

Unidad 3: Minería de Datos

Temario

1.- Proceso de Descubrimiento del Conocimiento
(KDD)

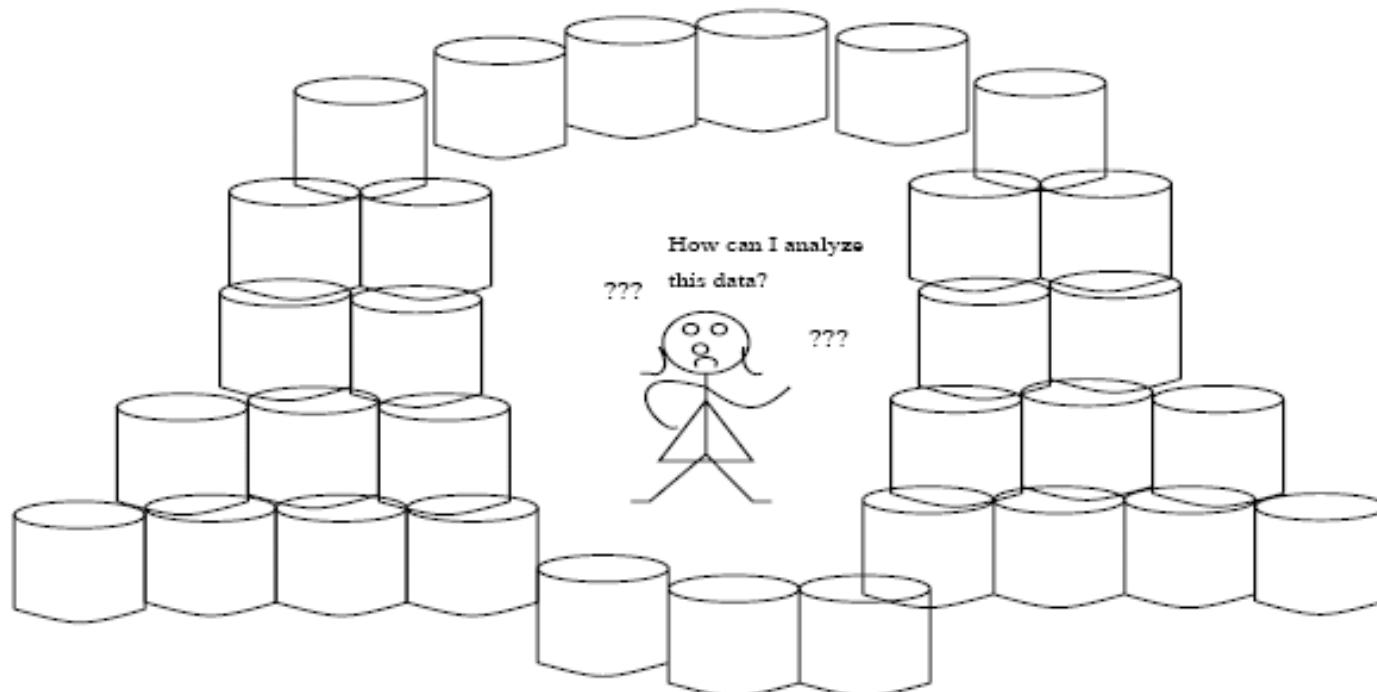
2.- Tareas y Métodos de la Minería de Datos

Proceso KDD

- Es el proceso de usar la base de datos en conjunto con cualquier selección, proprocesamiento, *sub-muestreo*, y transformaciones de ella; para aplicar métodos de minería de dato (algoritmos) y enumerar patrones desde ella; y para evaluar los productos de la minería de datos que identifican el subconjunto de patrones enumerados que llegarán a ser el “conocimiento”.
- El descubrimiento de conocimiento puede ser:
 - de Predicción: patrones para predecir comportamientos futuros.
 - de Descripción: patrones para explicar lo que sucede en un formato entendible por el ser humano.

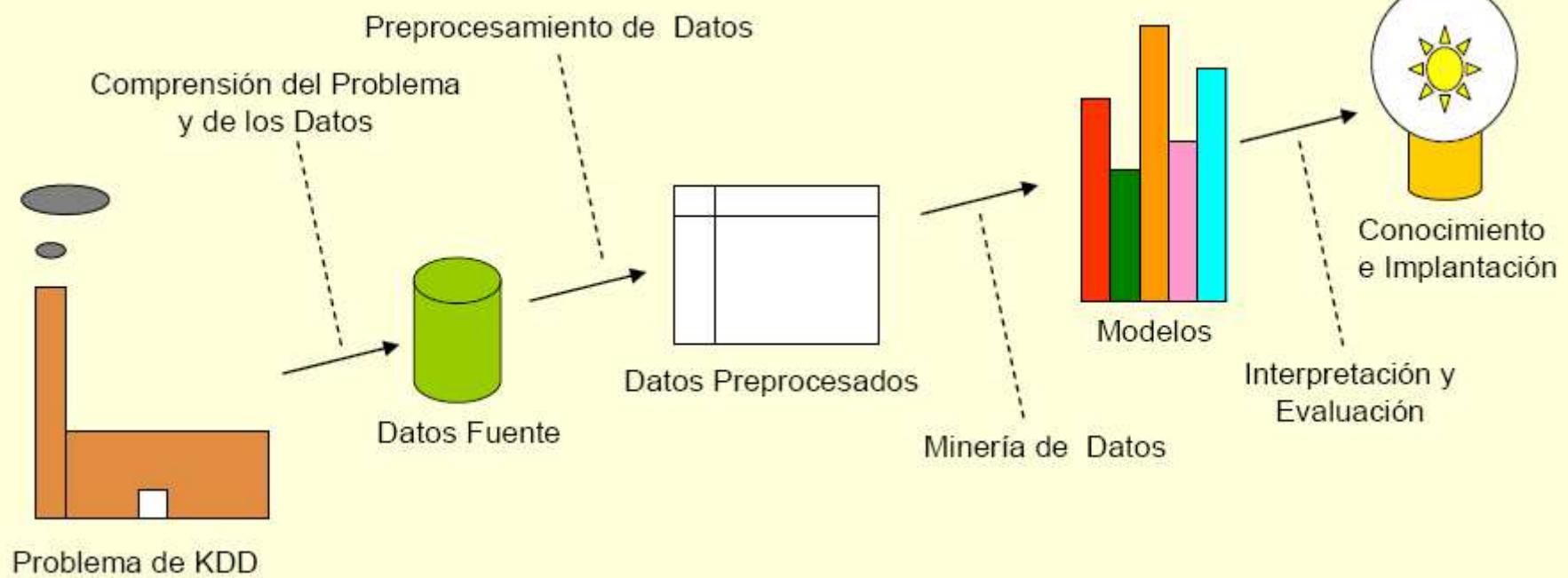
Proceso KDD

Antes de aplicarlo



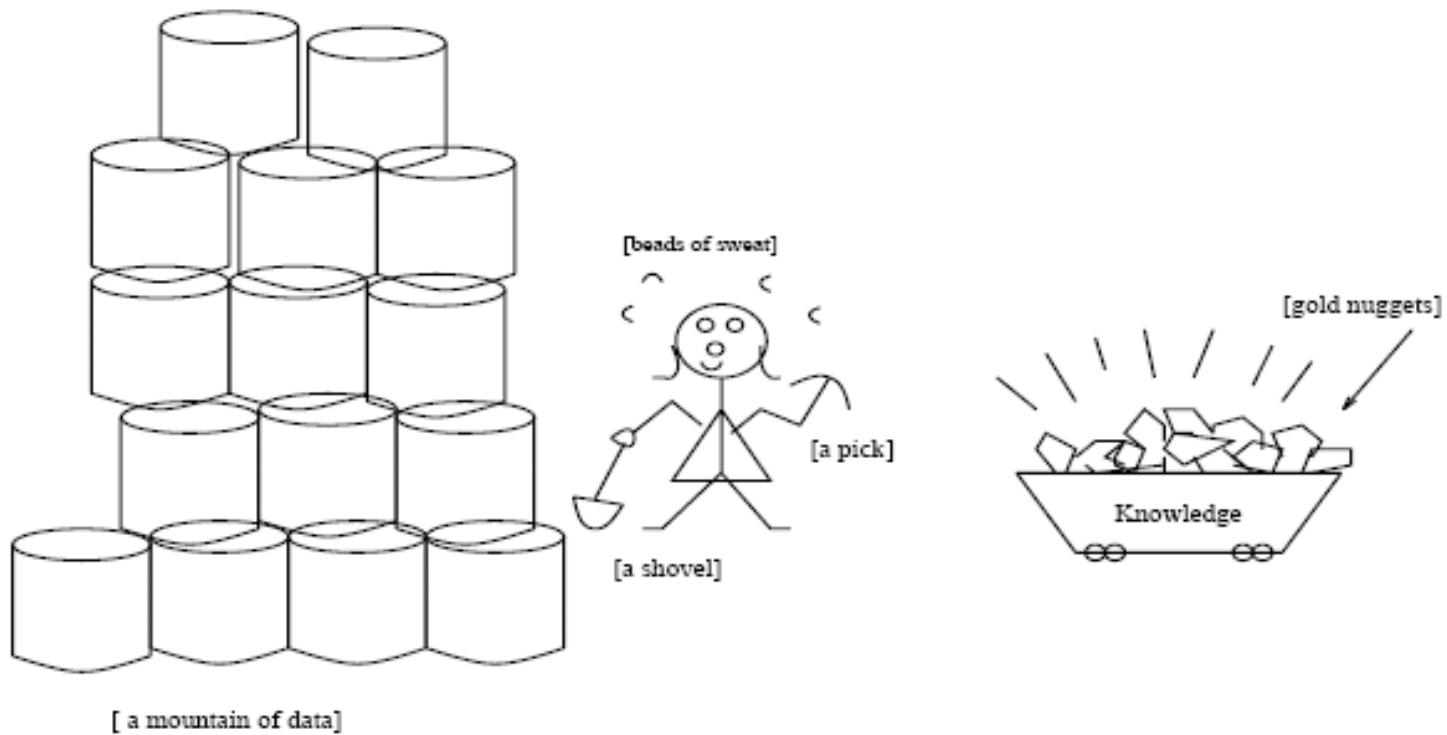
Proceso KDD

Etapas de su aplicación



Proceso KDD

Después de aplicarlo



Temario

1.- Proceso de Descubrimiento del Conocimiento
(KDD)

2.- Tareas y Métodos de la Minería de Datos

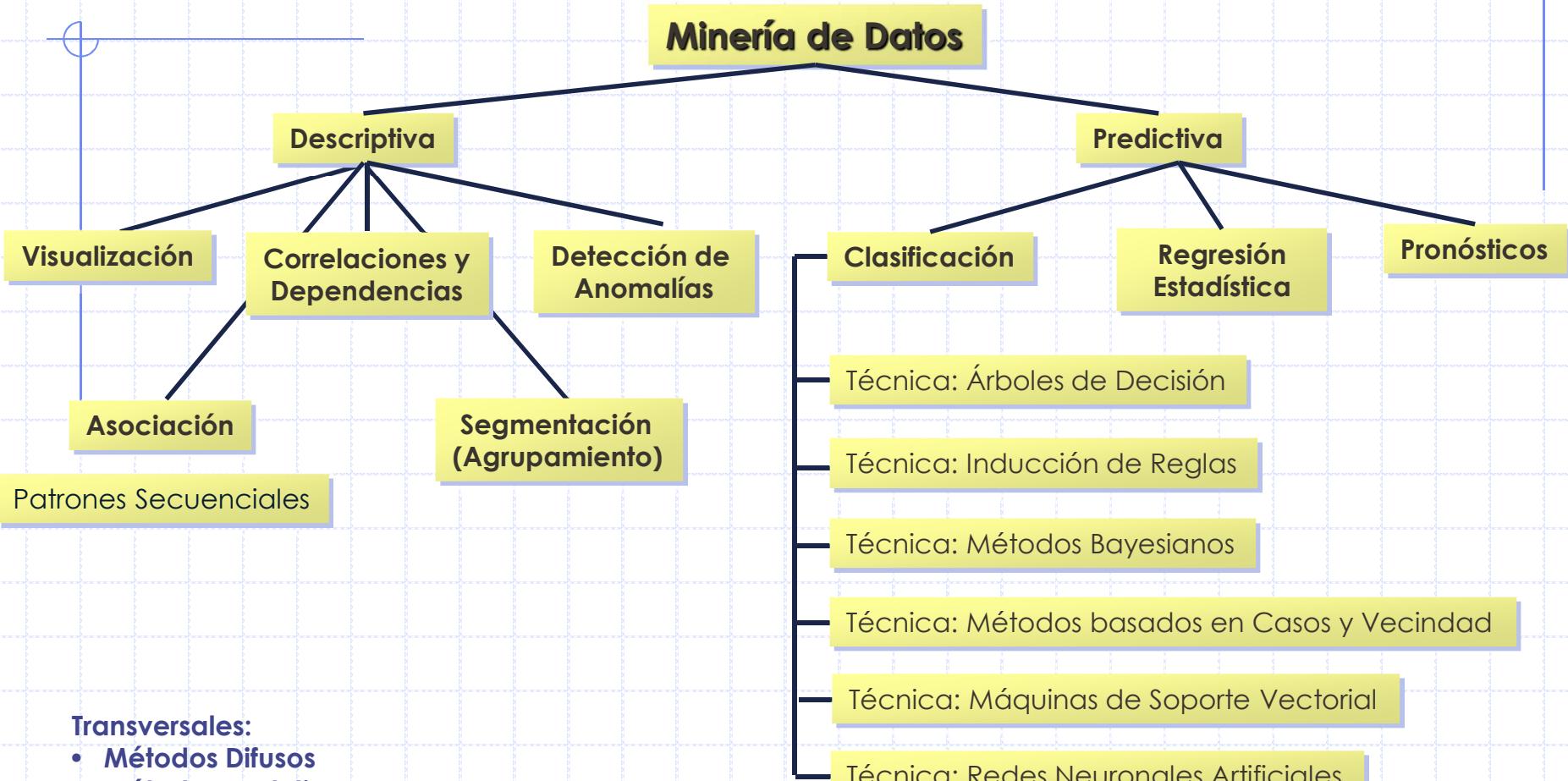
Minería de Datos

Tareas y Métodos

- Tarea: es un (tipo de) problema de minería de datos.
- Método: o técnica, permite resolver tareas.

Tareas

Minería de Datos



Minería de Datos

Tareas

- Preliminares:

- E : conjunto de todos los posibles elementos de entrada.
- A_i : atributo nominal o numérico, que describe una propiedad de cada elemento de E .
- $e = \langle a_1, a_2, \dots, a_n \rangle$ tal que a_i pertenece a A_i .

Minería de Datos

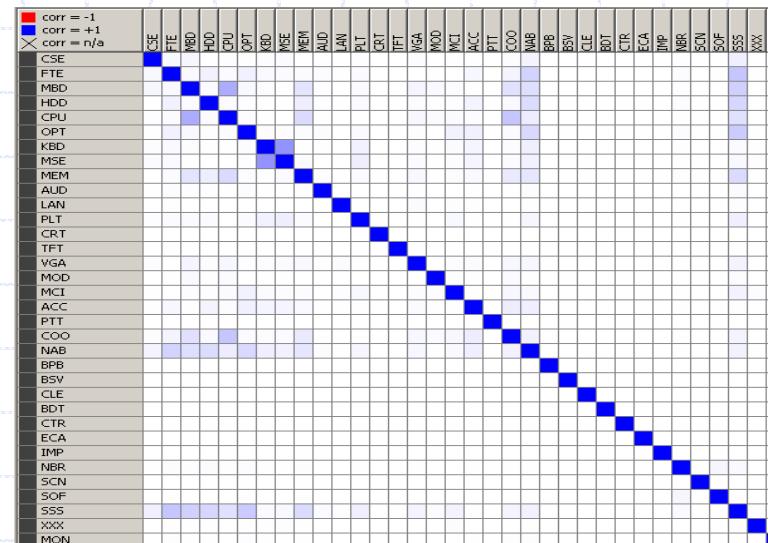
Tareas

- Tareas **Descriptivas**: los problemas se presentan como un conjunto $\delta = \{e: e \in E\}$, sin etiquetar ni ordenar de alguna forma. El objetivo consiste en describir los datos existentes (y no predecir nuevos datos).

Minería de Datos

Tareas

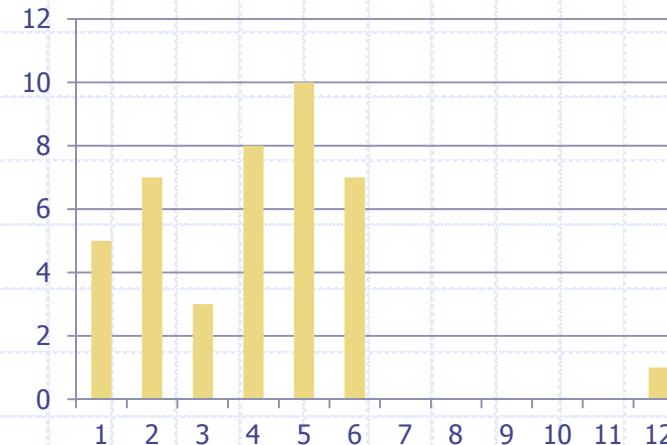
- Correlaciones y factorizaciones: se centran exclusivamente en los atributos numéricos. El objetivo es ver, dados los ejemplos del conjunto $E = A_1 \times A_2 \times \dots \times A_n$, si dos o más atributos numéricos A_i y A_j están correlacionados linealmente o relacionados de algún otro modo.



Minería de Datos

Tareas

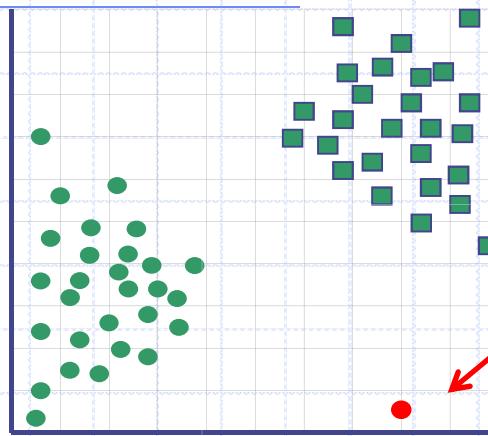
- Detección de valores e instancias anómalas: útil para detectar comportamientos anómalos como fraudes, fallas, intrusos.



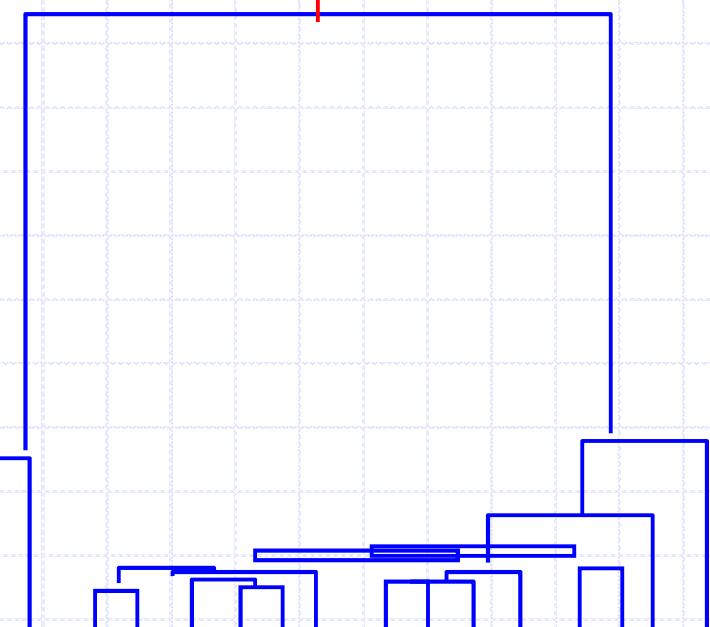
Minería de Datos

Tareas

...y detectar *outliers*.



Outlier



Minería de Datos

Tareas

- (Reglas de) Asociación: dados los ejemplos del conjunto $E = A_1 \times A_2 \times \dots \times A_n$, una regla de asociación se define generalmente como "SI $A_1=a$ and $A_2=b$ and ... x $A_k=h$ ENTONCES $A_r=u$ and $A_s=v$ and ... x $A_z=w$ ", donde todos los atributos son nominales.

- Negativas
- Multiniveles

- Secuenciales
- Direccionadas

Minería de Datos

Tareas

RUT	Ingreso Familiar	Ciudad	Actividad	Edad	Hijos	Sexo	Casado
10.251.545-3	5.000.000	Concepción	Ejecutivo	45	3	M	Sí
15.512.526-4	1.000.000	Valparaíso	Abogado	25	0	M	No
12.512.526-4	3.000.000	Talca	Ejecutivo	35	2	M	Sí
14.374.183-3	2.000.000	Valdivia	Camarero	30	0	M	Sí
14.572.904-1	1.500.000	Santiago	Animador Parque Temático	30	0	F	No

Asociaciones frecuentes:

Casado e (Hijos > 0) {40%, 2 casos}
sexo Masculino y Casado {60%, 3 casos}

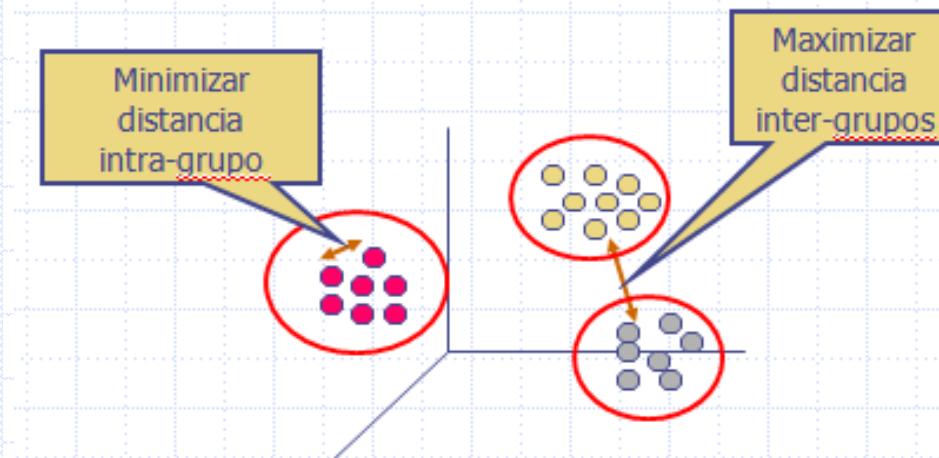
Dependencias:

$(\text{Hijos} > 0) \rightarrow \text{Casado}$ {100%, 2 casos}
 $\text{Casado} \rightarrow (\text{Hijos} > 0)$ {66.6%, 2 casos}
 $\text{Casado} \rightarrow \text{sexo Masculino}$ {100%, 3 casos}

Minería de Datos

Tareas

- Agrupamiento (Segmentación): el objetivo es obtener grupos o conjuntos entre los elementos de δ , de tal manera que los elementos asignados al mismo grupo sean similares.



Minería de Datos

Tareas

#Ej	Sueldo	Casado	Coche	Hijos	Alq/Prop	Sindic.	Bajas/Año	Antigüedad	Sexo
1	10000	Sí	No	0	Alquiler	No	7	15	H
2	20000	No	Sí	1	Alquiler	Sí	3	3	M
3	15000	Sí	Sí	2	Prop	Sí	5	10	H
4	30000	Sí	Sí	1	Alquiler	No	15	7	M
5	10000	Sí	Sí	0	Prop	Sí	1	6	H
6	40000	No	Sí	0	Alquiler	Sí	3	16	M
7	25000	No	No	0	Alquiler	Sí	0	8	H
8	20000	No	Sí	0	Prop	Sí	2	6	M
9	20000	Sí	Sí	3	Prop	No	7	5	H
10	30000	Sí	Sí	2	Prop	No	1	20	H
11	50000	No	No	0	Alquiler	No	2	12	M
12	8000	Sí	Sí	2	Prop	No	3	1	H
13	20000	No	No	0	Alquiler	No	27	5	M
14	10000	No	Sí	0	Alquiler	Sí	0	7	H
15	8000	No	Sí	0	Alquiler	No	3	2	H

cluster 1: 5 examples

Sueldo : 22600
Casado : No -> 0.8
 Sí -> 0.2
Coche : No -> 0.8
 Sí -> 0.2
Hijos : 0
Alq/Prop : Alquiler -> 1.0
Sindic. : No -> 0.8
 Sí -> 0.2
Bajas/Año : 8
Antigüedad : 8
Sexo : H -> 0.6
 M -> 0.4

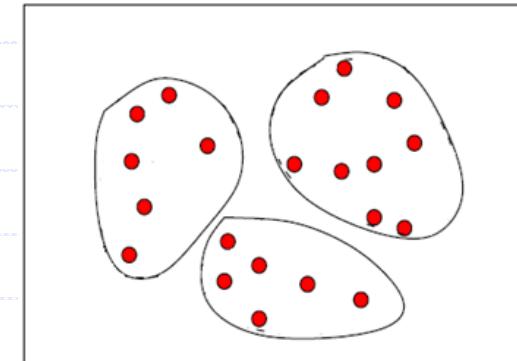
cluster 2: 4 examples

Sueldo : 22500
Casado : No -> 1.0
Coche : Sí -> 1.0
Hijos : 0
Alq/Prop :
 Alquiler -> 0.75
 Prop -> 0.25
Sindic. : Sí -> 1.0
Bajas/Año : 2
Antigüedad : 8
Sexo : H -> 0.25
 M -> 0.75

cluster 3: 6 examples

Sueldo : 18833
Casado : Sí -> 1.0
Coche : Sí -> 1.0
Hijos : 2
Alq/Prop :
 Alquiler -> 0.17
 Prop -> 0.83
Sindic. : No -> 0.67
 Sí -> 0.33
Bajas/Año : 5
Antigüedad : 8
Sexo : H -> 0.83
 M -> 0.17

- GRUPO 1: Sin hijos y de alquiler. Poco sindicalizados. Muchas bajas.
- GRUPO 2: Sin hijos y con coche. Muy sindicalizados. Pocas bajas. Normalmente de alquiler y mujeres.
- GRUPO 3: Con hijos, casados y con coche. Propietarios. Poco sindicados. Hombres.



Minería de Datos

Tareas

- Tareas **Predictivas**: son problemas y tareas en los que hay que predecir uno o más valores para un conjunto de ejemplos. Éstos van acompañados de una salida (clase, categoría, valor numérico) o un orden entre ellos.

Dependiendo de la correspondencia entre los ejemplos y los valores de salida, y la presentación de los ejemplos, las tareas predictivas pueden ser...

Minería de Datos

Tareas

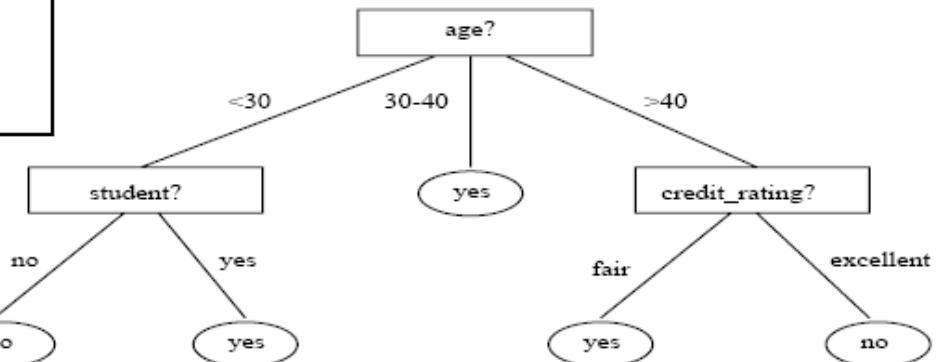
- Clasificación: los ejemplos son del tipo $\delta = \{<e,s>, e \in E, s \in S\}$, siendo S es el conjunto de salida, nominal. El objetivo es aprender una función $\lambda: E \rightarrow S$, llamada clasificador, que a cada valor de E se tiene un único valor para S .
 - Clasificación binaria: si S tiene sólo dos valores.
 - Clasificación suave: a la definición básica, se agrega una segunda función $\Theta: E \rightarrow R$, que representa el grado de precisión o certeza de la predicción de λ .
 - Estimación de probabilidad de clasificación: se trata de aprender m funciones $\Theta_i: E \rightarrow R$, donde m es el número de clases; es decir cada función retorna para cada ejemplo un valor p_i (grado de certeza – probabilidad, para la clase).

Minería de Datos

Tareas

IF $age = <30$ AND $student = no$ THEN $buys_computer = no$
 IF $age = <30$ AND $student = yes$ THEN $buys_computer = yes$
 IF $age = 30-40$ THEN $buys_computer = yes$
 IF $age = >40$ AND $credit_rating = excellent$ THEN $buys_computer = no$
 IF $age = >40$ AND $credit_rating = fair$ THEN $buys_computer = yes$

rid	age	income	student	credit_rating	Class: buys_computer
1	<30	high	no	fair	no
2	<30	high	no	excellent	no
3	30-40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	30-40	low	yes	excellent	yes
8	<30	medium	no	fair	no
9	<30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<30	medium	yes	excellent	yes
12	30-40	medium	no	excellent	yes
13	30-40	high	yes	fair	yes
14	>40	medium	no	excellent	no



Minería de Datos

Tareas

- Categorización: no se trata de aprender una función sino que una correspondencia. Tanto δ como λ pueden asignar varias categorías a un mismo e , a diferencia de la clasificación que sólo asigna una y no más; así, un ejemplo puede tener varias categorías asociadas.
- Preferencias o Priorización: consiste en determinar a partir de dos o más ejemplos, un orden de preferencia. Cada ejemplo es una secuencia $\langle e_1, e_2, \dots, e_k \rangle$, $e_i \in E$, $k \geq 2$, donde el orden de la secuencia representa la predicción.

Minería de Datos

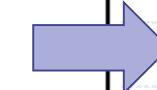
Tareas

- Regresión: el conjunto de evidencias son correspondencias entre dos conjuntos $\delta : E \rightarrow S$, siendo éste el conjunto de valores de salida, de tipo numérico; el objetivo es aprender una función $\lambda : E \rightarrow S$ que represente la correspondencia existente en los ejemplos.

→ Interpolación

→ Estimación

X years experience	Y salary (in \$1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83



$$Y = 21.7 + 3.7X$$

→ Logística: cuando se establece un valor umbral sobre el cual se compara la salida de la función λ . Dependiendo del valor se tiene una clase u otra como resultado.

TABLA 1. Operacionalización de las variables incluidas como posibles predictoras

Variáble	Operacionalización
Hábito de fumar	1. No 2. Sí
Práctica de ejercicios físicos	1. No 2. Sí
Frecuencia de visitas al médico	1. Hasta 2 2. Tres o más
Tipo de tratamiento	1. Dieta sola 2. Tabletas 3. Insulina 4. Tabletas e insulina
Edad	1. Menos de 40 2. De 40 a 59 3. De 60 o más
Conocimientos sobre la DM	1. No satisfactorios 2. Satisfactorios
Índice de masa corporal	1. Bajopeso 2. Nomopeso 3. Obesidad ligera 4. Obesidad moderada 5. Obesidad severa
Tiempo de evolución de la diabetes	1. Hasta 10 años 2. 11 años o más
Sexo	1. Masculino 2. Femenino
Plaza	1. No 2. Sí 3. Plaza 4. Moncada

TABLA 2. Resultados de la regresión logística

Variables	β	Significación	Exp (β)	Intervalo de confianza (90 %)
				Límite inferior Límite superior
Tipo de tratamiento		0,0284		
Tratamiento (2)	-0,6020	0,0168	0,5477	0,3620 0,8289
Tratamiento (3)	0,7681	0,0496	2,1556	1,1327 4,1023
Tratamiento (4)	0,0458	0,934	1,0469	0,4216 2,5996
Edad		0,1232		
Edad (2)	-0,3534	0,4153	0,7023	0,3441 1,4335
Edad (3)	-0,7801	0,0596	0,4583	0,2319 0,9059
Sexo	0,8030	0,0098	2,2323	1,3385 3,7229



Métodos

Minería de Datos

Métodos

- Son las técnicas o algoritmos con los cuales resolver una tarea de minería de datos. Algunas posibles son:

- Algebraicas y estadísticas: expresan los modelos y patrones mediante fórmulas algebraicas, funciones lineales y no lineales, distribuciones o valores agregados estadísticos (promedios, varianzas, correlaciones, etc.).

Ej.:

- Regresión Lineal (global, local), Logarítmica, Logística
- Discriminantes paramétricos
- algunas técnicas de Modelamiento Estadístico no paramétrico

Minería de Datos

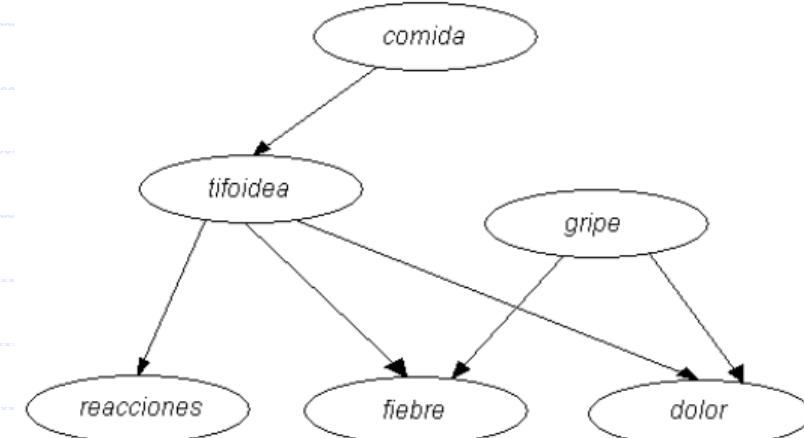
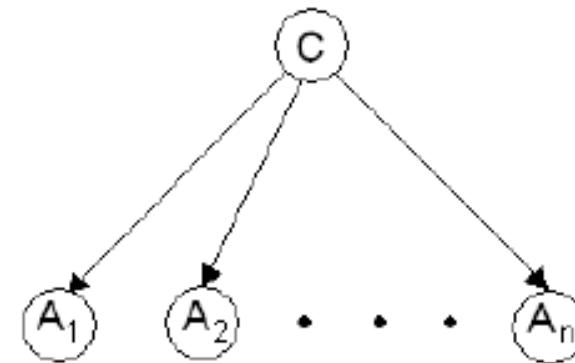
Métodos

- Bayesianas: estiman la probabilidad de pertenencia a una clase o grupo mediante la estimación de las probabilidades condicionales inversas o apriori, usando el teorema de Bayes. Ej.:
 - Clasificador bayesiano ingenuo
 - métodos basados en Máxima Verosimilitud
 - Algoritmo EM

Minería de Datos

Métodos

- Clasificador bayesiano ingenuo.
- Red bayesiana.



Minería de Datos

Métodos

- basadas en Conteos de Frecuencias y Tablas de Contingencia. Ej.: se basan en contar la frecuencia en que dos o más sucesos se presenten conjuntamente.
Ej.: algoritmo Apriori.

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

$$S1 = \{ \{1\}, \{2\}, \{3\}, \{4\}, \{5\} \}$$

$$S1': \text{soporte} = \{ \{1\}:2, \{2\}:3, \{3\}:3, \{5\}:3 \}$$

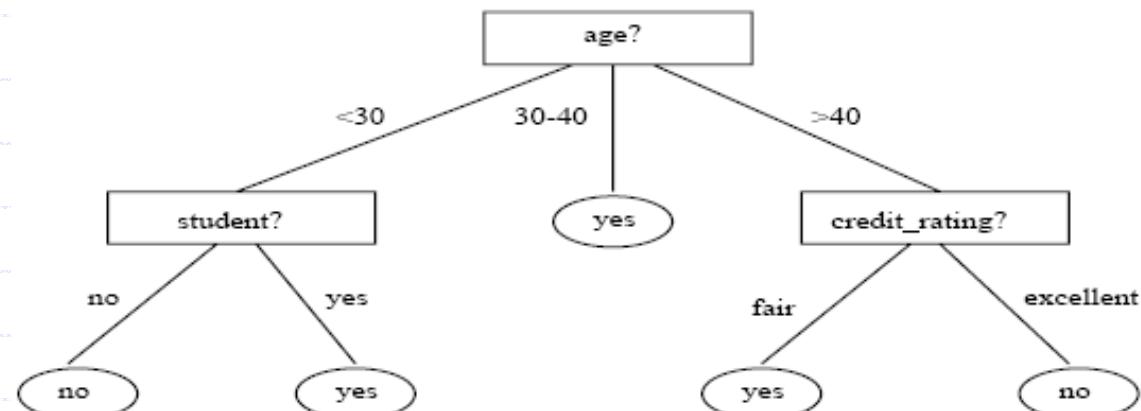
$$S2 = \{ \{1,2\}, \{1,3\}, \{1,5\}, \{2,3\}, \{2,5\}, \{3,5\} \}$$

$$S2': \text{soporte} = \{ \{1,3\}:2, \{2,3\}:2, \{2,5\}:3, \{3,5\}:2 \}$$

Minería de Datos

Métodos

- basadas en Árboles de Decisión y sistemas de Aprendizaje de Reglas: representan su resultado en forma de reglas. Ej.: algoritmos del tipo...
 - "dividir y conquistar": ID3, C4.5, CART.
 - "separar y conquistar": CN2.

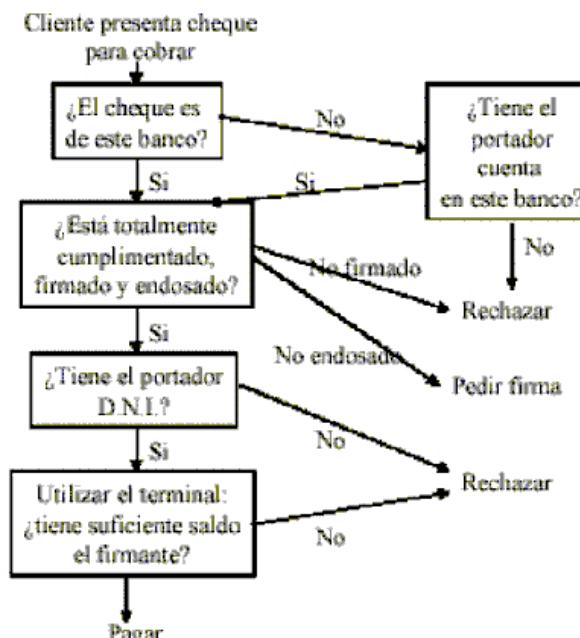


Minería de Datos

Métodos

- Relacionales, declarativas y estructurales: representan los modelos mediante lenguajes declarativos, lógicos, funcionales o mixtos.

PROCEDIMENTAL



DECLARATIVO

- (1) Si cheque completo y portador conocido y fondos suficientes entonces pagar
- (2) Si fecha correcta y firmado y fondos suficientes y portador identificado y ... entonces cheque completo
- (3) Si fecha cheque es hoy o fecha cheque entre 1 y 90 días antes de hoy entonces fecha correcta

Minería de Datos

Métodos

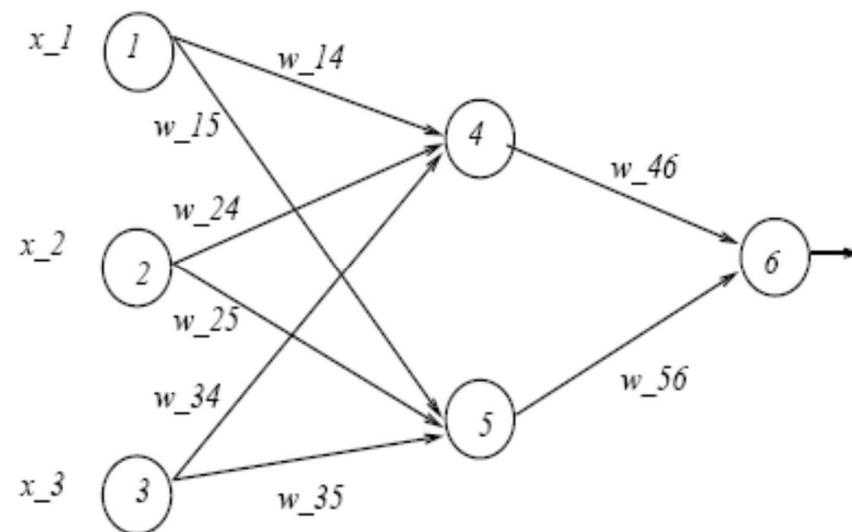
- basadas en Redes Neuronales Artificiales: aprenden un modelo mediante el entrenamiento de los pesos que conecten un conjunto de nodos o neuronas. La topología de la red y los pesos de las conexiones determinan el patrón aprendido. Ej.:

- Perceptrón simple
- Redes de base radial

- Redes multicapas
- Redes de Kohonen

Minería de Datos

Métodos



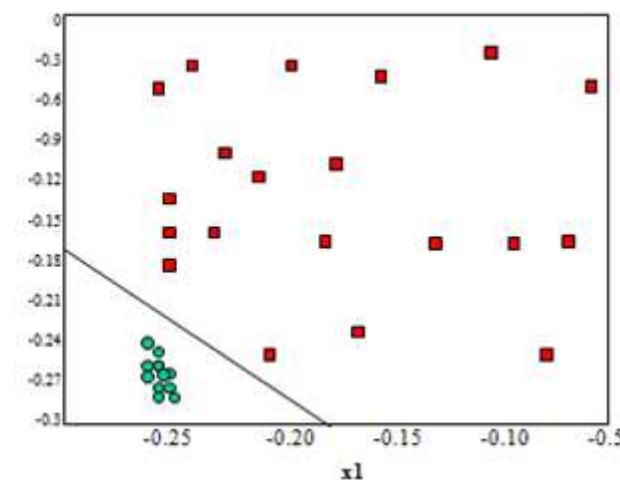
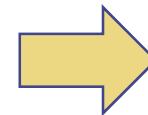
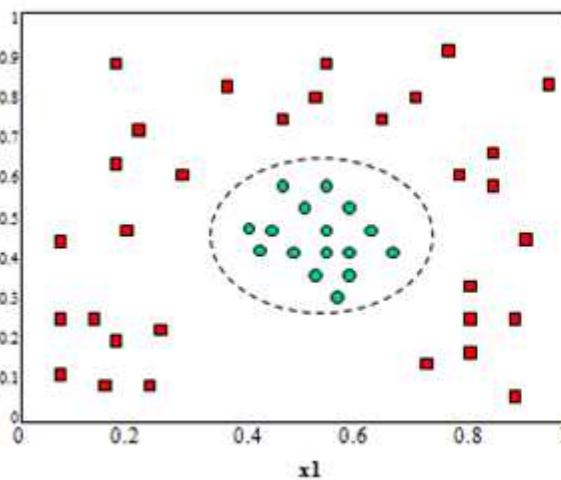
x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1

Unit j	Net Input, I_j	Output, O_j
4	$0.2 + 0 - 0.5 - 0.4 = -0.7$	$1/(1 + e^{-0.7}) = 0.33$
5	$-0.3 + 0 + 0.2 + 0.2 = 0.1$	$1/(1 + e^{-0.1}) = 0.52$
6	$(0.3)(0.33) - (0.2)(0.52) + 0.1 = 0.19$	$1/(1 + e^{-0.19}) = 0.55$

Minería de Datos

Métodos

- basadas en Núcleo y Máquinas de Soporte Vectorial: intentan maximizar el margen entre los grupos o las clases formadas, usando transformaciones que pueden aumentar la dimensionalidad (*kernel*).



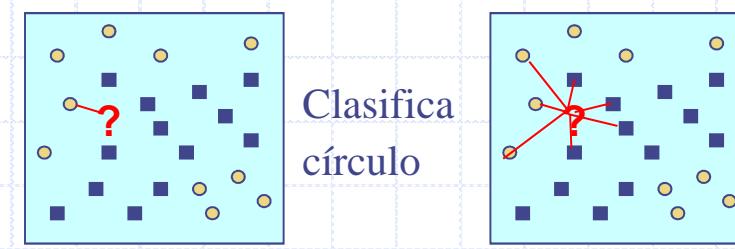
Ejemplo de conjunto linealmente separable

Minería de Datos

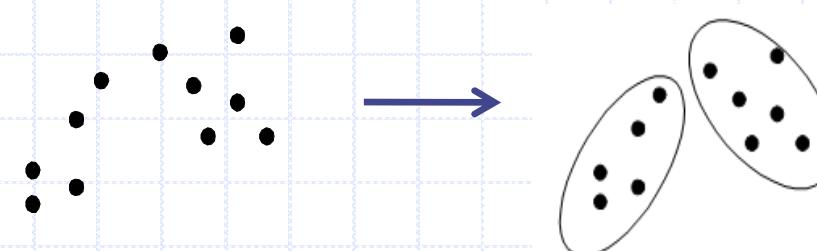
Métodos

- basadas en Casos, en Densidad o Distancia: se basan en distancias al resto de los elementos, ya sea directamente o de una forma más sofisticada. Ej.:

→ Vecinos más cercanos



→ Kmeans, Kmodes



Minería de Datos

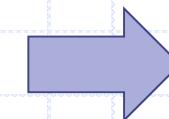
Métodos

- Estocásticas y Difusas: junto con las redes neuronales conforman la llamada computación flexible (*soft computing*). Ej.:
 - los componentes aleatorios son fundamentales: *simulated annealing*, métodos evolutivos y genéticos.
 - funciones de pertenencia difusas

Minería de Datos

Métodos

JUGADOR	RAPIDEZ	RESISTENCIA
1	0.58	0.33
2	0.90	0.11
3	0.68	0.17
4	0.11	0.44
5	0.47	0.81
6	0.24	0.83
7	0.09	0.18
8	0.82	0.11
9	0.65	0.50
10	0.09	0.63
11	0.98	0.24



Dato	Pertenencia	
	Cluster1	Cluster2
1	0.3085	0.6915
2	0.2016	0.7984
3	0.2665	0.7335
4	0.9762	0.0238
5	0.5481	0.4519
6	0.7927	0.2073
7	0.8412	0.1588
8	0.2213	0.7787
9	0.1000	0.9000
10	0.9473	0.0527
11	0.1289	0.8711

- Grupo 1 (jugadores resistentes): 1, 2, 3, 8, 9, 11.
- Grupo 2 (jugadores rápidos, poco resistentes): 4, 5, 6, 7, 10.

lentos,

Minería de Datos

Tareas y Métodos

Método	Técnicas	Descriptivas			Predictivas	
		Correlaciones	Reglas de Asociación	Segmentación	Clasificación	Regresión
Apriori			X			
Algoritmos Genéticos y Evolutivos		X	X	X	X	X
Análisis Discriminante Multivariante					X	
Análisis Factorial y de Componentes principales		X				
Árboles de decisión: CART					X	X
Árboles de decisión: ID3, C4.5					X	
Árboles de decisión: otros			X	X	X	X
Bayes Ingenuo (Naive)					X	
CobWeb, Two Step					X	
Kmeans					X	
Máquinas de Soporte Vectorial				X	X	X
Redes de Kohonen					X	
Redes Neuronales Artificiales				X	X	X
Reglas CN2			X		X	
Regresión Lineal y Logarítmica		X				X
Regresión Logística			X		X	
Vecinos más cercanos				X	X	X



Tarea Descriptiva: Asociación

Asociación

Definiciones básicas

- Tarea descriptiva, no supervisada.
- Posibilidades:
 - Reglas de Asociación: Se buscan asociaciones de la siguiente forma:
$$(X_1 = a) \leftrightarrow (X_4 = b)$$
 - Dependencias: asociaciones de la forma (if *Ante* then *Cons*):
$$\text{if } (X_1 = a, X_3 = c, X_5 = d) \rightarrow (X_4 = b, X_2 = a)$$

Asociación

Definiciones básicas

RUT	Ingreso Familiar	Ciudad	Actividad	Edad	Hijos	Sexo	Casado
10.251.545-3	5.000.000	Concepción	Ejecutivo	45	3	M	Sí
15.512.526-4	1.000.000	Valparaíso	Abogado	25	0	M	No
12.512.526-4	3.000.000	Talca	Ejecutivo	35	2	M	Sí
14.374.183-3	2.000.000	Valdivia	Camarero	30	0	M	Sí
14.572.904-1	1.500.000	Santiago	Animador Parque Temático	30	0	F	No

Asociaciones frecuentes:

Casado e (Hijos > 0)
sexo Masculino y Casado

{40%, 2 casos}

{60%, 3 casos}

Dependencias:

(Hijos > 0) → Casado
Casado → (Hijos > 0)
Casado → sexo Masculino

{100%, 2 casos}

{66.6%, 2 casos}

{100%, 3 casos}

Asociación

Tipos de Reglas de Asociación

- Basado en los Tipos de Valores manejados por la Regla:
 - Regla booleana: las asociaciones indican la ausencia o presencia del elementos, tal como:

computador → impresora

- Regla cuantitativa: las asociaciones describe relaciones entre atributos cuantitativos, como por ejemplo:

(30 < edad < 39) and (ingreso > 500.000) →

TV con pantalla plana

Asociación

Tipos de Reglas de Asociación

- Basado en las Dimensiones de los Datos Involucrados:
 - Regla unidimensional: los atributos hacen referencia a una única dimensión, como por ejemplo:

computador → impresora

- Regla multidimensional: se hace referencia a dos o más dimensiones, tal como:

(30 < edad < 39) and (ingreso > 500.000) →

TV con pantalla plana

Asociación

Tipos de Reglas de Asociación

- Instantáneas o Secuenciales.

- Instantánea: indica relaciones inmediatas, contemporáneas.

computador → impresora

- Secuencial: establece un orden temporal.

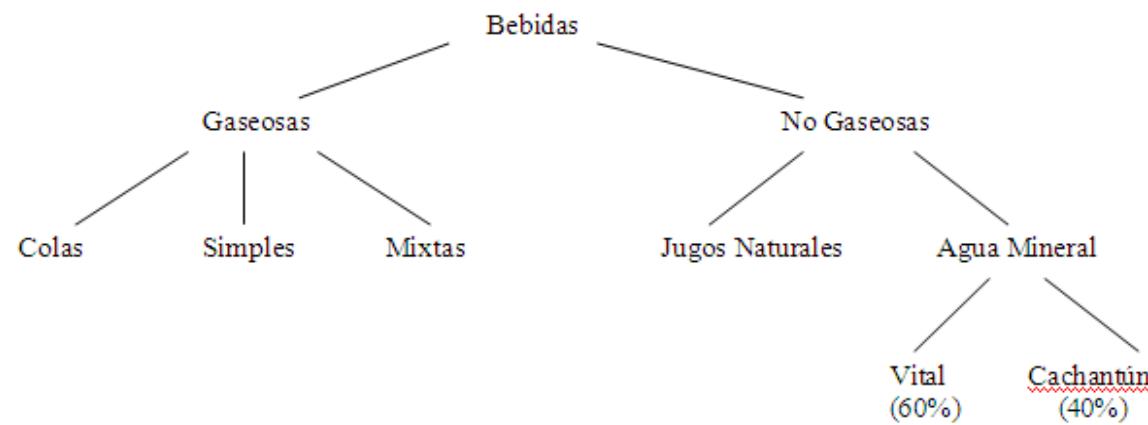
computador → impresora en próxima compra

computador → impresora antes de tres meses

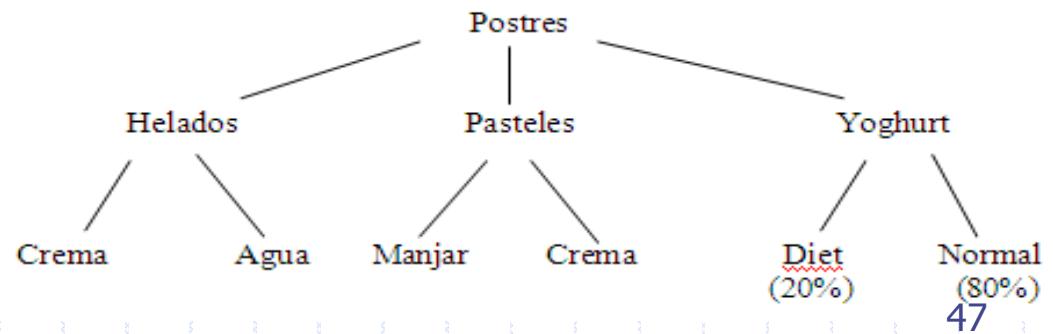
Asociación

Tipos de Reglas de Asociación

- Basado en los Niveles de Abstracción.



Bebidas → Postres



Asociación

Tipos de Reglas de Asociación

- Positivas o Negativas.

- Positiva: indica la ocurrencia o presencia de los ítems relaciones.

computador → impresora

- Negativa: señala la ausencia de al menos uno de los ítems de la regla

computador → not impresora

Asociación

Medidas para la Bondad de las Reglas de Asociación

- Medidas de Interés:

- Soporte: representa la utilidad de la regla.

soporte = número de casos o porcentaje en los que el antecedente se hace verdadero (r_c o r_c/n respectivamente), siendo n el número de datos en estudio.

- Confianza: refleja la certeza la regla.

confianza = corresponde al número de casos que habiendo cumplido el antecedente de la regla, cumplen también el consecuente (r_c/r_a).

$$\text{confianza } (X \rightarrow Y) = \text{soporte}(X \cup Y) / \text{soporte}(X)$$

Asociación

Medidas para la Bondad de las Reglas de Asociación

- Medidas de Interés (2):

- Elevación (lift): corresponde al cuociente entre el soporte observado y el soporte esperado si X e Y fueran independientes.

soporte($X \cup Y$)

$$\text{elevación}(X \rightarrow Y) = \frac{\text{soporte}(X \cup Y)}{\text{soporte}(X) * \text{soporte}(Y)}$$

Asociación

Medidas para la Bondad de las Reglas de Asociación

- Medidas de Interés (3):

- Convicción: corresponde al cuociente entre la frecuencia con que la regla hace una predicción incorrecta (siendo ambas partes de la regla independientes entre sí) y la frecuencia observada de las predicciones incorrectas.

$$1 - \text{soporte}(Y)$$

$$\text{conviccion}(X \rightarrow Y) = \frac{1 - \text{soporte}(Y)}{1 - \text{confianza}(X \rightarrow Y)}$$

Asociación

Algoritmos

- Los algoritmos de búsqueda de asociaciones y dependencias, en la mayoría se basa en descomponer el problema en dos fases:
 - FASE 1 - BÚSQUEDA DE **ITEMSETS** FRECUENTES. Se buscan conjuntos de ítems (o atributos) con 'soporte' mayor/igual al soporte deseado; de momento no se busca separarlos en parte izquierda y parte derecha.
 - FASE 2 - 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 mayor/igual a la confianza deseada.

Asociación

Algoritmos

- Algoritmo **Apriori**: método básico para encontrar reglas booleanas, unidimensionales y mononivel.
- Algunas ideas asociadas...
 - El algoritmo obtiene los llamados ***itemsets* *frecuentes*** para generar las reglas de asociación booleanas.
 - Su nombre es debido a que se basa en conocimientos previos sobre la frecuencia de los *itemsets*, al usar los *k-itemsets* para explorar los del siguiente nivel o paso ($k+1$).
 - Condición apriori: todos los subconjuntos de un *itemset* frecuente deben ser frecuentes.
 - Propiedad anti-monótona: si un conjunto no supera una prueba, los supra-conjuntos derivados tampoco la superarán.

Asociación

Algoritmos

- Algoritmo **Apriori**: dado un soporte mínimo s_{min} ...
 1. $i=1$ (tamaño de los conjuntos)
 2. Generar un conjunto unitario para cada atributo en S_i .
 3. Comprobar el soporte de todos los conjuntos en S_i . Eliminar aquellos cuyo soporte $< s_{min}$.
 4. Combinar los conjuntos en S_i para crear conjuntos de tamaño $i+1$ en S_{i+1} .
 5. **Si** S_i no es vacío **entonces** $i := i+1$. Ir a 3.
 6. **Si no**, retornar $S_2 \cup S_3 \cup \dots \cup S_i$
- Un ejemplo del uso de A priori para la generación de reglas de asociación es el siguiente...

Asociación

Algoritmos

FASE 1: BÚSQUEDA DE ITEMSETS FRECUENTES (Apriori)

Cliente	(Compra?) Producto				
	1	2	3	4	5
1	X		X	X	
2		X	X		X
3	X	X	X		X
4		X			X

soporte mínimo = 2

$$S_1 = \{ \{P1\}, \{P2\}, \{P3\}, \{P4\}, \{P5\} \}$$

$$\rightarrow S_1' = \{ \{P1\}:2, \{P2\}:3, \{P3\}:3, \{P5\}:3 \}$$

$$S_2 = \{ \{P1,P2\}, \{P1,P3\}, \{P1,P5\}, \{P2,P3\}, \{P2,P5\}, \{P3,P5\} \}$$

$$\rightarrow S_2' = \{ \{P1,P3\}:2, \{P2,P3\}:2, \{P2,P5\}:3, \{P3,P5\}:2 \}$$

$$S_3 = \{ \{P1,P2,P3\}, \{P1,P2,P5\}, \{P1,P3,P5\}, \{P2,P3,P5\} \}$$

$$\rightarrow S_3' = \{ \{P2,P3,P5\}:2 \}$$

$$S_{final} = S_2' \cup S_3' = \{ \{P1,P3\}, \{P2,P3\}, \{P2,P5\}, \{P3,P5\}, \{P2,P3,P5\} \}$$

Asociación

Algoritmos

FASE 2: ESCLARECIMIENTO DE DEPENDENCIAS (REGLAS)

Cliente	(Compra?) Producto				
	1	2	3	4	5
1	X		X	X	
2			X	X	X
3	X	X	X		X
4		X			X

confianza mínima = 0.75

$\{P1\} \rightarrow \{P3\}$: 1

$\{P2\} \rightarrow \{P3\}$: 0.67

$\{P2\} \rightarrow \{P5\}$: 1

$\{P3\} \rightarrow \{P5\}$: 0.67

$\{P3\} \rightarrow \{P1\}$: 0.67

$\{P3\} \rightarrow \{P2\}$: 0.67

$\{P5\} \rightarrow \{P2\}$: 1

$\{P5\} \rightarrow \{P3\}$: 0.67

$\{P2, P3\} \rightarrow \{P5\}$: 1

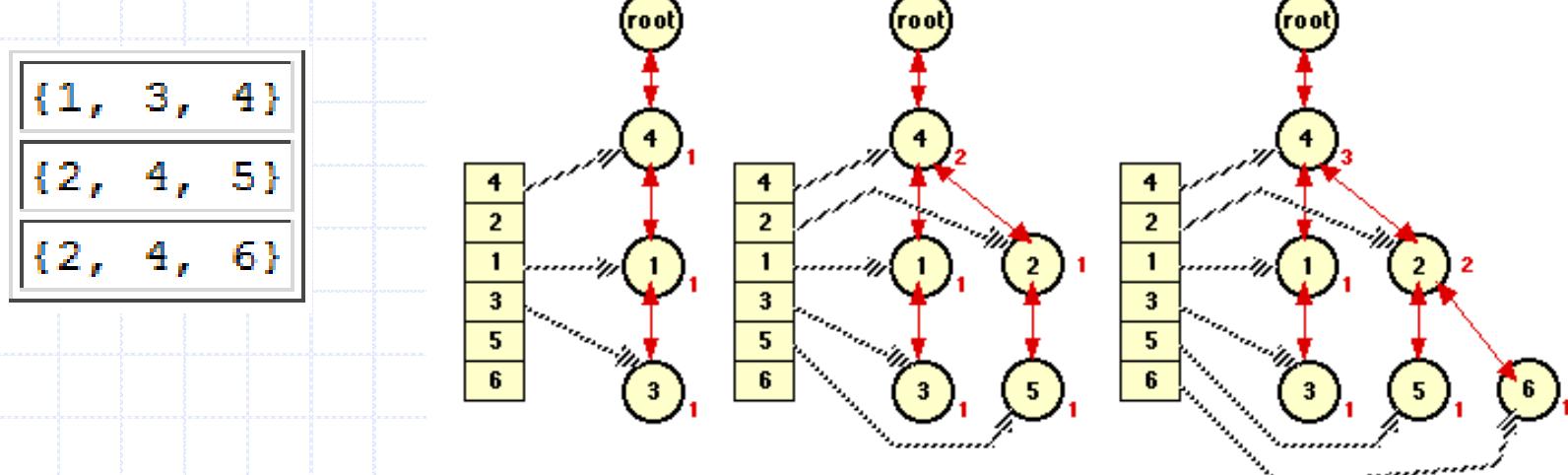
$\{P2, P5\} \rightarrow \{P3\}$: 0.67

$\{P3, P5\} \rightarrow \{P2\}$: 1

Asociación

Algoritmos

- Mejoras (extensiones)
 - Para reducir el número de accesos a la base de datos
 - estructuras de datos complejas (tablas hash).
 - FP-Tree (Frequent Pattern Tree) y el algoritmo **FP-growth**.



Asociación

Algoritmos

- Mejoras (extensiones)
 - Muestreo de la base de datos.
 - Filtro (selección) de atributos.
 - Paralelismo.
 - Aplicación a atributos numéricos → discretización; segmentación y asignar un valor discreto a cada grupo.

Asociación

Algoritmos

- algoritmo **AprioriAll**: trata de establecer asociaciones del estilo: "si compra X en T ... ¿comprará Y en T+P?"; es decir es para obtener patrones secuenciales.

Ejemplo:

Transaction Database

Customer	Transaction Time	Purchased Items
John	6/21/97 5:30 pm	Beer
	6/22/97 10:20 pm	Brandy
Frank	6/20/97 10:15 am	Juice, Coke
	6/20/97 11:50 am	Beer
	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
	6/21/97 6:17 pm	Wine, Cider
	6/22/97 5:05 pm	Brandy
Robin	6/20/97 11:05 pm	Brandy

Asociación

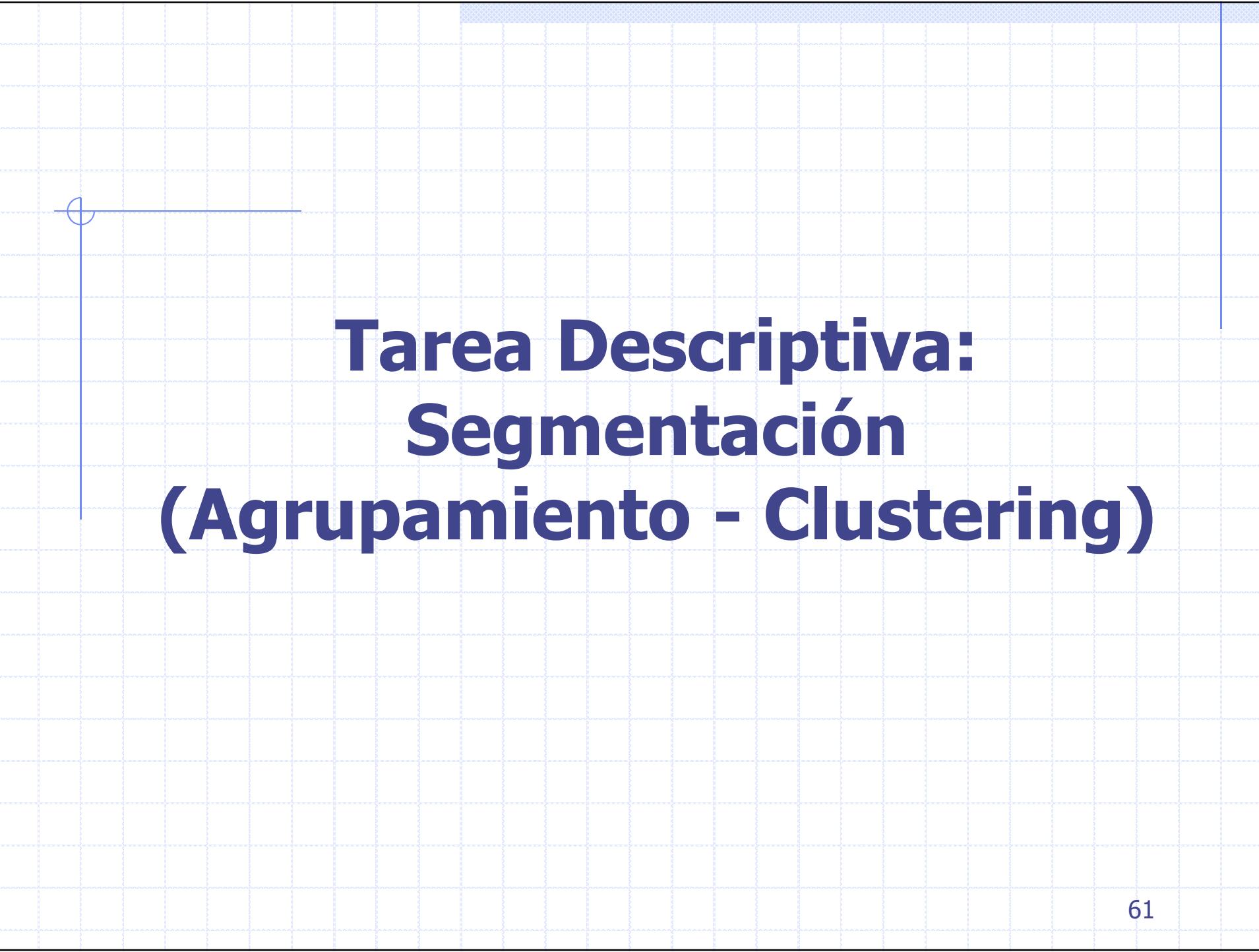
Algoritmos

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)

Mining Results

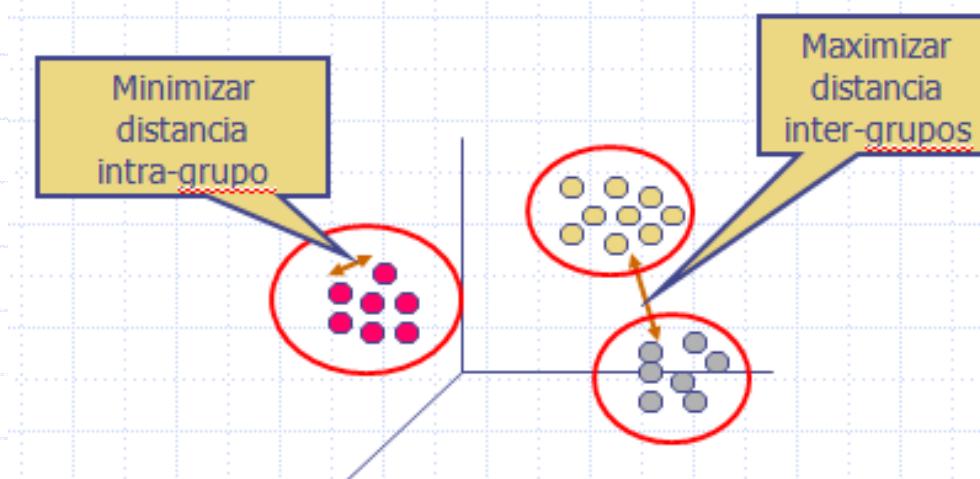
Sequential Patterns with Support $\geq 40\%$	Supporting Customers
(Beer) (Brandy) (Beer) (Wine, Cider)	John, Mary Frank, Mary



Tarea Descriptiva: Segmentación (Agrupamiento - Clustering)

Segmentación

- En este tipo de análisis se busca agrupar o segmentar los datos en grupos de acuerdo a la “relación” que se encuentre ellos.
- Encontrar agrupamientos de tal forma que los objetos de un grupo sean similares entre sí y diferentes de los objetos de otros grupos.



Segmentación

- Normalmente se refiere al llamado **aprendizaje no supervisado**, pues no descansa sobre clases predefinidas ni ejemplos de prueba en dichas clases.
- Por lo anterior, usa un esquema de **aprendizaje por observación** más que por ejemplos.

Segmentación

- El elemento clave es la elección de la distancia o medida de similitud entre objetos.

Distancia de Minkowski

$$d_r(x, y) = \left(\sum_{j=1}^J |x_j - y_j|^r \right)^{\frac{1}{r}}, \quad r \geq 1$$

➤ Distancia de Manhattan ($r=1$) / *city block* / *taxicab*

$$d_1(x, y) = \sum_{j=1}^J |x_j - y_j|$$

➤ Distancia euclídea ($r=2$):

$$d_2(x, y) = \sqrt{\sum_{j=1}^J (x_j - y_j)^2}$$

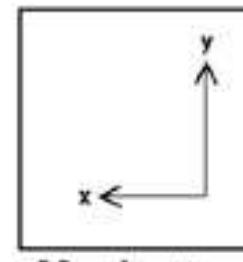
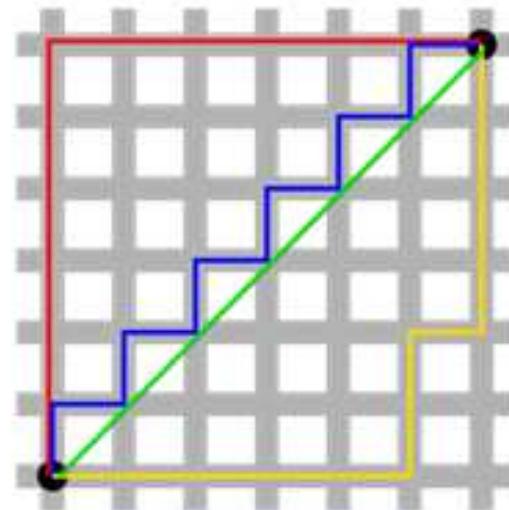
➤ Distancia de Chebyshev ($r \rightarrow \infty$) / *dominio* / *chessboard*

$$d_\infty(x, y) = \max_{j=1..J} |x_j - y_j|$$

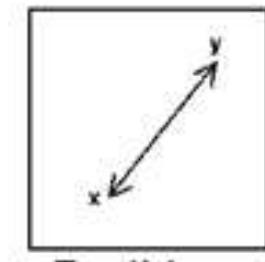
Segmentación

- Ejemplo:

Distancia de Minkowski



Manhattan



Euclidean

- Distancia de Manhattan = 12
- Distancia Euclídea ≈ 8.5
- Distancia de Chebyshev = 6

Segmentación

- Distancia de edición: de Levenshtein (número de operaciones necesario para transformar una cadena en otra).

$d("data mining", "data minino") = 1$

$d("efecto", "defecto") = 1$

$d("poda", "boda") = 1$

$d("night", "natch") = d("natch", "noche") = 3$

- Para datos binarios: Distancia de Hamming.

Segmentación

Tipos de Algoritmos

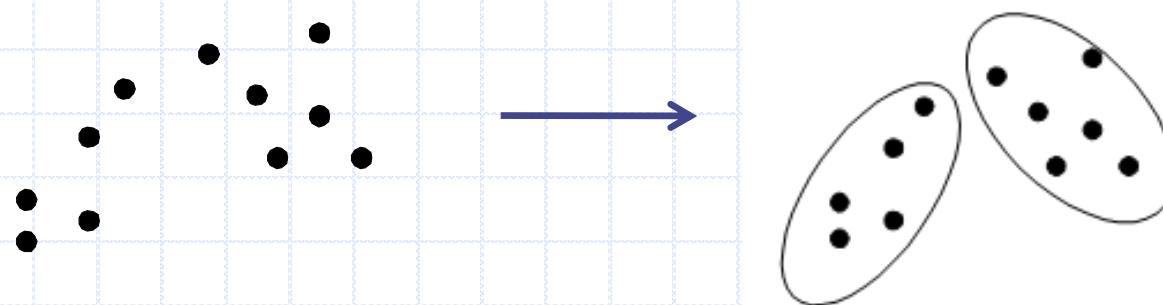
Un buen algoritmo de segmentación debe cumplir con:

- Escalabilidad
- Manejo de distintos tipos de datos
- Identificación de clusters con formas arbitrarias
- Número mínimo de parámetros
- Tolerancia frente a ruido y *outliers*
- Independencia con respecto al orden de presentación de los patrones de entrenamiento
- Posibilidad de trabajar en espacios con muchas dimensiones diferentes
- Capacidad de incorporar restricciones especificadas por el usuario
- Interpretabilidad / Usabilidad

Segmentación

Tipos de Algoritmos

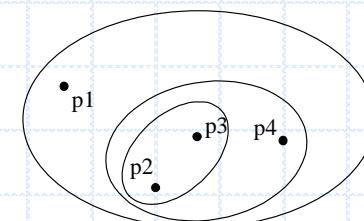
1) Métodos de **Particionamiento**: dada una base de datos con n objetos, un método de este tipo construye k particiones, donde cada una de éstas representa un grupo, siendo $k \leq n$. Ejs.: K-Means, K-Medoids (PAM).



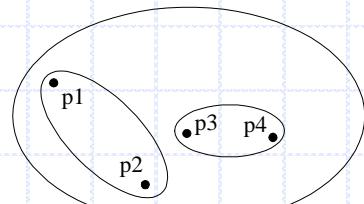
Segmentación

Tipos de Algoritmos

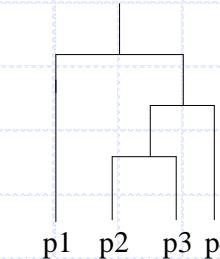
2) Métodos **Jerárquico**: crea una descomposición jerárquica del conjunto de datos dado. Ejs.: BIRCH, CURE.



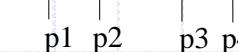
Tradicional



No tradicional



DENDOGRAMA



Segmentación

Tipos de Algoritmos

3) Métodos **basados en la Densidad**: la idea general es continuar creciendo el grupo dado tanto como la densidad (número de objetos o puntos de datos) en la vecindad exceda algún umbral. Ejs.: DBSCAN, OPTICS.

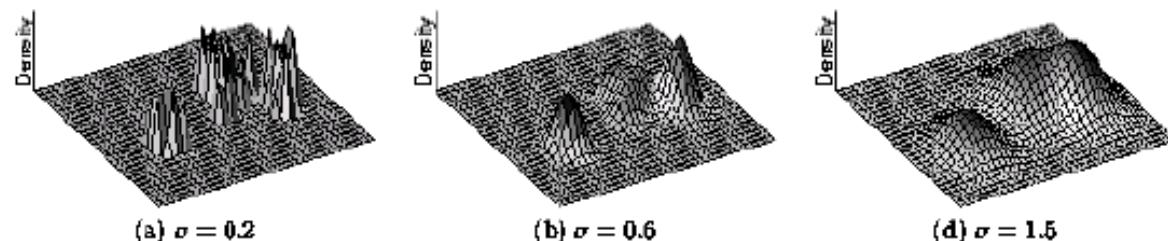


Figure 3: Example of Center-Defined Clusters for different σ

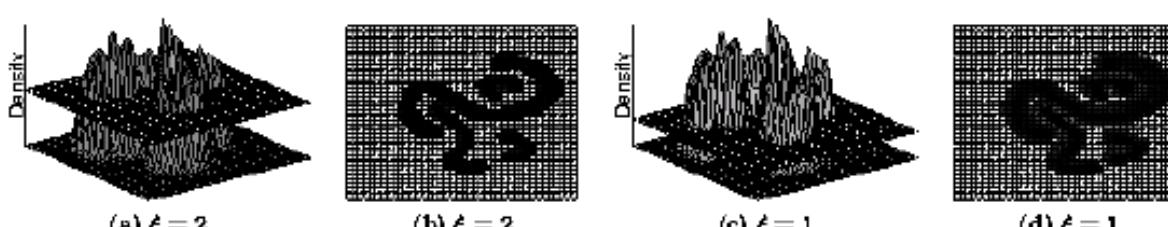


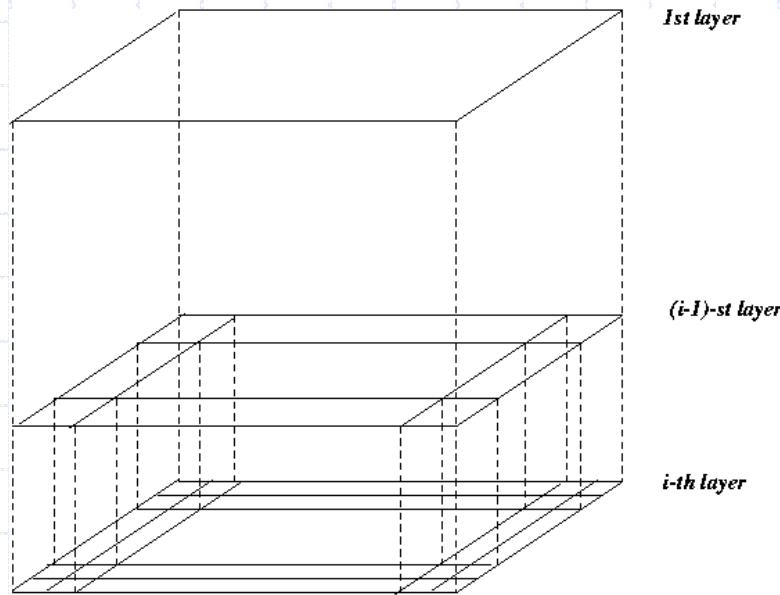
Figure 4: Example of Arbitrary-Shape Clusters for different ξ

Figure 8.11: Examples of center-defined clusters and arbitrary-shaped clusters

Segmentación

Tipos de Algoritmos

4) Métodos **basados en la Grilla**: cuantiza el espacio de objetos en un número finito de celdas que conforman una estructura de grilla. Entonces realizar todas las operaciones de agrupamiento en esta última. Ejs.: STING, CLIQUE, Wave-Cluster.



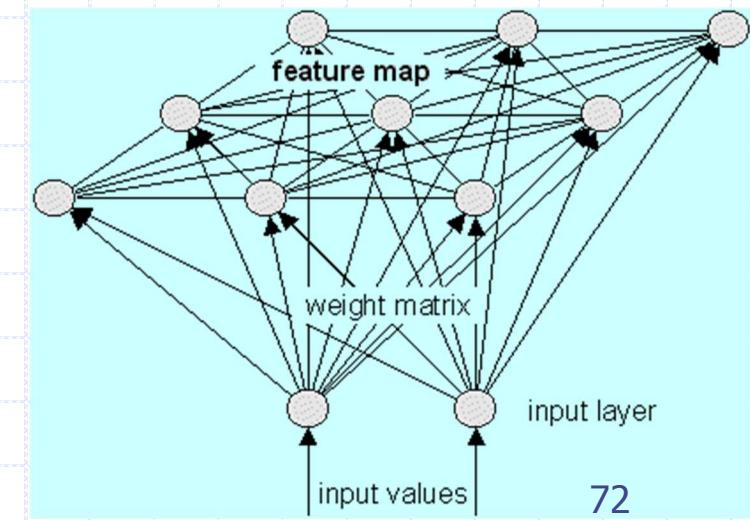
Segmentación

Tipos de Algoritmos

5) Métodos basados en **Modelo**: hipotetiza un modelo por cada uno de los grupos, y encuentra el mejor ajuste de los datos a ese modelo; puede localizar los grupos al construir una función de densidad que refleje la distribución espacial de los puntos de datos.

→ Enfoque Estadístico: algoritmos COBWEB, CLASSIT.

→ Enfoque de Red Neuronal:
SOM o Mapas AutoOrganizados
(Redes de Kohonen).



Segmentación

Algoritmos de Particionamiento

- **Algoritmo K-means:** basados en centroides.

- Procedimiento:
 - Dividir aleatoriamente los ejemplos en k conjuntos y calcular la media (el punto medio) cada conjunto.
 - Reasignar cada ejemplo al conjunto con el punto medio más cercano.
 - Calcular los puntos medios de los k conjuntos.
 - Repetir los pasos 2 y 3 hasta que los conjuntos no varíen.

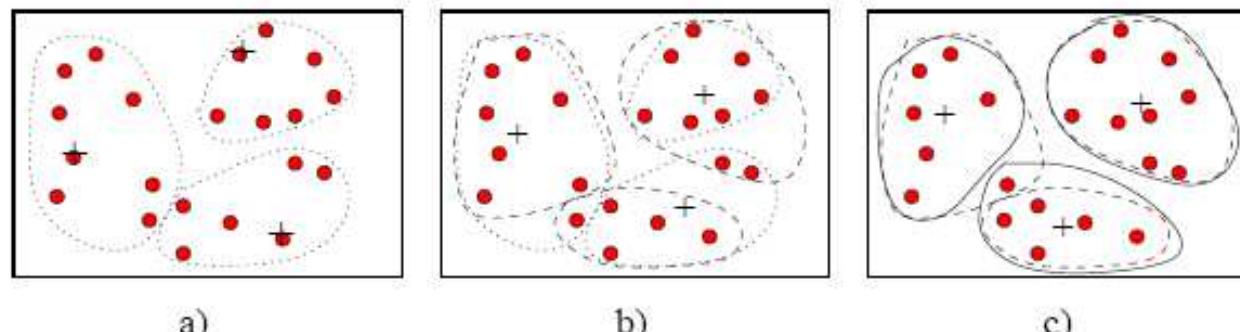


Figure 8.2: Clustering of a set of points based on the k -means method

Segmentación

Algoritmos de Particionamiento

- Algoritmo K-means:

Input: The number of clusters k , and a database containing n objects.

Output: A set of k clusters which minimizes the squared-error criterion.

Method: The k -means algorithm is implemented as follows.

- 1) arbitrarily choose k objects as the initial cluster centers;
- 2) repeat
- 3) (re)assign each object to the cluster to which the object is the most similar,
 based on the mean value of the objects in the cluster;
- 4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
- 5) until no change;

Segmentación

Algoritmos de Particionamiento

- Algoritmo K-means: más formalmente...
 - Si se considera sólo variables cuantitativas y la distancia Euclídea, el problema de optimización se puede plantear como:

$$\min_{C, \{m_k\}, k=1..K} \sum_{k=1..K} N_k \sum_{C(i)=k} \|x_i - m_k\|^2 \quad (*)$$

- Este problema puede abordarse en 2 pasos:
 - Dada una partición C , encontrar $\{m_1, \dots, m_K\}$ que minimicen $(*)$
 - Dadas las medias $\{m_1, \dots, m_K\}$, asignar cada observación al cluster cuya media es la más cercana:

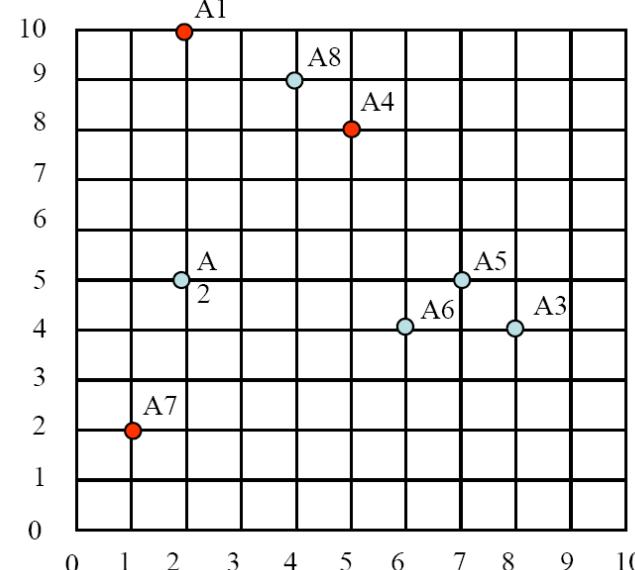
$$C(i) = \arg \min_{k=1..K} \|x_i - m_k\|^2$$

- Dichos pasos se repiten hasta que las asignaciones no cambien ("relocalización iterativa").

Segmentación

Algoritmos de Particionamiento

- Algoritmo K-means: ejemplo.



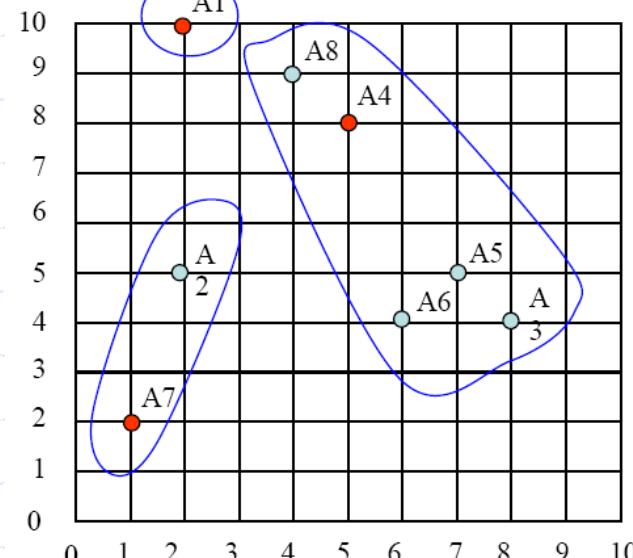
	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

Distancias Euclídeas

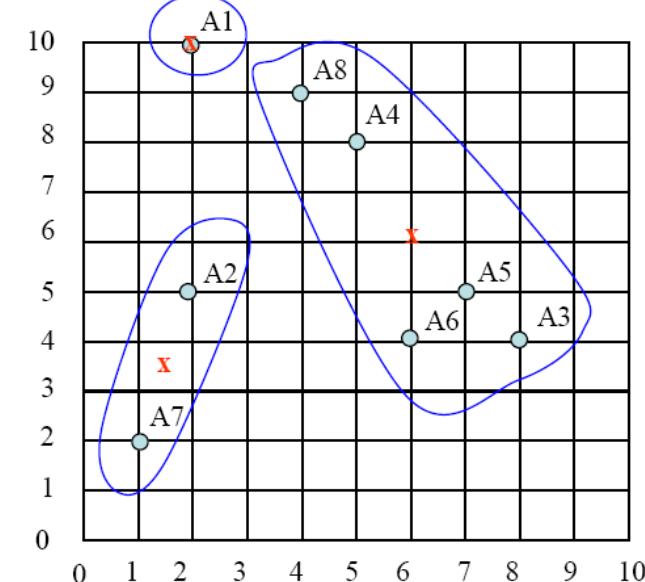
Segmentación

Algoritmos de Particionamiento

- Algoritmo K-means: ejemplo.



Primera Iteración

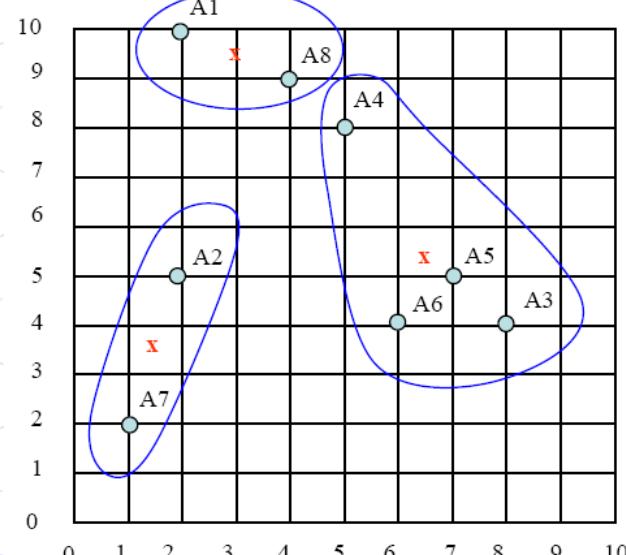


Segunda Iteración

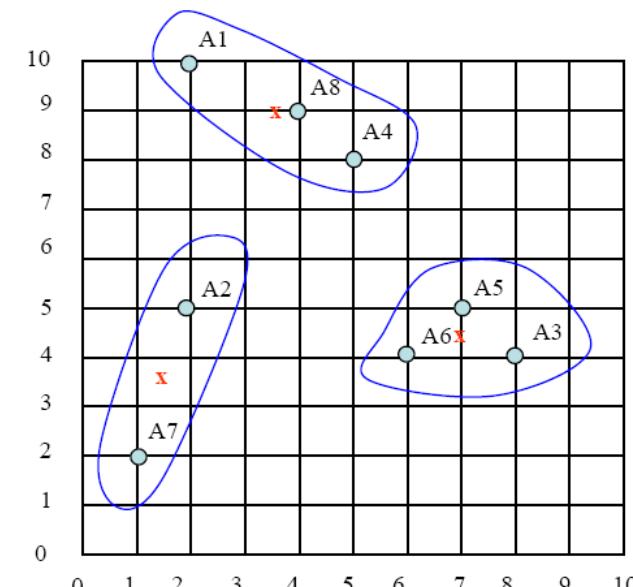
Segmentación

Algoritmos de Particionamiento

- Algoritmo K-means: ejemplo.



Tercera Iteración



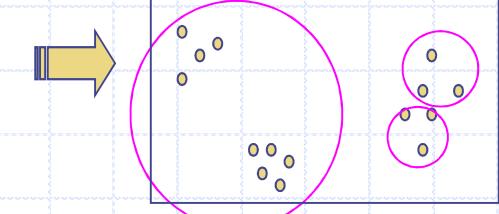
Configuración Final

Segmentación

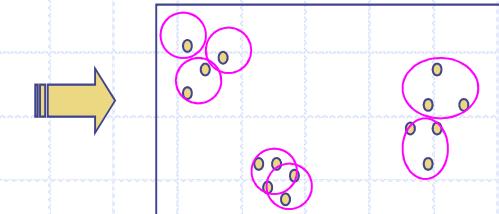
Algoritmos de Particionamiento

- Algoritmo K-means: problemas...

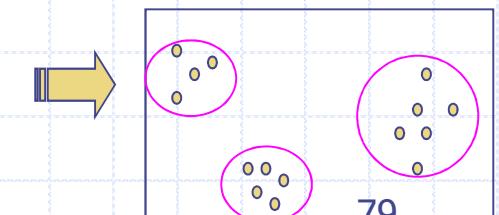
- Si se sabe que hay n clases, hacer $k=n$ puede resultar en que, algunas veces, algún grupo use dos centros y dos grupos separados tengan que compartir centro.



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



- Determinar el k ideal es difícil.



Segmentación

Algoritmos de Particionamiento

- Algoritmo K-means – ejercicio: considerar que se tienen cuatro tipos de datos, cada uno con dos atributos: X y Z.

Tipo de Datos	X	Z
A	1	1
B	2	1
C	4	3
D	5	4

Obtener dos grupos, usando K-means y distancias euclidianas, considerando a A y B como los puntos iniciales

Segmentación

Algoritmos de Particionamiento

- Algoritmo K-means: variaciones.
 - K-modes: para datos categóricos, al reemplazar los promedios de los grupos por las modas.
 - EM (*Expectation Maximization*): en vez de asignar cada punto a un grupo dedicado, asigna cada punto a un grupo de acuerdo a algún peso que represente la probabilidad de la membresía. En otras palabras no hay fronteras estrictas entre los grupos.

Segmentación

Algoritmos de Particionamiento

- Algoritmo **K-medoids**: basado en puntos representativos.
 - En este caso no hay restricciones sobre el tipo de variable, y además se consideran sólo las distancias o similitudes de entre observaciones.
 - Se reemplazan entonces las medias, por observaciones que estén en el centro de los grupos...con esto se soluciona el problema de los *outliers*, que tienden a distorsionar la distribución de los datos de un grupo.

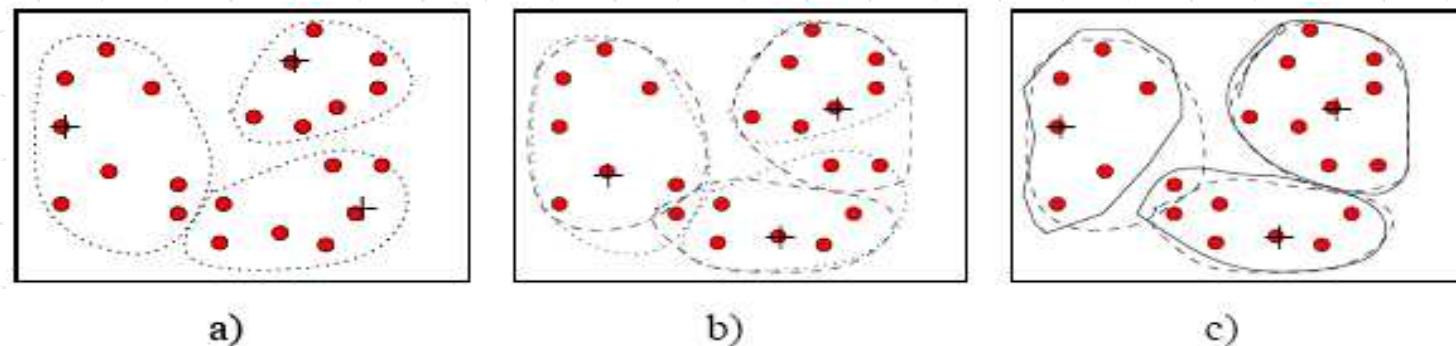


Figure 8.4: Clustering of a set of points based on the k -medoids method

Segmentación

Algoritmos de Particionamiento

- Algoritmo K-medoids: desarrolla dos pasos...
 - Dada una partición C , encontrar la observación en el grupo que minimice la distancia total a los otros puntos del grupo:

$$i_{k^*} = \arg \min_{\{i: C(i) = k\}} \sum_{C(i')=k} D(x_i, x_{i'})$$

Así, $m_k = x_{ik^*}$ es la estimación del centro del grupo

- Dados las centros $\{m_1, \dots, m_K\}$, asignar cada observación al grupo cuyo centro es el más cercano:

$$C(i) = \arg \min_{k=1..K} \|x_i - m_k\|^2$$

- Dichos pasos se repiten hasta que las asignaciones no cambien.

Segmentación

Algoritmos de Particionamiento

- Algoritmo K-medoids:

Input: The number of clusters k , and a database containing n objects.

Output: A set of k clusters which minimizes the sum of the dissimilarities of all the objects to their nearest medoid.

Method: The k -medoids algorithm is implemented as follows.

- 1) arbitrarily choose k objects as the initial medoids;
- 2) repeat
- 3) assign each object to the cluster corresponding to the nearest medoid;
- 4) calculate the objective function, which is the sum of dissimilarities
 of all the objects to their nearest medoid;
- 5) swap the medoid x by an object y if such a swap reduces the objective function;
- 6) until no change;

Segmentación

Algoritmos Jerárquicos

Crean una descomposición jerárquica del conjunto de datos dado. Enfoques.

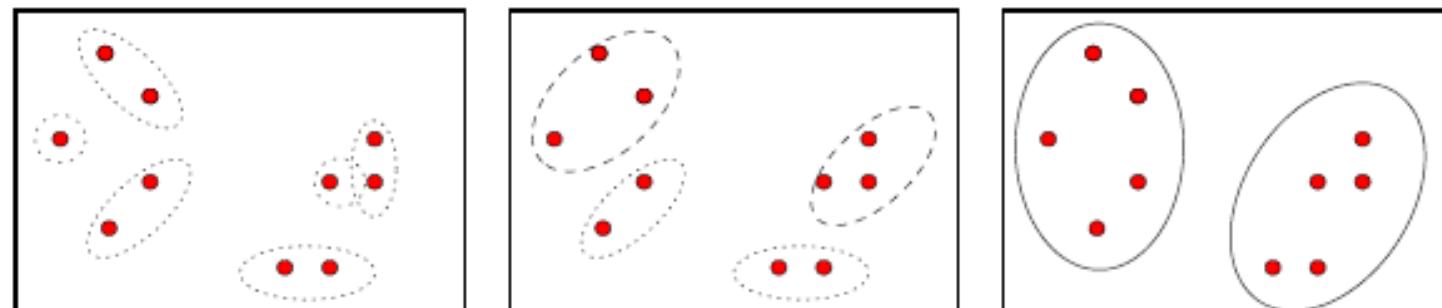
→ Aglomerativo (*bottom-up*): empieza con cada objeto formando un grupo separado; sucesivamente mezcla los (grupos de) objetos cercanos entre sí, hasta que se cumpla cierta condición dada.

→ Divisivo (*top-down*): empieza con todos los objetos en el mismo grupo; en cada iteración sucesiva, un grupo es dividido en otros más pequeños, hasta que eventualmente se cumpla cierta condición dada.

Segmentación

Algoritmos Jerárquicos

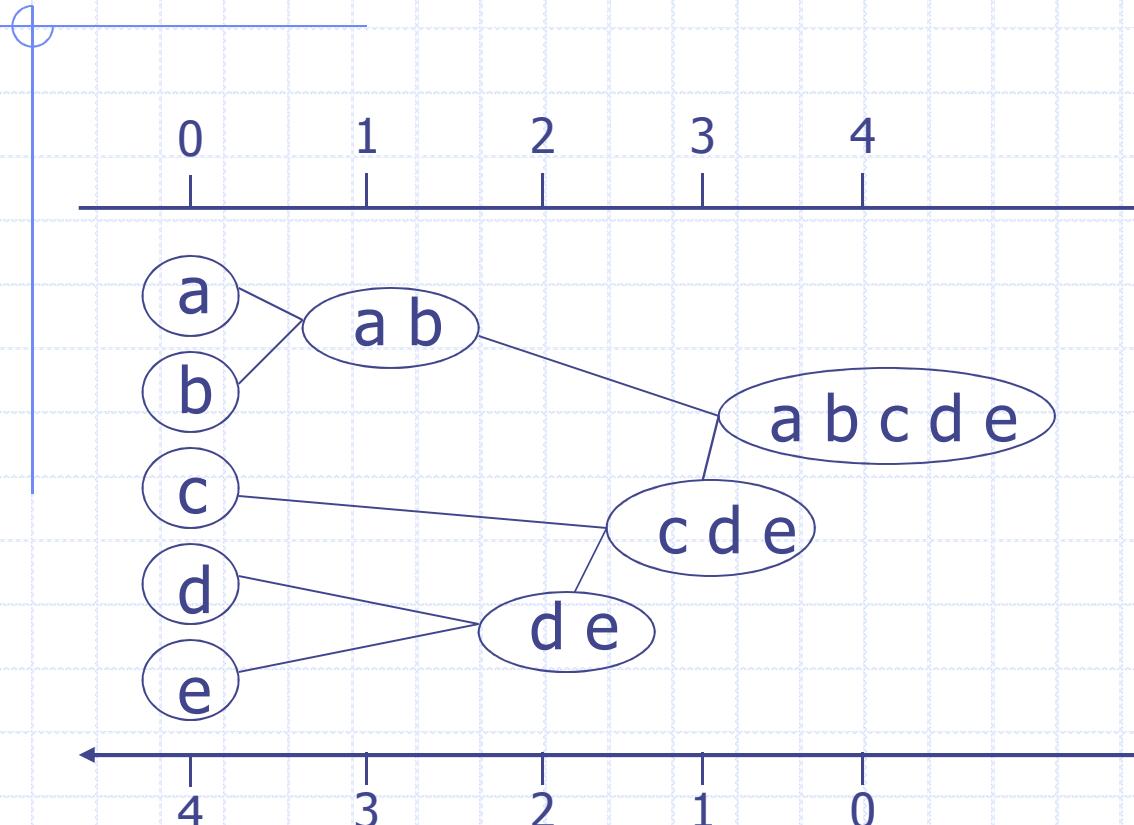
Enfoque Aglomerativo (*bottom-up*)



Enfoque Divisivo (*top-down*)

Segmentación

Algoritmos Jerárquicos



Algoritmo
Aglomerativo: AGNES
(AGglomerative NESting)

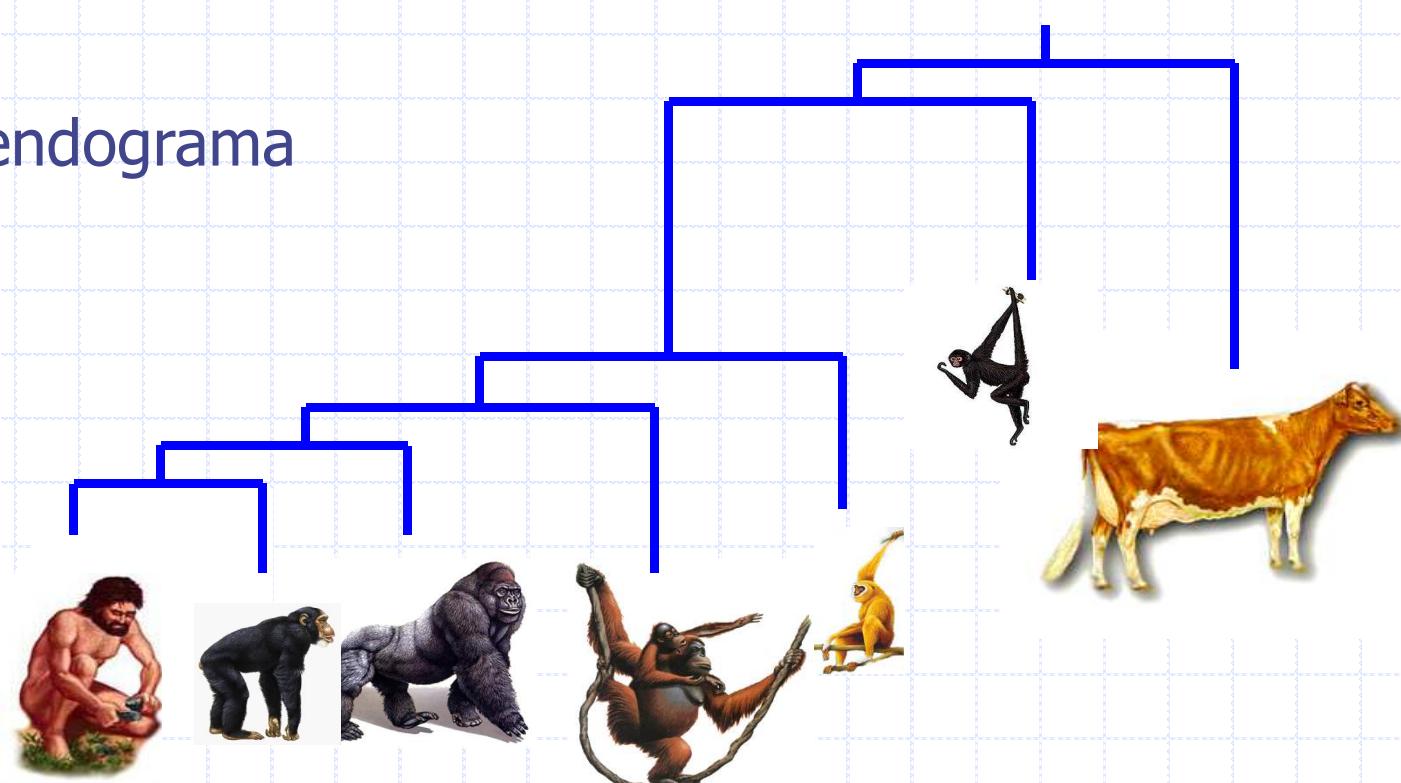
Algoritmo
Divisivo: DIANA
(DIvisive ANAlysis)

Cuál es el criterio de detención?

Segmentación

Algoritmos Jerárquicos

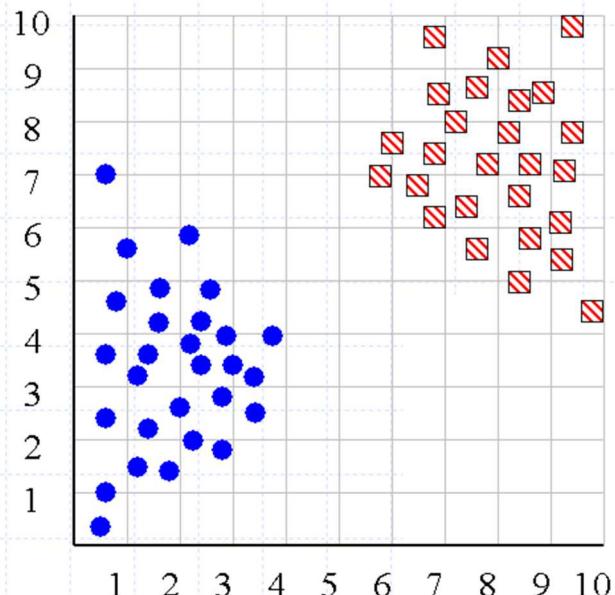
Dendograma



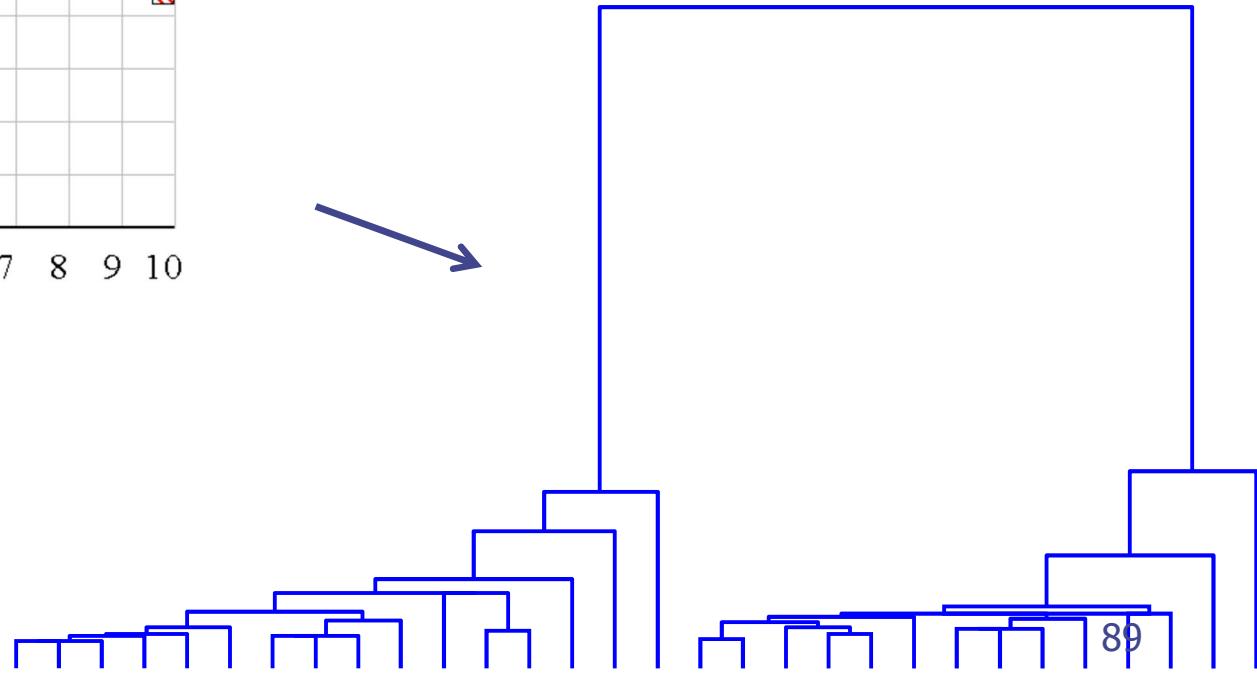
La similitud entre dos objetos viene dada por la “altura” del nodo común más cercano.

Segmentación

Algoritmos Jerárquicos



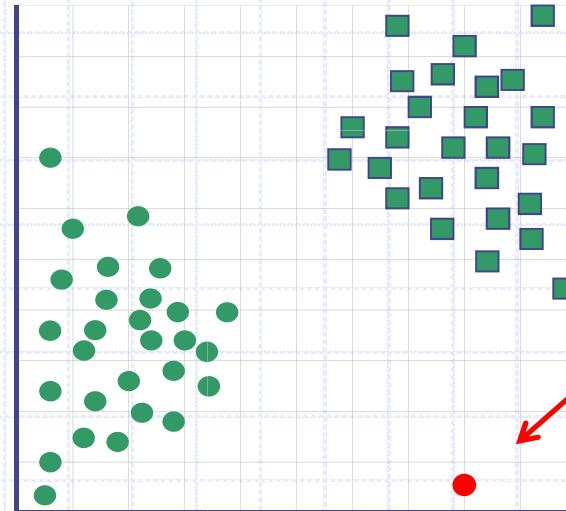
El dendograma puede ayudar a determinar el número de grupos...



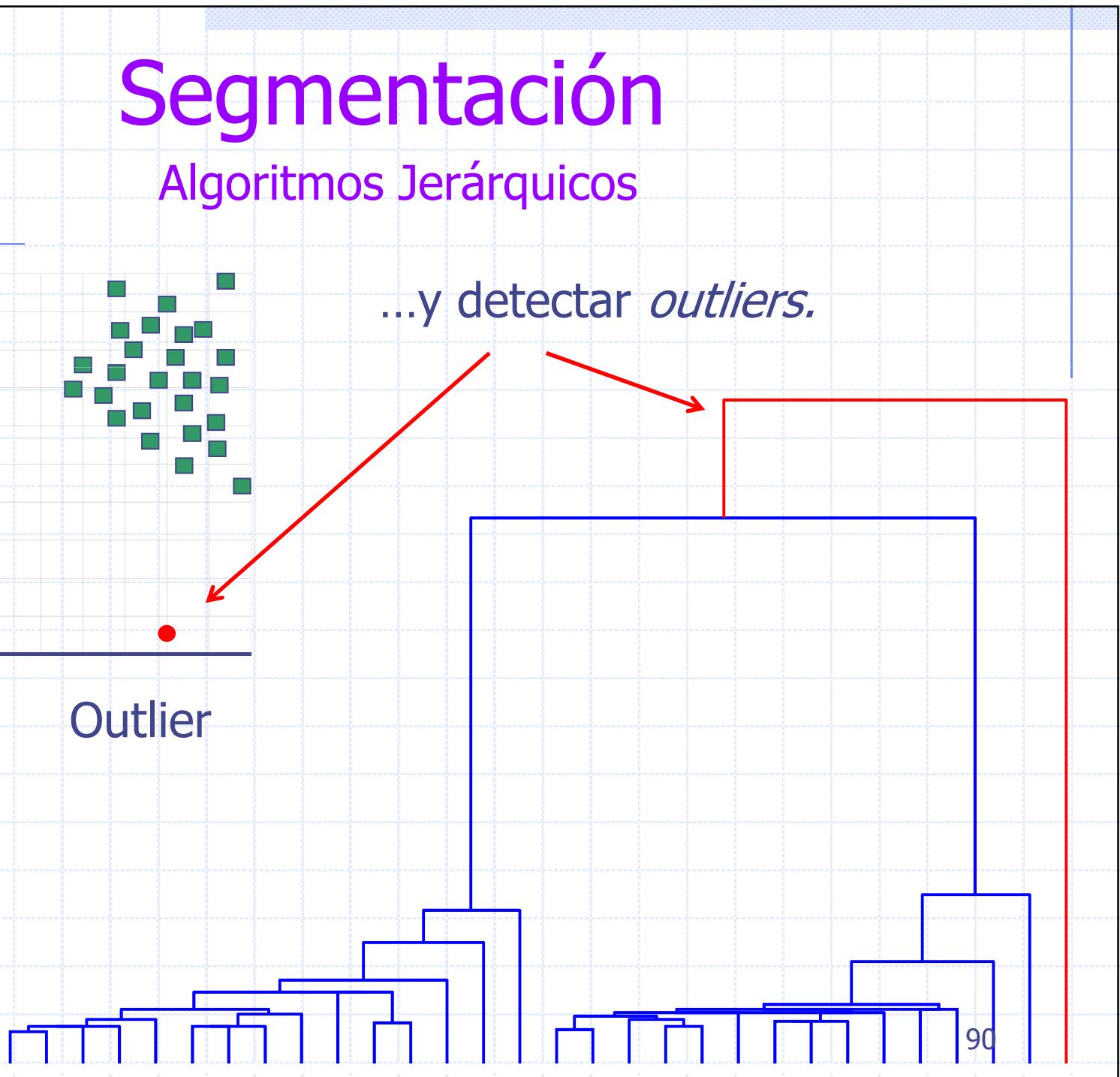
Segmentación

Algoritmos Jerárquicos

...y detectar *outliers*.



Outlier



Segmentación

Algoritmos Jerárquicos

Para construir un dendograma:

1. Calcular las distancias entre todos los pares de objetos – esto equivale a asumir que cada objeto constituye un grupo por si solo $\{C_1, \dots, C_N\}$.
2. Buscar los dos grupos más cercanos (C_i, C_j), y juntarlos para dejarlos como un único grupo.
3. Repetir el paso 2 hasta que no queden pares de comparación.

En general, la representación es mediante un árbol.

Segmentación

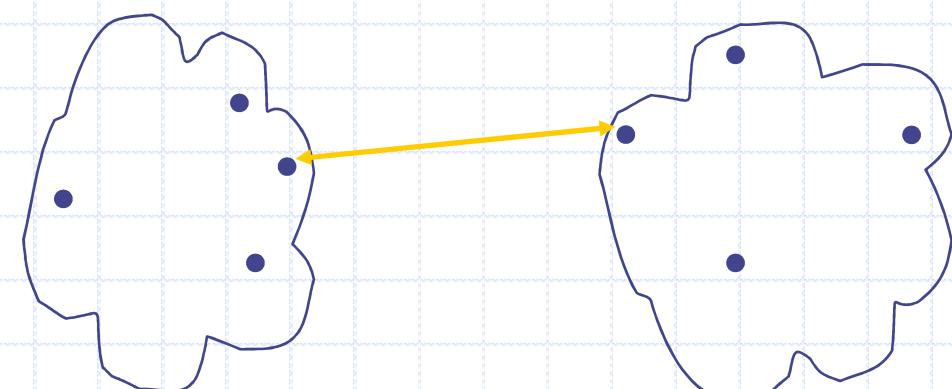
Algoritmos Jerárquicos

Para medir la distancia entre grupos...

- **MIN**

Enlace simple

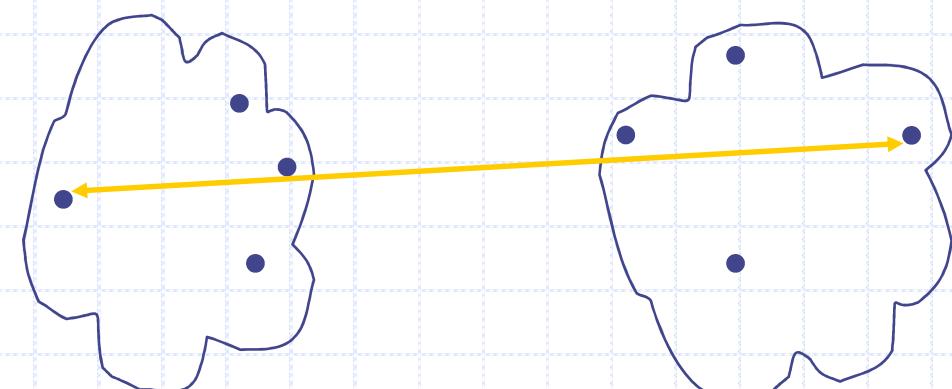
$$\min_{x_i \in S_i, x_j \in S_j} d(x_i, x_j)$$



- **MAX**

enlace completo
(diámetro)

$$\max_{x_i \in S_i, x_j \in S_j} d(x_i, x_j)$$



Segmentación

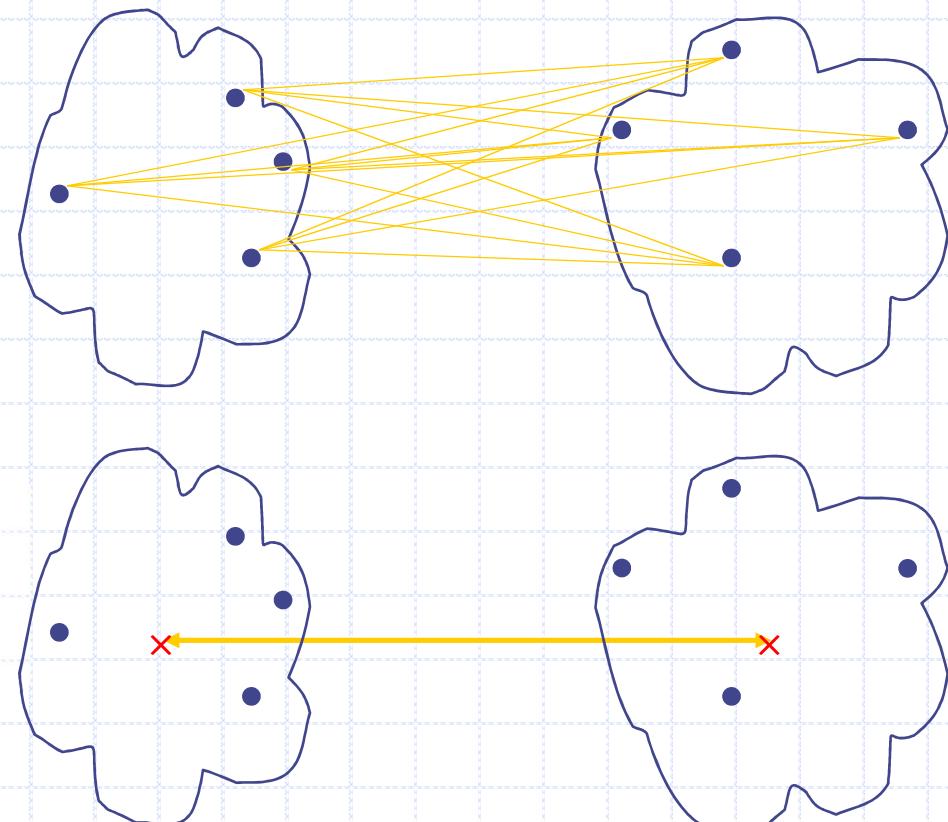
Algoritmos Jerárquicos

Para medir la distancia entre grupos...

- **PROMEDIO**
Enlace promediado

$$\frac{1}{|S_i||S_j|} \sum_{x_i \in S_i} \sum_{x_j \in S_j} d(x_i, x_j)$$

- basado en CENTROIDES
Ej.: **BIRCH**.



Segmentación

Algoritmos Jerárquicos

Ejercicio 1:

D1							
D2	0.3606						
D3	0.5000	0.4243					
D4	0.9220	0.7071	0.4472				
D5	1.3416	1.0440	0.9220	0.5000			
D6	1.8385	1.5524	1.3892	0.9434	0.5099		
D7	1.7263	1.5000	1.2369	0.8062	0.5831	0.4000	
D1	D2	D3	D4	D5	D6	D7	

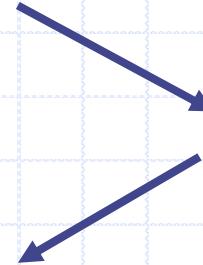
→

D1/D2=D8							
D3	0.4243						
D4	0.7071	0.4472					
D5	1.3416	0.9220	0.5000				
D6	1.5524	1.3892	0.9434	0.5099			
D7	1.5000	1.2369	0.8062	0.5831	0.4000		
D1/D2=D8	D3	D4	D5	D6	D7		

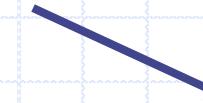
→

D8							
D3	0.4243						
D4	0.7071	0.4472					
D5	1.3416	0.9220	0.5000				
D6/D7=D9	1.5000	1.2369	0.8062	0.5831			
D8	D3	D4	D5	D6/D7=D9			

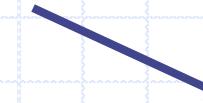
D8							
D3	0.4243						
D4	0.7071	0.4472					
D5	1.3416	0.9220	0.5000				
D6/D7=D9	1.5000	1.2369	0.8062	0.5831			
D8	D3	D4	D5	D6/D7=D9			



D3/D8=D10							
D4	0.4472						
D5	0.9220	0.5000					
D9	1.2369	0.8062	0.5831				
D3/D8=D10	D4	D5	D9				



D4/D10=D11							
D5	0.5000						
D9	0.8062	0.5831					
D4/D10=D11	D5	D9					



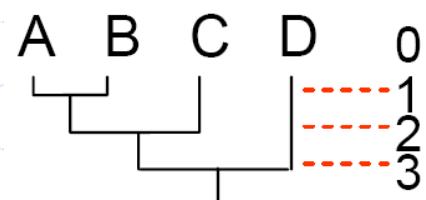
D5/D11							
D9	0.5831						
D5/D11	D9						

Segmentación

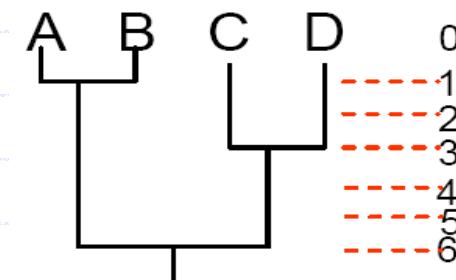
Algoritmos Jerárquicos

Ejercicio 2: aplicar un algoritmo aglomerativo para agrupar los datos descritos en la siguiente tabla de distancias.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0



Enlace único

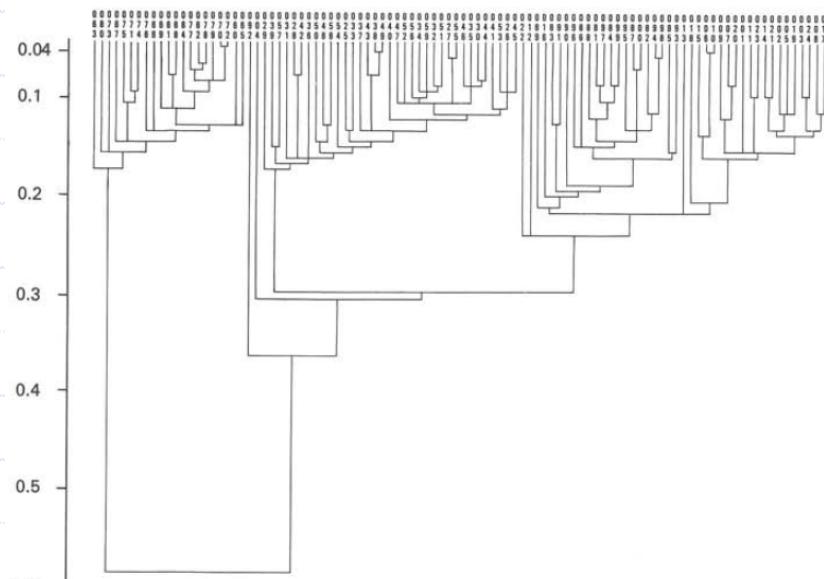


Enlace completo

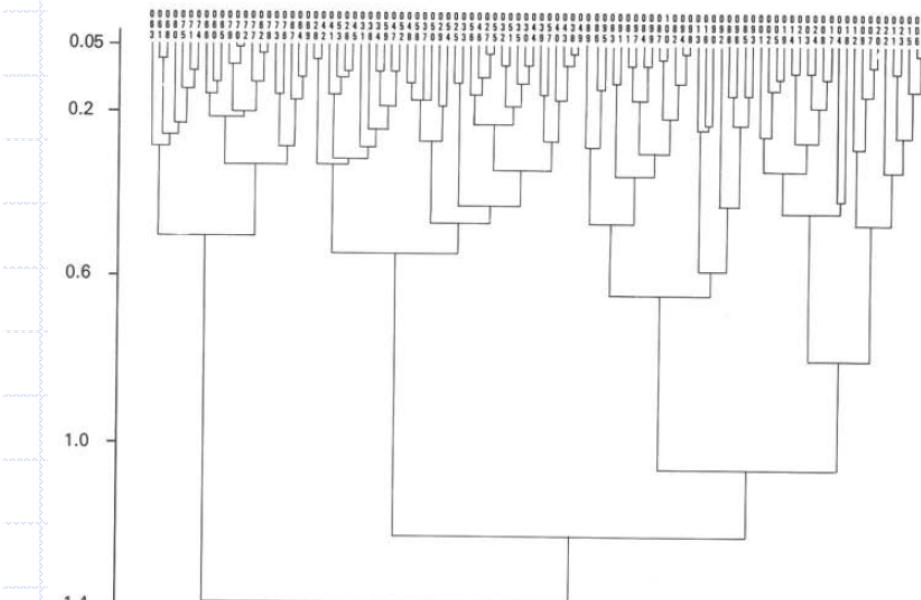
Segmentación

Algoritmos Jerárquicos

Ejemplo gráfico de dendograma: para cuatro grupos.



Enlace único

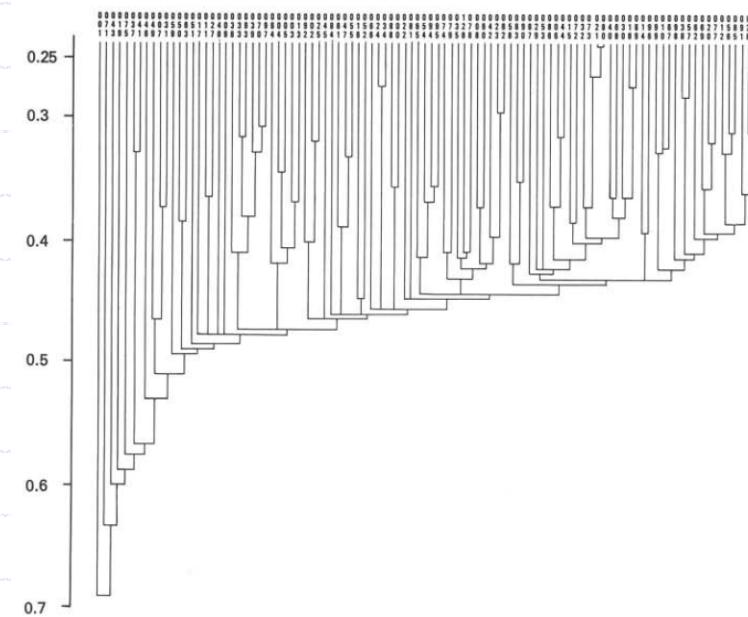


Enlace completo

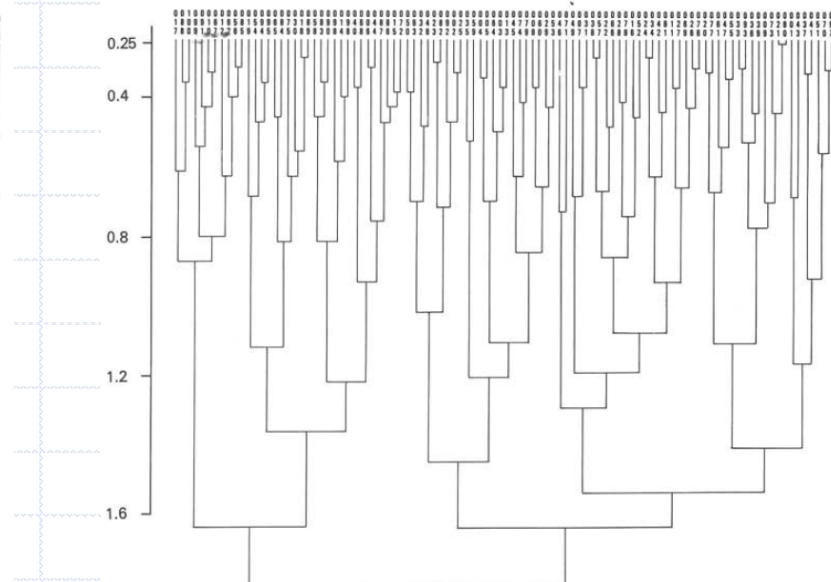
Segmentación

Algoritmos Jerárquicos

Ejemplo gráfico de dendograma: para datos aleatorios.



Enlace único



Enlace completo

Segmentación

Algoritmos Jerárquicos

CHAMELEON.

Grupos finales

Partición del grafo

**Combinar
particiones**

Segmentación

Algoritmos basados en la Densidad

Criterio de agrupamiento local: **densidad** de los puntos.

- Objetivo: identificar regiones densas de puntos separadas de otras también densas, por regiones de baja densidad.
- Características:
 - Identifica grupos de formas arbitrarias.
 - Robustos ante la presencia de ruido.
 - Escalables (realizan un único recorrido por los datos).

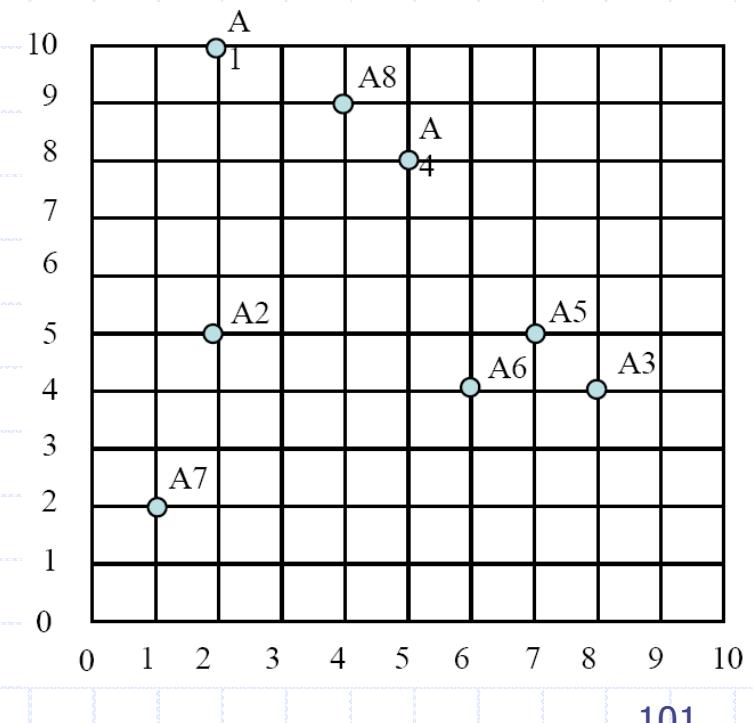
Segmentación

Algoritmos basados en la Densidad

- Algoritmo **DBSCAN**: ejemplo.

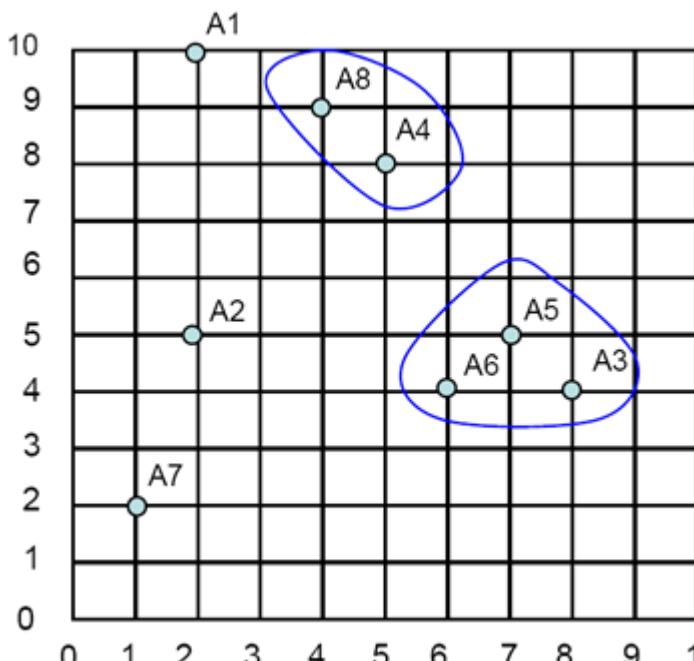
- Parámetro: número de puntos en el vecindario → 2

- Parámetro: radio del vecindario → $\sqrt{2}$, $\sqrt{10}$



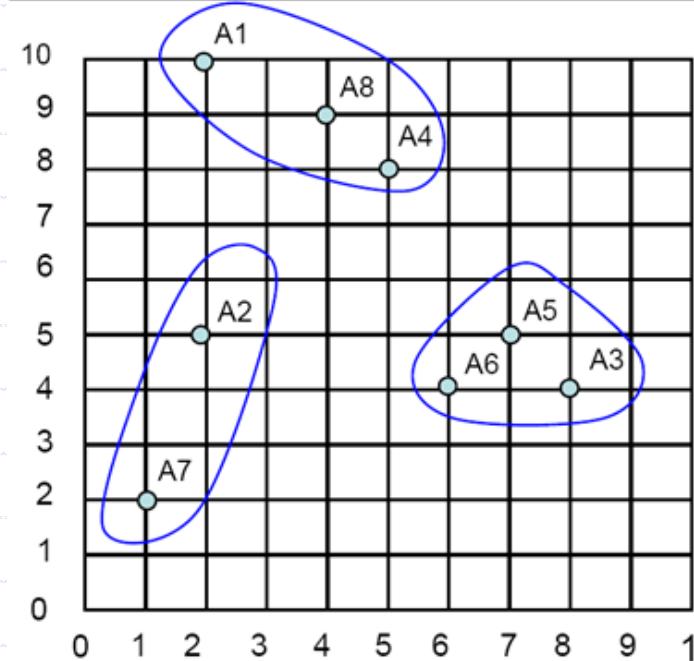
Segmentación

Algoritmos basados en la Densidad



Epsilon: $\sqrt{2}$

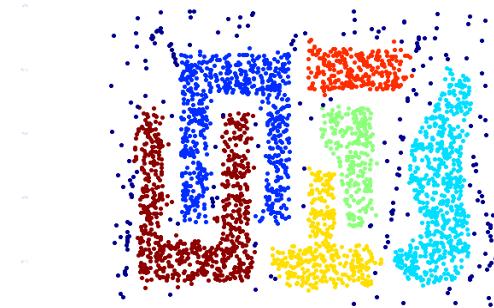
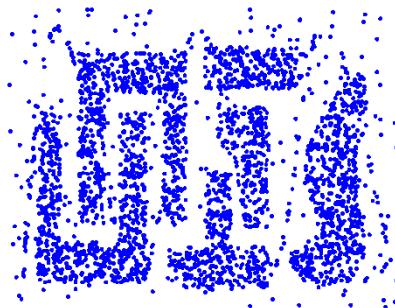
A1, A2 y A7 no tienen vecinos en su vecindario, por lo que se consideran "outliers" (no están en zonas densas).



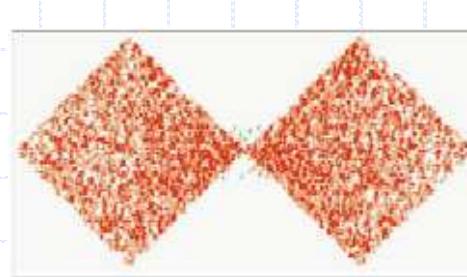
Epsilon: $\sqrt{10}$

Segmentación

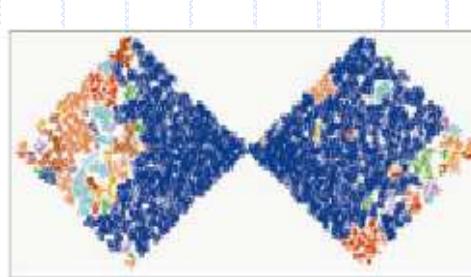
Algoritmos basados en la Densidad



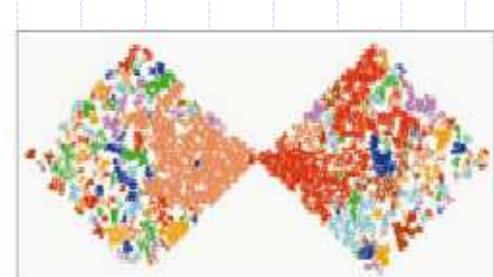
DBSCAN cuando
funciona bien...



(a)



(b)



(c)

...pues es sensible al valor inicial de sus parámetros

Tarea Predictiva: Clasificación

Clasificación

- Es el proceso de encontrar un modelo que describa y distinga clases de datos o conceptos, con el propósito de conocer la clase de otros objetos que aún no la tienen definida.
- En general:
 - Clasificación: predice el valor de un atributo categórico (discreto o nominal).
 - Predicción: construye funciones que toman valores continuos.
- Aunque sirve para conocer la clase de un objeto, en algunas aplicaciones puede predecir, en su lugar, algún valor “perdido” o no disponible.

Clasificación

- El modelo obtenido está basado en el análisis de un conjunto de datos de entrenamiento, que son objetos ya clasificados → **aprendizaje supervisado**.
- Para construir un modelo de clasificación:
 - Se divide el conjunto de datos disponible en un conjunto de entrenamiento (para construir el modelo) y un conjunto de prueba (para evaluar el modelo).
 - Se construye el modelo usando el conjunto de entrenamiento, y se valida con el conjunto de prueba, obteniéndose un porcentaje de clasificación asociado al número de aciertos obtenidos.
 - Si dicho porcentaje es aceptable, el modelo es considerado como útil para clasificar nuevos casos.

Clasificación

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Conjunto de entrenamiento

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Conjunto de prueba

Algoritmo
de
aprendizaje

Inducción

Aprender
modelo

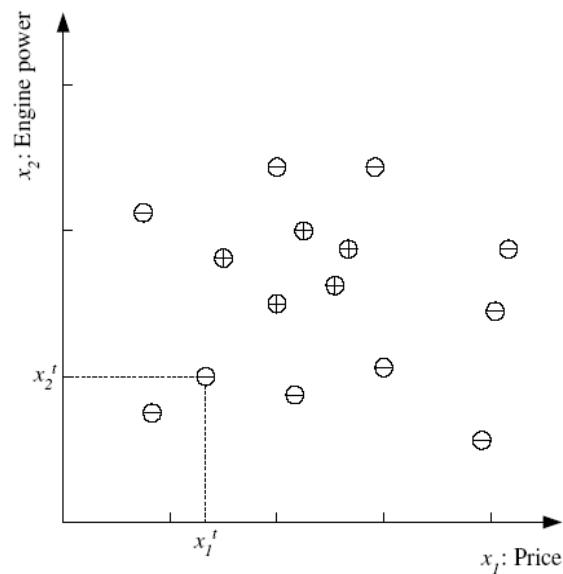
Modelo

Aplicar
modelo

Deducción

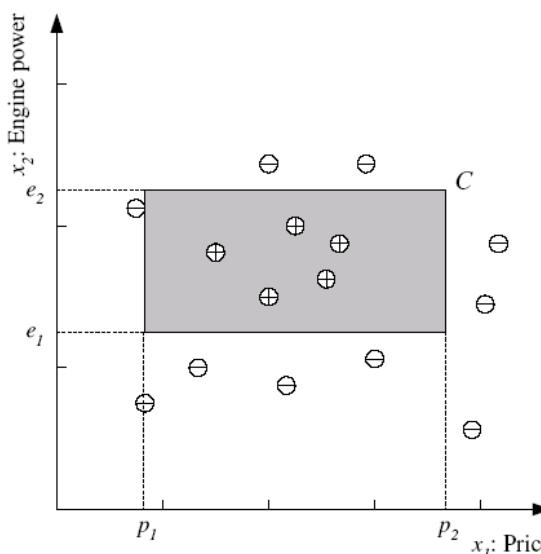
Clasificación

- Ejemplo: el ejemplo mas simple de aprendizaje supervisado es la generación de una regla de clasificación a partir de ejemplos positivos y negativos de una clase.
- Suponer que la clase en estudio es “Auto Familiar”; después de una encuesta entre usuarios de lo que define a un auto familiar, se distinguen dos criterios como los más mencionados: Precio y Potencia de la máquina (centímetros cúbicos de los cilindros).

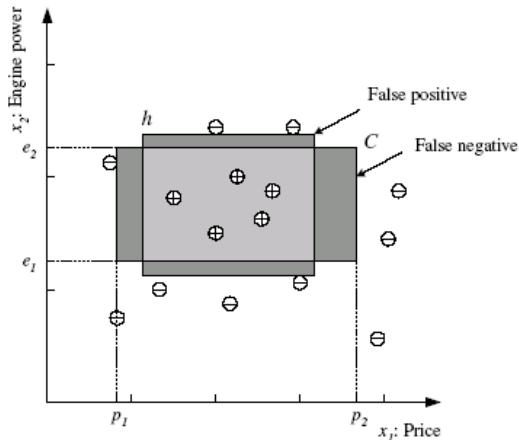


Clasificación

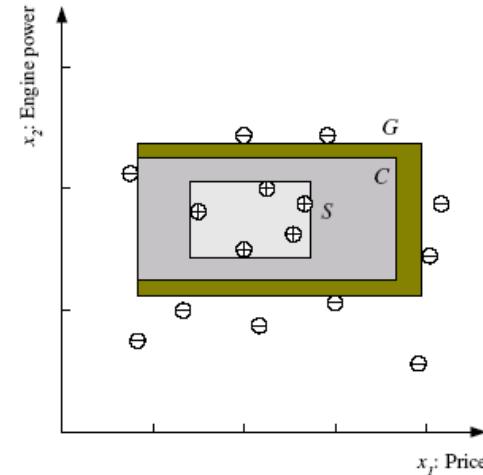
A partir de una discusión con los expertos y a partir de los datos, se podría inferir que los valores del Precio y Potencia debieran estar en un determinado rango para que el vehículo sea clasificado como “familiar”.



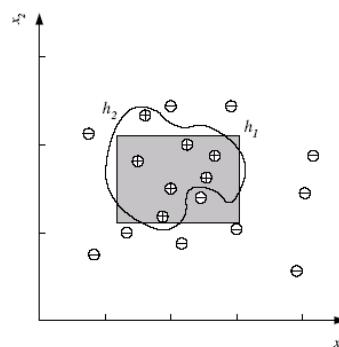
$$p_1 \leq p \leq p_2 \text{ y } e_1 \leq e \leq e_2$$



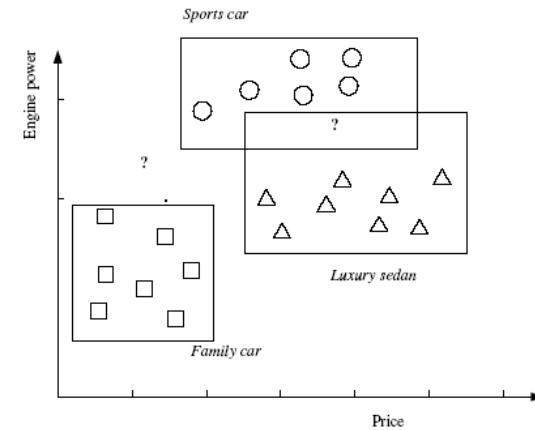
h es la hipótesis inducida y C es la verdadera clase.



Hipótesis más general vs hipótesis más específica.



¿Y si hay ruido?



¿Y si hay múltiples clases?



Algoritmos

Clasificación

- El modelo puede ser representado por varias formas:
 - Métodos básicos: ZeroR, OneR
 - Árboles de clasificación.
 - Reglas de clasificación (if-then).
 - Métodos bayesianos.
 - Redes neuronales artificiales.
 - ...
 - ...

Clasificación

Métodos Básicos

- Algoritmo **ZeroR** (Zero-attribute-rule): sólo retorna el valor de la clase de mayor frecuencia.
- Algoritmo **OneR** (One-attribute-rule): genera todas las reglas en base a un mismo atributo en el antecedente..

```
For each attribute A:  
  For each value V of that attribute, create a rule:  
    1. count how often each class appears  
    2. find the most frequent class, c  
    3. make a rule "if A=V then C=c"  
  Calculate the error rate of this rule  
  Pick the attribute whose rules produce the lowest error rate
```

Clasificación

Métodos Básicos

- Ejemplo de ZeroR y OneR.

Predictors				Target
Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

ZeroR

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Play Golf	
No	
No	
Yes	
Yes	
Yes	
No	
Yes	
No	
Yes	
No	

Sort

Play Golf	
No	
Yes	

$$5 / 14 = 0.36$$



$$9 / 14 = 0.64$$

OneR

IF Outlook = Sunny THEN PlayGolf = Yes
IF Outlook = Overcast THEN PlayGolf = Yes
IF Outlook = Rainy THEN PlayGolf = No

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

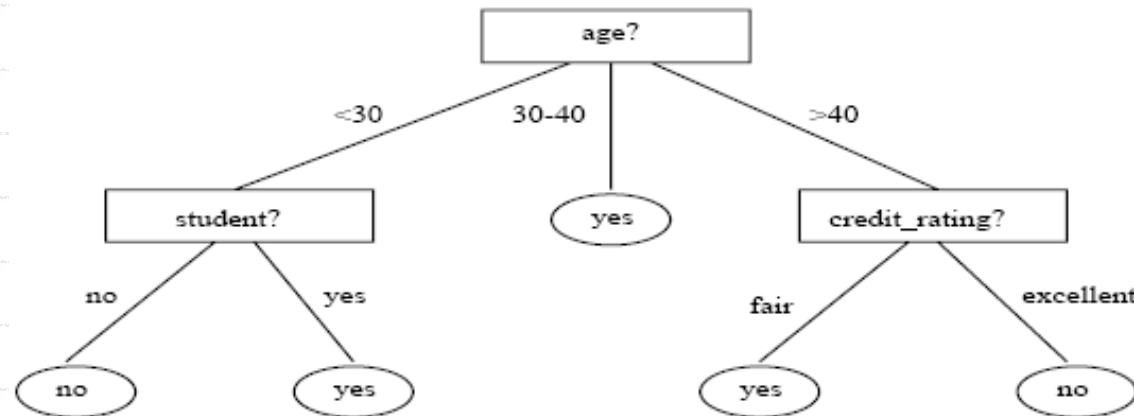
		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Clasificación

Árboles de Clasificación

- Árboles de Clasificación (o de decisión).

- Estructura similar a un diagrama de flujo, donde cada nodo interno denota una condición sobre un atributo, cada enlace representa una salida de la misma, y cada nodo hoja representa las clases.



- Pueden ser fácilmente convertidos en reglas de clasificación.
- Algoritmos más comunes: ID3, C4.5 (J48 de Weka).

Clasificación

Árboles de Clasificación

- Construcción: por lo general, una estrategia del tipo *dividir y conquistar*.
 - Se comienza con todos los ejemplos de entrenamiento en la raíz del árbol.
 - Los ejemplos se van dividiendo en función del atributo que se seleccione para ramificar el árbol en cada nodo.
 - Los atributos que se usan para ramificar se eligen en función de una heurística.

Clasificación

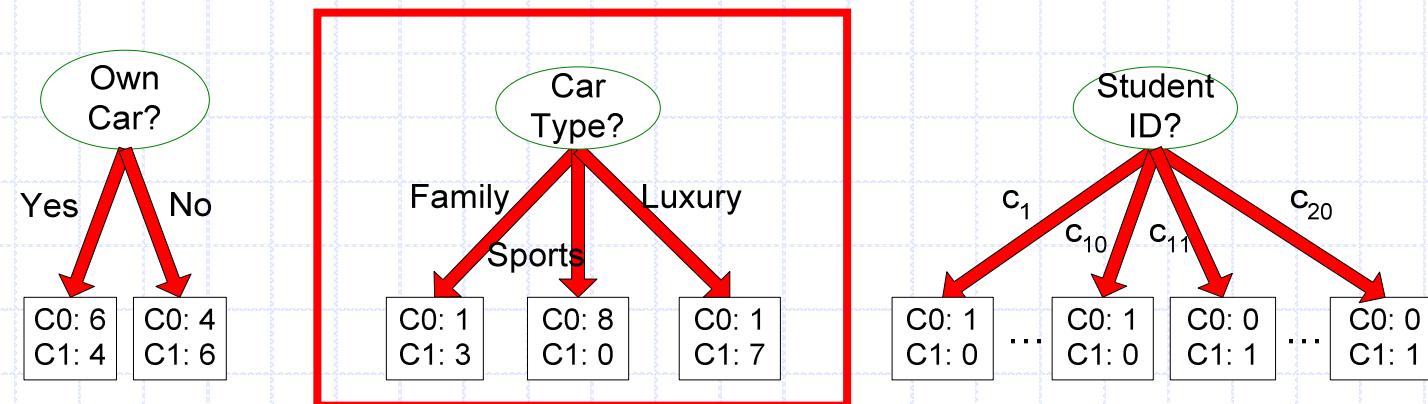
Árboles de Clasificación

- Construcción: posibles criterios de detención...
 - Todos los ejemplos que quedan pertenecen a la misma clase (se añade una hoja al árbol con la etiqueta de la clase).
 - No quedan más atributos por ramificar (se añade una hoja etiquetada con la clase más frecuente en el nodo).
 - No hay más datos que clasificar.

Clasificación

Árboles de Clasificación

- Construcción: Heurísticas
 - La heurística a escoger para seleccionar el atributo por el cual ramificar debe ser aquélla que entregue nodos más homogéneos.



- Ejemplos de Heurísticas:
 - Ganancia de Información (ejs.: algoritmos **ID3, C4.5**)
 - Índice de Gini (ejs.: algoritmos **CART, SLIQ, SPRINT**)
 - Otras como χ^2 , MDL (Minimum Description Length).

Clasificación

Árboles de Clasificación

- Construcción: Heurísticas (2)
 - **Ganancia de Información:** referida a una medida de la "bondad" de la división...a mayor ganancia de información, mayor reducción de la entropía.
 - Información esperada para clasificar una muestra:

$$I(s_1, s_2, \dots, s_m) = - \sum_{i=1}^m p_i \log_2(p_i)$$

- Entropía:

$$E(A) = \sum_{j=1}^v \frac{s_{1j} + \dots + s_{mj}}{s} I(s_{1j}, \dots, s_{mj}).$$

- Reducción esperada en la entropía causada al conocer el valor del atributo A:

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A)$$

Clasificación

Árboles de Clasificación

- Construcción: Heurísticas (3)
 - **Índice de Gini:** es una medida de la impureza.

$$gini(D) = 1 - \sum_{j=1}^n p_j^2$$

- Se escoge aquel atributo que entrega la mayor reducción de la impureza.

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

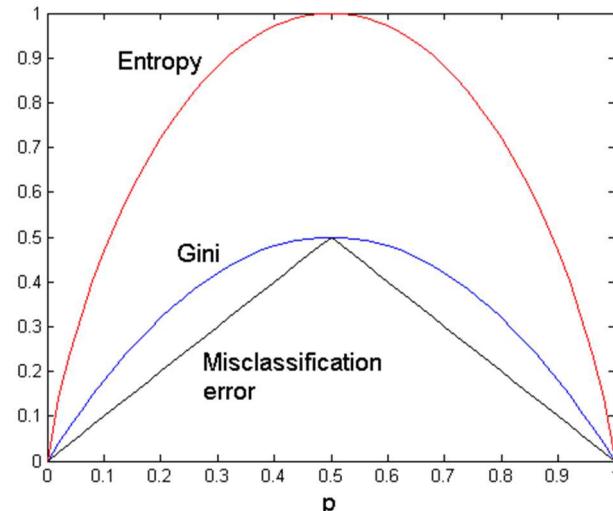
C1	2
C2	4
Gini=0.444	

C1	3
C2	3
Gini=0.500	

Clasificación

Árboles de Clasificación

- Construcción: Heurísticas (4)



Para problemas con
dos clases.

- Ganancia de información: sesgado hacia atributos con muchos valores diferentes.
- Índice de Gini: funciona peor cuando hay muchas clases y tiende a favorecer particiones de tamaño y pureza similares.

Clasificación

Árboles de Clasificación

- **ID3:** algoritmo básico para la inducción de árboles de decisión.

Algorithm 7.3.1 (Generate_decision_tree) Generate a decision tree from the given training data.

Input: The training samples, *samples*, represented by discrete-valued attributes; the set of candidate attributes, *attribute-list*.

Output: A decision tree.

Method:

- 1) create a node *N*;
- 2) if *samples* are all of the same class, *C* then
- 3) return *N* as a leaf node labeled with the class *C*;
- 4) if *attribute-list* is empty then
- 5) return *N* as a leaf node labeled with the most common class in *samples*; // majority voting
- 6) select *test-attribute*, the attribute among *attribute-list* with the highest information gain;
- 7) label node *N* with *test-attribute*;
- 8) for each known value *a_i* of *test-attribute* // partition the samples
- 9) grow a branch from node *N* for the condition *test-attribute=a_i*;
- 10) let *s_i* be the set of samples in *samples* for which *test-attribute=a_i*; // a partition
- 11) if *s_i* is empty then
- 12) attach a leaf labeled with the most common class in *samples*;
- 13) else attach the node returned by *Generate_decision_tree(s_i, attribute-list - test-attribute)*;

- Ejemplo: Sean $C_1 = \text{yes}$, $C_2 = \text{no}$.

age	income	student	credit_rating	buys_computer
<30	high	no	fair	no
<30	high	no	excellent	no
30-40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
30-40	low	yes	excellent	yes
<30	medium	no	fair	no
<30	low	yes	fair	yes
>40	medium	yes	fair	yes
<30	medium	yes	excellent	yes
30-40	medium	no	excellent	yes
30-40	high	yes	fair	yes
>40	medium	no	fair	no

$$I(s_1, s_2) = I(9, 5) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940$$

for $age = \text{"<30"}$: $s_{11} = 2$ $s_{21} = 3$ $I(s_{11}, s_{21}) = 0.971$
 for $age = \text{"30-40"}$: $s_{12} = 4$ $s_{22} = 0$ $I(s_{12}, s_{22}) = 0$
 for $age = \text{">40"}$: $s_{13} = 3$ $s_{23} = 2$ $I(s_{13}, s_{23}) = 0.971$

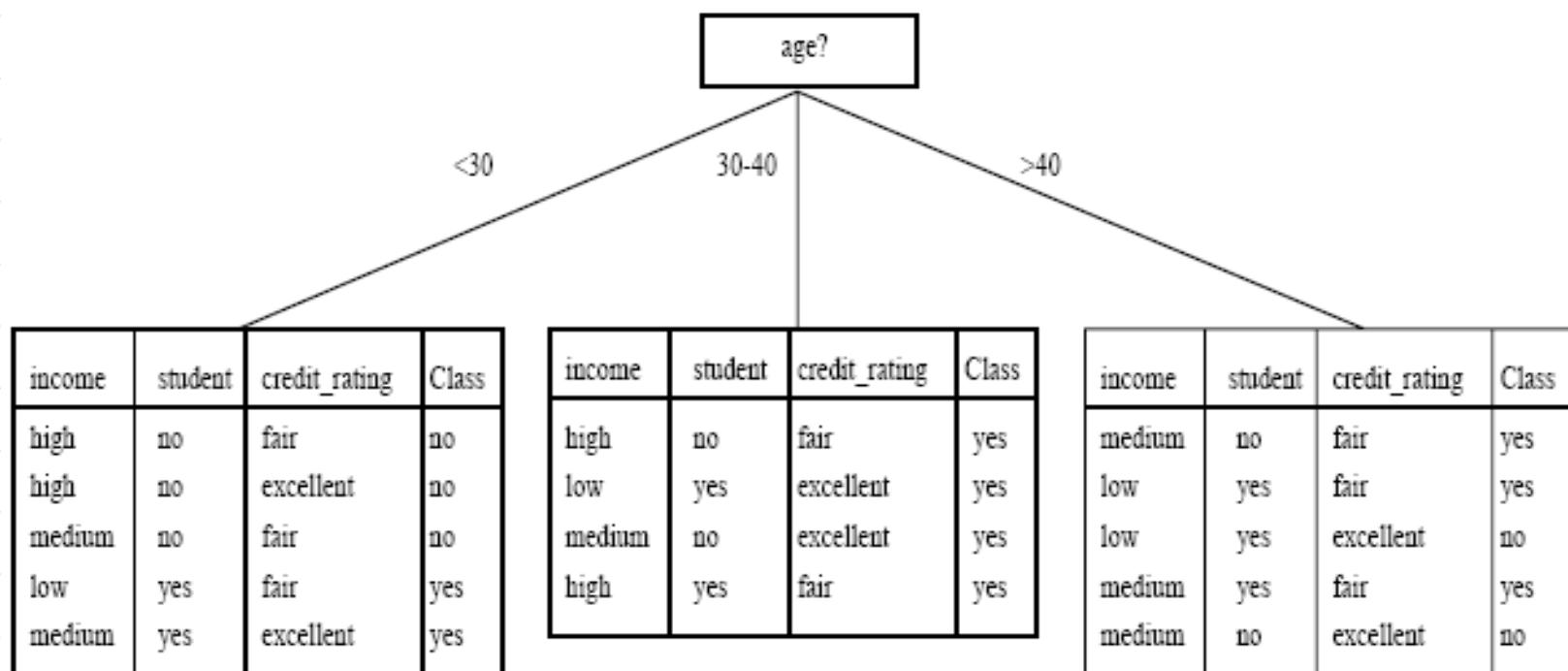
$$E(\text{age}) = \frac{5}{14}I(s_{11}, s_{21}) + \frac{4}{14}I(s_{12}, s_{22}) + \frac{5}{14}I(s_{13}, s_{23}) = 0.694.$$

$$\text{Gain}(\text{age}) = I(s_1, s_2) - E(\text{age}) = 0.246$$

Clasificación

Árboles de Clasificación

- Ejemplo: continuación...



Clasificación

Árboles de Clasificación

- Poda del Árbol:
 - Al construir un árbol de decisión, varias ramas reflejarán anomalías en los datos de entrenamiento, debido a ruido o *outliers*.
 - Los métodos de poda direccionan este problema de **sobreajustar** los datos.
 - Típicamente se usan medidas estadísticas para remover las ramas menos confiables, generalmente resultando en una clasificación más rápida y una mejora en la habilidad de clasificar correctamente datos de prueba independientes.
- Enfoques comunes:
 - Poda Previa, Poda Posterior.
 - Poda por costo-complejidad (CART), poda pesimista (C4.5).

Clasificación

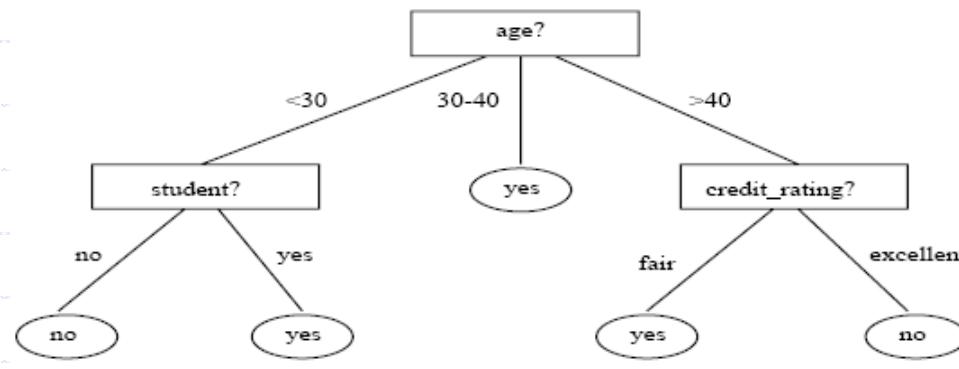
Reglas de Clasificación

- **Reglas de Clasificación:** existen diversas formas de obtener reglas.
 - A partir de un árbol de decisión
 - A través de algoritmos específicos de inducción de reglas (ejs.: STAR, Ripper)
 - A partir de reglas de asociación

Clasificación

Reglas de Clasificación

a) Extracción de Reglas de Clasificación a partir de un **Árbol de Decisión**: por cada camino que existe entre la raíz y una hoja del árbol.



IF $age = <30\text{ AND }student = no$
IF $age = <30\text{ AND }student = yes$
IF $age = 30-40$
IF $age = >40\text{ AND }credit_rating = excellent$
IF $age = >40\text{ AND }credit_rating = fair$

THEN $buys_computer = no$
THEN $buys_computer = yes$
THEN $buys_computer = yes$
THEN $buys_computer = no$
THEN $buys_computer = yes$

Las reglas son mutuamente excluyentes y exhaustivas.

Clasificación

Reglas de Clasificación

...Las reglas extraídas son mutuamente y exhaustivas. Si se simplifican...

- podrían dejar de ser mutuamente excluyentes (varias reglas serían válidas para un mismo ejemplo) → establecer un orden entre las reglas [lista de decisión] o realizar una votación.
- podrían dejar de ser exhaustivas (ninguna regla sea aplicable a un ejemplo concreto) → incluir una clase por defecto.

Clasificación

Reglas de Clasificación

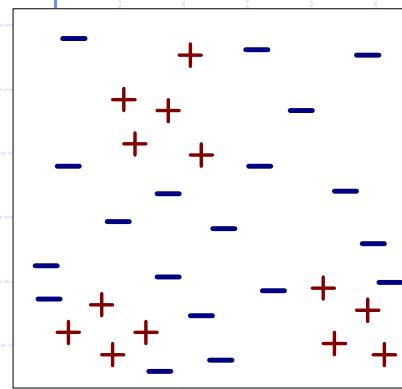
b) Obtención de Reglas de Clasificación mediante
Algoritmos de Inducción de Reglas.

- Las reglas se aprenden de una en una.
- Cada vez que se selecciona una regla, se eliminan del conjunto de entrenamiento todos los casos cubiertos por ella.
- El proceso se repite hasta que se cumpla alguna condición de detención.
- el aprendizaje comienza con la regla más general.
- ésta se le va agregando elementos a su antecedente para maximizar la “calidad” (cobertura, precisión).

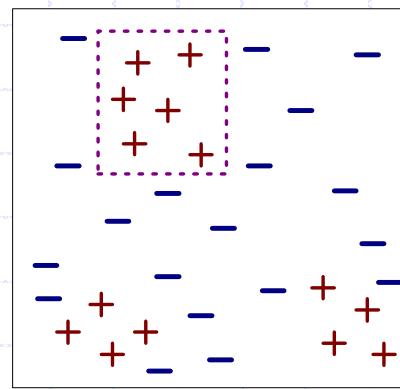
Clasificación

Reglas de Clasificación

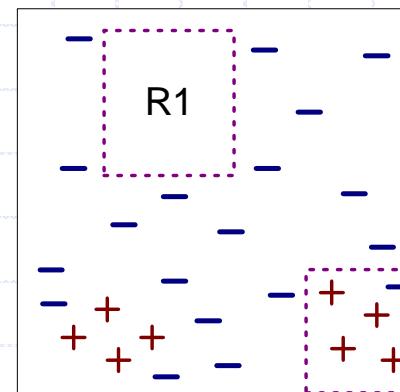
- Ejemplo:



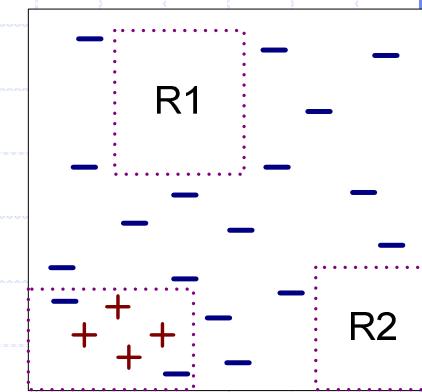
(i) Original Data



(ii) Step 1



(iii) Step 2



(iv) Step 3

- Algoritmos representativos: FOIL, CN2, RIPPER, PNRule.

Clasificación

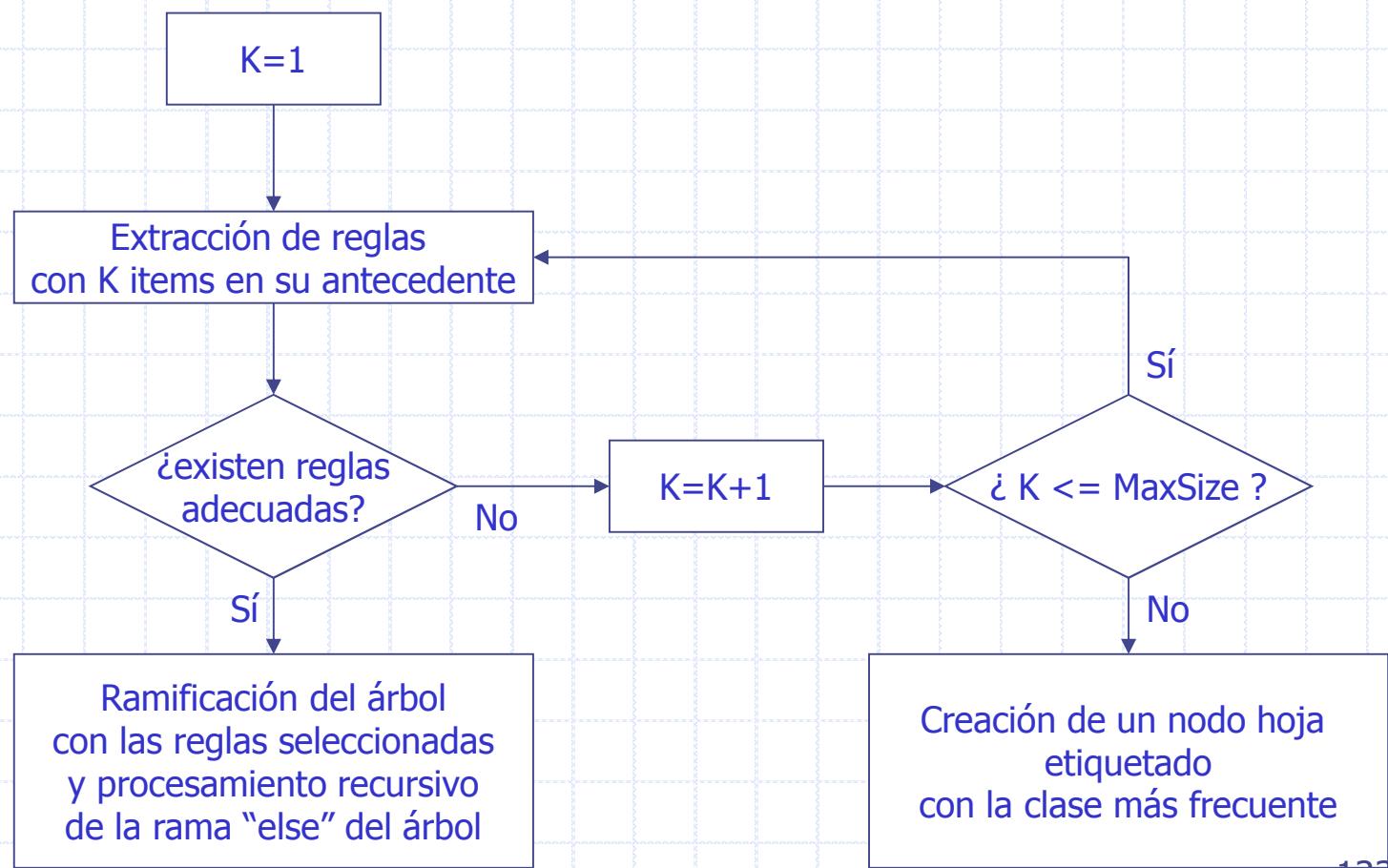
Reglas de Clasificación

- c) Reglas de Clasificación a partir de **Reglas de Asociación**: se buscan entre las mejores reglas de asociación, se superan algunas limitaciones de los árboles de decisión (que sólo consideran los atributos de uno en uno [y parcialmente]).
- Algunos algoritmos representativos: CBA, RCBT, CMAR, CPAR, ART.

Clasificación

Reglas de Clasificación

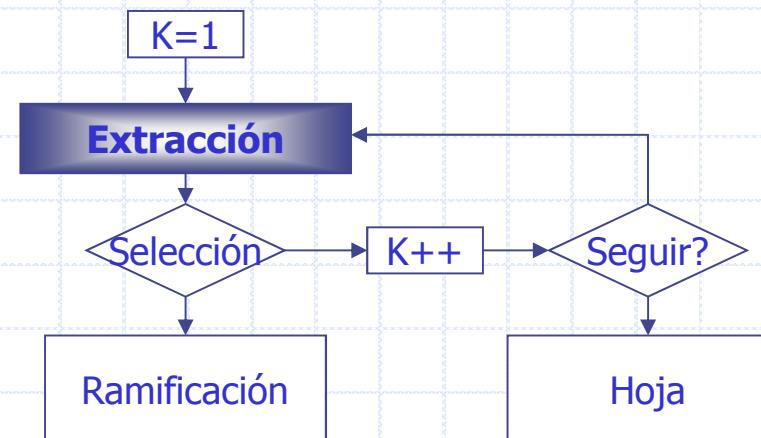
- Ejemplo: algoritmo ART.



Clasificación

Reglas de Clasificación

- Ejemplo: algoritmo **ART** (2).

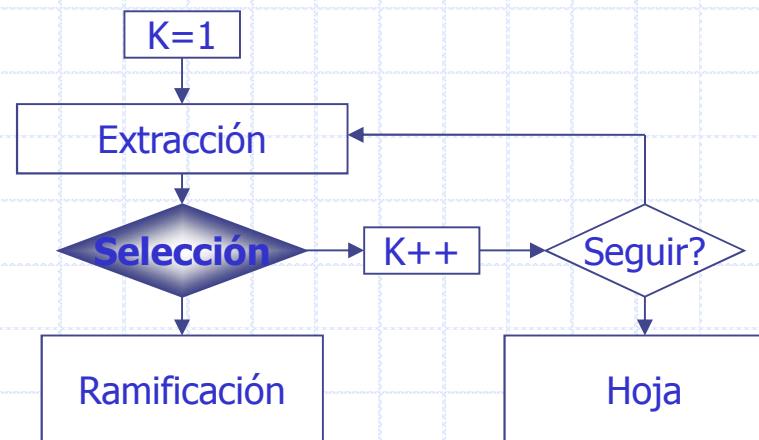


- a) Extracción de reglas: hipótesis candidatas
- Soporte mínimo
 - Confianza mínima

Clasificación

Reglas de Clasificación

- Ejemplo: algoritmo **ART** (3).



b) Selección de reglas:

- Reglas agrupadas por conjuntos de atributos
- Criterio de preferencia

Ejemplo:

Cliente	X ¿desempleado?	Y ¿con fondos?	Z ¿con inmuebles?	C ¿crédito?
I1	0	0	0	0
I2	0	0	1	0
I3	1	1	0	0
I4	1	0	0	0
I5	1	1	1	1
I6	1	0	1	1
I7	0	1	0	1
I8	0	1	1	1

NIVEL 1 - Extracción de reglas de asociación

- Umbral de soporte mínimo = 20%
- Selección automática del umbral de confianza

NIVEL 1, k = 1

S1: if (Y=0) then C=0 with confidence 75%
 if (Y=1) then C=1 with confidence 75%
S2: if (Z=0) then C=0 with confidence 75%
 if (Z=1) then C=1 with confidence 75%

NIVEL 1, k = 2

S1: if (X=0 and Y=0) then C=0 (100%)
 if (X=0 and Y=1) then C=1 (100%)
S2: if (X=1 and Z=0) then C=0 (100%)
 if (X=1 and Z=1) then C=1 (100%)
S3: if (Y=0 and Z=0) then C=0 (100%)
 if (Y=1 and Z=1) then C=1 (100%)

Cliente	X	Y	Z	C
	¿desempleado?	¿con fondos?	¿con inmuebles?	¿crédito?
I1	0	0	0	0
I2	0	0	1	0
I3	1	1	0	0
I4	1	0	0	0
I5	1	1	1	1
I6	1	0	1	1
I7	0	1	0	1
I8	0	1	1	1

NIVEL 1 - Selección del mejor conjunto de reglas p.ej. S1

NIVEL 1, k = 1

S1: if (Y=0) then C=0 with confidence 75%
 if (Y=1) then C=1 with confidence 75%
 S2: if (Z=0) then C=0 with confidence 75%
 if (Z=1) then C=1 with confidence 75%

NIVEL 1, k = 2

S1: if (X=0 and Y=0) then C=0 (100%)
 if (X=0 and Y=1) then C=1 (100%)

S2: if (X=1 and Z=0) then C=0 (100%)
 if (X=1 and Z=1) then C=1 (100%)

S3: if (Y=0 and Z=0) then C=0 (100%)
 if (Y=1 and Z=1) then C=1 (100%)

<i>Cliente</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	<i>C</i>
	¿desempleado?	¿con fondos?	¿con inmuebles?	¿crédito?
I1	0	0	0	0
I2	0	0	1	0
I3	1	1	0	0
I4	1	0	0	0
I5	1	1	1	1
I6	1	0	1	1
I7	0	1	0	1
I8	0	1	1	1



<i>Cliente</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	<i>C</i>
	¿desempleado?	¿con fondos?	¿con inmuebles?	¿crédito?
I3	1		1	0
I4	1		0	0
I5	1		1	1
I6	1		0	1

NIVEL 2 - Extracción de reglas de asociación

- Umbral de soporte mínimo = 20%
- Selección automática del umbral de confianza

NIVEL 2, k = 1

S1: if (*Z*=0) then *C*=0 with confidence 100%
 if (*Z*=1) then *C*=1 with confidence 100%

Resultado Final:

```

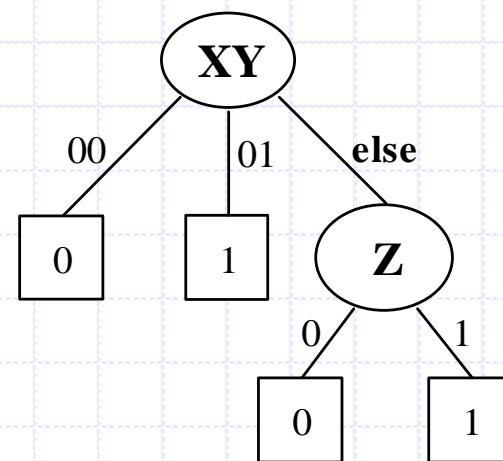
x=0 and Y=0: C=0 (2)
x=0 and Y=1: C=1 (2)
else
  Z=0: C=0      (2)
  Z=1: C=1      (2)
    
```

Clasificación

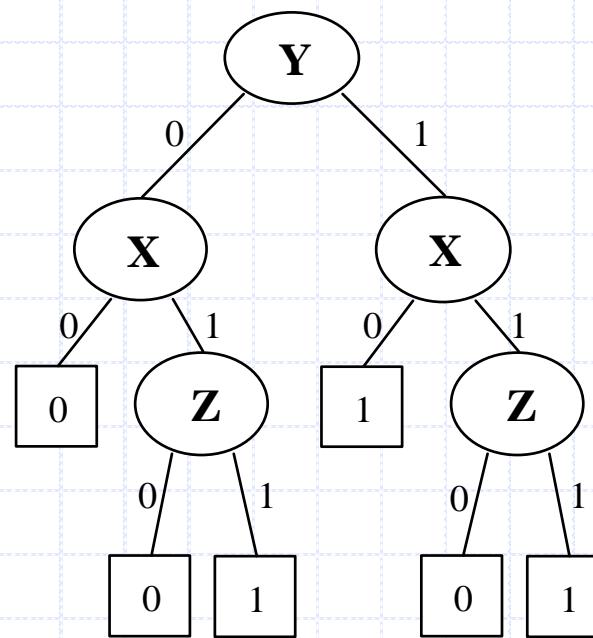
Reglas de Clasificación

- Reglas de Clasificación: los resultados de la aplicación de los algoritmos no siempre pueden coincidir.

ART



TDIDT



Clasificación

Métodos Bayesianos

- Los **clasificadores bayesianos** son clasificadores estadísticos, que pueden predecir las probabilidades de pertenencia a una clase. Particularmente:
 - Los clasificadores bayesianos ingenuos (*naive*) asumen que el efecto del valor de un atributo es independiente de los valores de los otros atributos, supuesto que es llamado *independencia condicional de clases*.
 - Las redes bayesianas permiten la representación de dependencias entre subconjuntos de atributos.

Clasificación

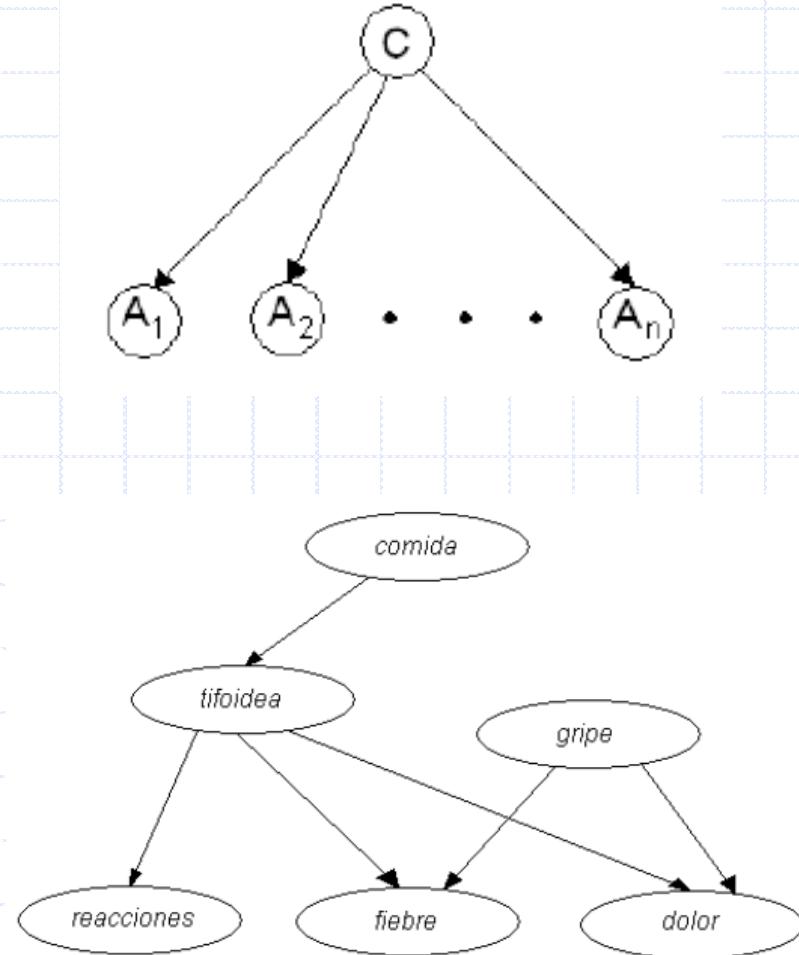
Métodos Bayesianos

- Uno de los principales problemas asociados a la minería de datos es cómo trabajar con incertidumbre.
- Lo anterior no es tal para los métodos y técnicas bayesianas, puesto que de sus principales características es el uso explícito de la teoría de la probabilidad para cuantificar dicha incertidumbre.
- Los métodos bayesianos permiten un doble uso:
 - descriptivo: se centran en el descubrimiento de relaciones de independencia y/o relevancia entre sus variables. Posteriormente, es posible un mejor estudio de las mismas mediante inferencia estadística.
 - predictivo: como clasificadores.

Clasificación

Métodos Bayesianos

- Ejemplos:
 - Clasificador bayesiano ingenuo.
 - Red bayesiana.



Clasificación

Métodos Bayesianos

- Se basan en el Teorema de Bayes:

$$P(h/O) = \frac{P(O/h) * P(h)}{P(O)}$$

donde:

- $P(h)$: probabilidad a priori de la hipótesis h .
- $P(O)$: probabilidad a priori de las observaciones.
- $P(O/h)$: probabilidad condicional que corresponde a la verosimilitud de que la hipótesis h haya producido el conjunto de observaciones O .

Clasificación

Métodos Bayesianos

- Teorema de Bayes: ejemplo de su aplicación.

Una fábrica produce un artículo en tres diferentes máquinas. Del total de la producción el 30% es producido en la maquina A, el 50% en la B y el 20% lo produce la máquina C. La probabilidad de que un artículo producido por una máquina específica sea de primera calidad, se muestra en la siguiente tabla :



Maquina	Probabilidad
A	0.8
B	0.7
C	0.9

Si se selecciona un artículo aleatoriamente de la linea de produccion:

- Cual es la probabilidad de que sea de primera calidad?
- Si el artículo seleccionado es de primera calidad, cual es la probabilidad de que haya sido producido por la maquina A?

Clasificación

Métodos Bayesianos

- En un problema de clasificación, con una variable de clase (C) y un conjunto de variables predictoras o atributos $\{A_1, \dots, A_n\}$, el teorema de Bayes adopta la forma general:

$$P(A_1, \dots, A_n | C) * P(C)$$
$$P(C | A_1, \dots, A_n) = \frac{P(A_1, \dots, A_n | C) * P(C)}{P(A_1, \dots, A_n)}$$

- Si C tiene k posibles valores $\{c_1, \dots, c_k\}$, interesa identificar el que tenga la mayor probabilidad a posterior, para devolverlo como resultado de la clasificación (llamada hipótesis MAP).

$$C_{MAP} = \arg \max_{C \in \{c_1, \dots, c_k\}} p(A_1, \dots, A_n | C) * p(C)$$

Clasificación

Métodos Bayesianos

- **Clasificador Bayesiano Ingenuo (*Naïve*).**

- Variables discretas (nominales): usando Estimador de Máxima Verosimilitud (EMV) o Suavizamiento mediante Corrección de Laplace (Estimador basado en la Ley de Sucesión de Laplace).
- Variables continuas: se asume una distribución Normal.

$$P(x|c_i) = N(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Clasificación

Métodos Bayesianos

- Clasificador Bayesiano Ingenuo (*Naive*): Estimador de Máxima Verosimilitud.

- Sea $n(x_j)$ el número de veces que aparece $X_i = x_j$ en el conjunto de datos, y $n(x_i, x_j)$ el número de veces que aparece el par $(X_i = x_i, X_j = x_j)$ en el mismo conjunto. Luego,

$$p(x_i/x_j) = \frac{n(x_i, x_j)}{n(x_j)}$$

Clasificación

Métodos Bayesianos

- Clasificador Bayesiano Ingenuo (*Naive*): Suavizamiento mediante Corrección de Laplace.

$$p(x_i/x_j) = \frac{n(x_i, x_j) + 1}{n(x_j) + |\Omega_{x_i}|}$$

donde $|\Omega_{x_i}|$ corresponde al número de valores distintos de X_i .

Cuando hay muchos casos, tiende al EMV; de haber pocos, tiende a la Uniforme.

Clasificación

Métodos Bayesianos

- Clasificador Bayesiano Ingenuo (*Naïve*): a modo de ejemplo...

outlook	temperature	humidity	windy	PLAY
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
rainy	65	70	TRUE	no
sunny	72	95	FALSE	no
rainy	71	91	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
overcast	64	65	TRUE	yes
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes

Estimación para *PLAY*, *outlook* y *windy*:

PLAY	EMV	Lapl.	outlook	EMV	Lapl.	windy	EMV	Lapl.				
				no	yes	no	yes	no	yes			
no	5/14	6/16	sunny	3/5	2/9	4/8	3/12	true	3/5	3/9	4/7	4/11
yes	9/14	10/16	overcast	0	4/9	1/8	5/12	false	2/5	6/9	3/7	7/11

Clasificación

Métodos Bayesianos

- Clasificador Bayesiano Ingenuo (*Naïve*): a modo de ejemplo...

outlook	temperature	humidity	windy	PLAY
sunny	85	85	FALSE	no
sunny	80	90	TRUE	no
rainy	65	70	TRUE	no
sunny	72	95	FALSE	no
rainy	71	91	TRUE	no
overcast	83	86	FALSE	yes
rainy	70	96	FALSE	yes
rainy	68	80	FALSE	yes
overcast	64	65	TRUE	yes
sunny	69	70	FALSE	yes
rainy	75	80	FALSE	yes
sunny	75	70	TRUE	yes
overcast	72	90	TRUE	yes
overcast	81	75	FALSE	yes

Estimación para *temperature* y *humidity*:

- ▶ Temperature (PLAY=no): $\mu = 74.6, \sigma = 7.89$
- ▶ Temperature (PLAY=yes): $\mu = 73, \sigma = 6.16$
- ▶ Humidity (PLAY=no): $\mu = 86.6, \sigma = 9.73$
- ▶ Humidity (PLAY=yes): $\mu = 79.11, \sigma = 10.21$

Clasificación

Métodos Bayesianos

- Clasificador Bayesiano Ingenuo (*Naive*): suponer que se necesita clasificar un nuevo dato →

$x = (\text{outlook} = \text{sunny}, \text{temperature} = 87, \text{humidity} = 90, \text{windy} = \text{false})$

- Usando EMV, para ambas categorías de la clase:

$$\begin{aligned} P(\text{play} = \text{No}) &= P(\text{sunny}/\text{No}) * P(87/\text{No}) * P(90/\text{No}) * P(\text{false}/\text{No}) * P(\text{No}) \\ &= 0.6 * N_{\text{temp}}(87, 74.6, 7.89) * N_{\text{hum}}(90, 86.6, 9.73) * 0.4 * 0.36 = \mathbf{4.08 e^{-0.05}} \end{aligned}$$

$$\begin{aligned} P(\text{play} = \text{Yes}) &= P(\text{sunny}/\text{Yes}) * P(87/\text{Yes}) * P(90/\text{Yes}) * P(\text{false}/\text{Yes}) * P(\text{Yes}) \\ &= 0.22 * N_{\text{temp}}(87, 73.6, 6.16) * N_{\text{hum}}(90, 79.11, 10.21) * 0.67 * 0.64 = \mathbf{1.02 e^{-0.05}} \end{aligned}$$

Normalizando:

$$P(\text{play} = \text{No}) = 0.8$$

$$P(\text{play} = \text{Yes}) = 0.2$$

Clasificación

Métodos Bayesianos

- **Redes Bayesianas:** útiles para...
 - Tomar decisiones en base a las probabilidades de la red.
 - Saber qué tantas más variables de evidencia hay que observar para conseguir información útil.
 - Realizar un análisis de sensibilidad y saber qué aspectos del modelo tienen más peso en las probabilidades de las variables de consulta.
 - Explicar al usuario los resultados mediante inferencia probabilista

Clasificación

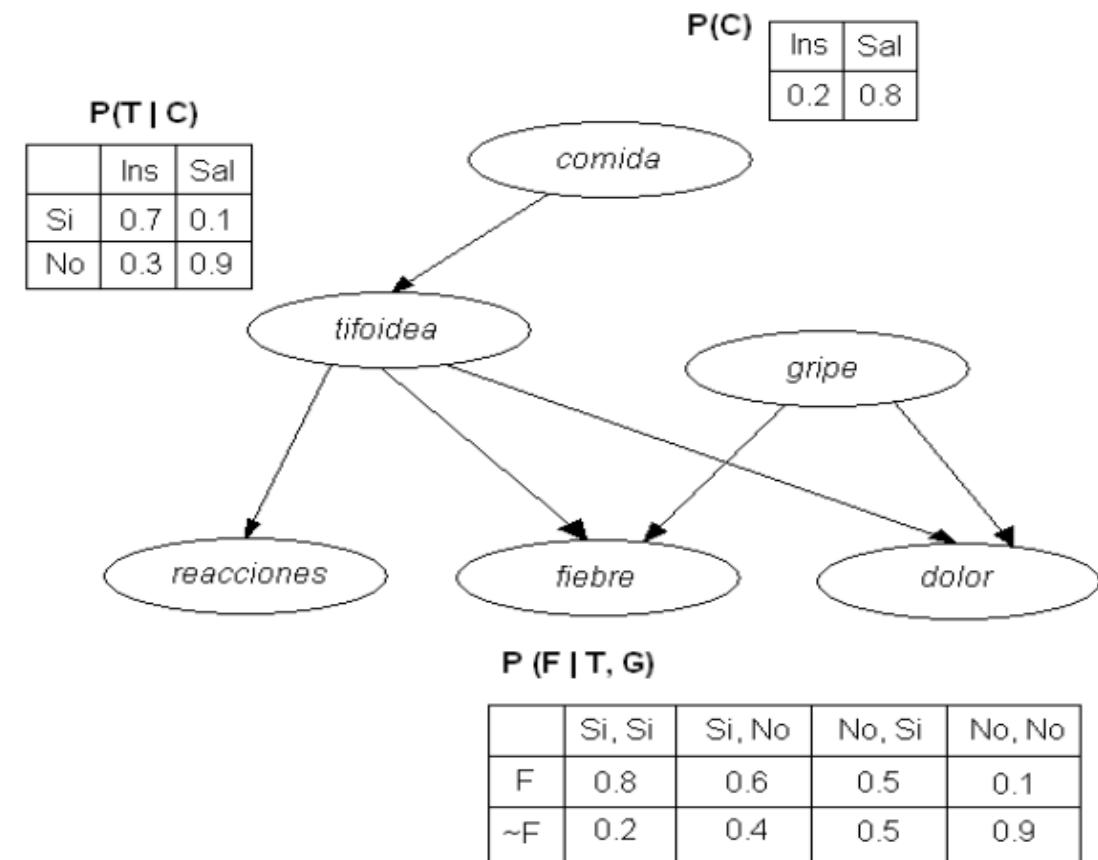
Métodos Bayesianos

- Redes Bayesianas.
 - Los nodos de la red están formados por un conjunto de variables aleatorias.
 - Cada par de nodos se conecta entre sí mediante enlaces o flechas. Si X ejerce una influencia directa sobre Y, entonces se dibujará una flecha desde X a Y, y se dirá que X es *padre* de Y.
 - Por cada nodo, hay una tabla de probabilidad que cuantifica la influencia de los padres sobre el nodo.
- Componentes:
 - Grafo acíclico dirigido: donde cada nodo representa una variable aleatoria, y cada arco una dependencia probabilística.
 - Tabla de probabilidades condicionales.

Clasificación

Métodos Bayesianos

- Redes Bayesianas: ejemplo.



Clasificación

Métodos Bayesianos

- Redes Bayesianas: ejemplo.

Un señor piensa que su esposa le está siendo infiel, con una probabilidad del 0,1. Dos acciones que podrían dar pie a esto es que ella cene con otro hombre o que reciba llamadas telefónicas sospechosas. Después de un análisis, estimó las siguientes probabilidades:

$$P(\text{Esposa cene con otro hombre} | \text{Esposa es infiel}) = 0.7$$

$$P(\text{Esposa cene con otro hombre} | \text{Esposa no es infiel}) = 0.2$$

$$P(\text{Esposa reciba llamada sospechosa} | \text{Esposa es infiel}) = 0.8$$

$$P(\text{Esposa reciba llamada sospechosa} | \text{Esposa no es infiel}) = 0.4$$

Clasificación

Métodos Bayesianos

- Redes Bayesianas: ejemplo (continuación).

Por otro lado, está la posibilidad de que la esposa sea vista realmente cenando con otro hombre, a partir de las sospechas de que lo podría hacer, para lo cual se estima:

$$P(\text{Esposa sea vista cenando con otro hombre} | \text{Esposa cene con otro hombre}) = 0.4$$

$$P(\text{Esposa sea vista cenando con otro hombre} | \text{Esposa no cene con otro hombre}) = 0.001$$

- Cuál es la red bayesiana asociada?
- Cuál es la probabilidad de que la esposa sea vista cenando con otro hombre? Y la de que reciba llamadas telefónicas sospechosas?

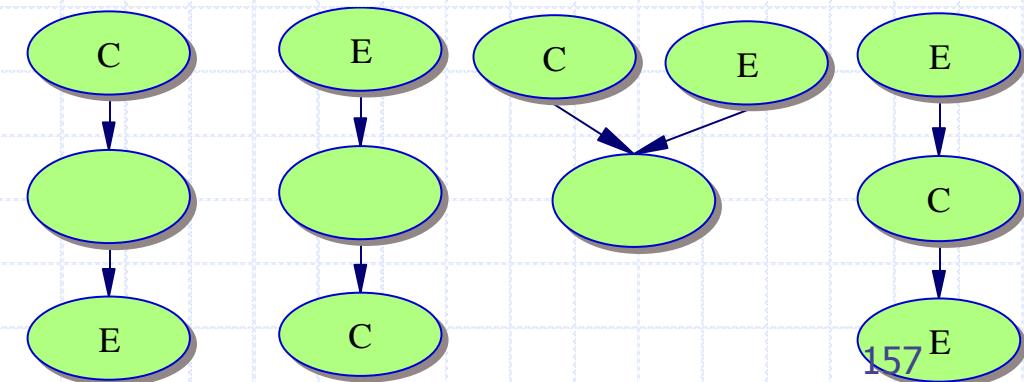
Clasificación

Métodos Bayesianos

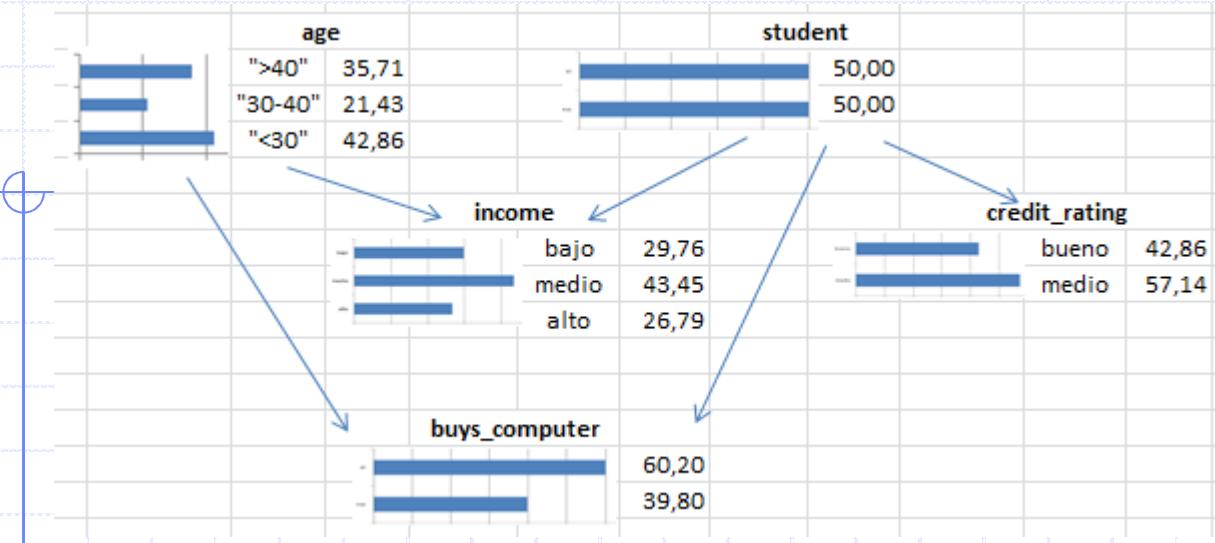
- Redes Bayesianas: inferencia.
 - Se trata de obtener la distribución de probabilidad de un conjunto de variables llamadas *de consulta*, con base en el valor de las variables *de evidencia*, es decir: $P(X_{\text{consulta}} | X_{\text{evidencia}})$.

Se distingue:

- Inferencia por diagnóstico: de los efectos a las causas.
- Inferencias causales : de las causas a los efectos.
- Inferencias intercausales: entre las causas de un efecto común.
- Inferencias mixtas.

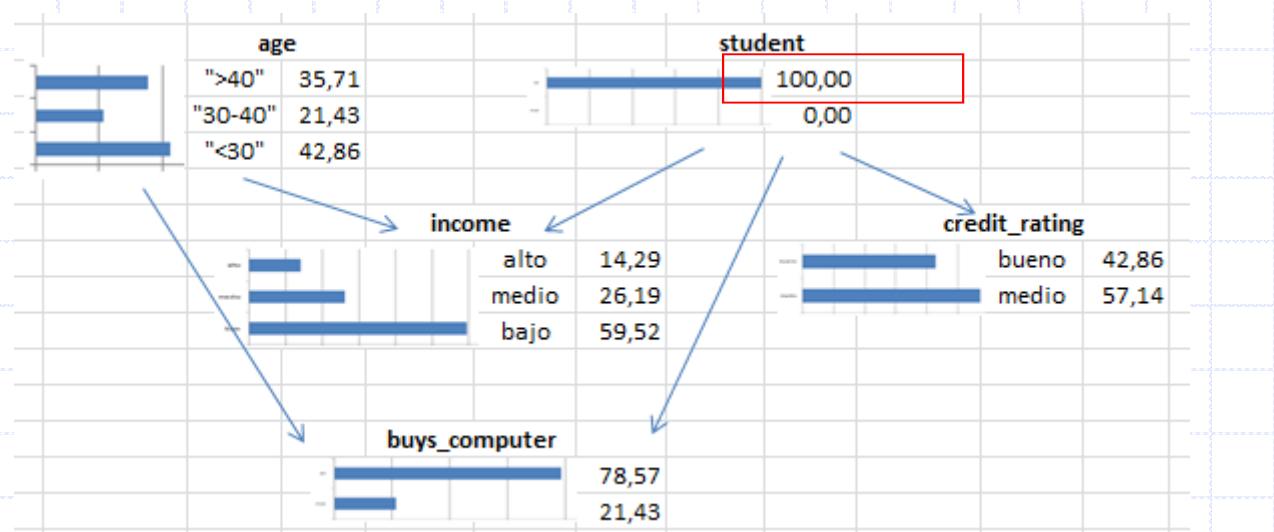


- Redes Bayesianas: otro ejemplo.



Ejemplo de
Probabilidades
sin evidencia

Ejemplo de
Probabilidades
con evidencia



Clasificación

Métodos Bayesianos

- Redes Bayesianas: aprendizaje de dos tipos:
 - Aprendizaje estructural: aprende enlaces, habiendo de dos tipos:
 - Basados en pruebas de independencia (algoritmos PC, NPC...).
 - Puntuación y búsqueda (*Score & Search*): para la primera parte se tienen medidas bayesianas tales como IAC y BIC; para la segunda: K2, LK2, Montecarlo, B.
 - Aprendizaje paramétrico: aprende las probabilidades de la red en base a casos dados. Ejemplos de algoritmos de este tipo: EM, ML.

Clasificación

Métodos Bayesianos

- Redes Bayesianas: aprendizaje estructural.
 - Algoritmo K2: algoritmo codicioso, considera la ganancia de incluir un nuevo parente, dada ya la presencia de otro(s), con respecto a un mismo nodo de referencia.

ALGORITMO K2(X :nodos (variables ordenadas), D :Datos)

Fase de Inicialización:

PARA CADA X_i ($i=0$ hasta n)

$Pa(X_i)$ = conjunto vacío

FIN PARA CADA

Fase Iterativa:

PARA CADA X_i ($i=0$ hasta n)

$ok := true$

MIENTRAS ok

Sea X_j el nodo tal que $j < i$ y X_j no pertenezca a $Pa(X_i)$ que maximiza

$f_j(X_j | Pa(X_i) \cup X_j; D)$.

SI $f_j(X_j | Pa(X_i) \cup X_j; D) > f_j(X_j | Pa(X_i); D)$ **ENTONCES** $Pa(X_i) := Pa(X_i) \cup X_j$

EN CASO CONTRARIO $ok := false$

FIN MIENTRAS

FIN PARA CADA

FIN ALGORITMO

Clasificación

Métodos Bayesianos

- Redes Bayesianas: aprendizaje estructural.
 - Algoritmo B: algoritmo codicioso, se diferencia del anterior por no imponer la restricción de tener como entradas un orden específico entre las variables.

ALGORITMO B(X :nodos (variables), D :Datos)

Fase de Inicialización:

PARA CADA X_i ($i=0$ hasta n)

$Pa(X_i)$ = conjunto vacío

FIN PARA CADA

Fase Iterativa:

$ok := true$

MIENTRAS ok

Sea $X_j \rightarrow X_i$ el enlace que maximiza (de todos los enlaces que no formen un ciclo dirigido acíclico y previamente no incluido) la medida:

$$f_i(X_i | Pa(X_i) \cup X_j; D) - f_i(X_i | Pa(X_i); D).$$

SI $f_i(X_i | Pa(X_i) \cup X_j; D) - f_i(X_i | Pa(X_i); D) > 0$ **ENTONCES** $Pa(X_i) := Pa(X_i) \cup X_j$

EN CASO CONTRARIO $ok := false$

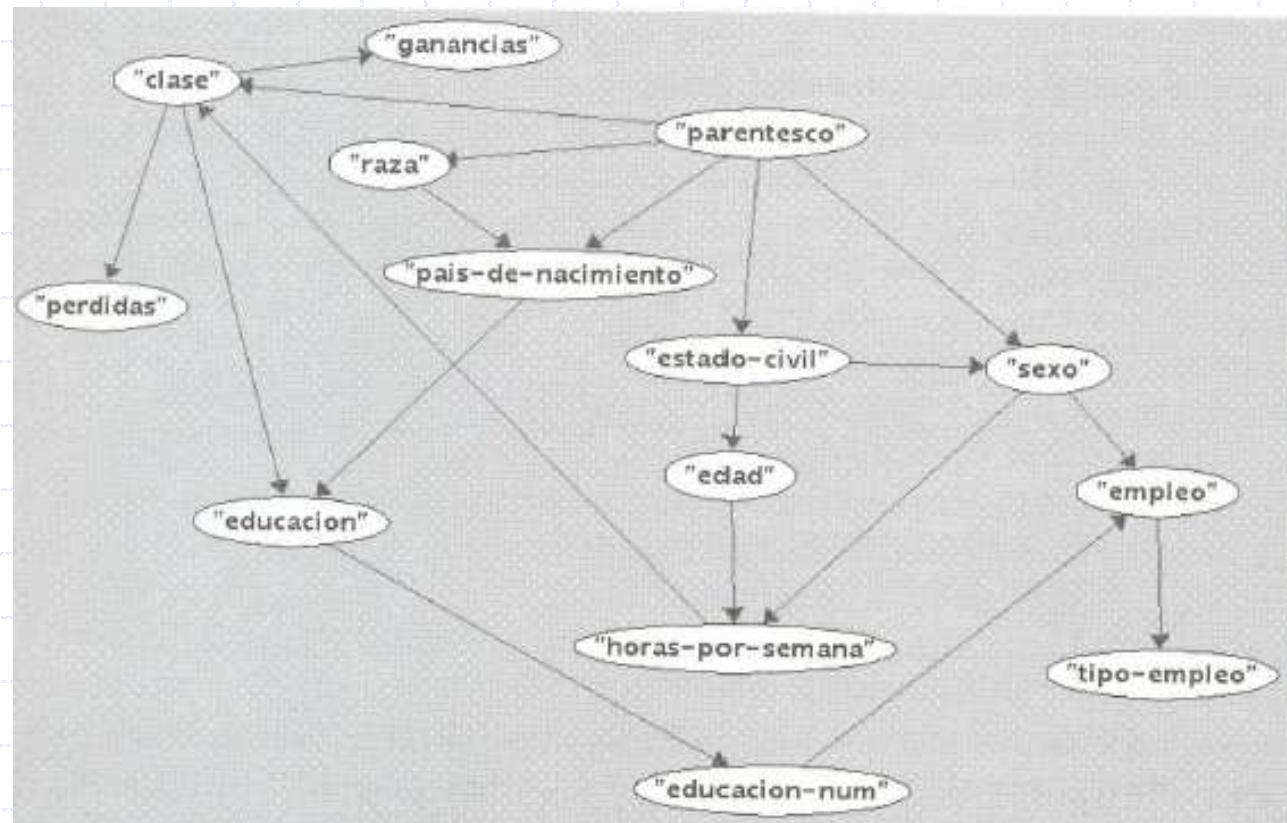
FIN MIENTRAS

FIN ALGORITMO

Clasificación

Métodos Bayesianos

- Redes Bayesianas: aprendizaje estructural... ejemplo...



Clasificación

Métodos Bayesianos

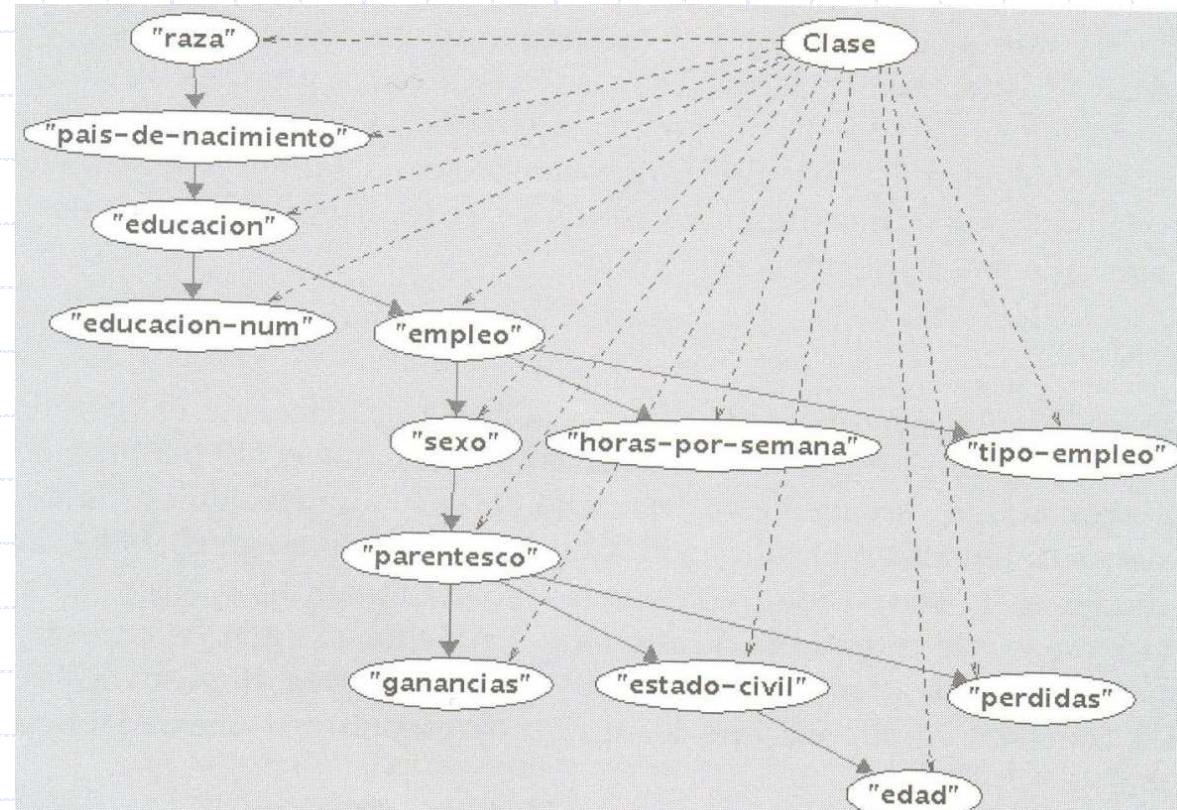
- Redes Bayesianas: aprendizaje estructural para clasificación...un algoritmo conocido es:
 - TAN (*Tree Augmented Naive Bayes*): extiende al clasificador NB admitiendo la posibilidad de ciertas dependencias entre los atributos.
 - Construye un árbol con los atributos, sin la clase, la cual se agrega al final como padre de todos los atributos.
 - Se basa en el concepto de información mutua:

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p_1(x) p_2(y)} \right)$$

ALGORITMO TAN

1. Calcular $I_{P(A_i, A_j|C)}$ para cada par de atributos ($i \neq j$)
2. Crear un grafo no dirigido con todos los atributos como conjunto de nodos y añadir aristas entre cada par de nodos.
3. Asociar a cada arista (i, j) del grafo el peso $I_{P(A_i, A_j|C)}$.
4. Construir un árbol expandido de máximo peso a partir del grafo anterior.
5. Elegir un nodo cualquiera del árbol anterior como raíz y direccionar a partir de él el resto de aristas.
6. Añadir la variable clase C y el conjunto de aristas dirigidas ($C \rightarrow A_i$) para todo atributo A_i .
7. Devolver el modelo TAN obtenido.

FIN ALGORITMO



Clasificación

Métodos Bayesianos

- Redes Bayesianas: aprendizaje paramétrico...un algoritmo conocido, que puede usarse también para el Tratamiento de Datos faltantes es:
 - EM (*Expectation Maximization*).

ALGORITMO EM ($B=(G, \Theta)$: Red Bayesiana, D :Datos)

Fase de Inicialización:

Iniciar el conjunto de parámetros $\Theta^{(0)}$

Iniciar contador de etapas $e=0$

Fase Iterativa:

MIENTRAS no convergencia

Paso E: Etapa de cálculo de Esperanzas $E[N_{ijk} | \Theta^{(e)}]$

Paso M: Etapa de maximización

$$\Theta_{ijk}^{(e+1)} = E[N_{ijk} | \Theta^{(e)}] / E[N_i | \Theta^{(e)}]$$

FIN MIENTRAS

FIN ALGORITMO

Clasificación

Métodos Basados en Casos y Vecindad

- **Basada en Casos y Vecindad:** almacenan (una parte de) el conjunto de entrenamiento y lo utilizan directamente para clasificar nuevos datos.
- **Técnicas para Segmentación:**
 - Mapas Auto-Organizativos de Kohonen (SOM)
 - K-Means
 - Agrupamiento Jerárquico (dendogramas)
- **Técnicas para Clasificación:**
 - Estimación Bayesiana de Funciones de Densidad
 - K-NN (K Vecinos más Cercanos)
 - LVQ

Clasificación

Métodos Basados en Casos y Vecindad

- **Algoritmo K Vecinos más Cercanos.**

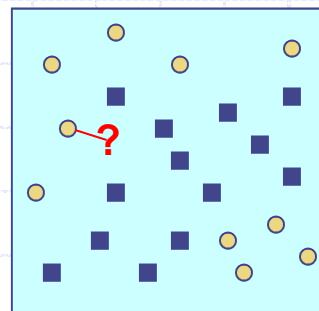
- Están basados en el aprendizaje por analogía.
- Las muestras de entrenamiento son descritas por atributos numéricos n-dimensionales, donde cada muestra representa un punto en el espacio n-dimensional.
- Son “basados en las instancias” pues ellos almacenan todas las muestras de entrenamiento, asignándole igual peso a cada uno de los atributos.

Clasificación

Métodos Basados en Casos y Vecindad

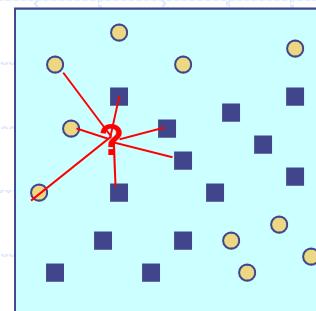
- **Algoritmo K Vecinos más Cercanos:** algoritmo.

1. Se miran los k casos más cercanos (distancia Euclídea).
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.



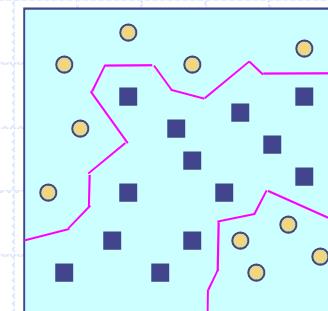
1-nearest neighbor

Clasifica
círculo



7-nearest neighbor

Clasifica
cuadrado



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

Clasificación

Métodos Basados en Casos y Vecindad

- Algoritmo **K Vecinos más Cercanos**: algoritmo (2).
 - Pueden ser usados, también, para predicción, es decir para retornar un valor real asociado a los k vecinos más cercanos a la muestra desconocida.
 - El valor de retorno normalmente es el promedio de los vecinos.
- Consideraciones:
 - K demasiado pequeño: sensible a ruido.
 - K muy grande: el vecindario puede incluir puntos de otras clases

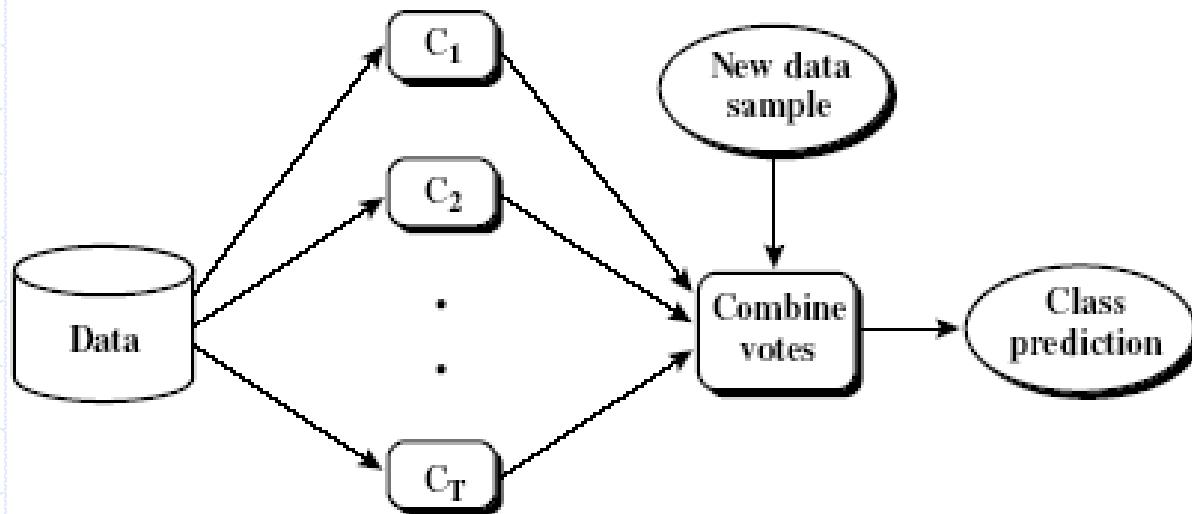
Breve Ejercicio:

http://www.theparticle.com/applets/ml/nearest_neighbor/

Clasificación

Métodos Basados en Casos y Vecindad

- **Ensamblajes:** combinan varios modelos con el objetivo de mejorar la precisión final del clasificador.



Clasificación

Métodos Basados en Casos y Vecindad

- Estrategias:

- **Bagging:** aplica una votación por mayoría → varios clasificadores diferentes votan para decidir la clase de un caso de prueba (ej.: bootstrapping)
- **Boosting:** se realiza una votación ponderada → los clasificadores tienen distintos pesos en la votación, en función de su precisión (ej.: AdaBoost)

Evaluación de Modelos

Clasificación

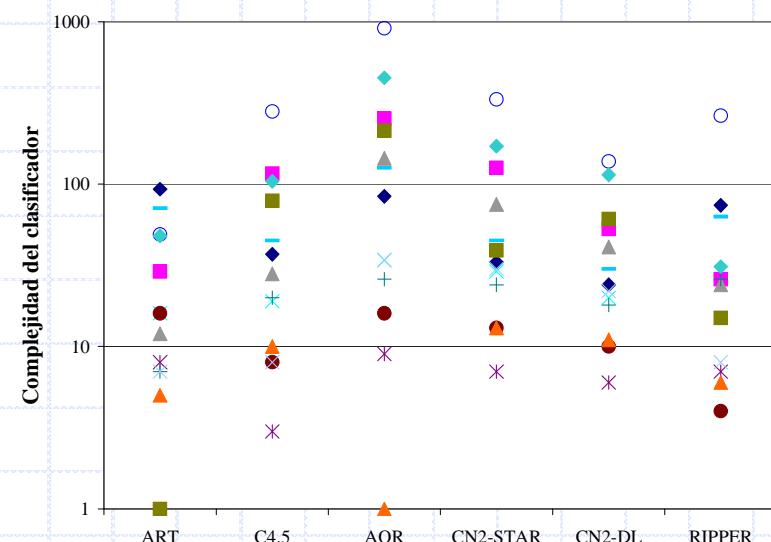
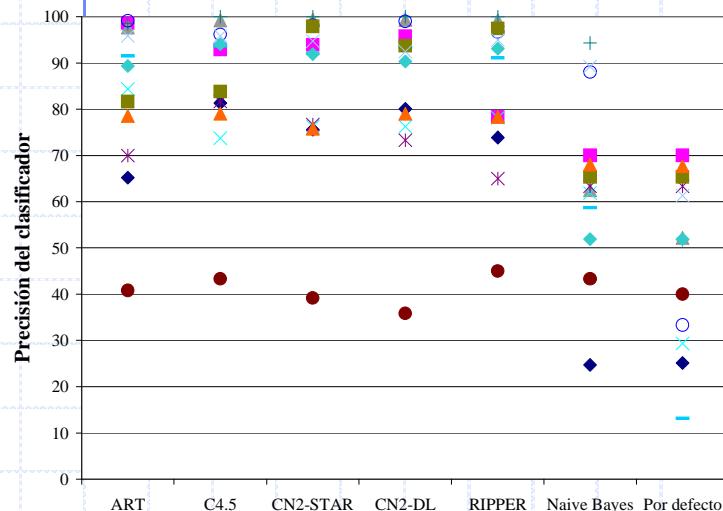
Evaluación de Modelos

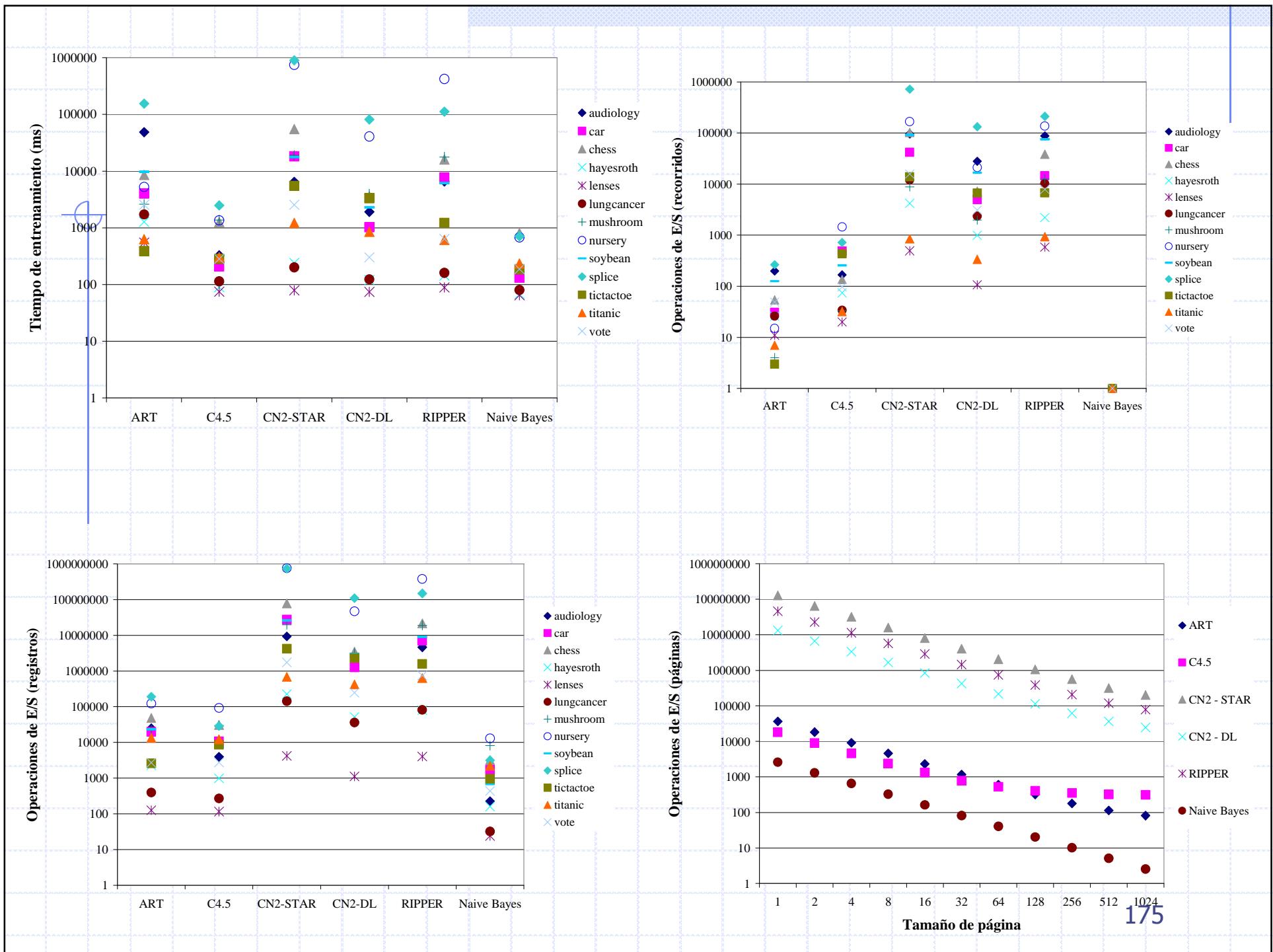
- La evaluación de un algoritmo de clasificación se hace a través de distintos aspectos del modelo creado o del proceso utilizado para crearlo:
 - **Precisión:** porcentaje de casos clasificados correctamente.
 - **Eficiencia:** tiempo necesario para construir/uso el clasificador.
 - **Robustez:** frente a ruido y valores nulos.
 - **Escalabilidad:** utilidad en grandes bases de datos.
 - **Interpretabilidad:** el clasificador, ¿es sólo una caja negra?).
 - **Complejidad:** del modelo de clasificación.

Clasificación

Evaluación de Modelos

- Comparación de métodos de evaluación.





Clasificación

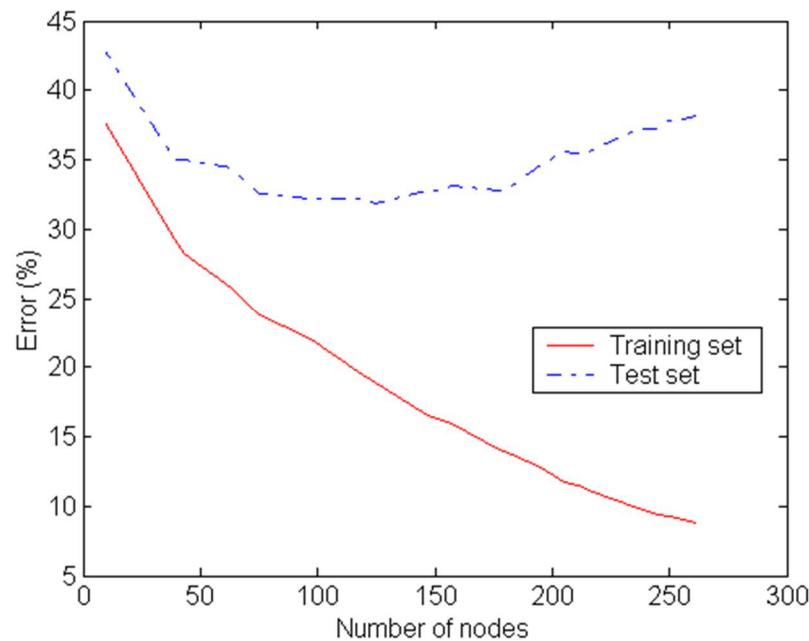
Evaluación de Modelos

- Estimación de la precisión del modelo:
 - Previo a construir el modelo de clasificación, se divide el conjunto de datos disponible en un conjunto de **entrenamiento** (para construir el modelo) y un conjunto de **prueba** (para evaluar el modelo).
 - Posterior a la construcción del modelo, se aplica el modelo al conjunto de prueba y los resultados se comparan con las etiquetas ya conocidas de los mismos datos de prueba, obteniéndose así un **porcentaje de clasificación**.
 - Si la precisión del clasificador es aceptable, tiene sentido usar el modelo para clasificar nuevos casos (ahora, sin clase conocida).

Clasificación

Evaluación de Modelos

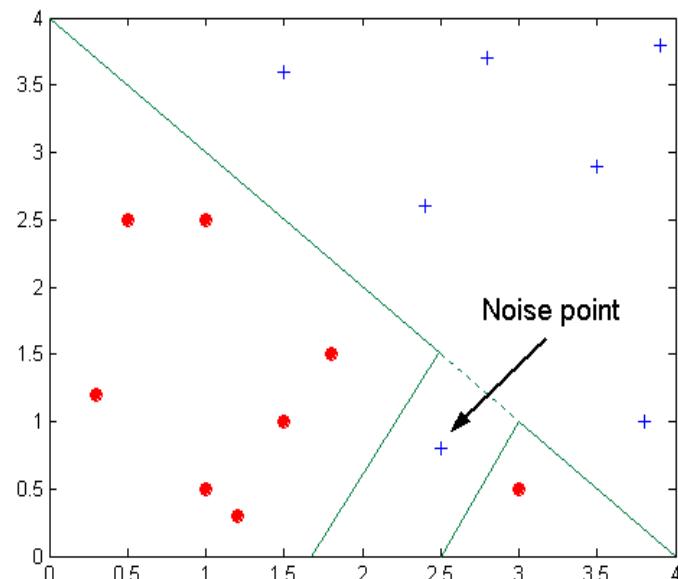
- Estimación de la precisión del modelo: problema...
 - Cuanto mayor sea su complejidad, los modelos de clasificación tienden a ajustarse más al conjunto de entrenamiento utilizado en su construcción (problema de sobreaprendizaje).



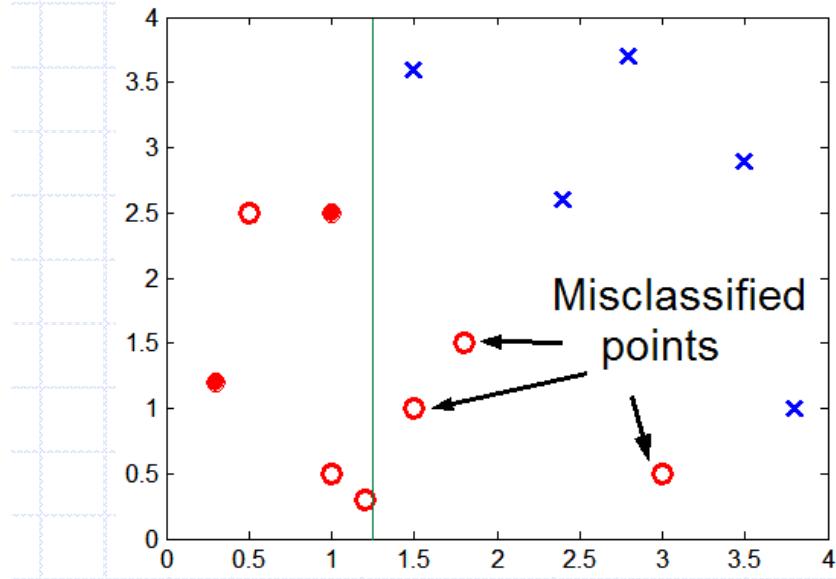
Clasificación

Evaluación de Modelos

- Estimación de la precisión del modelo: problema...



Sobreaprendizaje debido
a la presencia de ruido en los datos



Sobreaprendizaje debido
a la escasez de muestras

Clasificación

Evaluación de Modelos

- Estimación de la precisión del modelo: mejoras...
 - El error de clasificación en el conjunto de entrenamiento **NO** es un buen estimador de la precisión del clasificador.
 - Para una mejor estimación, considerar:
 - Métricas: cómo evaluar la “calidad” de un modelo de clasificación?.
 - Métodos: cómo estimar, de forma fiable, la calidad de un modelo?.
 - Comparación: cómo comparar el rendimiento relativo de dos modelos de clasificación alternativos?.

Clasificación

Evaluación de Modelos

- Estimación de la precisión del modelo: métricas.
 - Matriz de Confusión: la precisión del clasificador viene dada por la expresión...

$(TP+TN)$

$$\text{accuracy} = \frac{(TP+TN)}{(TP+TN+FP+FN)}$$

		Predicción	
		C_P	C_N
Clase real	C_P	TP : True positive	FN : False negative
	C_N	FP : False positive	TN : True negative

$$\text{precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

$$\text{recall} = \text{sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

$$\text{specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})}$$

$$2 * \text{precision} * \text{recall} = \frac{2 \text{TP}}{\text{precision} + \text{recall}}$$

$$\text{F-measure} = \frac{2 \text{TP}}{\text{precision} + \text{recall}} = \frac{2 \text{TP}}{2\text{TP} + \text{FP} + \text{FN}}$$

		Predicción	
		C_P	C_N
Real	C_P	TP	FN
	C_N	FP	TN

Accuracy

		Predicción	
		C_P	C_N
Real	C_P	TP	FN
	C_N	FP	TN

Recall

		Predicción	
		C_P	C_N
Real	C_P	TP	FN
	C_N	FP	TN

Precision

		Predicción	
		C_P	C_N
Real	C_P	TP	FN
	C_N	FP	TN

F-measure

Clasificación

Evaluación de Modelos

- Estimación de la precisión del modelo: métricas.
 - Matriz de Costos: el costo de la clasificación será proporcional a la precisión del clasificador sólo si

$$\forall i, j: i \neq j \quad C(i|j) = C(j|i)$$

$$C(i|i) = C(j|j)$$

		Predicción	
		C_P	C_N
$C(i j)$	C_P	$C(P P)$	$C(N P)$
	C_N	$C(P N)$	$C(N N)$

Clasificación

Evaluación de Modelos

- Al evaluar la precisión de un modelo de clasificación, nunca se debe usar el conjunto de entrenamiento (lo que generaría el “error de resustitución” del clasificador), sino un conjunto de prueba independiente...
...por ejemplo, reservar 2/3 de los datos de ejemplo disponibles para construir el clasificador, y tercio restante usarlo como conjunto de prueba para estimar la precisión del clasificador.

Clasificación

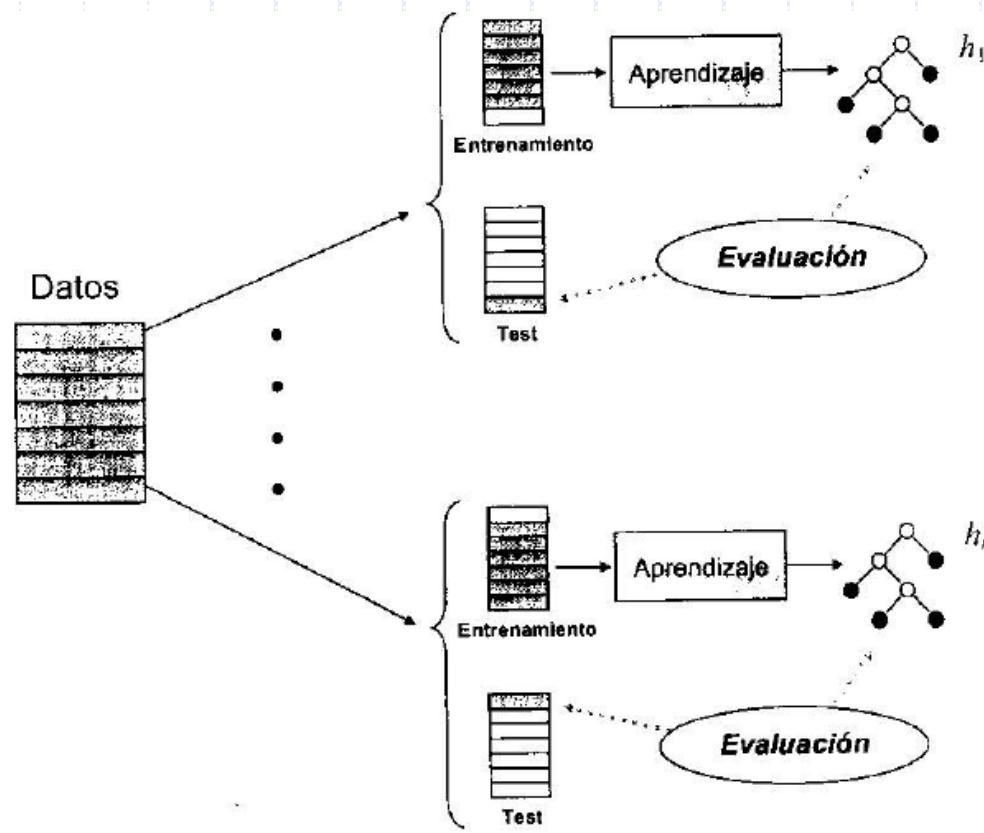
Evaluación de Modelos

- Otros métodos de evaluación: Validación Cruzada (*k-fold Cross-Validation*).
 - Se divide el conjunto de datos, aleatoriamente, en k subconjuntos de intersección vacía (más o menos del mismo tamaño); por lo general, $k=10$.
 - En la iteración i , se usa el subconjunto i como conjunto de prueba y los $k-1$ restantes como conjunto de entrenamiento.
 - Como medida de evaluación del método de clasificación se toma el promedio aritmético de las k iteraciones realizadas.

Clasificación

Evaluación de Modelos

- Otros métodos de evaluación: Validación Cruzada (*k-fold Cross-Validation*).



Clasificación

Evaluación de Modelos

- Otros métodos de evaluación: Validación Cruzada – Variantes.
 - ***Leave one out:*** se realiza una validación cruzada con k particiones del conjunto de datos, donde k coincide con el número de ejemplos disponibles.
 - **Validación cruzada estratificada:** las particiones se realizan intentando mantener, en todas ellas, la misma proporción de clases que aparece en el conjunto de datos completo.

Clasificación

Evaluación de Modelos

- Otros métodos de evaluación: *Bootstrapping*.
 - Corresponde a un muestreo uniforme con reemplazo de los ejemplos disponibles; es decir, una vez que se escoge un dato, se devuelve al conjunto de entrenamiento y puede que se vuelva a escoger.

Clasificación

Evaluación de Modelos

- Otros métodos de evaluación: Curvas ROC (*Receiver Operating Character*).
 - Se desarrollan en los años 50 para analizar señales con ruido y así caracterizar el compromiso entre aciertos y falsas alarmas.
 - Permiten comparar visualmente distintos modelos de clasificación, al analizar el área que queda bajo la curva el cual es una medida de la precisión (accuracy) del clasificador:
 - Cuanto más cerca se esté de la diagonal (área cercana a 0.5), menos preciso será el modelo.
 - Un modelo “perfecto” tendrá área igual a 1.

Clasificación

Evaluación de Modelos

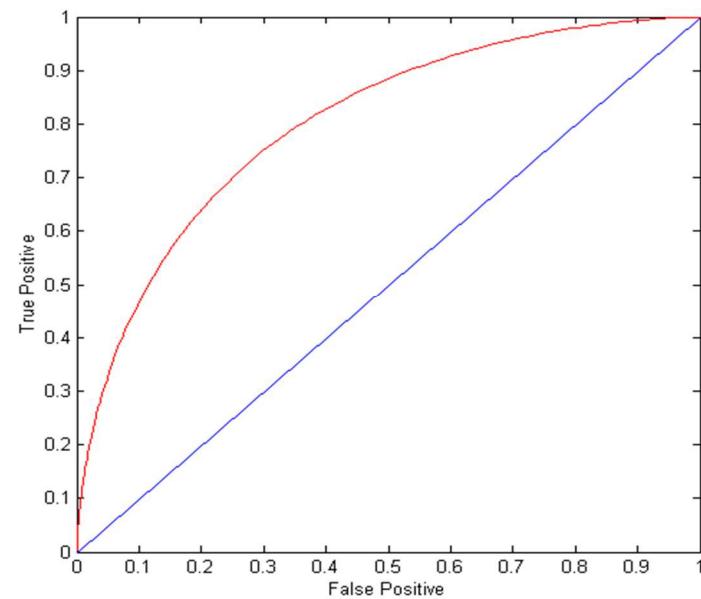
- Otros métodos de evaluación: Curvas ROC.

Tasa de Verdaderos
Positivos

$TP/(TP+FN)$

Tasa de Falsos Positivos

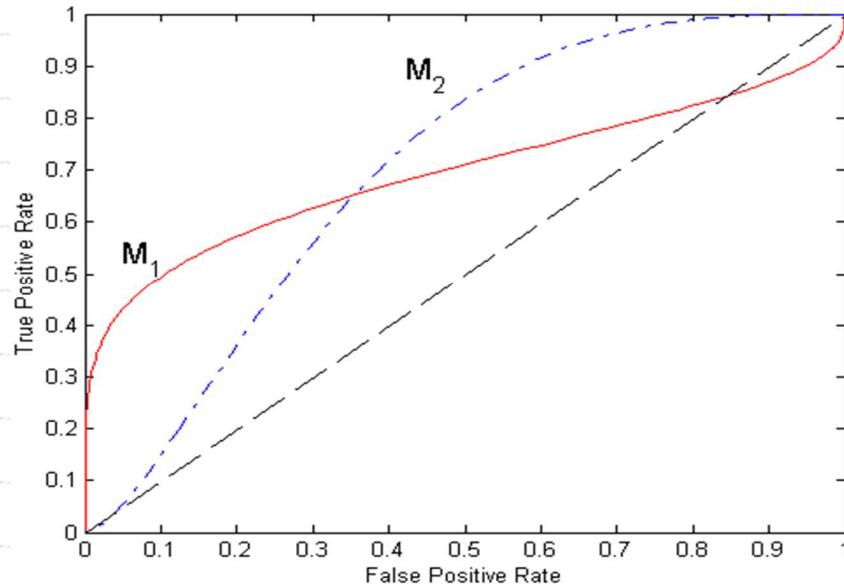
$FP/(FP+TN)$



Clasificación

Evaluación de Modelos

- Otros métodos de evaluación: Curvas ROC – ejemplo.



Ningún modelo es consistentemente mejor que el otro:
 M_1 es mejor para FPR bajos, M_2 para FPR altos.

Clasificación

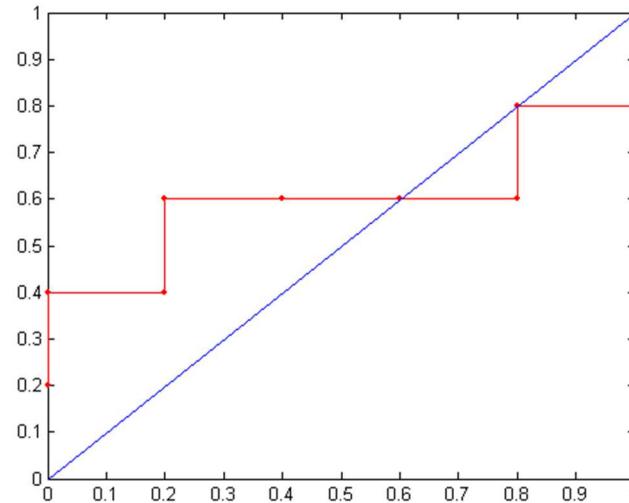
Evaluación de Modelos

- Otros métodos de evaluación: Curvas ROC – construcción.
 - Se usa un clasificador que prediga la probabilidad de que un ejemplo E pertenezca a la clase positiva $P(+|E)$.
 - Se ordenan los ejemplos en orden decreciente del valor estimado $P(+|E)$.
 - Se aplica un umbral para cada valor distinto de $P(+|E)$, donde se cuenta el número de TP, FP, TN y FN.

$$TPR = TP / (TP + FN)$$

$$FPR = FP / (FP + TN)$$

Ejemplo	$P(+ E)$	Clase
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+



Class	+	-	+	-	-	-	+	-	+	+	
	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

Bibliografía

- *Data Mining: Concepts and Techniques.* J. Han & M. Kamber. Morgan Kaufmann Publishers. Segunda Edición, 2006.
- *Introducción a la Minería de Datos* . J. Hernández, M. J. Ramírez, C. Ferri. Editorial Pearson, 2004.