backpropagation).
  - Radial Basis Functions.
  - Support Vector Machines
  - Decision Tree Learning (e.g. ID3, C4.5, CART).
  - Bayes Classifiers.
  - Center Splitting Methods.
  - Rules (CN2)
  - Pseudo-relational: Supercharging, Pick-and-Mix.
  - Relational: ILP, IFLP, *SCIL*.

10

## El Problema de la Extracción Automática de Conocimiento

---

### ***DESCRIPTIVO: Análisis Exploratorio***

- Técnicas:
  - Estudios correlacionales
  - Asociaciones.
  - Dependencias.
  - Detección datos anómalos.
  - Análisis de dispersión.

11

## El Problema de la Extracción Automática de Conocimiento

---

### ***DESCRIPTIVO: Segmentación (Aprendizaje no supervisado)***

- Técnicas de *clustering*.
  - Jerárquico
  - No jerárquico
    - k-means (competitive learning).
    - redes neuronales de Kohonen
    - EM (Estimated Means) (Dempster et al. 1977).
    - Cobweb (Fisher 1987).
    - AUTOCLASS
    - ...

12

## Similitud/Distancia

Un concepto importante en el aprendizaje supervisado (clasificación) y no supervisado (segmentación) es el concepto de similitud:

- La razón de este uso es que, intuitivamente, datos similares tendrán clases/grupos similares. ¿Cómo se mide la similitud?
- DISTANCIA inversa a SIMILITUD.
- Los métodos de similitud (o de distancia) se basan en almacenar los ejemplos vistos, y calcular la similitud/distancia del nuevo caso con el resto de ejemplos.

13

## Similitud/Distancia

- Muchísimas formas de calcular la distancia:

- **Distancia Euclídea:**  $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
  - **Distancia de Manhattan:**  $\sum_{i=1}^n |x_i - y_i|$
  - **Distancia de Chebychev:**  $\max_{i=1..n} |x_i - y_i|$
  - **Distancia del coseno:**  
*cada ejemplo es un vector y la distancia es el coseno del ángulo que forman*
  - **Distancias por Diferencia:**  
*ejemplo: if  $x=y$  then  $D=0$  else  $D=1$*
  - **Distancia de Edición:**
  - **Distancias Específicas:** para los ejemplos complejos de CBR.
- Valores Continuos (conveniente normalizar entre 0-1 antes)
- Valores Continuos. No es necesario normalizar
- Valores Discretos

14

## 3. Técnicas de Minería de Datos

### 3.1. El Problema de la Extracción Automática de Conocimiento.

### 3.2. Evaluación de Hipótesis

### 3.3. Técnicas no supervisadas y descriptivas.

### 3.4. Técnicas supervisadas y predictivas.

15

## Evaluación de Hipótesis

### ¿Cómo se validan/descartan las hipótesis para conformar el conocimiento adquirido?

- Principio ('escándalo') de la Inducción: las hipótesis pueden ser refutadas, pero nunca confirmadas.
- Y para las que todavía no han sido refutadas, ¿cuál elegimos?
  - Necesidad de criterios de selección: simplicidad, refuerzo, ...
  - Existencia de métodos de validación: estadísticos, cross-validation, informacionales, ...
- ¿Cuánto afecta a la plausibilidad el número de ejemplos?
- ¿Cómo afecta la presencia de ruido?

16

## Evaluación de Hipótesis

**El problema del aprendizaje NO está especificado completamente.**

- Si sólo nos basamos en la evidencia, una solución al problema sería *cualquier hipótesis que cubre la evidencia*.
- Si el lenguaje es expresivo, pueden existir infinitas hipótesis.
- Objetivo: Elegir la hipótesis  $h$  que MINIMIZA EL ERROR de la hipótesis  $h$  respecto la función objetivo  $f$ ,

¿Qué error?

17

## Evaluación de Hipótesis

### Evaluación de modelos predictivos:

¿Qué medida usamos para comparar el valor correcto "f" del valor estimado "h" ?

#### Clasificación:

- %Acierto o, inversamente, %Error
- Alcance y precisión (recall & precision).
- Área bajo la curva ROC.
- ...

#### Regresión:

- Error cuadrático medio.
- Error absoluto medio.
- ...

18

## Evaluación de Hipótesis

### Evaluación de modelos predictivos:

- Dado un conjunto  $S$  de  $n$  datos, el error se define:

#### Clasificación: Error

$$error_S(h) = \frac{1}{n} \sum_{x \in S} \delta(f(x), h(x))$$

donde  $\delta(a,b)=0$  si  $a=b$  y 1 en caso contrario.

Clase predicha ( $h(x)$ )	Clase real ( $f(x)$ )	Error
Compra	Compra	No
No Compra	Compra	Si
Compra	No Compra	Si
Compra	Compra	No
No Compra	No Compra	No
No Compra	Compra	Si
No Compra	No Compra	No
Compra	Compra	No
Compra	Compra	No
No Compra	No Compra	No

Fallos / Total



Error = 3/10 = 0.3

19

## Evaluación de Hipótesis

### Evaluación de modelos predictivos:

- Dado un conjunto  $S$  de  $n$  datos, el error se define:

#### Regresión: Error Cuadrático Medio

$$error_S(h) = \frac{1}{n} \sum_{x \in S} (f(x) - h(x))^2$$

Valor predicho ( $h(x)$ )	Valor real ( $f(x)$ )	Error	Error <sup>2</sup>
100 mill. €	102 mill. €	2	4
102 mill. €	110 mill. €	8	64
105 mill. €	95 mill. €	10	100
95 mill. €	75 mill. €	20	400
101 mill. €	103 mill. €	2	4
105 mill. €	110 mill. €	5	25
105 mill. €	98 mill. €	7	49
40 mill. €	32 mill. €	8	64
220 mill. €	215 mill. €	5	25
100 mill. €	103 mill. €	3	9



Error = 744/10 = 74,4

20

## Evaluación de Hipótesis

### Medidas de Error para evaluar Hipótesis

$D$  : dominio

$S$  : sample (muestra)

#### TRUE ERROR:

caso discreto

caso continuo (p.ej. error cuadrático medio)

$$error_D(h) = \Pr_{x \in D}[f(x) \neq h(x)]$$

$$error_D(h) = \lim_{S \rightarrow D} \frac{1}{n} \sum_{x \in S} (f(x) - h(x))^2$$

#### SAMPLE ERROR :

caso discreto

caso continuo (p.ej. error cuadrático medio)

$$error_{train}(h) = \frac{1}{n} \sum_{x \in trainSet} \delta(f(x) \neq h(x))$$

$$error_{train}(h) = \frac{1}{n} \sum_{x \in trainSet} (f(x) - h(x))^2$$

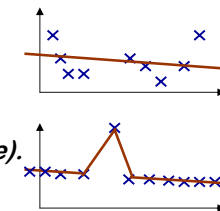
donde  $(\delta(true)=1, \delta(false)=0)$  y  $n = |trainSet|$

21

## Evaluación de Hipótesis

### Problemas típicos:

- under-fitting  
(sobregeneralización o subajuste)
- over-fitting  
(sobreespecialización o superajuste).



- Definición de **over-fitting**: Una hipótesis  $h \in H$  sobre-especializa o superajusta si existe una hipótesis alternativa  $h' \in H$  tal que:

$$error_{train}(h) < error_{train}(h')$$

y

$$error_D(h) > error_D(h')$$

Sample or train error

True error

22

## Evaluación de Hipótesis

- **Problema:**  $f$  (la función objetivo) no se conoce!!!
- Podemos calcular el SAMPLE ERROR pero no el TRUE ERROR.
- Si nos fijamos sólo en toda la muestra y minimizamos el SAMPLE ERROR, aparecerán dos problemas:
  - si la evidencia es sólo positiva: under-fitting o sobregeneralización.
  - Si la evidencia tiene más de una clase: over-fitting o sobreespecialización.

23

## Evaluación de Hipótesis

¿Qué hipótesis elegimos?

### APROXIMACIONES:

- Asumir distribuciones a priori.
- Criterio de simplicidad, de descripción o transmisión mínimas.
- Separar: Training Set y Test Set.
  - Cross-validation.
- Basadas en refuerzo.

En frío  
En caliente  
En frío  
En caliente

Otras preguntas importantes:

¿Cómo sabemos lo bien que se comportará en el futuro?

24

## Evaluación de Hipótesis

### Evaluación por técnicas bayesianas.

- La mejor hipótesis es la más probable.
- Basadas en el teorema de Bayes. Despejan  $P(h/D)$ .
  - La distribución de hipótesis a priori  $P(h)$  y la probabilidad de unas observaciones respecto a cada hipótesis  $P(D/h)$  deben ser conocidas.
  - Son sólo técnicas evaluadoras aunque si el conjunto de hipótesis  $H$  es reducido se pueden utilizar en algoritmos de aprendizaje.
  - Permiten acomodar hipótesis probabilísticas tales como “este paciente de neumonía tiene un 93% de posibilidades de recuperarse”.
  - Muchas veces no se conoce  $P(h)$  o incluso  $P(D/h)$ . Se hacen suposiciones: distribución uniforme, normal o universal.

25

## Evaluación de Hipótesis

### Teorema de Bayes, MAP y Maximum Likelihood:

- $P(h/D)$ : probabilidad de una hipótesis dado un cijo. de datos.
- $P(h)$ : probabilidad a priori de las hipótesis.
- $P(D/h)$ : probabilidad de  $D$  dada la hipótesis.
- $P(D)$ : probabilidad a priori de los datos (sin otra información).

- Teorema de Bayes: (prob. a posteriori a partir de a priori)

$$P(h | D) = \frac{P(D | h)P(h)}{P(D)}$$

- Criterio MAP (Maximum a Posteriori) ( $h$  es indep. de  $P(D)$ ):

El Naive Bayes Classifier es un caso particular de esto.

$$h_{MAP} = \arg \max_{h \in H} P(h | D) = \arg \max_{h \in H} \frac{P(D | h)P(h)}{P(D)} = \arg \max_{h \in H} P(D | h)P(h)$$

- Maximum Likelihood (asumiendo  $P(h)$  uniforme):

$$h_{ML} = \arg \max_{h \in H} P(D | h)$$

26

## Evaluación de Hipótesis

### Evaluación bayesiana:

Si el cijo. de hipótesis  $H$  es pequeño y conocido:

- Se puede asumir la distribución uniforme:

$$P(h) = \frac{1}{|H|}$$

Si  $H$  es infinito:

- La distribución uniforme no está bien definida ( $P=0$ ).
- Aunque el maximum likelihood se puede seguir utilizando.

27

## Evaluación de Hipótesis

### El principio MDL (Minimum Description Length):

- Asumimos  $P(h)$  como la distribución universal (Occam's Razor):

$$P(h) = 2^{-K(h)}$$

donde  $K(\cdot)$  es la complejidad descriptiva (Kolmogorov) de  $H$ .

### FORMALIZACIÓN DE LA NAVAJA DE OCCAM:

“Las hipótesis con mínima descripción más pequeña son más probables”.

- Asumimos  $P(D/h)$  de la misma manera:

$$P(D | h) = 2^{-K(D|h)}$$

28

## Evaluación de Hipótesis

El principio MDL:

- A partir de MAP tenemos:

$$\begin{aligned} h_{MAP} &= \arg \max_{k \in H} P(D | h) P(h) = \arg \max_{k \in H} \log [P(D | h) P(h)] = \\ &= \arg \max_{k \in H} \log P(D | h) + \log P(h) = \arg \max_{k \in H} \log 2^{-K(D|h)} + \log 2^{-K(h)} = \\ &= \arg \max_{k \in H} (-K(D | h) - K(h)) \end{aligned}$$

- Resulta en:

$$h_{MDL} = \arg \min_{k \in H} (K(h) + K(D | h))$$

**PRINCIPIO MDL: La hipótesis más probable es la que minimiza la suma de su descripción y la descripción de los datos respecto a ella.**

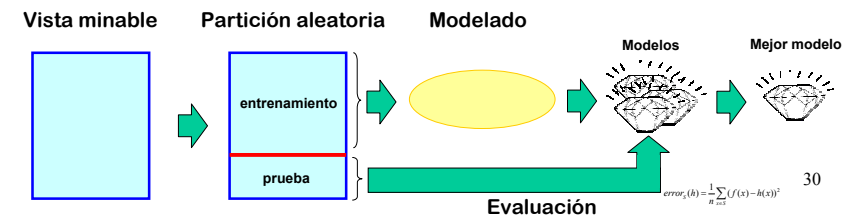
29

## Evaluación de Hipótesis

- Evaluación de modelos predictivos.

- PARTICIÓN DE LOS DATOS:

- Separación de los datos en:
  - Conjunto de entrenamiento (train).
    - Los modelos se entrenan con estos datos
  - Conjunto de prueba (test).
    - Los modelos se evalúan con estos datos.



30

## Evaluación de Hipótesis

- Evaluación de modelos predictivos.

- Particiones más elaboradas:

- Validación cruzada: Se parten los datos aleatoriamente en n pliegues de igual tamaño.
- Bootstrap: se realizan n muestras con repetición de los datos iniciales.

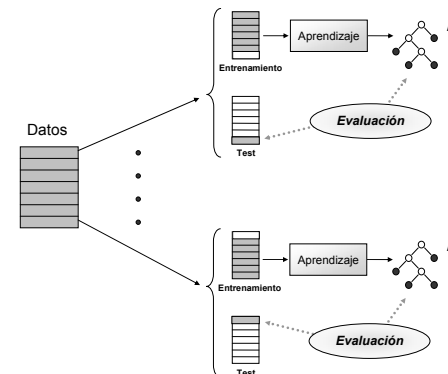
🔗 Especialmente útiles si hay pocos datos.  
🐢 Proceso mucho más lento

31

## Evaluación de Hipótesis

- Evaluación de modelos predictivos.

- Validación cruzada (detalle):



- Se realizan n particiones, incluyendo n-1 pliegues para entrenamiento y 1 para evaluación.
- El error medio se calcula promediando las 10 veces.
- Se reentrena un último modelo con todos los datos.

32



## Evaluación de Hipótesis

*Una vez obtenida una hipótesis...*

¿cómo obtener su precisión (accuracy) para datos futuros?

- Utilizar la precisión para el *training data* puede ser una aproximación, ¿pero cuán buena?
- La estadística nos da soluciones para esto:
  - Suponiendo la muestra  $S$  de  $n$  ejemplos, la hipótesis  $h$  es discreta y son independientes.
  - Si  $n \geq 30$ , nos permite aproximar la distribución binomial con la normal.
  - Calculado el  $error_s(h)$  sobre la muestra como  $n^\circ \text{errores}/n$

33

## Evaluación de Hipótesis

Podemos obtener un intervalo de confianza a un nivel  $\alpha$ :

$$error_s(h) \pm Z_c \cdot \sqrt{\frac{error_s(h)(1 - error_s(h))}{n}}$$

donde  $Z_c$  es la constante obtenida de la tabla de confianzas de la normal.

- Algunos valores de la tabla normal:

Nivel de confianza $\alpha$ :	50%	68%	80%	90%	95%	98%	99%
Constante $Z_c$ :	0.67	1.00	1.28	1.64	1.96	2.33	2.58

34

## Evaluación de Hipótesis

### EJEMPLO:

- Considerando que una hipótesis da 12 errores sobre 40 ejemplos, tenemos un  $error_s(h) = 0.30$ .
- Tenemos, por tanto, que con confianza 95% ( $Z_c = 1.96$ ), el intervalo del error será:  
 $0.30 \pm 0.14$
- lo que quiere decir que, para el 95% de otras muestras de 40 ejemplos que probáramos, el error estaría dentro de ese intervalo.

*En general, una mejor regla para saber si se puede aplicar la evaluación anterior es que:*

$$n \cdot error_s(h)(1 - error_s(h)) \geq 5$$

*(si no, habría que utilizar la dist. binomial)*

35

## Evaluación de Hipótesis

### DATOS IMPERFECTOS:

- Tipos de Datos Imperfectos:
  - Ruido:
    - en la evidencia o ejemplos de entrenamiento.
      - Valores erróneos de argumentos de los ejemplos.
      - Clasificación errónea de algún ejemplo.
    - en el conocimiento previo.
  - Ejemplos de entrenamiento muy dispersos.
  - Conocimiento previo o características correcto pero inapropiado.
    - Existencia de mucho conocimiento/características previo irrelevante para el problema a aprender.
    - Conocimiento previo insuficiente (faltan características) para el problema a aprender (algunos predicados/funciones auxiliares serían necesarios).
  - Argumentos faltantes en los ejemplos.

36

## Evaluación de Hipótesis

### DATOS IMPERFECTOS:

- Consecuencias:
  - Ruido o dispersión de datos  $\Rightarrow$  OVERFITTING.
    - Es necesario podar las hipótesis, eliminando partes de la hipótesis muy ad-hoc (cubren uno o pocos ejemplos). El criterio MDL es un buen método para esto.
  - Conocimiento previo / características inapropiado  $\Rightarrow$  INTRATABILIDAD
    - Demasiado conocimiento previo: se necesita metaconocimiento o priorización de los predicados / características.
    - Poco conocimiento previo o del dominio: se necesita invención de nuevas funciones/conceptos/predicados.
  - Argumentos faltantes en los ejemplos  $\Rightarrow$  Se pierde tamaño de muestra si no se es capaz de aprovecharlos.
    - Los sistemas basados en árboles de decisión los tratan.

37

## Evaluación de Hipótesis

### ■ Evaluación de Modelos Descriptivos:

- Agrupamiento: mucho más compleja

Concepto de error más difícil de definir

- En los métodos basados en distancia se puede mirar:
  - GRUPOS:
    - distancia entre bordes de los clusters
    - distancia entre centros (de haberlos)
    - radio y densidad (desv. típica de la dist.) de los clusters.
  - Para cada ejemplo a agrupar se comprueba su distancia con el centro o con el borde de cada cluster.
- Se pueden hacer diferentes agrupamientos con distintas técnicas y comparar los grupos formados (matriz de confusión)

38

## Evaluación de Hipótesis

### ■ Evaluación con sesgos o desequilibrios.

- Desequilibrios:
  - En clasificación puede haber muchos ejemplos de una clase y muy pocos del resto.
    - Problema: la clase escasa se puede tomar como ruido y ser ignorada por la teoría.
    - Ejemplo: si un problema binario (*sí/no*) sólo hay un 1% de ejemplos de la clase *no*, el modelo "todo es de la clase *sí*" tendría un 99% de acierto.

Este modelo es inútil

- Soluciones:
  - Utilizar sobremuestreo...
  - Macromedia,
  - Análisis ROC

39

## 3. Técnicas de Minería de Datos

### 3.1. El Problema de la Extracción Automática de Conocimiento.

### 3.2. Evaluación de Hipótesis

### 3.3. Técnicas no supervisadas y descriptivas.

### 3.4. Técnicas supervisadas y predictivas.

40

## Métodos Descriptivos

### Correlación y Asociaciones (análisis exploratorio):

- **Coefficiente de correlación:**

$$Cor(\bar{x}, \bar{y}) = \frac{Cov(\bar{x}, \bar{y})}{\sigma_x \cdot \sigma_y}$$

donde

$$Cov(\bar{x}, \bar{y}) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)$$

- **Asociaciones (cuando los atributos son discretos).**
  - Ejemplo: tabaquismo y alcoholismo están asociados.
- **Dependencias funcionales: asociación unidireccional.**
  - Ejemplo: el nivel de riesgo de enfermedades cardiovasculares depende del tabaquismo y alcoholismo (entre otras cosas).

41

## Métodos Descriptivos

### Correlaciones y Estudios Factoriales:

- Permiten establecer relevancia/irrelevancia de factores y si aquella es positiva o negativa respecto a otro factor o variable a estudiar.

Ejemplo (Kiel 2000): Estudio de visitas: 11 pacientes, 7 factores:

- Health: salud del paciente (referida a la capacidad de ir a la consulta). (1-10)
- Need: convicción del paciente que la visita es importante. (1-10)
- Transportation: disponibilidad de transporte del paciente al centro. (1-10)
- Child Care: disponibilidad de dejar los niños a cuidado. (1-10)
- Sick Time: si el paciente está trabajando, puede darse de baja. (1-10)
- Satisfaction: satisfacción del cliente con su médico. (1-10)
- Ease: facilidad del centro para concertar cita y eficiencia de la misma. (1-10)
- No-Show: indica si el paciente no se ha pasado por el médico durante el último año (0-se ha pasado, 1 no se ha pasado)

42

## Métodos Descriptivos

### Correlaciones y Estudios Factoriales. Ejemplo (cont.):

Matriz de correlaciones:

	Health	Need	Transp'tion	Child Care	Sick Time	Satisfaction	Ease	No-Show
Health	1							
Need	-0.7378	1						
Transportation	0.3116	-0.01041	1					
Child Care	0.3116	-0.01041	1	1				
Sick Time	0.2771	0.0602	0.6228	0.6228	1			
Satisfaction	0.22008	-0.1337	0.6538	0.6538	0.6257	1		
Ease	0.3887	-0.0334	0.6504	0.6504	0.6588	0.8964	1	
No-Show	0.3955	-0.5416	-0.5031	-0.5031	-0.7249	-0.3988	-0.3278	1

### Coefficientes de Regresión:

Independent Variable	Coefficient
Health	.6434
Need	.0445
Transportation	-.2391
Child Care	-.0599
Sick Time	-.7584
Satisfaction	.3537
Ease	-.0786

Indica que un incremento de 1 en el factor Health aumenta la probabilidad de que no aparezca el paciente en un 64.34%

43

## Métodos Descriptivos

### Reglas de Asociación y Dependencia:

- La terminología no es muy coherente en este campo (Fayyad, p.ej. suele llamar asociaciones a todo y regla de asociación a las dependencias):

### Asociaciones bidireccionales:

Se buscan asociaciones de la siguiente forma:

$$(X_1 = a) \leftrightarrow (X_2 = b)$$

De los  $n$  casos de la tabla, que las dos comparaciones sean verdaderas o falsas será cierto en  $r_c$  casos:

Un parámetro  $T_c$  (confidence):

$$T_c = \text{certeza de la regla} = r_c / n$$

- si consideramos valores nulos, tenemos también un número de casos en los que se aplica satisfactoriamente (diferente de  $T_c$ ) y denominado  $T_s$ .

## Métodos Descriptivos

### Reglas de Asociación y Dependencia de Valor:

#### Dependencias de Valor o Asociaciones orientadas:

Se buscan dependencias de la siguiente forma (if *Ante* then *Cons*):

P.ej. if (X1=a, X3=c, X5=d) then (X4=b, X2=a)

*De los n casos de la tabla, el antecedente se puede hacer cierto en  $r_a$  casos y de estos en  $r_c$  casos se hace también el consecuente, tenemos:*

Dos parámetros  $T_c$  (confidence/accuracy) y  $T_s$  (support):

$T_c$  = certeza de la regla =  $r_c/r_a$ , fuerza o confianza  $P(Cons|Ante)$

$T_s$  = mínimo n° de casos o porcentaje en los que se aplica satisfactoriamente ( $r_c$  o  $r_c/n$  respectivamente).

Llamado también prevalencia:  $P(Cons \wedge Ante)$

45

## Métodos Descriptivos

### Reglas de Asociación y Dependencia de Valor.

#### Ejemplo:

DNI	Renta Familiar	Ciudad	Profesión	Edad	Hijos	Obeso	Casado
11251545	5.000.000	Barcelona	Ejecutivo	45	S	S	S
30512526	1.000.000	Melilla	Abogado	25	N	S	N
22451616	3.000.000	León	Ejecutivo	35	S	S	S
25152516	2.000.000	Valencia	Camarero	30	N	S	S
23525251	1.500.000	Benidorm	Animador Parque Temático	30	N	N	N

#### Asociaciones bidireccionales (ítems frecuentes):

Casado e Hijos están asociados (80%, 4 casos).

Obeso y casado están asociados (80%, 4 casos).

#### Dependencias (Asociaciones orientadas):

Hijos → Casado (100%, 2 casos).

Casado → Obeso (100%, 3 casos).

46

## Métodos Descriptivos

### Algoritmos de búsqueda de asociaciones y dependencias.

La mayoría se basa en descomponer el problema en dos fases:

- FASE A: BÚSQUEDA DE "LARGE ITEMSETS". Se buscan conjuntos de atributos con 'support'  $\geq$  al support deseado, llamados 'large itemsets' (conjuntos de atributos grandes). De momento no se busca separarlos en parte izquierda y parte derecha.
- FASE B: ESCLARECIMIENTO DE DEPENDENCIAS (REGLAS). Se hacen particiones binarias y disjuntas de los itemsets y se calcula la confianza de cada uno. Se retienen aquellas reglas que tienen confianza  $\geq$  a la confianza deseada.

Propiedad: cualquier subconjunto de un conjunto grande es también grande.

47

## Métodos Descriptivos

### Algoritmos de búsqueda de asociaciones.

#### FASE A:

#### Método genérico de búsqueda de "LARGE ITEMSETS"

Dado un support mínimo  $s_{\min}$ :

- $i=1$  (tamaño de los conjuntos)
- Generar un conjunto unitario para cada atributo en  $S_i$ .
- Comprobar el support de todos los conjuntos en  $S_i$ . Eliminar aquellos cuyo support  $< s_{\min}$ .
- Combinar los conjuntos en  $S_i$  para crear conjuntos de tamaño  $i+1$  en  $S_{i+1}$ .
- Si  $S_i$  no es vacío entonces  $i:=i+1$ . Ir a 3.
- Si no, retornar  $S_2 \cup S_3 \cup \dots \cup S_i$

Hay refinamientos que permiten una mejor paralelización (dividen en subproblemas con menos tuplas y luego comprueban para todo el problema). El más famoso es el algoritmo "APRIORI" (Agrawal & Srikant 1994).

48

## Métodos Descriptivos

Algoritmos de búsqueda de asociaciones. Ejemplo:

FASE A:

tabla:

Fila	1	2	3	4	5
1	x		x	x	
2		x	x		x
3	x	x	x		x
4		x			x

support = 2

confidence = 0.75

$S_1 = \{ \{1\}, \{2\}, \{3\}, \{4\}, \{5\} \}$

$S'_1: \text{support} = \{ \{1\}:2, \{2\}:3, \{3\}:3, \{5\}:3 \}$

$S_2 = \{ \{1,2\}, \{1,3\}, \{1,5\}, \{2,3\}, \{2,5\}, \{3,5\} \}$

$S'_2: \text{support} = \{ \{1,3\}:2, \{2,3\}:2, \{2,5\}:3, \{3,5\}:2 \}$

$S_3 = \{ \{1,2,3\}, \{1,2,5\}, \{1,3,5\}, \{2,3,5\} \}$

$S'_3: \text{support} = \{ \{2,3,5\}:2 \}$

$S_{\text{final}} = S'_2 \cup S'_3 = \{ \{1,3\}, \{2,3\}, \{2,5\}, \{3,5\}, \{2,3,5\} \}$

FASE B: Se evalúa la confianza:

$\{1\} \rightarrow \{3\} : 1$

$\{3\} \rightarrow \{1\} : 0.67$

$\{2\} \rightarrow \{3\} : 0.67$

$\{3\} \rightarrow \{2\} : 0.67$

$\{2\} \rightarrow \{5\} : 1$

$\{5\} \rightarrow \{2\} : 1$

$\{3\} \rightarrow \{5\} : 0.67$

$\{5\} \rightarrow \{3\} : 0.67$

$\{2,3\} \rightarrow \{5\} : 1$

$\{2,5\} \rightarrow \{3\} : 0.67$

$\{3,5\} \rightarrow \{2\} : 1$

49

## Métodos Descriptivos

Otro Ejemplo:

	VINO "EL CABEZÓN"	GASEOSA "CHISPA"	VINO "TÍO PACO"	HORCHATA "XUFER"	BIZCOCHOS "GOLOSO"	GALLETAS "TRIGO"	CHOCOLATE "LA VACA"
T1	1	1	0	0	0	1	0
T2	0	1	1	0	0	0	0
T3	0	0	0	1	1	1	0
T4	1	1	0	1	1	1	1
T5	0	0	0	0	0	1	0
T6	1	0	0	0	0	1	1
T7	0	1	1	1	1	0	0
T8	0	0	0	1	1	1	1
T9	1	1	0	0	1	0	1
T10	0	1	0	0	1	0	0

50

## Métodos Descriptivos

Ejemplo:

Si definimos la cobertura mínima igual a dos:

- Siete conjuntos de sólo un ítem (siete atributos)
- De los  $7!/5!=42$  posibles casos de conjuntos formados por dos ítems, tenemos 15 conjuntos que superan la cobertura mínima
- 11 conjuntos de tres ítems.
- 2 conjuntos de cuatro ítems.

51

## Métodos Descriptivos

Ejemplo:

- La siguiente fase consiste en la creación de reglas a partir de los conjuntos de ítems frecuentes
- Por ejemplo, si tenemos el conjunto de ítems *horchata "Xufer"* y *bizcochos "Goloso"* y *galletas "Trigo"* se construyen las reglas:

Si *bizcochos "Goloso"* y *horchata "Xufer"* ENTONCES *galletas "Trigo"*      Cb=3, Cf=3/4  
 Si *bizcochos "Goloso"* y *galletas "Trigo"* ENTONCES *horchata "Xufer"*      Cb=3, Cf=3/3  
 Si *galletas "Trigo"* y *horchata "Xufer"* ENTONCES *bizcochos "Goloso"*      Cb=3, Cf=3/3

52

## Métodos Descriptivos

### Otros tipos de asociaciones:

- Asociaciones entre jerarquías. Si existen jerarquías entre los ítems (p.ej. las familias de productos de un comercio o de un supermercado) a veces sólo es interesante buscar asociaciones inter-jerarquía y no intra-jerarquía. Esto puede reducir mucho el espacio de búsqueda.
- Asociaciones negativas. A veces nos interesa conocer asociaciones negativas, p.ej. “80% de los clientes que compran pizzas congeladas no compran lentejas”. El problema es mucho más difícil en general, porque, cuando hay muchos ítems, existen muchas más combinaciones que no se dan que las que se dan.
- Asociaciones con valores no binarios y/o continuos: se deben binarizar. P.ej. Si se tiene un atributo  $a$  con  $k$  posibles valores  $v_1, \dots, v_k$  ( $k > 2$ ) se sustituye por  $k$  atributos con la condición ( $a = v_i$ ). Con los atributos continuos se discretizan en rangos (0-5, 6-10, 11-15, ...) y luego se hace el mismo procedimiento.
- Asociaciones relacionales (Dehaspe and de Raedt 1997b).

53

## Métodos Descriptivos

### Patrones Secuenciales:

Se trata de establecer asociaciones del estilo:

*“si compra  $X$  en  $T$  comprará  $Y$  en  $T+P$ ”*

Ejemplo:

Transaction Database			
Customer	Transaction Time		Purchased Items
John	6/21/97	5:30 pm	Beer
John	6/22/97	10:20 pm	Brandy
Frank	6/20/97	10:15 am	Juice, Coke
Frank	6/20/97	11:50 am	Beer
Frank	6/21/97	9:25 am	Wine, Water, Cider
Mitchell	6/21/97	3:20 pm	Beer, Gin, Cider
Mary	6/20/97	2:30 pm	Beer
Mary	6/21/97	6:17 pm	Wine, Cider
Mary	6/22/97	5:05 pm	Brandy
Robin	6/20/97	11:05 pm	Brandy

54

## Métodos Descriptivos

### Patrones Secuenciales:

Ejemplo (cont.):

Customer Sequence

Customer	Customer Sequences
John	(Beer) (Brandy)
Frank	(Juice, Coke) (Beer) (Wine, Water, Cider)
Mitchell	(Beer, Gin, Cider)
Mary	(Beer) (Wine, Cider) (Brandy)
Robin	(Brandy)

55

## Métodos Descriptivos

### Patrones Secuenciales:

Ejemplo (cont.):

Mining Results

Sequential Patterns with Support $\geq 40\%$	Supporting Customers
(Beer) (Brandy) (Beer) (Wine, Cider)	John, Mary Frank, Mary

56

## Métodos Descriptivos

### Patrones Secuenciales:

#### Métodos Representativos (Agrawal Srikant 1995, 1996)

- *AprioriAll*
- *AprioriSome*
- *DynamicSome*

Problema: los usuarios quieren especificar restricciones sobre el tiempo máximo y mínimo entre eventos secuenciales.

#### Extensiones:

- Minería de patrones secuenciales con restricciones.  
P.ej. Sólo permitir las secuencias si los elementos adyacentes (p.ej. compras) suceden en un intervalo menor a dos meses.

57

## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación):

Se trata de buscar agrupamientos naturales en un conjunto de datos tal que tengan semejanzas.

#### Métodos de Agrupamiento:

- Jerárquicos: los datos se agrupan de manera arborescente (p.ej. el reino animal).
- No jerárquicos: generar particiones a un nivel.
  - (a) Paramétricos: se asumen que las densidades condicionales de los grupos tienen cierta forma paramétrica conocida (p.e. Gaussiana), y se reduce a estimar los parámetros.
  - (b) No paramétricos: no asumen nada sobre el modo en el que se agrupan los objetos.

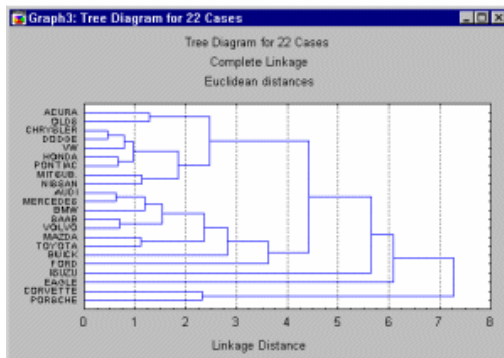
58

## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos jerárquicos:

Un método sencillo consiste en ir separando individuos según su distancia (en concreto medidas derivadas de enlazado, *linkage*) e ir aumentando el límite de distancia para hacer grupos. Esto nos da diferentes agrupaciones a distintos niveles, de una manera jerárquica:

Se denomina  
*Dendograma* o  
*Hierarchical Tree*  
*Plot*.



59

## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos jerárquicos:

#### Minimal Spanning Tree Clustering Algorithm

Algoritmo (dado un número de clusters deseado C).

Inicialmente considera cada ejemplo como un clúster.

- Agrupa el par de clusters más cercanos para formar un nuevo cluster.
- Repite el proceso anterior hasta que el número de clusters = C.

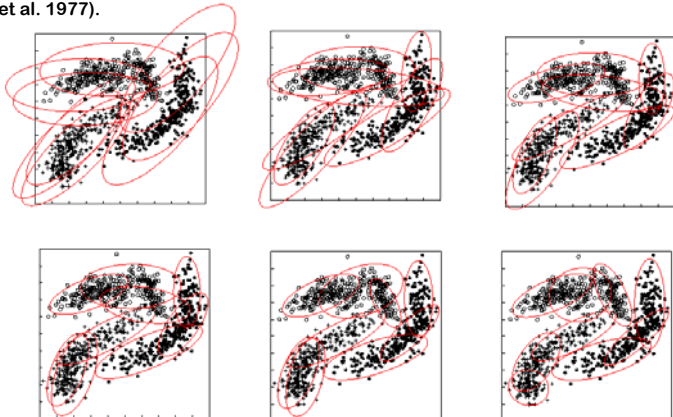
60



## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos paramétricos:

**El algoritmo EM** (Expectation Maximization, Maximum Likelihood Estimate) (Dempster et al. 1977).



Gráficas:  
Enrique Vidal

61

## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos No Paramétricos

**Métodos:**

- $k$ -NN
- $k$ -means clustering,
- online  $k$ -means clustering,
- centroides
- SOM (Self-Organizing Maps) o Redes Kohonen.

**Otros específicos:**

- El algoritmo Cobweb (Fisher 1987).
- El algoritmo AUTOCLASS (Cheeseman & Stutz 1996)

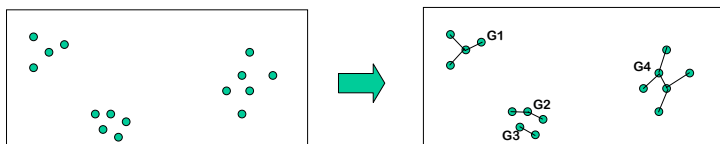
62

## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos No Paramétricos

**1-NN (Nearest Neighbour):**

Dado una serie de ejemplos en un espacio, se conecta cada punto con su punto más cercano:



La conectividad entre puntos genera los grupos.

A veces hace grupos pequeños.

Existen variantes:  $k$ -NN o como el spanning tree que para de agrupar cuando llega a un número de grupos.

63

## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos No Paramétricos

**$k$ -means clustering:**

- Se utiliza para encontrar los  $k$  puntos más densos en un conjunto arbitrario de puntos.
- Algoritmo:
  1. Dividir aleatoriamente los ejemplos en  $k$  conjuntos y calcular la media (el punto medio) de cada conjunto.
  2. Reasignar cada ejemplo al conjunto con el punto medio más cercano.
  3. Calcular los puntos medios de los  $k$  conjuntos.
  4. Repetir los pasos 2 y 3 hasta que los conjuntos no varíen.

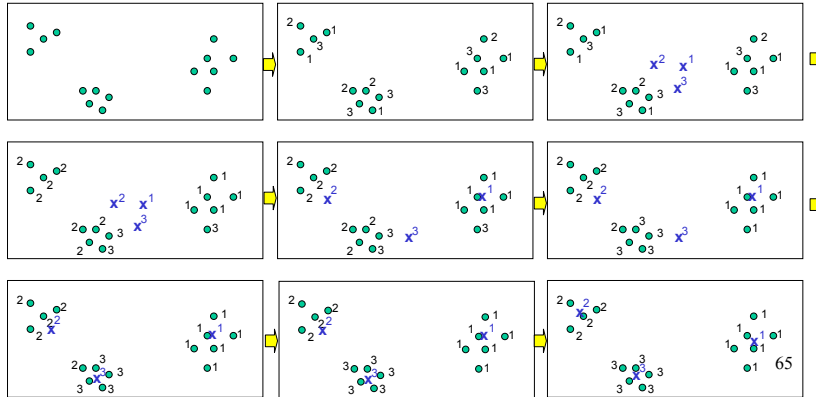
64



## Modelado: Métodos Descriptivos

### Agrupamiento. Métodos No Paramétricos

#### *k*-medias:



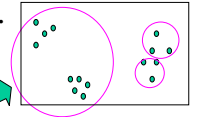
## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos No Paramétricos

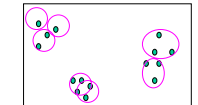
#### *k*-means clustering:

- El valor de *k* se suele determinar heurísticamente.
- Problemas:

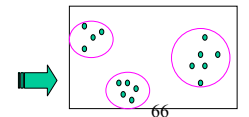
- Si se sabe que hay *n* clases, hacer *k=n* puede resultar en que, algunas veces, algún grupo use dos centros y dos grupos separados tengan que compartir centro.



- Si *k* se elige muy grande, la generalización es pobre y las agrupaciones futuras serán malas.



- Determinar el *k* ideal es difícil.



## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos No Paramétricos

#### *On-line k*-means clustering (competitive learning):

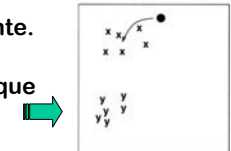
- Refinamiento incremental del anterior.
- Algoritmo:
  1. Inicializar aleatoriamente *k* puntos, llamados *centros*.
  2. Elegir el siguiente ejemplo y ver cuál es el centro más cercano. Mover el centro hacia el ejemplo. (p.ej. Distancia/2)
  3. Repetir el paso 2 para cada ejemplo.
  4. Repetir los pasos 2 y 3 hasta que los ejemplos capturados por cada centro no varíen.

## Métodos Descriptivos Aprendizaje No Supervisado

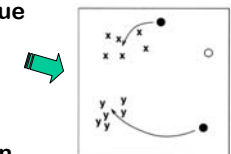
### Clustering (Segmentación). Métodos No Paramétricos

El valor de *k* se suele determinar heurísticamente.

- Problemas:
  - Si *k* se elige muy pequeño, hay grupos que se quedan sin centro.

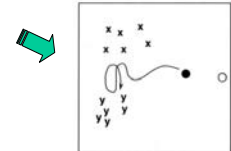


- Si *k* se elige muy grande, hay centros que se quedan huérfanos.



*Aunque esto es preferible a...*

- Incluso con *k* exacto, puede haber algún centro que quede huérfano.



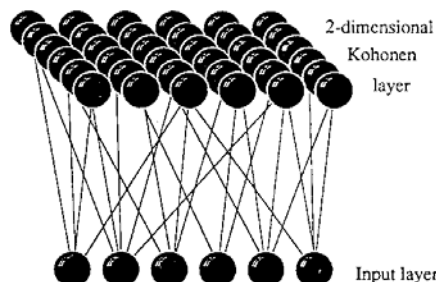
Variación muy popular: LVQ (linear-vector quantization) (Kohonen 1984).

## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos No Paramétricos

#### SOM (Self-Organizing Maps) o Redes Kohonen

- También conocidos como LVQ (linear-vector quantization) o redes de memoria asociativa (Kohonen 1984).



La matriz de neuronas de la última capa forma un grid bidimensional.

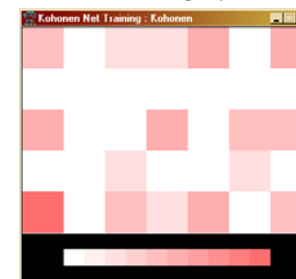
69

## Métodos Descriptivos Aprendizaje No Supervisado

### Clustering (Segmentación). Métodos No Paramétricos

#### SOM (Self-Organizing Maps) o Redes Kohonen

Durante el entrenamiento cada uno de los nodos de este grid compite con los demás para ganar cada uno de los ejemplos. Finalmente los nodos fuertes (representados con colores más oscuros) ganan más ejemplos que los nodos débiles. Al final del aprendizaje la red se estabiliza y sólo unas pocas combinaciones de pares (X,Y) obtienen registros. Estos son los grupos formados.



También puede verse como una red que reduce la dimensionalidad a 2. Por eso es común realizar una representación bidimensional con el resultado de la red para buscar grupos visualmente.

70

## Otros Métodos Descriptivos

### Análisis Estadísticos:

- Estudio de la distribución de los datos.
- Estimación de Densidad
- Detección datos anómalos.
- Análisis de dispersión (p.ej. las funciones de separabilidad pueden considerarse como técnicas muy simples no supervisadas).
- Muchas veces, estos análisis se pueden utilizar previamente para determinar el método más apropiado para un aprendizaje supervisado
- También se utilizan mucho para la limpieza y preparación de datos para el uso de métodos supervisados.

71

### 3. Técnicas de Minería de Datos

#### 3.1. El Problema de la Extracción Automática de Conocimiento.

#### 3.2. Evaluación de Hipótesis

#### 3.3. Técnicas no supervisadas y descriptivas.

#### 3.4. Técnicas supervisadas y predictivas.

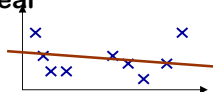
72

## Métodos Predictivos. Interpolación y Predicción Secuencial

### Regresión Lineal Global.

Se buscan los coeficientes de una función lineal

$$\hat{f}(x) = w_0 + w_1 x_1 + \dots + w_n x_n$$



Una manera fácil (si es *lineal simple*, sólo dos dimensiones  $x$  e  $y$ ):

$$w_1 = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \quad w_0 = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

obteniendo  $y = w_0 + w_1 x$

Error típico de una regresión lineal simple:

$$E_{\text{típico}} = \sqrt{\left[ \frac{1}{n(n-2)} \right] \left[ n(\sum y^2) - (\sum y)^2 - \frac{[(\sum xy) - (\sum x)(\sum y)]^2}{n(\sum x^2) - (\sum x)^2} \right]}$$

73

## Métodos Predictivos. Interpolación y Predicción Secuencial

### Regresión Lineal Global por Gradient Descent.

Una manera usual es utilizando "gradient descent".  
Se intenta minimizar la suma de cuadrados:

$$E = \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2$$

Derivando,

$$\Delta w_j = r \cdot \sum_{x \in D} (f(x) - \hat{f}(x)) x_j$$

Iterativamente se van ajustando los coeficientes y reduciendo el error.

74

## Métodos Predictivos. Interpolación y Predicción Secuencial

### Regresión No Lineal.

- Estimación Logarítmica (se sustituye la función a obtener por  $y = \ln(f)$ ):

$$y = w_0 + w_1 x_1 + \dots + w_m x_m$$

- Se hace regresión lineal para calcular los coeficientes y a la hora de predecir se calcula la  $f = e^y$ .

**Regresión Logística.** (variación que se usa para clasificación entre 0 y 1 usando la  $f = \ln(p/(1-p))$ )

### Pick and Mix - Supercharging

- Se añaden dimensiones, combinando las dadas. P.ej. si tenemos cuatro dimensiones:  $x_1, x_2, x_3$  (además de  $y$ ) podemos definir  $x_4 = x_1 \cdot x_2$ ,  $x_5 = x_3^2$ ,  $x_6 = x_1 x_2$  y obtener una función lineal de  $x_1, x_2, x_3, x_4, x_5, x_6$

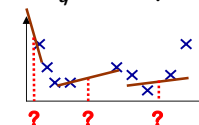
75

## Métodos Predictivos. Interpolación y Predicción Secuencial

### Regresión Lineal Ponderada Localmente.

La función lineal se aproxima para cada punto  $x_q$  a interpolar:

$$\hat{f}(x) = w_0 + w_1 x_1 + \dots + w_m x_m$$



Se intenta minimizar la suma de cuadrados de los  $k$  más cercanos

$$E = \frac{1}{2} \sum_{x \in \{\text{los } k \text{ puntos más cercanos}\}} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$$

donde  $d(\cdot, \cdot)$  es una distancia y  $K$  es una función que disminuye con la distancia (una función Kernel), p.ej.  $1/d^2$

Gradient Descent:

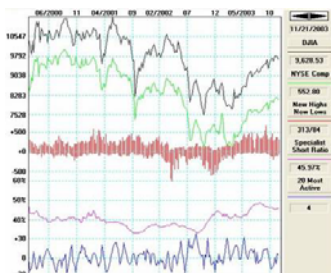
$$\Delta w_j = r \cdot \sum_{x \in \{\text{los } k \text{ puntos más cercanos}\}} (f(x) - \hat{f}(x)) \cdot K(d(x_q, x)) \cdot x_j$$

A mayor  $k$  más global, a menor  $k$  más local (pero ojo con el overfitting)

## Métodos Predictivos. Interpolación y Predicción Secuencial

### Regresión Adaptativa:

- Son casos particulares de regresión local, en el que se supone un orden y se utiliza preferentemente para predecir futuros valores de una serie:
- Muy utilizada en compresión de sonido y de vídeo, en redes, etc. (se predicen las siguientes tramas)



Algoritmos mucho más sofisticados:

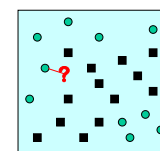
- cadenas de Markov,
- Vector Quantization
- Algoritmo MARS (Multiple Adaptive Regression Splines) (Friedman 1991).

77

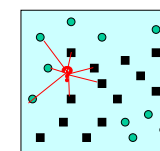
## Métodos Predictivos. Aprendizaje Supervisado

### k-NN (Nearest Neighbour):

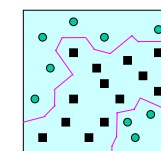
1. Se miran los  $k$  casos más cercanos.
2. Si todos son de la misma clase, el nuevo caso se clasifica en esa clase.
3. Si no, se calcula la distancia media por clase o se asigna a la clase con más elementos.



1-nearest neighbor



7-nearest neighbor



PARTICIÓN DEL 1-nearest neighbor (Poliédrica o de Voronoi)

- El valor de  $k$  se suele determinar heurísticamente.

78

## Aprendizaje Supervisado

**k-NN (Nearest Neighbour).** Mejora (ponderar los más cercanos):

$$\text{Atracción}(c_j, x_q) = |\{x_i : x_i \in c_j\}| \cdot \text{krnl}_i \quad \text{donde: } \text{krnl}_i = \frac{1}{d(x_q, x_i)^2}$$

Se calcula la fuerza de atracción de cada clase  $c_j$  para el nuevo punto  $x_q$ . Y se elige la clase que más atrae.

(Si el punto  $x_q$  coincide con un punto  $x_p$  la clase es la de  $x_p$ )

(Si el punto  $x_q$  coincide con más de un punto  $x_p$  se procede de la forma anterior)

Para valores continuos (sirve para interpolar):

Si la clase es un valor real, el  $k$ -NN es fácilmente adaptable:

$$\hat{f}(x_q) = \frac{\sum_{i=1}^k \text{krnl}_i f(x_i)}{\sum_{i=1}^k \text{krnl}_i}$$

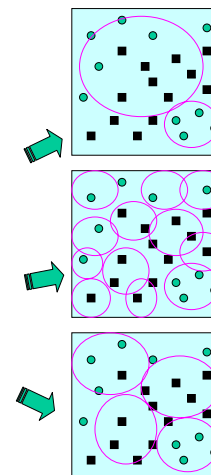
donde los  $x_i$  son los  $k$  vecinos más próximos y  $f(\cdot)$  es la función que da el valor real de cada uno.

79

## Aprendizaje Supervisado

### (On-line) k-means clustering:

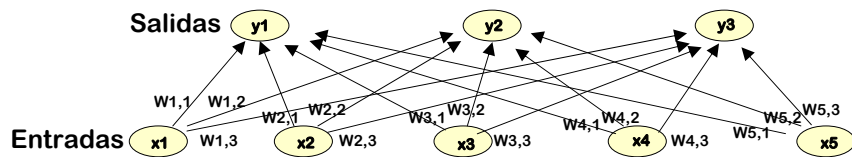
- Aunque lo vimos como una técnica no supervisada, también se puede utilizar para aprendizaje supervisado, si se utiliza convenientemente.
- Elegir un  $k$  mayor que el número de clases pero no mucho mayor.



80

## Aprendizaje Supervisado

### Perceptron Learning.



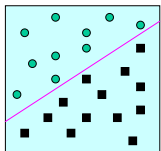
- Computan una función lineal para cada  $y_j$  es:

$$y'_j = \sum_{i=1}^n w_{i,j} \cdot x_i$$

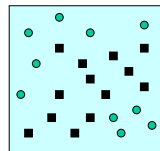
Se añade un threshold escalón:

$$\text{output}_j = \text{sgn}(y'_j)$$

$$\text{sgn}(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si no} \end{cases}$$



PARTICIÓN LINEAL POSIBLE

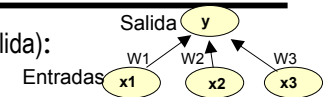


PARTICIÓN LINEAL IMPOSIBLE

81

## Aprendizaje Supervisado

### Gradient Descent (formul. para una sola salida):



- El error de Least Mean Squares de los  $p$  ejemplos se define como:

$$E(\vec{w}) = \frac{1}{2} \sum_{k=1..p} (e_k)^2 = \frac{1}{2} \sum_{k=1..p} (y_k - y'_k)^2$$

- Si queremos disminuir el error poco a poco. El gradiente es la derivada por cada componente del vector.

$$\begin{aligned} \frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{k=1..p} (y_k - y'_k)^2 = \frac{1}{2} \frac{\partial}{\partial w_i} \sum_{k=1..p} (y_k - y'_k)^2 = \frac{1}{2} \sum_{k=1..p} 2(y_k - y'_k) \frac{\partial}{\partial w_i} (y_k - y'_k) = \\ &= \sum_{k=1..p} (y_k - y'_k) \frac{\partial}{\partial w_i} (y_k - \vec{w} \cdot \vec{x}_k) = \sum_{k=1..p} (y_k - y'_k) (-x_{i,k}) \end{aligned}$$

- Queda:

$$\Delta w_i = \sum_{k=1..p} (y_k - y'_k) x_{i,k} = \sum_{k=1..p} x_{i,k} \cdot e_k$$

82

## Aprendizaje Supervisado

### Perceptron Learning (Gradient Descent).

- El algoritmo Gradient Descent ajusta así:
  1. Se asigna a los  $w_{i,j}$  un valor pequeño aleatorio entre 0 y 1.
  2. Hasta que la condición de terminación se cumple, hacer:
  3. Para todos los  $p$  ejemplos  $(x_k, y_k)^t$  se calcula la matriz de error  $(\vec{e}^t = \vec{y}^t - \vec{y}'^t)$
  4. Se recalculan los pesos siguiendo Least-Mean Squares (LMS), con un learning rate ( $r$ ):

$$w_{i,j}^{t+1} = w_{i,j}^t + \sum_{k=1..p} r(x_{i,k}^t \cdot e_{j,k}^t)$$

5.  $t := t+1$ , ir a 2.

$r$  es un valor generalmente pequeño (0.05) y se determina heurísticamente. A mayor  $r$  converge más rápido pero puede perderse en valles locales.

83

## Aprendizaje Supervisado

### Perceptron Learning:

- El algoritmo Perceptron (*versión incremental o aproximación estocástica al gradient descent*):

1. Se asignan aleatoriamente los  $w_{i,j}$  entre 0 y 1 (o se pone .5)
2.  $t = 1$  (se toma el primer ejemplo).
3. Para el ejemplo  $(\vec{x}, \vec{y})^t$  se calcula el vector error  $(\vec{e}^t = \vec{y}^t - \vec{y}'^t)$
4. Se recalculan los pesos siguiendo Least-Mean Squares (LMS), también llamada regla delta, Adaline o *Widrow-Hoff*:

$$w_{i,j}^{t+1} = w_{i,j}^t + r(x_{i,j}^t \cdot e_j^t)$$

5.  $t := t+1$ , ir a 2 hasta que no queden ejemplos o el error medio se ha reducido a un valor deseado.

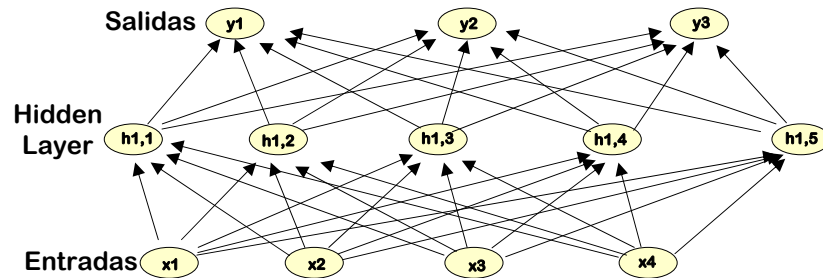
En general, esta versión es más eficiente que la anterior y evita algunos mínimos locales.

84

## Aprendizaje Supervisado

### Multilayer Perceptron (redes neuronales artificiales, ANN).

- El perceptron de una capa no es capaz de aprender las funciones más sencillas.
- Se añaden capas internas, se introducen diferentes funciones de activación e incluso se introducen bucles y retardos.

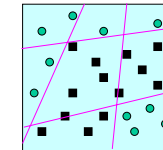


85

## Aprendizaje Supervisado

### Multilayer Perceptron (redes neuronales artificiales, ANN).

- En el caso más sencillo, con la función de activación sgn, el número de unidades internas  $k$  define exactamente el número de *boundaries* que la función global puede calcular por cada salida.



PARTICIÓN POLIGONAL  
POSIBLE CON 4  
UNIDADES INTERNAS

- El valor de  $k$  se suele determinar heurísticamente.
- Pero, ¿cómo entrenar este tipo de red?

86

## Aprendizaje Supervisado

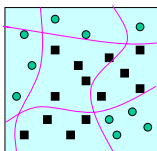
### Multilayer Perceptron (redes neuronales artificiales, ANN).

- Para poder extender el gradient descent necesitamos una función de activación continua:
- La más usual es la función sigmoideal:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



- Esto permite particiones no lineales:



PARTICIÓN NO LINEAL  
MÚLTIPLE POSIBLE  
CON 4 UNIDADES  
INTERNAS

87

## Aprendizaje Supervisado

### Algoritmo Backpropagation (Rumelhart et al. 1986)

- Inicializar los pesos a valores pequeños aleatorios (entre  $-0.05$  y  $0.05$ )
- Hasta que se cumpla la condición de terminación hacer:
  - Para cada ejemplo  $(\vec{x}, \vec{y})$ :
    - Se calculan las salidas de todas las unidades  $o_u$
    - Se calcula el error en cada salida  $k$ :

$$\delta_k = o_k(1 - o_k)(y_k - o_k)$$

- Para cada unidad oculta  $h$  se calcula su error:

$$\delta_h = o_h(1 - o_h) \sum_{k \in \text{outputs}} (w_{k,h} \times \delta_k)$$

- Se actualizan los pesos:

$$w_{j,i} = w_{j,i} + r \times \delta_j \times x_{j,i}$$

Se necesitan muchos ejemplos: al menos 10 ejemplos por cada peso y output a aprender. P.ej, una red con 50 entradas y 10 nodos internos, necesita 10.220 ejemplos por lo menos.

88



## Aprendizaje Supervisado

### Variaciones:

- Si hay más de una capa oculta:

$$\delta_h = o_h(1 - o_h) \sum_{k \in \text{outputs\_of\_next\_layer}} (w_{k,h} \times \delta_k)$$

- Si la red es no recursiva, pero no está organizada en capas (se trata de cualquier árbol acíclico), también se puede:

$$\delta_h = o_h(1 - o_h) \sum_{k \in \text{downstream}(h)} (w_{k,h} \times \delta_k)$$

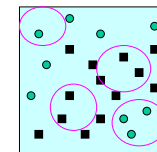
- Existe una variante que va añadiendo capas según se necesitan, denominado cascade-correlation (Fahlman and Lebiere 1990), resolviendo el problema de determinar el número de unidades ocultas.

89

## Aprendizaje Supervisado

### Radial-Basis Function (Clustering Method + LMS).

- PRIMER PASO: Algoritmo Clustering:
  1. Dividir aleatoriamente los ejemplos en  $k$  conjuntos y calcular la media (el punto medio) de cada conjunto.
  2. Reasignar cada ejemplo al cto. con punto medio más cercano.
  3. Calcular los puntos medios de los  $k$  conjuntos.
  4. Repetir los pasos 2 y 3 hasta que los conjuntos no varíen.
- SEGUNDO PASO: Recodificar los ejemplos como distancias a los centros y normalizar (cada ejemplo pasa a ser un vector de  $k$  eltos).
- TERCER PASO: Con un perceptrón de  $k$  elementos de entrada y una salida, aplicar el algoritmo visto antes.



PARTICIÓN  
HIPERESFÉRICA  
CON 4 centros.

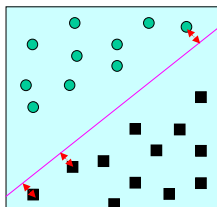
Se convierte en una partición lineal (hiperplano) en un espacio de 4 dimensiones con los ejemplos siendo las distancias a los centros.

90

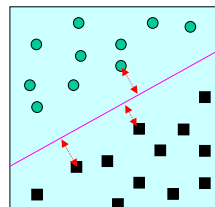
## Aprendizaje Supervisado

### Máquinas de vectores soporte (métodos basados en núcleo).

- Se basan en un clasificador lineal muy sencillo (el que maximiza la distancia de los tres ejemplos (vectores) soporte), precedido de una transformación de espacio (a través de un núcleo) para darle potencia expresiva.
- El clasificador lineal que se usa simplemente saca la línea (en más dimensiones, el hiperplano) que divida limpiamente las dos clases y además que los tres ejemplos más próximos a la frontera estén lo más distantes posibles.



Separa perfectamente, pero los tres ejemplos más cercanos (vectores soporte) están muy cerca de la frontera.



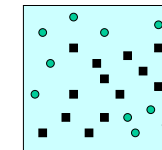
Separa perfectamente, pero además los ejemplos más cercanos (vectores soporte) están lo más lejos posible de la frontera.

91

## Aprendizaje Supervisado

### Máquinas de vectores soporte (métodos basados en núcleo).

- Son eficientes (incluso para cientos de dimensiones), pues el separador lineal sólo tiene que mirar unos pocos puntos (vectores soporte) y puede descartar muchos que estarán lejos de la frontera.
- ¿Pero qué ocurre si los datos no son separables linealmente?



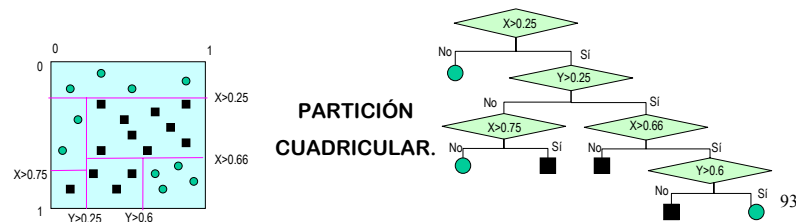
- Se aplica una función núcleo ("kernel") que suele aumentar el número de dimensiones de tal manera que los datos sean separables.

92

## Aprendizaje Supervisado

### Árboles de Decisión (ID3 (Quinlan), C4.5 (Quinlan), CART).

- Algoritmo Divide y Vencerás:
  - Se crea un nodo raíz con  $S :=$  todos los ejemplos.
  - Si todos los elementos de  $S$  son de la misma clase, el subárbol se cierra. Solución encontrada.
  - Se elige una condición de partición siguiendo un criterio de partición (split criterion).
  - El problema (y  $S$ ) queda subdividido en dos subárboles (los que cumplen la condición y los que no) y se vuelve a 2 para cada uno de los dos subárboles.



## Aprendizaje Supervisado

### Árboles de Decisión.

- Ejemplo C4.5 con datos discretos:

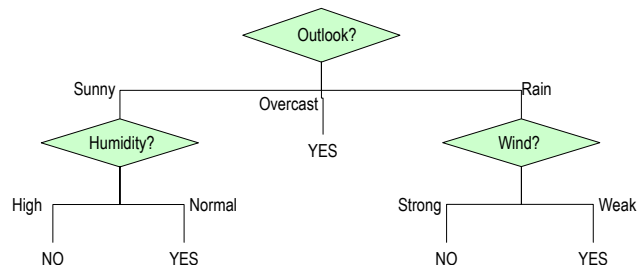
Example	Sky	Temperature	Humidity	Wind	PlayTennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

94

## Aprendizaje Supervisado

### Árboles de Decisión.

- Ejemplo C4.5 con datos discretos:



- Representación Lógica:  
 (Outlook=Sunny AND Humidity=Normal) OR (Outlook=Overcast) OR  
 (Outlook=Rain AND Wind=Weak)

P.ej., la instancia (Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong) es NO.

## Aprendizaje Supervisado

### Árboles de Decisión (ID3, C4.5, CART).

- El criterio GANANCIA DE INFORMACIÓN (C4.5) ha dado muy buenos resultados. Suele derivar en una preferencia en árboles pequeños (navaja de Occam).
- VENTAJAS:**
  - Muy fácil de entender y de visualizar el resultado.
  - Son robustos al ruido. Existen algoritmos de *post-pruning* para podar hojas poco significativas (que sólo cubren uno o muy pocos ejemplos).
- DESVENTAJAS:**
  - Suele ser muy voraz (no hace backtracking: mínimos locales).
  - Si el criterio de partición no está bien elegido, las particiones suelen ser muy ad-hoc y generalizan poco.

96



## Aprendizaje Supervisado

### Naive Bayes Classifiers.

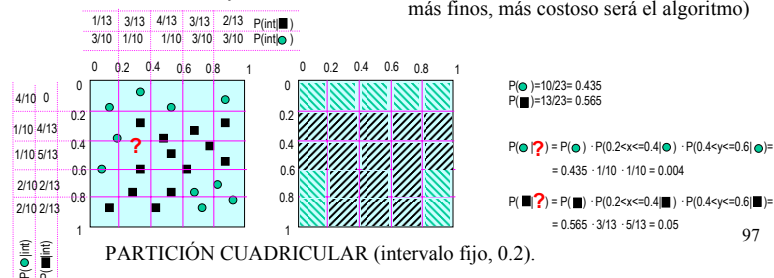
$$\arg \max_{c_i \in C} P(c_i | (x_1, x_2, \dots, x_m)) \stackrel{\text{Bayes}}{=} \arg \max_{c_i \in C} \frac{P((x_1, x_2, \dots, x_m) | c_i) \cdot P(c_i)}{P(x_1, x_2, \dots, x_m)}$$

$$= \arg \max_{c_i \in C} P((x_1, x_2, \dots, x_m) | c_i) \cdot P(c_i)$$

- Asumiendo independencia entre los atributos, tenemos:

$$V_{NB} = \arg \max_{c_i \in C} P(c_i) \prod_j P(x_j | c_i)$$

Si estos  $x_j$  son continuos podemos discretizar en intervalos y calcular  $P(c_i | t_k < x_j \leq t_{k+1})$  (cuanto más finos, más costoso será el algoritmo)



## Aprendizaje Supervisado

### Naive Bayes Classifiers.

- Otra manera es hacer los intervalos continuos y calcular la frecuencia acumulada  $f(c_i / x_j \leq t)$ . Tenemos  $f(c_i / s < x_j \leq t) = f(c_i / x_j \leq t) - f(c_i / x_j \leq s)$ .
- Se puede fijar un radio  $r$ .
- O podemos utilizar una función de densidad

$$p(x_0) = \lim_{\varepsilon \rightarrow \infty} \frac{1}{\varepsilon} P(x_0 \leq x < x_0 + \varepsilon)$$

- Así las particiones son más ajustadas.
- En el último caso (función de densidad), a partir del Maximum Likelihood obtendríamos la hipótesis Least-Squared Error:

$$h_{ML} = \arg \max_{h \in H} p(D | h) = \dots = \arg \min_{h \in H} \sum_{i=1 \dots m} (d_i - h(x_i))^2$$

donde  $d_i$  representa el dato  $i$ .

98

## Aprendizaje Supervisado

### Naive Bayes Classifiers.

- Se utilizan más con variables discretas. Ejemplo del playtennis:
- Queremos clasificar una nueva instancia:  
(Outlook = sunny, Temperature = cool, Humidity = high, Wind = strong)

$$V_{NB} = \arg \max_{c_i \in \{yes, no\}} P(c_i) \prod_j P(x_j | c_i) =$$

$$= \arg \max_{c_i \in \{yes, no\}} P(c_i) \cdot P(Outlook = sunny | c_i) \cdot P(Temperature = cool | c_i) \cdot P(Humidity = high | c_i) \cdot P(Wind = strong | c_i)$$

- Estimando las 10 probabilidades necesarias:

$$P(Playtennis=yes)=9/14=.64, \quad P(Playtennis=no)=5/14=.36$$

$$P(Wind=strong|Playtennis=yes)=3/9=.33$$

$$P(Wind=strong|Playtennis=no)=3/5=.60$$

- Tenemos que:

$$P(yes)P(sunny|yes)P(cool|yes)P(high|yes)P(strong|yes)=0.0053$$

$$P(no)P(sunny|no)P(cool|no)P(high|no)P(strong|no)=0.206$$

99

## Aprendizaje Supervisado

### Naive Bayes Classifiers. $m$ -estimate.

- Generalmente, si hay pocos datos, es posible que alguna probabilidad condicional sea 0 (p.ej.  $P(\text{water=cool} | \text{enjoysport=no})$ ), porque no ha aparecido un determinado valor de un atributo para una cierta clase.
- Para evitar este problema se utiliza un  $m$ -estimado de la probabilidad:

$$\frac{n_c + mp}{n + m}$$

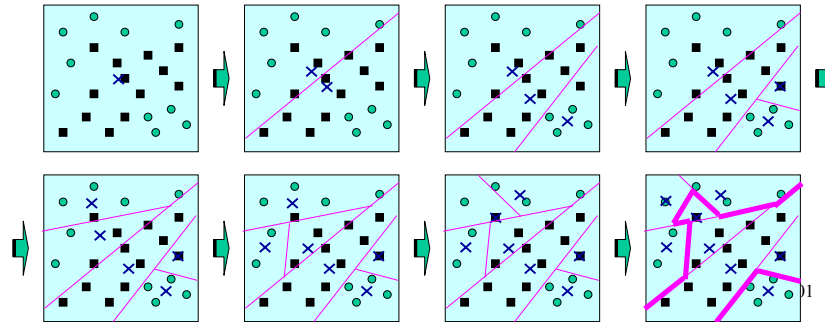
- donde  $n$  son los casos de la clase considerada,  $n_c$  son los casos de esta clase en los que el atributo considerado toma un cierto valor,  $m$  es una constante denominada "tamaño equivalente de muestra" y  $p$  es la probabilidad de cada valor del atributo a priori.
- Generalmente  $p$  se escoge uniformemente, es decir, si hay  $k$  valores posibles,  $p = 1/k$ .
- El valor de  $m$  se escoge lo más pequeño posible (1 a 10) para no interferir en la proporción observada ( $n_c/n$ ).

100

## Aprendizaje Supervisado

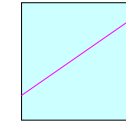
Center Splitting (es un híbrido LVQ/C4.5).

1. Inicializar el primer centro en la media de los ejemplos.
2. Asignar todos los ejemplos a su centro más cercano.
3. Si hay algún centro que tiene ejemplos de diferente clase, borrar el centro y crear tantos nuevos centros distintos como clases haya, cada uno siendo la media de los ejemplos de la clase. Ir a 2.

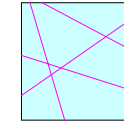


## Aprendizaje Supervisado

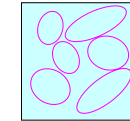
Comparación de representación:



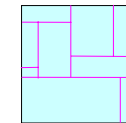
Perceptron / LMS



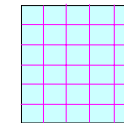
Redes  
Neuronales  
Multicapa



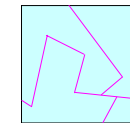
RBF



C4.5/ID3/CART



Naive Bayes  
Classifier



k-NN, LVQ, CS

## Aprendizaje Supervisado

Comparación de métodos no relacionales:

- **k-NN:**
  - Muy fácil de usar
  - Eficiente si el nº de ejemplos no es excesivamente grande.
  - Gran expresividad de la partición.
  - Inteligible sólo visualmente.
  - Robusto al ruido pero no a atributos no significativos (las distancias aumentan, conocido como "the curse of dimensionality")
- **RBF**  
(combinaciones de k-means clustering + perceptron):
  - Preferibles a cualquiera de las dos técnicas por separado.
  - Difícil de ajustar el  $k$ .
  - Poca inteligibilidad.
- **Redes neuronales (multicapa):**
  - El número de capas y elementos por capa difíciles de ajustar.
  - Apropiado para clasificación o regresión.
  - Poca inteligibilidad.
  - Muy sensibles a outliers (datos anómalos).
  - Se necesitan muchos ejemplos ( $\sqrt{n}$ ).

## Aprendizaje Supervisado

Comparación de métodos no relacionales (cont.):

- **Naive Bayes:**
  - Muy fácil de usar.
  - Muy eficiente.
  - NO HAY MODELO.
  - Robusto al ruido.
- **Árboles de decisión: (C4.5):**
  - Muy fácil de usar.
  - Admite atributos discretos y continuos.
  - La clase debe ser discreta y finita. (aunque tb. existen los árboles de regresión que permiten clase continua)
  - Es tolerante al ruido, a atributos no significativos y a *missing attribute values*.
  - Alta inteligibilidad.
- **SVM:**
  - Muy eficientes cuando hay muchos atributos (no sufre el "curse of dimensionality").
  - Hay que saber elegir el kernel adecuadamente
  - No inteligible.

## Aprendizaje Supervisado

Comparación de *accuracy* (k-NN, C4.5 y CS) (de Thornton 2000):

Dataset (del UCI repository)	C4.5	1-NN	CS
BC (Breast Cancer)	72.0	67.31	70.6
CH (chess endgames)	99.2	82.82	89.9
GL (glass)	63.2	73.6	67.19
G2 (GL con clases 1 y 3 combinadas, y 4 a 7 borradas)	74.3	81.6	78.87
HD (heart disease)	73.6	76.24	78.77
HE (hepatitis)	81.2	61.94	62.62
HO (horse colic)	83.6	76.9	77.73
HY (hypothyroid)	99.1	97.76	96.1
IR (iris)	93.8	94.0	95.76
LA (labor negotiations)	77.2	94.74	90.7
LY (lymphography)	77.5	77.03	79.4
MU (agaricus-lepiota)	100.0	100.0	100.0
SE (sick-euthyroid)	97.7	93.19	91.3
SO (soybean-small)	97.5	100.0	99.38
VO (house votes, 1984)	95.6	92.87	92.59
V1 (VO con "physician fee freeze" borrado)	89.4	87.47	89.46
Media:	85.9	84.8	85

105

## Aprendizaje Supervisado. Sobremuestreo

### Sobremuestreo (oversampling):

- En problemas de clasificación sobre bases de datos es posible que haya muchísima más proporción de algunas clases sobre otras. Esto puede ocasionar que haya muy pocos casos de una clase:
- Problema: la clase escasa se puede tomar como ruido y ser ignorada por la teoría. Ejemplo: si un problema binario (*yes / no*) sólo hay un 1% de ejemplos de la clase *no*, la teoría "todo es de la clase *yes*" tendría un 99% de precisión (*accuracy*).

#### Soluciones:

- Utilizar sobremuestreo...
- Análisis ROC

106

## Aprendizaje Supervisado. Sobremuestreo

### Sobremuestreo (oversampling / balancing):

- El sobremuestreo consiste en filtrar los ejemplos (tuplas) de las clases con mayor proporción, manteniendo las tuplas de las clases con menor proporción.
- Esto, evidentemente, cambia la proporción de las clases, pero permite aprovechar a fondo los ejemplos de las clases más raras.

#### ¿Cuándo se debe usar sobremuestreo?

- Cuando una clase es muy extraña: p.ej. predecir fallos de máquinas, anomalías, excepciones, etc.
- Cuando todas las clases (especialmente las escasas) deben ser validadas. P.ej. si la clase escasa es la de los clientes fraudulentos.
- Pegas: hay que ser muy cuidadoso a la hora de evaluar los modelos.

107

## Aprendizaje Supervisado. Macro-average

### Macro-average:

- Una alternativa al sobremuestreo consiste en calcular la precisión de una manera diferente.
- Habitualmente, la precisión (*accuracy*) se calcula:

$$acc(h) = \text{aciertos} / \text{total}$$

(conocido como *micro-averaged accuracy*)

- La alternativa es calcular la precisión como:

$$acc(h) = \frac{\text{aciertos}_{\text{clase1}} / \text{total}_{\text{clase1}} + \text{aciertos}_{\text{clase2}} / \text{total}_{\text{clase2}} + \dots + \text{aciertos}_{\text{clase-n}} / \text{total}_{\text{clase-n}}}{n^{\circ} \text{clases}}$$

(conocido como *macro-averaged accuracy*)

De esta manera se obtiene un resultado mucho más compensado

108

## Aprendizaje Supervisado. Matrices de Coste y Confusión.

### Errores de Clasificación (confusión de clases) :

- En muchos casos de minería de datos, el error de clasificación sobre una clase no tiene las mismas consecuencias económicas, éticas o humanas que con otras.
- Ejemplo: clasificar una partida de neumáticos en perfectas condiciones como defectuoso o viceversa.

109

## Aprendizaje Supervisado. Matrices de Coste y Confusión.

### Matrices de Confusión y Coste:

- Existen técnicas para ponderar las clases → se combinan las “matrices de confusión” con las “matrices de costes”:

COST		actual		
		low	medium	high
predicted	low	0€	5€	2€
	medium	200€	-2000€	10€
	high	10€	1€	-15€

ERROR		actual		
		low	medium	high
predicted	low	20	0	13
	medium	5	15	4
	high	4	7	60

Coste  
total:  
-29787€

110

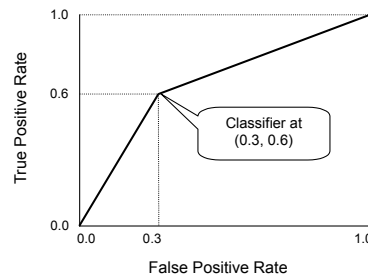
## Aprendizaje Supervisado. Análisis ROC.

### Análisis ROC (Receiver Operating Characteristic):

- Se basa en dibujar el “true-positive rate” en el eje  $y$  y el “false-positive rate” en el eje  $x$ . Por ejemplo, dada la siguiente matriz de confusión:

		Actual	
		T	F
Predicted	T	30	30
	F	20	70

- Tendríamos  $TPR = 0.6$  y  $FPR = 0.3$ .



111

## Aprendizaje Supervisado. Mailings.

### Mailings:

- Existen técnicas específicas para evaluar la conveniencia de campañas de ‘mailings’ (propaganda por correo selectiva):
  - EJEMPLO: Supongamos que una empresa de venta de productos informáticos por catálogo posee una base de datos de clientes. Esta empresa desea promocionar la venta de un nuevo producto: un mando de piloto para ser utilizado en programas de simulación de vuelo.
- Podríamos enviar propaganda a todos sus clientes:
  - Solución poco rentable
- Podemos utilizar técnicas de aprendizaje automático para poder predecir la respuesta de un determinado cliente al envío de la propaganda y utilizar esta información para optimizar el diseño de la campaña.

112

## Aprendizaje Supervisado. Mailings.

- Mailings:

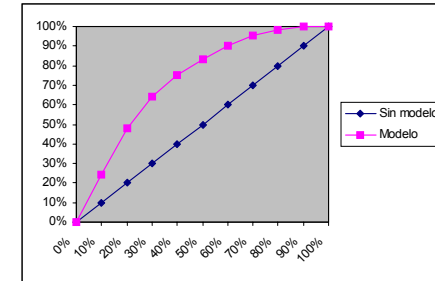
1. Selección de una muestra aleatoria y suficientemente numerosa de clientes
2. Se realiza el envío de la propaganda a los clientes seleccionados
3. Una vez pasado un tiempo prudencial etiquetamos a los clientes de la muestra: 1 ha comprado el producto, 0 no ha comprado el producto
4. Con la muestra etiquetada aprendemos un clasificador probabilístico
  - Asigna a cada ejemplo (cliente) no la clase predicha, sino una estimación de la probabilidad de respuesta de ese cliente

113

## Aprendizaje Supervisado. Mailings.

- Mailings:

- Con el clasificador probabilístico podemos ordenar a los clientes según su interés y dibujar un gráfico de respuesta acumulada

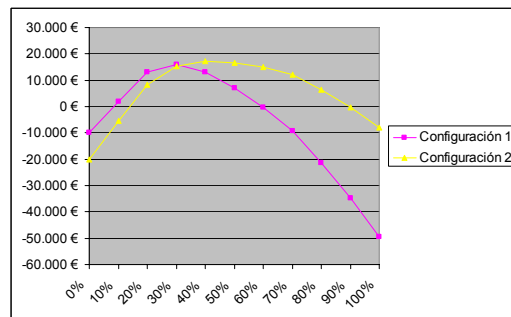


- Nos indican qué porcentaje de las posibles respuestas vamos a obtener dependiendo del porcentaje de envíos que realicemos sobre la población total

114

## Aprendizaje Supervisado. Mailings.

- Además si estimamos la matriz de coste, podemos conocer la configuración óptima mediante los gráficos de beneficio
  - Configuración 1: Coste inicial de la campaña 10.000€, coste de envío de cada folleto 1,5€. Por cada producto vendido ganamos 3€
  - Configuración 2: Coste inicial de la campaña 20.000€, coste de envío de cada folleto 0,8€. Por cada producto vendido ganamos 2,5€



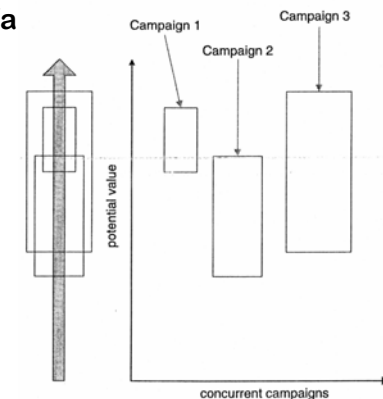
115

## Aprendizaje Supervisado. Mailings.

### Secuenciación de Mailings:

- No sobrecargar los clientes con demasiados mensajes de marketing... O bien acabarán ignorándolos o bien se cambiarán de compañía

El mismo pequeño grupo de gente se elige una y otra vez y otros no se eligen nunca.



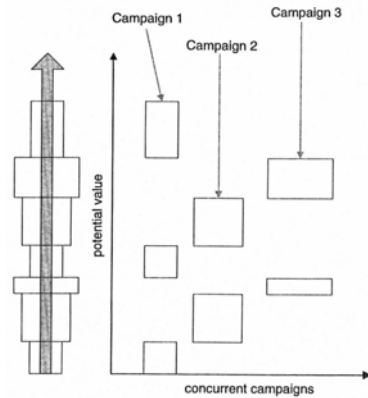
116

## Aprendizaje Supervisado. Mailings.

### Secuenciación de Mailings:

- Hay que intentar abarcar mejor los clientes:

Ahora todos los clientes participan en una campaña.



117

## Métodos Predictivos Combinación de Hipótesis

### Combinación de Hipótesis:

- VOTING/ARBITER/COMBINER:**
  - Se utiliza **DISTINTOS** algoritmos para aprender distintas hipótesis sobre todo el conjunto de los datos.
  - Luego se *combinan* las distintas hipótesis.
- Maneras de **COMBINAR** hipótesis:
  - WEIGHTING MAJORITY:** el valor se obtiene haciendo la media (caso continuo) o la mediana (caso discreto).
  - STACKING/CASCADE:** se utiliza cada hipótesis como una variable y se utiliza otro algoritmo (p.ej. una red neuronal para asignar diferentes pesos a las diferentes hipótesis).

118

## Métodos Predictivos Combinación de Hipótesis

### Voting y el Clasificador Bayesiano Óptimo:

- Una pregunta es: “Qué hipótesis es más probable?”
- Otra pregunta es: “Qué predicción es más probable?”
- Consideremos una evidencia  $D$  y tres hipótesis  $h_1$ ,  $h_2$  y  $h_3$ , cuyas probabilidades a posteriori se han calculado:
 
$$P(h_1 | D) = 0.4, \quad P(h_2 | D) = 0.3, \quad P(h_3 | D) = 0.3$$
- Para la próxima observación  $h_1$  la clasifica como positiva (true), mientras que  $h_2$  y  $h_3$  la clasifican como negativa (false).
- Según MAP y suponiendo  $P(h)$  uniforme, la mejor hipótesis es  $h_1$  y la nueva observación debería clasificarse como positiva...
  - Sin embargo...

La mejor clasificación de una nueva instancia se obtiene combinando las predicciones de las distintas hipótesis consideradas (*voting*).

119

## Métodos Predictivos Combinación de Hipótesis

### Voting y el Clasificador Bayesiano Óptimo:

- Justificación:
 
$$P(v_j | D) = \sum_{h_i \in H} P(v_j | h_i) P(h_i | D)$$
- Para el ejemplo anterior:
 
$$P(h_1 | D) = 0.4, \quad P(h_2 | D) = 0.3, \quad P(h_3 | D) = 0.3$$

$$P(\text{false} | h_1) = 0, \quad P(\text{false} | h_2) = 1, \quad P(\text{false} | h_3) = 1$$

$$P(\text{true} | h_1) = 1, \quad P(\text{true} | h_2) = 0, \quad P(\text{true} | h_3) = 0$$
- Por tanto:

$$P(\text{false} | D) = \sum_{h_i \in H} P(\text{false} | h_i) P(h_i | D) = 0.6$$

$$P(\text{true} | D) = \sum_{h_i \in H} P(\text{true} | h_i) P(h_i | D) = 0.4$$

120

## Métodos Predictivos

### Combinación de Hipótesis

---

#### Potenciación mediante Combinación de Hipótesis:

- **BOOSTING:** Se utiliza el MISMO algoritmo para aprender distintas hipótesis sobre los mismos datos, aumentando el peso de aquellos ejemplos que han sido clasificados incorrectamente. Luego se *combinan* las distintas hipótesis.
- **BAGGING:** Se utiliza el MISMO algoritmo para aprender distintas hipótesis sobre  $n$  muestras de  $m$  de los  $m$  datos con reemplazamiento (bootstrap). Luego se *combinan* las distintas hipótesis.
- **RANDOMISING:** Se utiliza el MISMO algoritmo para aprender distintas hipótesis sobre los mismos datos, pero variando aleatoriamente alguna característica del algoritmo (restringiendo los atributos que se usan cada vez, variando el criterio de selección, etc.). Luego se *combinan* las distintas hipótesis. 121

## Métodos Predictivos

### Combinación de Hipótesis

---

#### Ejemplo: Boosting (reiteración):

- *A veces unos malos resultados se pueden mejorar mediante la técnica de BOOSTING:*

*Se da peso a los ejemplos y para cada iteración se aprende una nueva hipótesis, siendo los ejemplos reponderados para que el sistema se centre en los ejemplos que han sido mal clasificados.*

- Algoritmo más simple:
  - Dar a todos los ejemplos el mismo peso.
  - De  $i:1$  hasta  $T$  (nº de reiteraciones)
    - Aprender una hipótesis  $h_i$  con los pesos actuales.
    - Disminuir los pesos de los ejemplos que  $h_i$  clasifica correctamente.
  - Después se promedia una solución ponderada a partir de las  $T$  hipótesis, teniendo en cuenta la precisión de cada una. 122