Extracción Automática de Conocimiento en Bases de Datos e Ingeniería del Software

T.2 Minería de Datos y Extracción de Conocimiento de Bases de Datos

---

José Hernández Orallo

Cèsar Ferri Ramírez

Programas:

- Programación Declarativa e Ingeniería de la Programación (Dep. de Sistemes Informàtics i Computació)

Objetivos

- Conocer las características especiales de la extracción automática de conocimiento de bases de datos.

- Ver las técnicas de aprendizaje automático más apropiadas y su adaptación a estos problemas.

Temario

2.1. Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos. Los Data-Warehouses y el KDD.
2.2. El Proceso de Extracción de Conocimiento de Bases de Datos.
2.3. Métodos Específicos de Prospección de Datos.

Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos

- El **aumento del volumen y variedad de información** que se encuentra informatizada en bases de datos digitales ha crecido espectacularmente en la última década.
- Gran parte de esta **información es histórica**, es decir, representa transacciones o situaciones que se han producido.
- Aparte de su función de "memoria de la organización", la información histórica es útil **para predecir la información futura**.

Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos

- La mayoría de ***decisiones*** de empresas, organizaciones e instituciones se basan también en información de experiencias pasadas extraídas de fuentes muy diversas.
- las **decisiones colectivas** suelen tener consecuencias mucho más graves, especialmente económicas, y, recientemente, se deben basar en **volúmenes de datos que desbordan la capacidad humana**.

El área de la extracción (semi-)automática de conocimiento de bases de datos ha adquirido recientemente una importancia científica y económica inusual

Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos

- "Descubrimiento de Conocimiento a partir de Bases de Datos" (KDD, del inglés *Knowledge Discovery from Databases*).

  "*proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles a partir de los datos*". Fayyad et al. 1996

- Diferencia clara con métodos estadísticos: la estadística se utiliza para validar o parametrizar un *modelo sugerido y preexistente*, no para generarlo.
- Diferencia sutil "Análisis Inteligente de Datos" (IDA, del inglés Intelligent Data Analysis) que correspondía con el uso de técnicas de inteligencia artificial en el análisis de los datos.

Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos

- Además el resultado de KDD debe ser COMPRENSIBLE.
- Se excluyen, a priori, por tanto, muchos métodos de aprendizaje automático (redes neuronales, CBR, k-NN, Radial Basis Functions, Bayes Classifiers...).

- Cambia la Manera de Extraer el Conocimiento:
  - Eficiente.
  - Entornos de Descubrimiento ('Navegación').
  - Consultas Inductivas.

Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos

- KDD nace como interfaz y se nutre de diferentes disciplinas:
  - estadística.
  - sistemas de información / bases de datos.
  - aprendizaje automático / IA.
  - visualización de datos.
  - computación paralela / distribuida.
  - interfaces de lenguaje natural a bases de datos.

Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos

- Datos poco habituales para algoritmos clásicos:
  - número de registros (ejemplos) muy largo (10^8-10^{12} bytes).
  - datos altamente dimensionales (nº de columnas/atributos): 10^2-10^4.
- El usuario final no es un experto en ML ni en estadística.
- El usuario no se puede perder más tiempo analizando los datos:
  - industria: ventajas competitivas, decisiones más efectivas.
  - ciencia: datos nunca analizados, bancos no cruzados, etc.
  - personal: "information overload"...

Los sistemas clásicos de estadística son difíciles de usar y no escalan al número de datos típicos en bases de datos.

Nuevas Necesidades del Análisis de Grandes Volúmenes de Datos

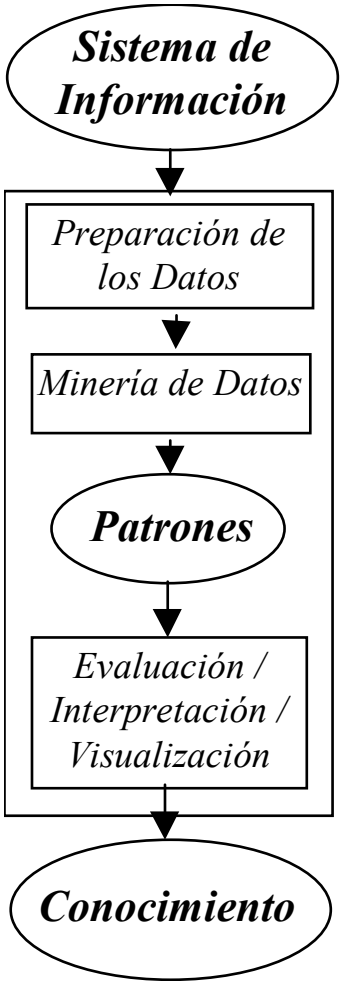
Evaluación del 'conocimiento':

- válido?
- útil?
- inteligible?
- novedoso?
- interesante?

Uso del 'conocimiento' obtenido:

- hacer predicciones sobre nuevos datos.
- explicar los datos existentes
- resumir una base de datos masiva para facilitar la toma de decisiones.
- visualizar datos altamente dimensionales, extrayendo estructura *local* simplificada.

FASES DEL KDD

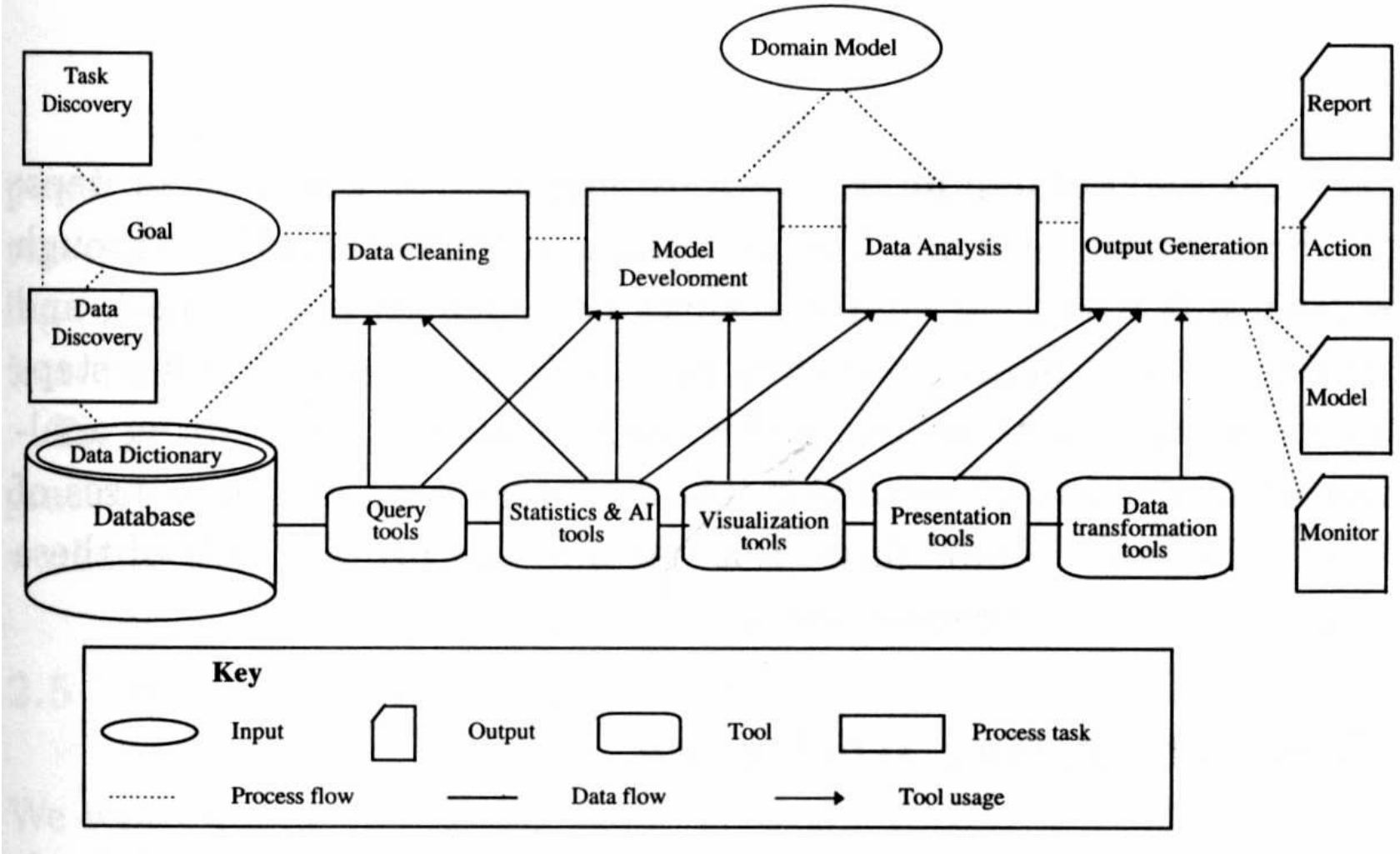


KDD

1. Determinar las fuentes de información que pueden ser útiles y dónde conseguirlas.
2. Diseñar el esquema de un almacén de datos (Data Warehouse) que consiga unificar de manera operativa toda la información recogida.
3. Implantación del almacén de datos que permita la "navegación" y visualización previa de sus datos, para discernir qué aspectos puede interesar que sean estudiados.
4. Selección, limpieza y transformación de los datos que se van a analizar. La selección incluye tanto una criba o fusión horizontal (filas) como vertical (atributos).
5. Seleccionar y aplicar el método de minería de datos apropiado.
6. Interpretación, transformación y representación de los patrones extraídos.
7. Difusión y uso del nuevo conocimiento.

Fases y Técnicas del KDD

Las distintas técnicas de distintas disciplinas se utilizan en distintas fases:



Recogida de Datos

Las primeras fases del KDD determinan que las fases sucesivas sean capaces de extraer conocimiento válido y útil a partir de la información original.

Generalmente, la información que se quiere investigar sobre un cierto dominio de la organización se encuentra en bases de datos y otras fuentes muy diversas, tanto internas como externas.

Muchas de estas fuentes son las que se utilizan para el trabajo diario.

OLAP

- Sobre estas mismas bases de datos de trabajo ya se puede extraer conocimiento (visión tradicional).
  - Se mantiene el trabajo transaccional diario de los sistemas de información originales (conocido como **OLTP, *On-Line Transactional Processing***).
  - Se hace análisis de los datos en tiempo real sobre la misma base de datos (conocido como **OLAP, *On-Line Analytical Processing***),
- PROBLEMAS:
  - perturba el trabajo transaccional diario de los sistemas de información originales ("*killer queries*"). Se debe hacer por la noche o en fines de semana.
  - la base de datos está diseñada para el trabajo transaccional, no para el análisis de los datos. Generalmente no puede ser en tiempo real (era AP pero no OLAP).

Data-Warehousing

- Para poder operar eficientemente con esos datos y debido a que los costes de almacenamiento masivo y conectividad se han reducido drásticamente en los últimos años, parece razonable recoger (copiar) los datos en un sistema unificado.

NACE EL DATA-WAREHOUSING

- DATA-WAREHOUSES (Almacenes de Datos): Se separan de los datos a analizar con respecto a sus fuentes transaccionales (se copia/almacena toda la información histórica).

Existe toda una tecnología creciente de cómo organizarlos y sobretodo de cómo tenerlos actualizados (cargas periódicas) respecto a los datos originales.

Data-Warehousing

Data-warehousing:

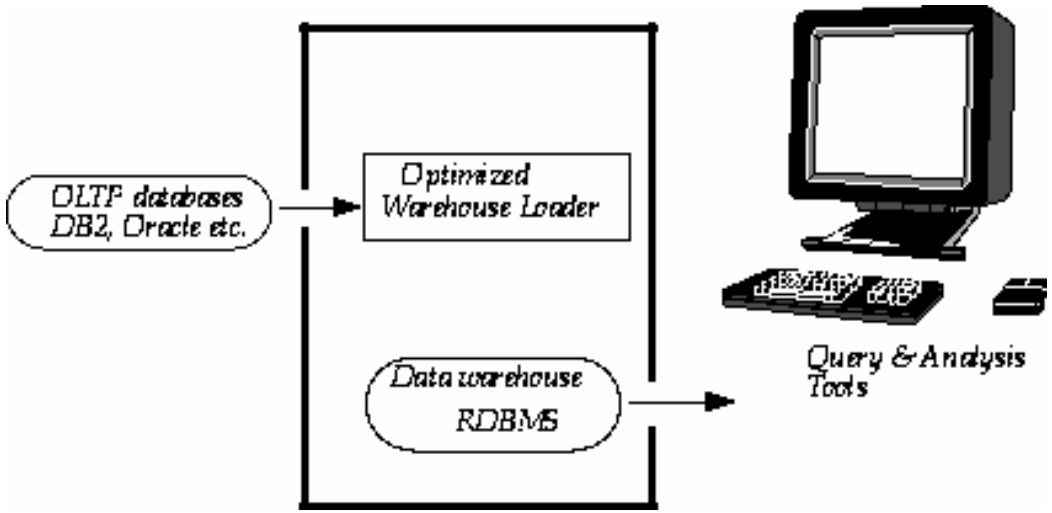
- Facilita el análisis de los datos en tiempo real (**OLAP**),
- No disturba el **OLTP** de las bases de datos originales.

A partir de ahora diferenciaremos entre bases de datos para OLTP (tradicional) y almacenes de datos (KDD sobre data-warehouses).

	BASE DE DATOS TRANSACCIONAL	**ALMACÉN DE DATOS**
Propósito	Operaciones diarias. Soporte a las aplicaciones.	Recuperación de información, informes, análisis y minería de datos.
Tipo de datos	Datos de funcionamiento de la organización.	Datos útiles para el análisis, la sumarización, etc.
Características de los datos	Datos de funcionamiento, cambiantes, internos, incompletos...	Datos históricos, datos internos y externos, datos descriptivos...
Modelo de datos	Datos normalizados.	Datos en estrella, en copo de nieve, parcialmente desnormalizados, multidimensionales...
Número y tipo de usuarios	Cientos/miles: aplicaciones, operarios, administrador de la base de datos.	Decenas: directores, ejecutivos, analistas (granjeros, mineros).
Acceso	SQL. Lectura y escritura.	SQL y herramientas propias (*slice & dice*, *drill*, *roll, pivot*...). Lectura.

Data-Warehousing

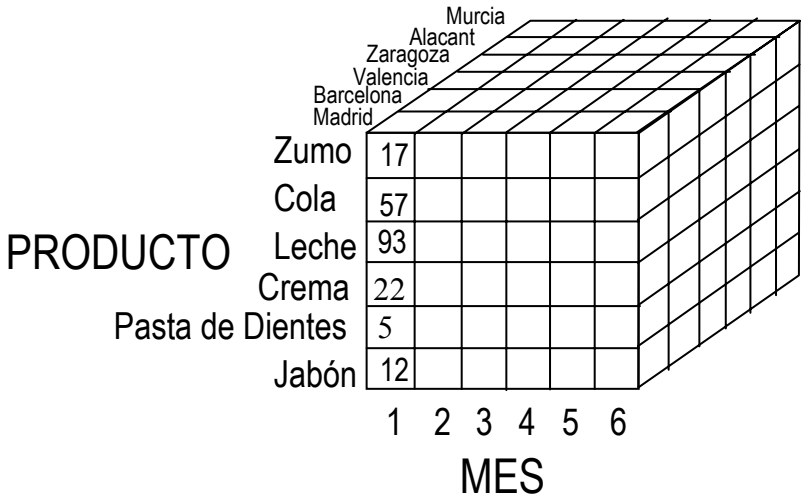
- Según la organización de la información copiada se distingue:
  - ROLAP (Relational OLAP): el almacén de datos es relacional.
  - MOLAP (Multidimensional OLAP): el almacén de datos es una matriz multidimensional.
- Aunque un MOLAP puede estar implementado sobre un sistema de gestión de base de datos relacional (RDBMS).



Data-Warehousing

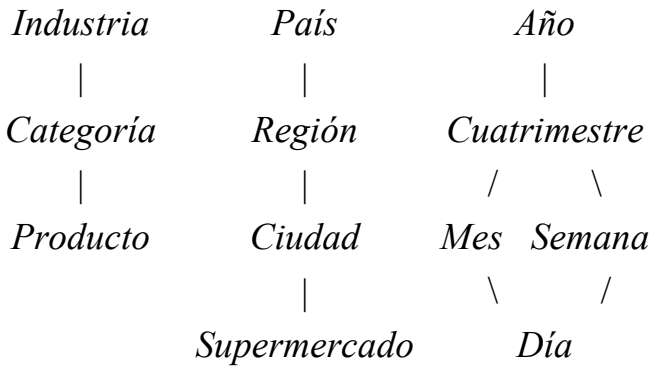
Ejemplo de MOLAP:

- Cada atributo relevante se establece en una dimensión, que se puede agregar o desagregar. La base de datos está completamente desnormalizada.



Ventas en millones de Euros

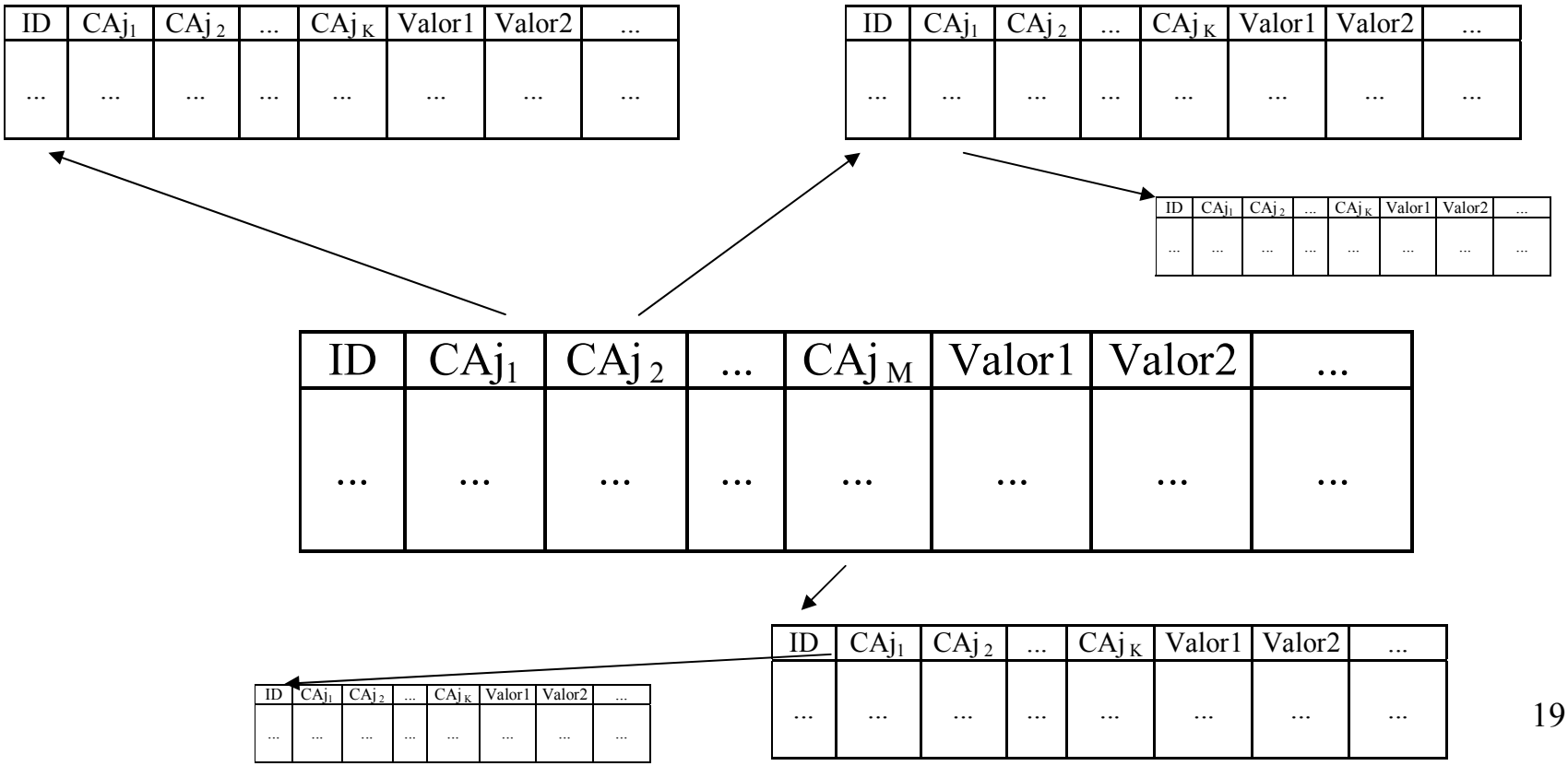
Las dimensiones se agregan:



Data-Warehousing

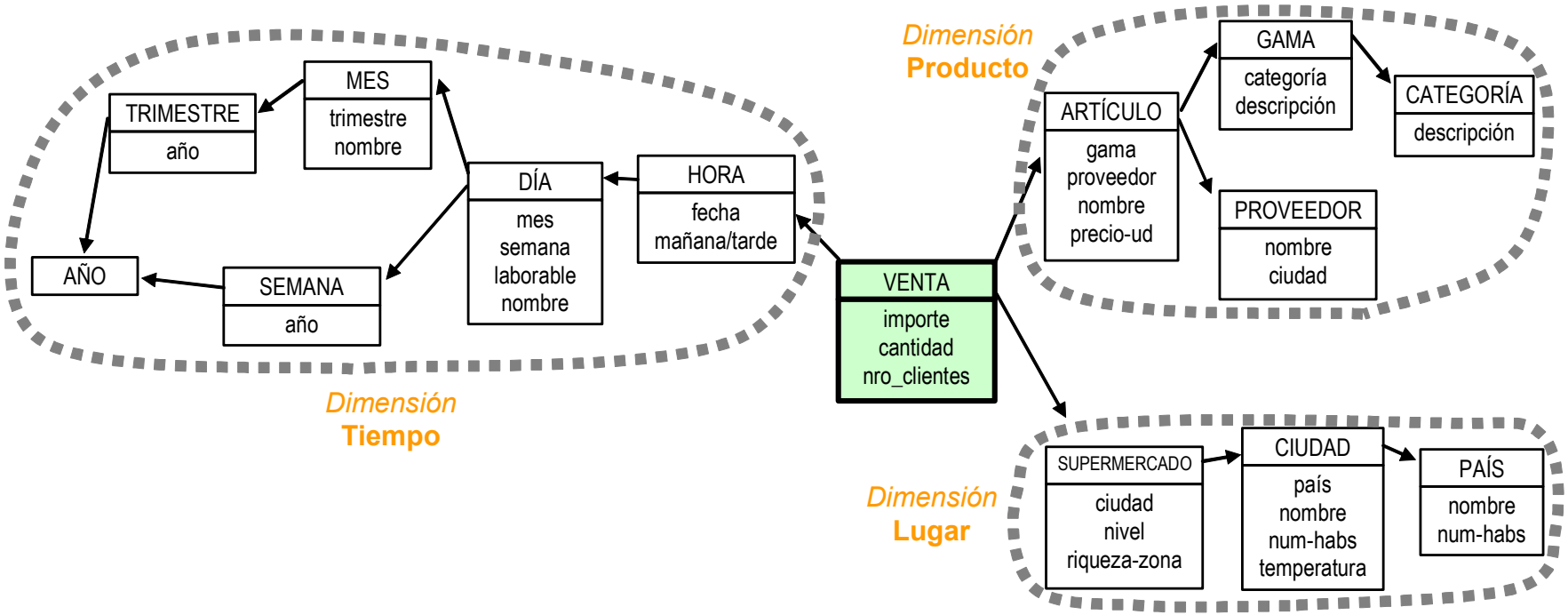
Ejemplo de ROLAP:

- Las dimensiones, que se puede agregar o desagregar, siguiendo claves ajenas. Se conserva parte de la normalización...



Data-Warehousing

Ejemplo de ROLAP:



De esta estructura suele venir el nombre de estrella...

Data-Warehousing

Esquemas de almacenes de datos más comunes:

- estrella simple
- estrella jerárquica (copo de nieve).

Esta estructura permite la sumarización, la visualización y la navegación según las dimensiones de la estrella.

Data-Warehousing

Sumarización y Operadores

- Estas estructuras permiten 'navegar' sumarizando (agregando) o desagregando.
- *Drill.* Se utiliza para desagregar dimensiones. Este operador permite entrar más al detalle en el informe.

CATEGORÍA	TRIMESTRE	IMPORTE
Refrescos	T1	150.323 euros
Refrescos	T2	233.992 euros
Refrescos	T3	410.497 euros
Refrescos	T4	203.400 euros
Congelados	T1	2.190.103 euros
Congelados	T2	1.640.239 euros
Congelados	T3	1.904.401 euros
Congelados	T4	2.534.031 euros

drill

categoría= "refrescos"

ciudad= {"Valencia", "León"}

CATEGORÍA	TRIMESTRE	CIUDAD	IMPORTE
Refrescos	T1	Valencia	13.267
Refrescos	T1	León	3.589
Refrescos	T2	Valencia	27.392
Refrescos	T2	León	4.278
Refrescos	T3	Valencia	73.042
Refrescos	T3	León	3.780
Refrescos	T4	Valencia	18.391
Refrescos	T4	León	3.629

Data-Warehousing

Sumarización y Operadores

- *Roll.* Operador inverso a *drill*. Obtiene información más agregada.

CATEGORÍA	TRIMESTRE	IMPORTE
Refrescos	T1	150.323 euros
Refrescos	T2	233.992 euros
Refrescos	T3	410.497 euros
Refrescos	T4	203.400 euros
Congelados	T1	2.190.103 euros
Congelados	T2	1.640.239 euros
Congelados	T3	1.904.401 euros
Congelados	T4	2.534.031 euros

roll

un nivel por "tiempo"

CATEGORÍA	IMPORTE
Refrescos	998.212 euros
Congelados	10.458.877 euros

Data-Warehousing

Sumarización y Operadores

- El operador *pivot* permite cambiar algunas filas por columnas.

CATEGORÍA	TRIMESTRE	Valencia	León
Refrescos	T1	13.267	3.589
Refrescos	T2	27.392	4.278
Refrescos	T3	73.042	3.780
Refrescos	T4	18.391	3.629
Congelados	T1	150.242	4.798
Congelados	T2	173.105	3.564
Congelados	T3	163.240	4.309
Congelados	T4	190.573	4.812

pivot

categoría × ciudad

CATEGORÍA	TRIMESTRE	Refrescos	Congelados
Valencia	T1	13.267	150.242
Valencia	T2	27.392	173.105
Valencia	T3	73.042	163.240
Valencia	T4	18.391	190.573
León	T1	3.589	4.798
León	T2	4.278	3.564
León	T3	3.780	4.309
León	T4	3.629	4.812

Data-Warehousing

Sumarización y Operadores

- *slice & dice.* Este operador permite escoger parte de la información mostrada, no por agregación sino por selección.

CATEGORÍA	TRIMESTRE	Valencia	León
Refrescos	T1	13.267	3.589
Refrescos	T2	27.392	4.278
Refrescos	T3	73.042	3.780
Refrescos	T4	18.391	3.629
Congelados	T1	150.242	4.798
Congelados	T2	173.105	3.564
Congelados	T3	163.240	4.309
Congelados	T4	190.573	4.812

slice & dice

trimestre = {T1, T4}
ciudad = Valencia

CATEGORÍA	Trimestre	Valencia
Refrescos	T1	13.267
Refrescos	T4	18.391
Congelados	T1	150.242
Congelados	T4	190.573

Data-Warehousing

Necesidad de los Almacenes de Datos:

- No son imprescindibles para hacer KDD pero sí convenientes.

Especialmente indicada para las dos tipologías de usuarios:

- 'picapedreros' (o 'granjeros'): se dedican fundamentalmente a realizar informes periódicos, ver la evolución de determinados parámetros, controlar valores anómalos, etc.
- 'exploradores': encargados de encontrar nuevos patrones significativos utilizando técnicas de minería de datos.

Data-Warehousing

Recogida de Información Externa:

- Aparte de información interna de la organización, los almacenes de datos pueden recoger información externa:
  - Demografías (censo), páginas amarillas, psicografías, gráficos web, información de otras organizaciones.
  - Datos compartidos en una industria o área de negocio, organizaciones y colegios profesionales, catálogos, etc.
  - Datos resumidos de áreas geográficas, distribución de la competencia, evolución de la economía, información de calendarios y climatológicas, programaciones televisivas-deportivas, catástofres,..
  - Bases de datos externas compradas a otras compañías.

Selección, Limpieza y Transformación de Datos

Limpieza (data cleansing) y criba (selección) de datos:

Se deben eliminar el mayor número posible de datos erróneos o inconsistentes (limpieza) e irrelevantes (criba).

Métodos estadísticos casi exclusivamente.

- histogramas (detección de datos anómalos).
- selección de datos (ya sea verticalmente, eliminando atributos u horizontalmente, eliminando tuplas).
- redefinición de atributos (agrupación o separación).

Muy relacionado con la disciplina de "Calidad de Datos".

Si los datos del sistema OLTP se han recogido con cuidado, intentanto mantener su calidad (evitando datos erróneos, obsoletos, inconsistencias). Mucho mejor si se tienen metadatos acerca de la calidad de datos (frec. de uso, etc.)

Selección, Limpieza y Transformación de Datos

El primer paso en la limpieza de datos consiste en la elaboración de un resumen de características

ATRIBUTO	TIPO	# TOTAL	# NULS	# DIST	MEDIA	DESV.	MODA	MIN	MAX
Código postal	Nominal	10320	150	1672	-	-	“46003”	“01001”	“50312”
Sexo	Nominal	10320	23	6	-	-	“V”	“E”	“M”
Estado civil	Nominal	10320	317	8	-	-	Casado	“Casado”	“Viudo”
Edad	Numérico	10320	4	66	42,3	12,5	37	18	87
Total póliza p/a	Numérico	17523	1325	142	737,24€	327€	680€	375€	6200€
Asegurados	Numérico	17523	0	7	1,31	0,25	1	0	10
Matrícula	Nominal	16324	0	16324	-	-	-	“A-0003-BF”	“Z-9835-AF”
Modelo	Nominal	16324	1321	2429	-	-	“O. Astra”	“Audi A3”	“VW Polo”
...	...	...	...	...	...	...	...	...	...

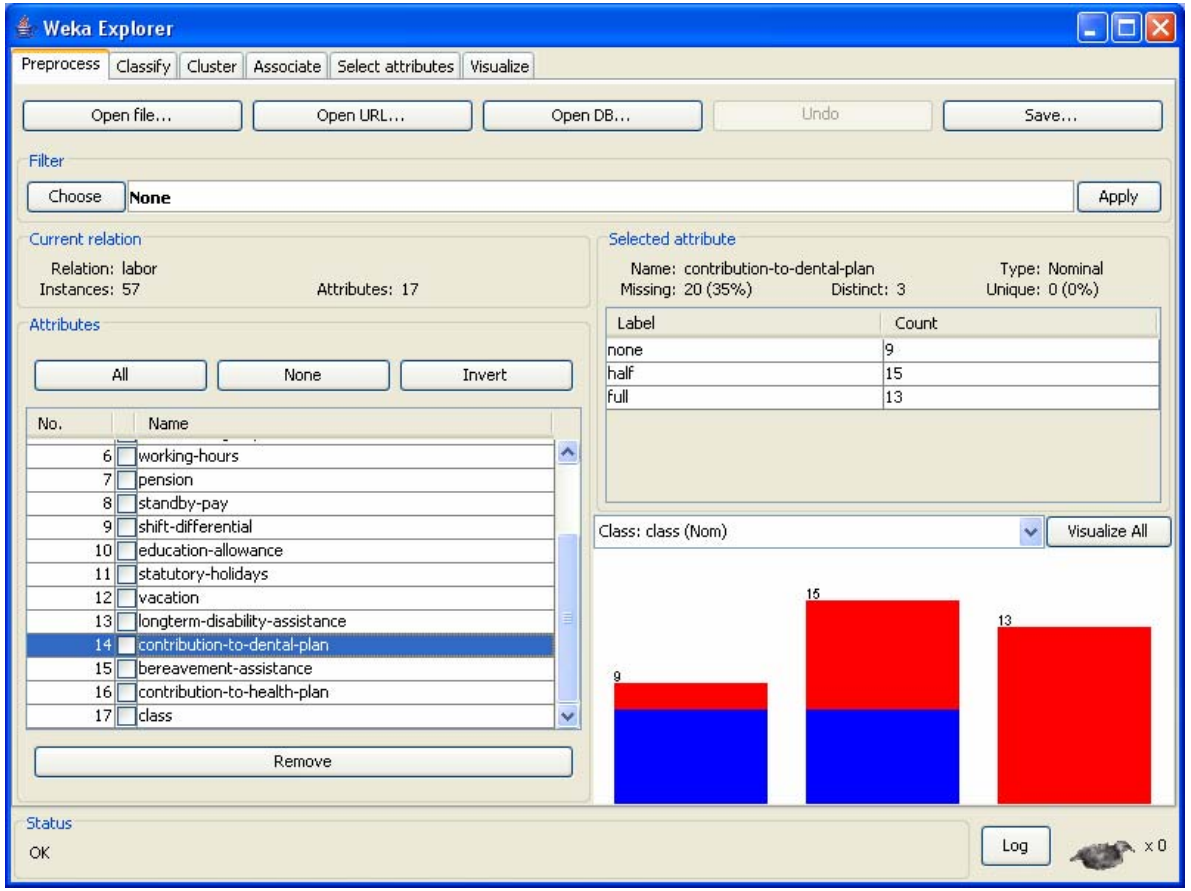
Selección, Limpieza y Transformación de Datos

Atributos Nominales: Debemos analizar con detalle cada uno de los atributos:

Podemos detectar:

•Valores redundantes: (Hombre,Varón)

•Valores despreciables (agrupar valores como *otros*)



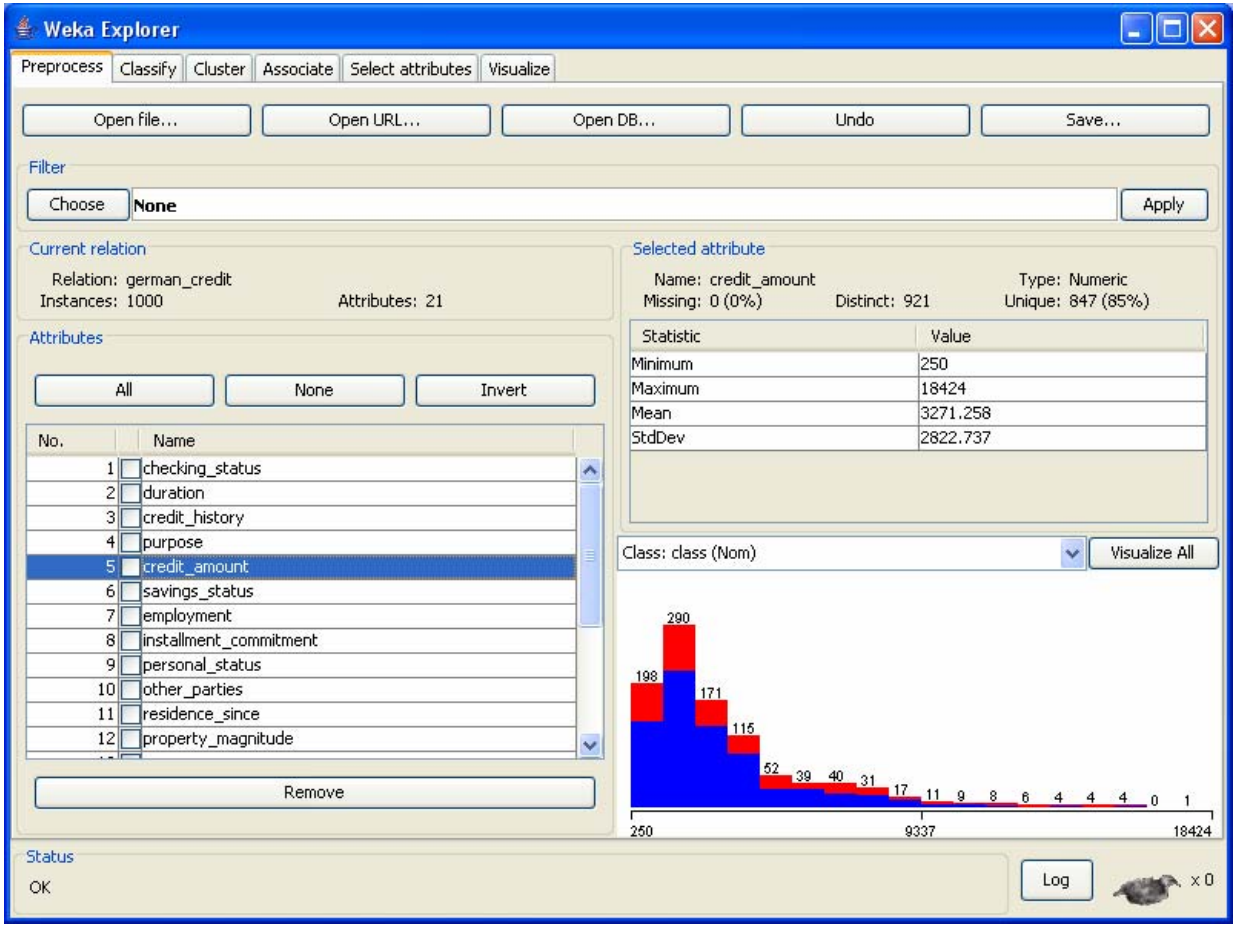
Selección, Limpieza y Transformación de Datos

Atributos Numéricos: Debemos analizar con detalle cada uno de los atributos:

Podemos detectar:

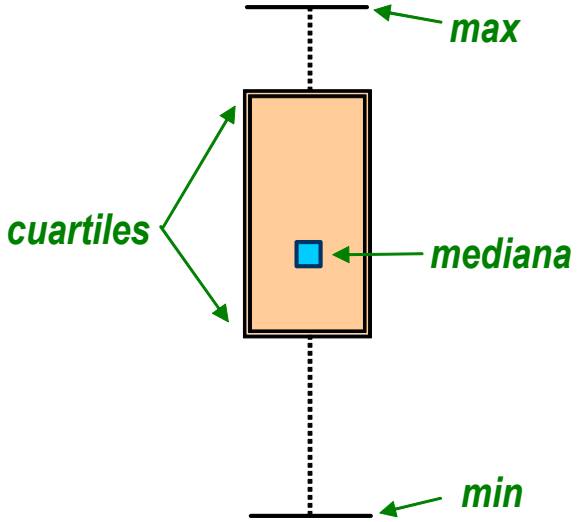
•Valores anómalos

•Distribuciones en los datos



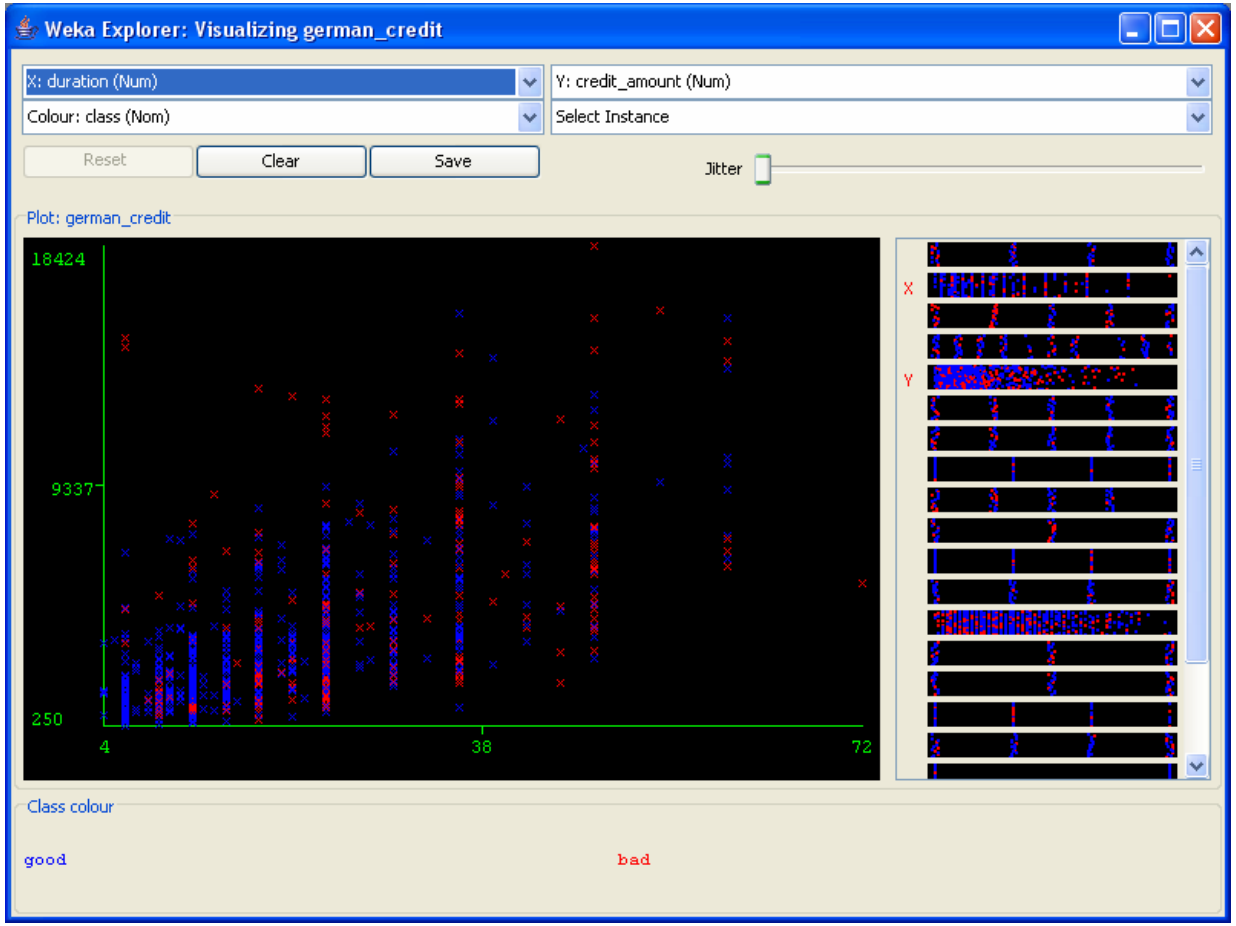
Selección, Limpieza y Transformación de Datos

Atributos Numéricos: Otra alternativa para los atributos numéricos son los diagramas de caja (*box plot*) o de bigotes (*whisker plots*).



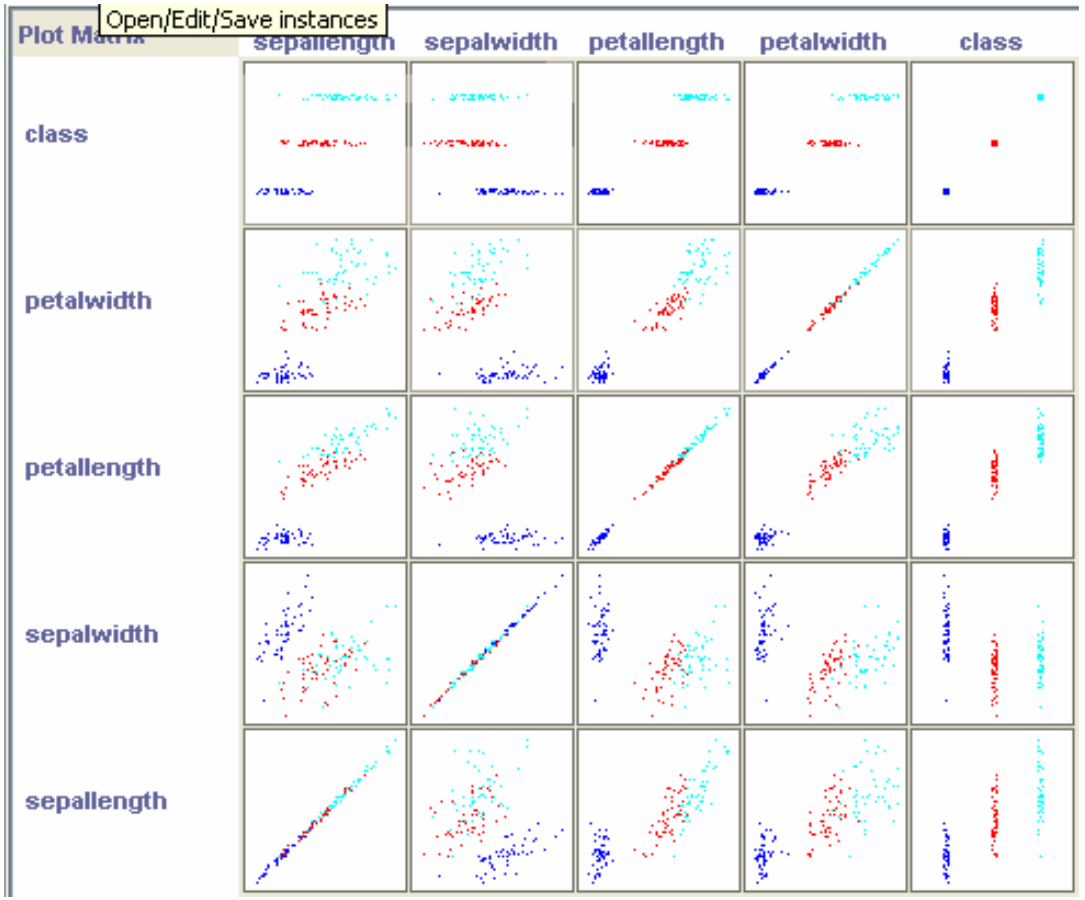
Selección, Limpieza y Transformación de Datos

Atributos Numéricos: Otra alternativa especialmente útil para los atributos numéricos son las gráficas de dispersión.



Selección, Limpieza y Transformación de Datos

Atributos Numéricos: Cuando tenemos más de dos variables el gráfico anterior se puede repetir para todas las combinaciones posibles.



Selección, Limpieza y Transformación de Datos

Acciones ante datos anómalos (outliers):

- **ignorar**: algunos algoritmos son robustos a datos anómalos (p.ej. árboles)
- **filtrar** (eliminar o reemplazar) la columna: solución extrema, pero a veces existe otra columna dependiente con datos de mayor calidad. Preferible a eliminar la columna es reemplazarla por una columna discreta diciendo si el valor era normal o outlier (por encima o por debajo).
- **filtrar la fila**: claramente sesga los datos, porque muchas veces las causas de un dato erróneo están relacionadas con casos o tipos especiales.
- **reemplazar el valor**: por el valor 'nulo' si el algoritmo lo trata bien o por máximos o mínimos, dependiendo por donde es el outlier, o por medias. A veces se puede *predecir* a partir de otros datos, utilizando cualquier técnica de ML.
- **discretizar**: transformar un valor continuo en uno discreto (p.ej. muy alto, alto, medio, bajo, muy bajo) hace que los outliers caigan en 'muy alto' o 'muy bajo' sin mayores problemas.

Selección, Limpieza y Transformación de Datos

Acciones ante datos faltantes (missing values):

- **ignorar**: algunos algoritmos son robustos a datos faltantes (p.ej. árboles).
- **filtrar (eliminar o reemplazar) la columna**: solución extrema, pero a veces existe otra columna dependiente con datos de mayor calidad. Preferible a eliminar la columna, es reemplazarla por una columna booleana diciendo si el valor existía o no.
- **filtrar la fila**: claramente sesga los datos, porque muchas veces las causas de un dato faltante están relacionadas con casos o tipos especiales.
- **reemplazar el valor**: por medias. A veces se puede *predecir* a partir de otros datos, utilizando cualquier técnica de ML.
- **segmentar**: se segmentan las tuplas por los valores que tienen disponibles. Se obtienen modelos diferentes para cada segmento y luego se combinan.
- **modificar la política** de calidad de datos y esperar hasta que los datos faltantes estén disponibles.

Selección, Limpieza y Transformación de Datos

Razones sobre datos faltantes (missing values):

A veces es importante examinar las razones tras datos faltantes y actuar en consecuencia:

- algunos valores faltantes expresan características relevantes: p.ej. la falta de teléfono puede representar en muchos casos un deseo de que no se moleste a la persona en cuestión, o un cambio de domicilio reciente.
- valores no existentes: muchos valores faltantes existen en la realidad, pero otros no. P.ej. el cliente que se acaba de dar de alta no tiene consumo medio de los últimos 12 meses.
- datos incompletos: si los datos vienen de fuentes diferentes, al combinarlos se suele hacer la unión y no la intersección de campos, con lo que muchos datos faltantes representan que esas tuplas vienen de una/s fuente/s diferente/s al resto.

Selección, Limpieza y Transformación de Datos

Inconsistencias:

Un problema grave que afecta a varios métodos de aprendizaje predictivo son los registros inconsistentes, es decir, dos o más registros con los mismo valores en los atributos, pero diferente valor en el atributo clase.

Algunas técnicas no soportan las inconsistencias en los datos. Por lo que se deben eliminar unificando (siempre que se pueda) los registros en una única clase.

Transformación de Atributos

La transformación de datos engloba, en realidad, cualquier proceso que modifique la forma de los datos.

Por transformación entendemos aquellas técnicas que transforman un conjunto de atributos en otros, o bien derivan nuevos atributos, o bien cambian el tipo o el rango.

La selección de atributos (eliminar los menos relevantes) en realidad no transforma atributos y, en consecuencia, no entra en este grupo de técnicas.

Reducción de Dimensionalidad

Si tenemos muchas dimensiones (atributos) respecto a la cantidad de instancias, pueden existir demasiados grados de libertad, por lo que los patrones extraídos pueden ser poco robustos.

Este problema se conoce popularmente como "**la maldición de la dimensionalidad**" ("*the curse of dimensionality*"). Una manera de intentar resolver este problema es mediante la reducción de dimensiones.

La reducción se puede realizar por selección de un subconjunto de atributos, o bien la sustitución del conjunto de atributos iniciales por otros diferentes.

Análisis de Componentes Principales

La técnica más conocida para reducir la dimensionalidad por transformación se denomina "**análisis de componentes principales**" ("*principal component analysis*"), **PCA**.

PCA transforma los m atributos originales en otro conjunto de atributos p donde $p \leq m$.

Este proceso se puede ver geométricamente como un cambio de ejes en la representación (proyección).

Los nuevos atributos se generan de tal manera que son independientes entre sí y, además, los primeros tienen más relevancia (más contenido informacional) que los últimos.

Análisis de Componentes Principales

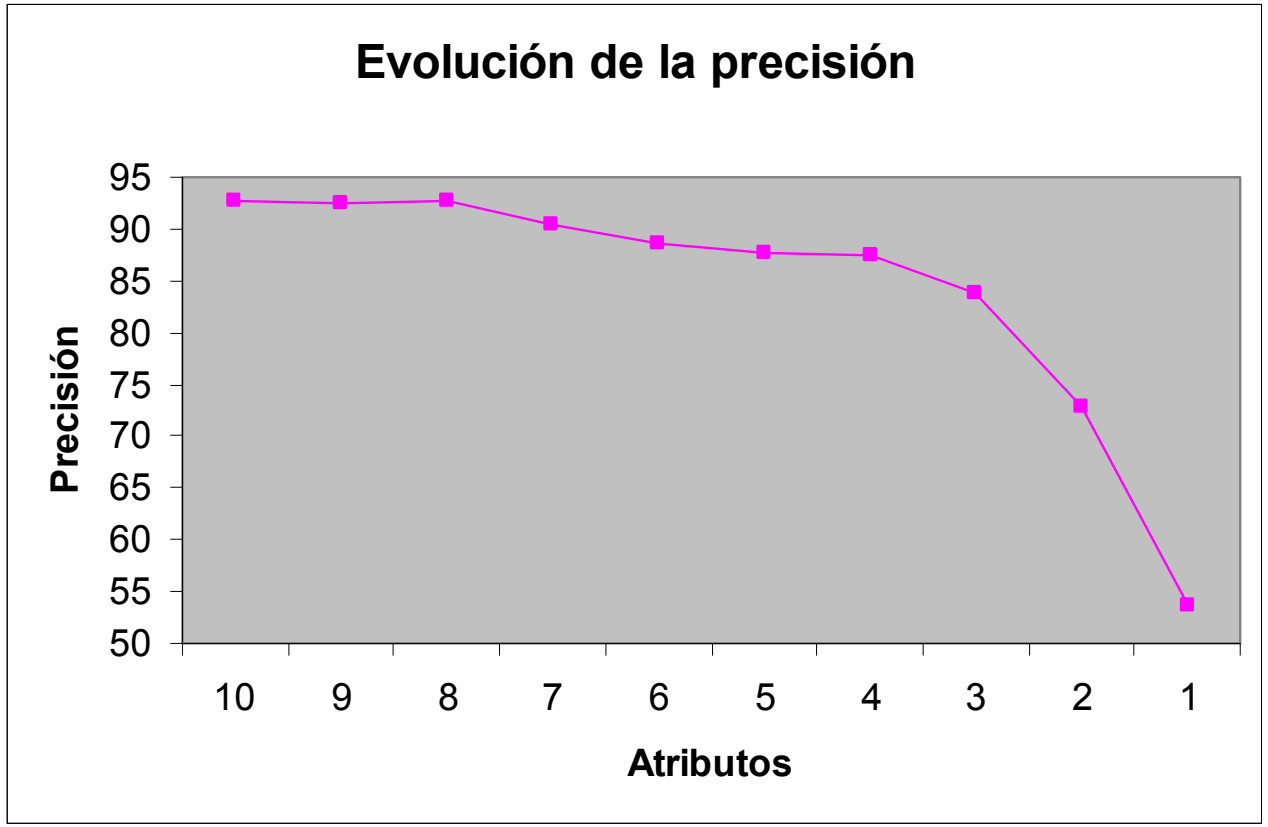
Ejemplo: Dataset *Segment*, 19 Atributos

El método Principal Components genera 10 atributos cubriendo el 97% de la varianza

eigenvalue	proportion	cumulative	
7.6214	0.42341	0.42341	-0.357rawblue-mean-0.355value-mean-0.351intensity-mean-0.348rawred-mean-0.343rawgreen-mean...
2.91666	0.16204	0.58545	0.495hedge-sd+0.481vegde-sd+0.472hedge-mean+0.466vedge-mean+0.257short-line-density-2...
1.7927	0.09959	0.68504	0.596hue-mean+0.428exgreen-mean+0.373region-centroid-row-0.363saturation-mean-0.192exblue-mean...
1.05431	0.05857	0.74362	0.714short-line-density-5-0.677region-centroid-col+0.127short-line-density-2-0.07vegde-sd-0.057exgre...
0.93564	0.05198	0.7956	-0.63region-centroid-col-0.462short-line-density-5-0.453short-line-density-2+0.213exgreen-mean-0...
0.90907	0.0505	0.8461	-0.694short-line-density-2+0.483short-line-density-5+0.323region-centroid-col+0.282vegde-sd...
0.72745	0.04041	0.88651	0.456region-centroid-row-0.434saturation-mean-0.428short-line-density-2-0.387exgreen-mean...
0.56163	0.0312	0.91771	-0.514vedge-mean-0.438saturation-mean+0.406exred-mean+0.357hedge-sd-0.331region-centroid-row...
0.53996	0.03	0.94771	0.491saturation-mean-0.438vedge-mean+0.418hedge-mean+0.317hedge-sd-0.297exred-mean...
0.39511	0.02195	0.96966	0.498hedge-mean-0.473vegde-sd-0.424region-centroid-row+0.392vedge-mean-0.29hedge-sd...

Análisis de Componentes Principales

Evolución de la precisión de J48 dependiendo del número de atributos



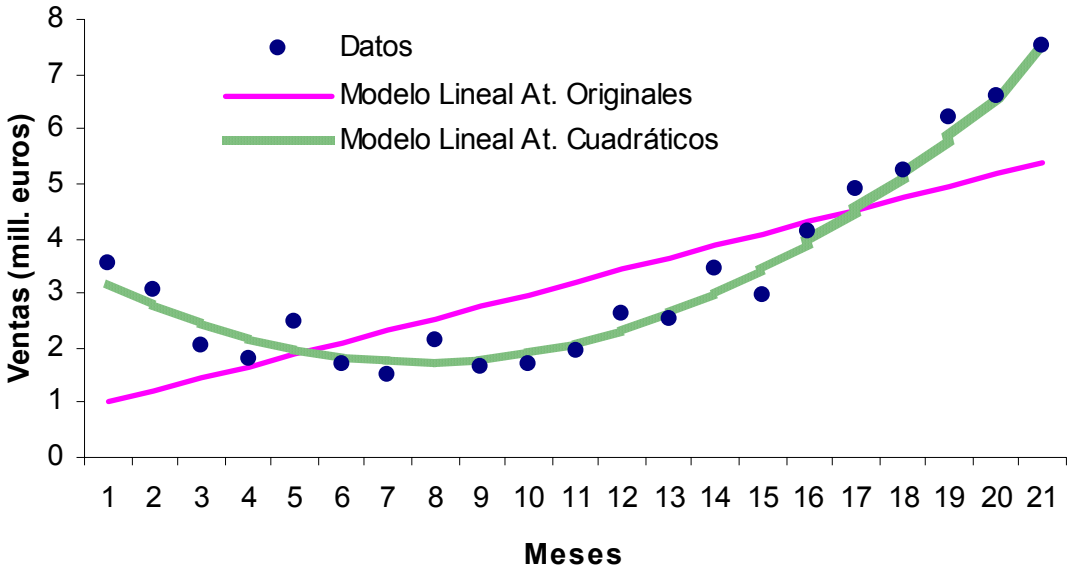
Kernels

El aprendizaje SVM se consigue mediante una transformación no lineal del espacio de atributos de entrada (*input space*) en un espacio de características (*feature space*) de dimensionalidad mucho mayor y donde sí es posible separar linealmente los ejemplos.

El uso de las denominadas funciones núcleo (*kernel functions*), que calculan el producto escalar de dos vectores en el espacio de características, permite trabajar de manera eficiente en el espacio de características sin necesidad de calcular explícitamente las transformaciones de los ejemplos de aprendizaje.

Aumento de Dimensionalidad

En otras ocasiones añadir atributos nuevos puede mejorar el proceso de aprendizaje



- La regresión lineal no se aproxima a la solución

- Añadiendo un nuevo atributo z=meses^2 se obtiene un buen modelo

Aumento de Dimensionalidad

Técnicas de generación de nuevos atributos:

•**Numéricos:** Se utilizan, generalmente, operaciones matemáticas básicas de uno o más argumentos

•**Nominales:** Operaciones lógicas: conjunción, disyunción, negación, implicación, condiciones M-de-N (M-de-N es cierto si y sólo si al menos M de las N condiciones son ciertas), igualdad o desigualdad

Aumento de Dimensionalidad

El conocimiento del dominio es el factor que más determina la creación de buenos atributos derivados

Atributo Derivado	Fórmula
Índice de obesidad	Altura^2 / peso
Hombre familiar	Casado, varón e “hijos>0”
Síntomas SARS	3-de-5 (fiebre alta, vómitos, tos, diarrea, dolor de cabeza)
Riesgo póliza	X-de-N (edad < 25, varón, años de carné < 2, vehículo deportivo)
Beneficios brutos	Ingresos - Gastos
Beneficios netos	Ingresos – Gastos – Impuestos
Desplazamiento	Pasajeros * kilómetros
Duración media	Segundos de llamada / número de llamadas
Densidad	Población / Área
Retardo compra	Fecha compra – Fecha campaña

Discretización de Atributos

La discretización, o cuantización (también llamada "*binning*") es la conversión de un valor numérico en un valor nominal ordenado

La discretización se debe realizar cuando:

- El error en la medida puede ser grande
- Existen umbrales significativos (p.e. notas)
- En ciertas zonas el rango de valores es más importante que en otras (interpretación no lineal)
- Aplicar ciertas tareas de MD que sólo soportan atributos nominales (p.e. reglas de asociación)

Discretización de Atributos

Ejemplo con el dataset *iris*: 150 ejemplos de lirios de tres tipos, con cuatro atributos numéricos (*sepallength, sepalwidth, petallength, petalwidth*) y un quinto nominal, que representa la clase o tipo de lirio (*setosa, versicolour, virginica*):

	Min	Max	Media	Desv.Típ.
sepallength	4,3	7,9	5,84	0,83
sepalwidth	2,0	4,4	3,05	0,43
petallength	1,0	6,9	3,76	1,76
petalwidth	0,1	2,5	1,20	0,76

Discretización de Atributos

La discretización más sencilla (*simple binning*) es aquella que realiza intervalos del mismo tamaño y utilizando el mínimo y el máximo como referencia.

Atributo	1	2	3	4	5
sepallength	(-inf-5,02): 32	(5,02-5,74): 41	(5,74-6,46): 42	(6,46-7,18): 24	(7,18-inf): 11
sepalwidth	(-inf-2,48): 11	(2,48-2,96): 46	(2,96-3,44): 69	(3,44-3,92): 20	(3,92-inf): 4
petallength	(-inf-2,18): 50	(2,18-3,36): 3	(3,36-4,54): 34	(4,54-5,72): 47	(5,72-inf): 16
petalwidth	(-inf-0,58): 49	(0,58-1,06): 8	(1,06-1,54): 41	(1,54-2,02): 29	(2,02-inf): 23

Discretización de Atributos

Otra técnica de discretización sencilla es intentar obtener intervalos con el mismo número de registros (*Equal-frequency binning*) .

Sin embargo estos técnicas de discretización ignoran la clase, por lo que pueden dar lugar a intervalos no adecuados.

Discretización de Atributos

Existen opciones más elaboradas. Como por ejemplo las basadas en el principio MDL. [Fayyad & Irani 1993]

Atributo	Discretización
sepallength	(-inf, 5.55] (59) [5.55, 6.15) (36) [6.15,inf] (55)
sepalwidth	(-inf, 2.95] (57) [2.95, 3.35) (57) [3.35,inf] (36)
petallength	(-inf, 2.45] (50) [2.45, 4.75) (45) [4.75,inf] (55)
petalwidth	(-inf, 5.55] (59) [5.55, -6.15) (36) [6.15,inf] (55)

Numerización de Atributos

La numerización es el proceso inverso a la discretización, es decir, convertir un atributo nominal en numérico.

La discretización se debe realizar cuando se quieren aplicar ciertas técnicas de MD que sólo soportan atributos numéricos (p.e. Regresión, métodos basados en distancias)

Numerización de Atributos

numerización "1 a n": Si una variable nominal x tiene posibles valores creamos n variables numéricas, con valores 0 o 1 dependiendo de si la variable nominal toma ese valor o no. Podemos también prescindir del último atributo pues es dependiente del resto (numerización "1 a n-1").

numerización "1 a 1": Se aplica si existe un cierto orden o magnitud en los valores del atributo nominal. Por ejemplo, si tenemos categorías del estilo {niño, joven, adulto, anciano} podemos numerar los valores de 1 a 4.

Normalización de Atributos

Algunos métodos de aprendizaje funcionan mejor con los atributos normalizados. Por ejemplo, los métodos basados en distancias.

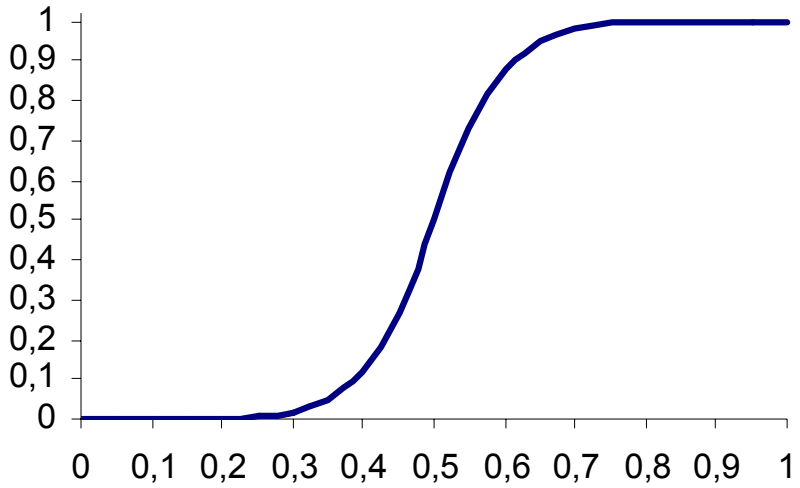
La normalización más común es la **normalización lineal uniforme**:

$$v' = \frac{v - \min}{\max - \min}$$

Esta normalización es muy sensible a la presencia de valores anómalos (*outliers*).

Normalización de Atributos

Una solución a este problema es el escalado *softmax* o escalado sigmoidal, en el que no se usa una transformación que es más pronunciada en el centro y más aplanada en los bordes



Métodos de Selección de Características

Existen dos tipos generales de métodos para seleccionar características:

- **Métodos de filtro** o métodos previos: se filtran los atributos irrelevantes antes de cualquier proceso de minería de datos y, en cierto modo, independiente de él.

- **Métodos basados en modelo** o métodos de envolvente (*wrapper*): la bondad de la selección de atributos se evalúa respecto a la calidad de un modelo de extraído a partir de los datos (utilizando, lógicamente, algún buen método de validación).

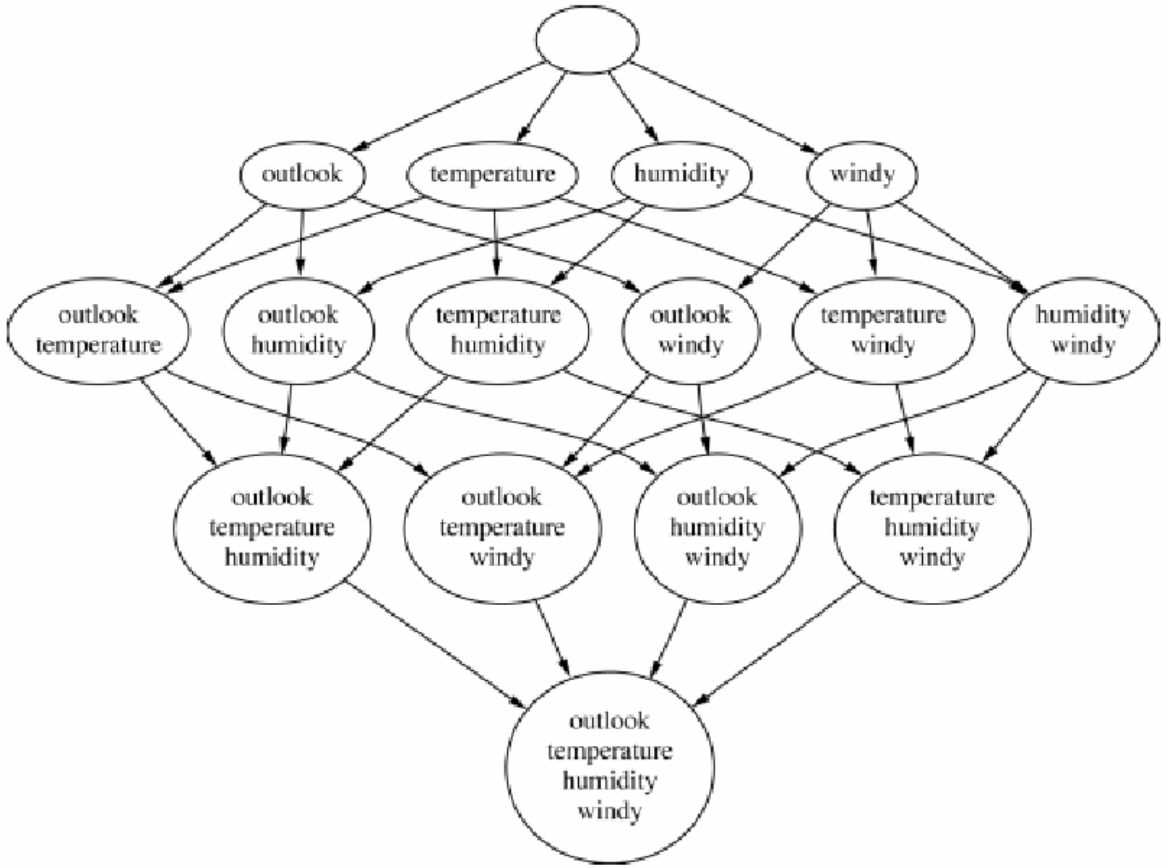
Métodos de Selección de Características

En cualquier caso se emplea una estrategia iterativa, es decir, se van eliminando atributos y se va observando el resultado. Se van recuperando o eliminando más atributos de una manera iterativa, hasta que se obtiene una combinación que maximiza la calidad.

De acuerdo con la medida de evaluación, estrategia y dirección de búsqueda, podemos establecer taxonomías más refinadas.

Métodos de Selección de Características

El número de subconjuntos de atributos crece exponencialmente con respecto al número de atributos



Métodos de Selección de Características

Medida de Evaluación:

•**Clásicas**: ganancia de información, o medidas de dependencia entre características

•**Acierto**: Precisión u cualquier otra medida de evaluación de calidad medido sobre un conjunto de test

•**Consistencia**: medidas que miden el grado de inconsistencias (registros iguales salvo en la clase) en el conjunto de datos

Métodos de Selección de Características

Estrategia de Búsqueda:

•**Completa**: Se cubren todas las combinaciones posibles de selección.

•**Heurística**: reduce el número de combinaciones a evaluar basándose en algún tipo de información.

•**No determinista (estocástico)**: basada en algoritmos de búsqueda globales. Intentan evitar el problema de mínimos locales.

Métodos de Selección de Características

Dirección de Búsqueda:

• **Forward**: Empezando con el mejor atributo y añadir el atributo que dé mayor calidad de selección con dos atributos, y así hasta que no se mejore la calidad o se llegue al número deseado de atributos

• **Backward**: Se inicia el proceso con todos los atributos eliminando uno a uno el menos relevante.

• **Aleatoria**: Se producen patrones de búsqueda mediante la creación de conjuntos de manera aleatoria

Análisis Correlacional

Una técnica sencilla consiste en utilizar una matriz de correlaciones.

ATRIBUTO	EDAD	TENSIÓN	OBES.	COLEST.	TABAQ.	ALCOHOL.	PULS.	HIERRO
Edad		0,63	0,34	0,42	-0,02	0,15	0,12	-0,33
Tensión	0,63		0,22	0,56	0,72	0,43	0,27	-0,08
Obesidad	0,34	0,22		0,67	0,72	0,32	0,32	0,21
Colesterol	0,42	0,56	0,67		0,52	0,27	0,40	0,45
Tabaquismo	-0,02	0,72	0,72	0,52		0,58	0,39	-0,12
Alcoholismo	0,15	0,43	0,32	0,27	0,58		0,23	-0,22
Pulsaciones	0,12	0,27	0,32	0,40	0,39	0,23		-0,15
Hierro	-0,33	-0,08	0,21	0,45	-0,12	-0,22	-0,15	

Análisis por modelo lineal

Queremos obtener un modelo de predicción de azúcar en la sangre a partir de los datos anteriores.

$$y = a_0 + a_1 x_1 + a_2 x_2 + \ldots + a_n x_n$$

Obtenemos los siguientes coeficientes:

ATRIBUTO	EDAD	TENSIÓN	OBES.	COLEST.	TABAQ.	ALCOHOL.	PULS.	HIERRO
Azúcar	2,23	-1,63	3,23	0,42	-0,12	2,23	0,00	-3,01

Esto, en realidad, es sólo el principio de múltiples técnicas del análisis multivariante. Si quisiéramos saber si el colesterol, el tabaquismo y las pulsaciones no influyen en el azúcar y, son, por tanto, descartables, deberíamos usar, por ejemplo, el Análisis de la Varianza (ANOVA).

Muestreo

La manera más directa de reducir el tamaño de una población o conjuntos de individuos es realizar una selección o muestreo.

Nos podemos plantear dos situaciones, dependiendo de la disponibilidad de la población:

- Se dispone de la población: en este caso se ha de determinar qué cantidad de datos son necesarios y cómo hacer la muestra
- Los datos son ya una muestra de realidad y sólo representan una parte de esta realidad

Tipos de Muestreo

•**Muestreo Aleatorio Simple**: Cualquier instancia tiene la misma probabilidad de ser extraída en la muestra. Dos versiones, con reemplazamiento y sin reemplazamiento.

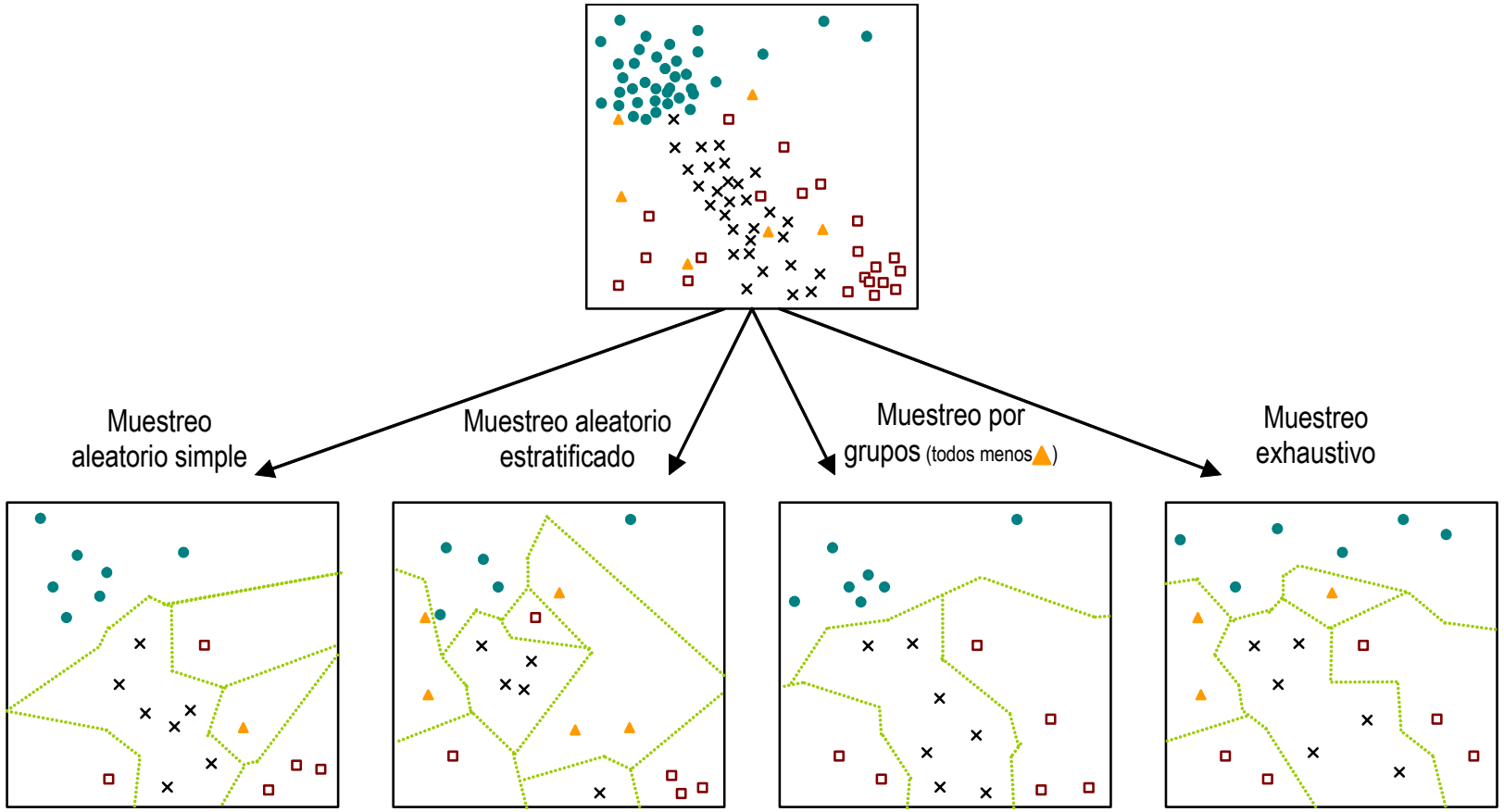
•**Muestreo Aleatorio Estratificado**: El objetivo de este muestreo es obtener una muestra balanceada con suficientes elementos de todos los estratos, o grupos. Una versión simple es realizar un muestreo aleatorio simple sin reemplazamiento de cada estrato hasta obtener los *n* elementos de ese estrato. Si no hay suficientes elementos en un estrato podemos utilizar en estos casos muestreo aleatorio simple con reemplazamiento (sobremuestreo).

Tipos de Muestreo

•**Muestreo de Grupos**: El muestreo de grupos consiste en elegir sólo elementos de unos grupos. El objetivo de este muestreo es generalmente descartar ciertos grupos que, por diversas razones, pueden impedir la obtención de buenos modelos.

•**Muestreo Exhaustivo**: Para los atributos numéricos (normalizados) se genera al azar un valor en el intervalo posible; para los atributos nominales se genera al azar un valor entre los posibles. Con esto obtenemos una instancia ficticia y buscamos la instancia real más similar a la ficticia. Se repite este proceso hasta tener *n* instancias. el objetivo de este método es cubrir completamente el espacio de instancias.

Tipos de Muestreo



Tipos de Muestreo

•Cúantos datos son necesarios mantener?

Depende, en general, del número de “grados de libertad” (número de atributos y valores) y del método de aprendizaje y de su expresividad (por ejemplo una regresión lineal requiere muchos menos ejemplos que una red neuronal).

Se utiliza una estrategia incremental, en el que se va haciendo la muestra cada vez más grande (y diferente si es posible) hasta que se vea que los resultados no varían significativamente entre un modelo y otro.

La Minería de Datos

Características Especiales de los Datos:

Aparte del gran volumen, ¿por qué algunas técnicas de aprendizaje automático y estadísticas no son directamente aplicables?

- Los datos residen en el disco. No se pueden escanear múltiples veces.
- Algunas técnicas de muestreo no son compatibles con algoritmos no incrementales.
- Muy alta dimensionalidad.
- Evidencia POSITIVA.
- DATOS IMPERFECTOS...

La Minería de Datos

Características Especiales de los Datos (cont.):

TIPOS DE DATOS IMPERFECTOS:

- Ruido:
  - en la evidencia o ejemplos de entrenamiento.
    - Erróneos valores de argumentos de los ejemplos.
    - Clasificación errónea de algún ejemplo.
  - en el conocimiento previo.
- Ejemplos de entrenamiento muy dispersos.
- Conocimiento previo correcto pero inapropiado.
  - Existencia de muchos predicados irrelevantes para el problema a aprender.
  - Conocimiento previo insuficiente para el problema a aprender (algunos predicados auxiliares serían necesarios).
- Argumentos faltantes (nulos) en los ejemplos.

La Minería de Datos

PATRONES A DESCUBRIR:

- Una vez recogidos los datos de interés en un almacén de datos, un explorador puede decidir qué tipos de patrón quiere descubrir.
- El tipo de conocimiento que se desea extraer va a marcar claramente la *técnica* de minería de datos a utilizar.
- Según como sea la búsqueda del conocimiento se puede distinguir:
  - *Directed data mining*: se sabe claramente lo que se busca; generalmente predecir unos ciertos datos o clases.
  - *Undirected data mining*: no se sabe lo que se busca, se trabaja con los datos (hasta que confiesen!).
- En el primer caso, los propios sistemas de minería de datos se encargan generalmente de elegir el *algoritmo* más idóneo entre los disponibles para un determinado tipo de patrón a buscar.

La Minería de Datos

Tipos de conocimiento:

- Asociaciones: Una asociación entre dos atributos ocurre cuando la frecuencia con la que se dan dos valores determinados de cada uno conjuntamente es relativamente alta.
    - Ejemplo: en un supermercado se analiza si los pañales y los potitos de bebé se compran conjuntamente.
- Dependencias: Una dependencia funcional (aproximada o absoluta) es un patrón en el que se establece que uno o más atributos determinan el valor de otro. Ojo! Existen muchas dependencias nada interesantes (ojo con causalidades inversas).
    - Ejemplo: que un paciente haya sido ingresado en maternidad determina su sexo

La búsqueda de asociaciones y dependencias se conoce a veces como análisis exploratorio.

La Minería de Datos

Tipos de conocimiento (cont.):

- Clasificación: Una clasificación se puede ver como el esclarecimiento de una dependencia, en la que el atributo dependiente puede tomar un valor entre varias clases, ya conocidas.
  - Ejemplo: se sabe (por un estudio de dependencias) que los atributos edad, número de dioptrías y astigmatismo han determinado los pacientes para los que su operación de cirugía ocular ha sido satisfactoria.
  - Podemos intentar determinar las reglas exactas que clasifican un caso como positivo o negativo a partir de esos atributos.
- Segmentación: La segmentación (o clustering) es la detección de grupos de individuos. Se diferencia de la clasificación en el que no se conocen ni las clases ni su número (aprendizaje no supervisado), con lo que el objetivo es determinar grupos o racimos (clusters) diferenciados del resto.

La Minería de Datos

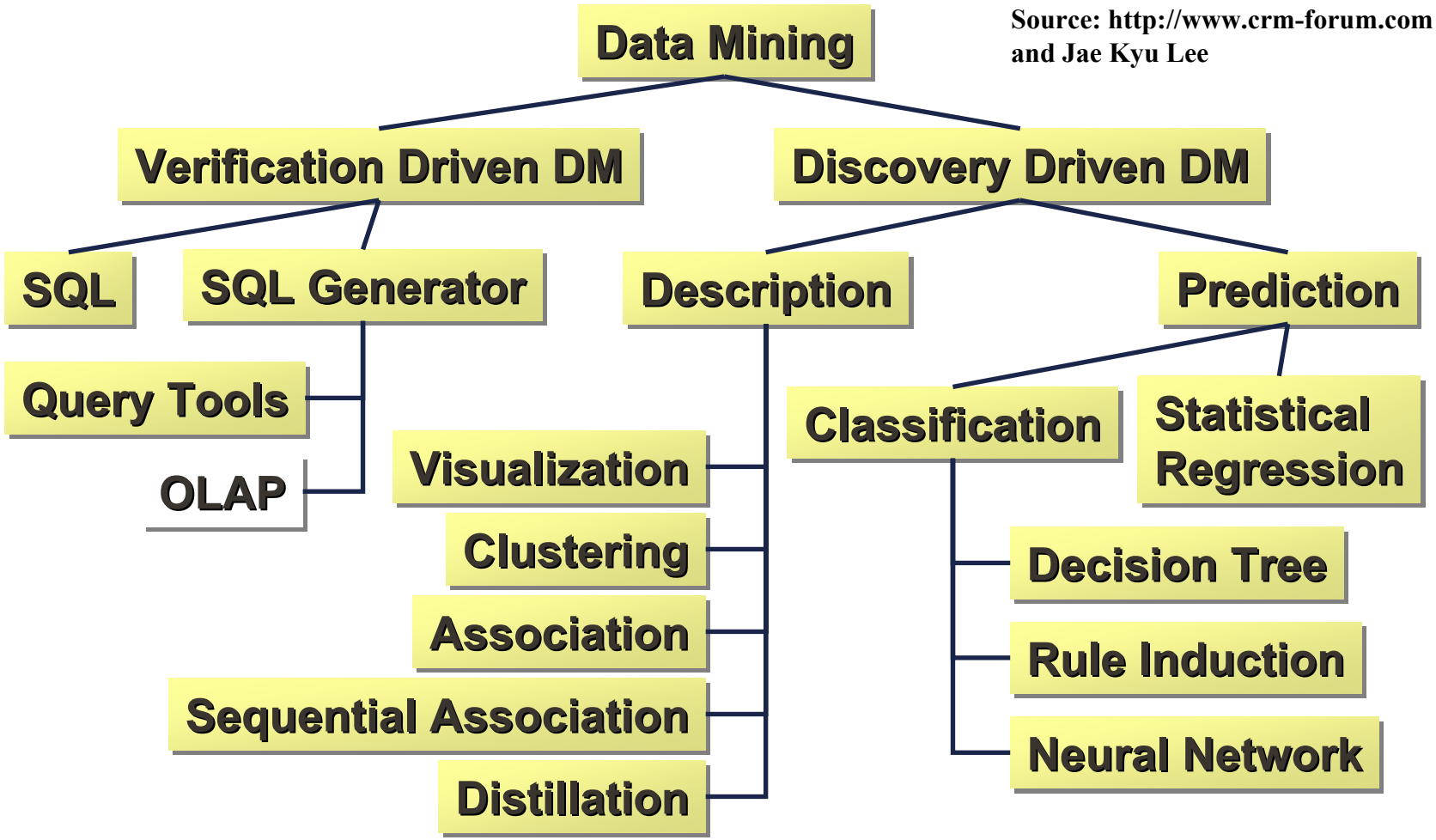
Tipos de conocimiento (cont.):

- Tendencias: El objetivo es predecir los valores de una variable continua a partir de la evolución de otra variable continua, generalmente el tiempo.
  - Ejemplo, se intenta predecir el número de clientes o pacientes, los ingresos, llamadas, ganancias, costes, etc. a partir de los resultados de semanas, meses o años anteriores.

- Información del Esquema: (descubrir claves primarias alternativas, R.I.).

- Reglas Generales: patrones que no se ajustan a los tipos anteriores. Recientemente los sistemas incorporan capacidad para establecer otros patrones más generales.

Taxonomía Técnicas de Minería de Datos.



Source: http://www.crm-forum.com and Jae Kyu Lee

Métodos Específicos de Minería de Datos.

Algoritmos de Aprendizaje de Árboles de Decisión Escalables:

- Diseñados con los siguientes requerimientos:
  - *No requieren que los datos quepan en memoria.*
  - *Los chequeos de consistencia se hacen eficientemente, utilizando índices, con el objetivo de agilizar los escaneos de los datos.*
  - *Las condiciones sobre índices son preferibles sobre aquellas que no permiten indización. P.ej.:*
    - *$x<3$, $x \geq 3$ es un split mucho más eficiente de comprobar que:*
    - *$x<y$, $x \geq y$ para el cual no se puede definir índices.*

Ejemplo: (Mehta, Agrawal and Rissanen 1996)

Métodos Específicos de Minería de Datos.

Algoritmos de Aprendizaje de Árboles de Decisión Escalables:

- Se utilizan las cuentas de correlación:

Variable a predecir

Una tabla para cada atributo

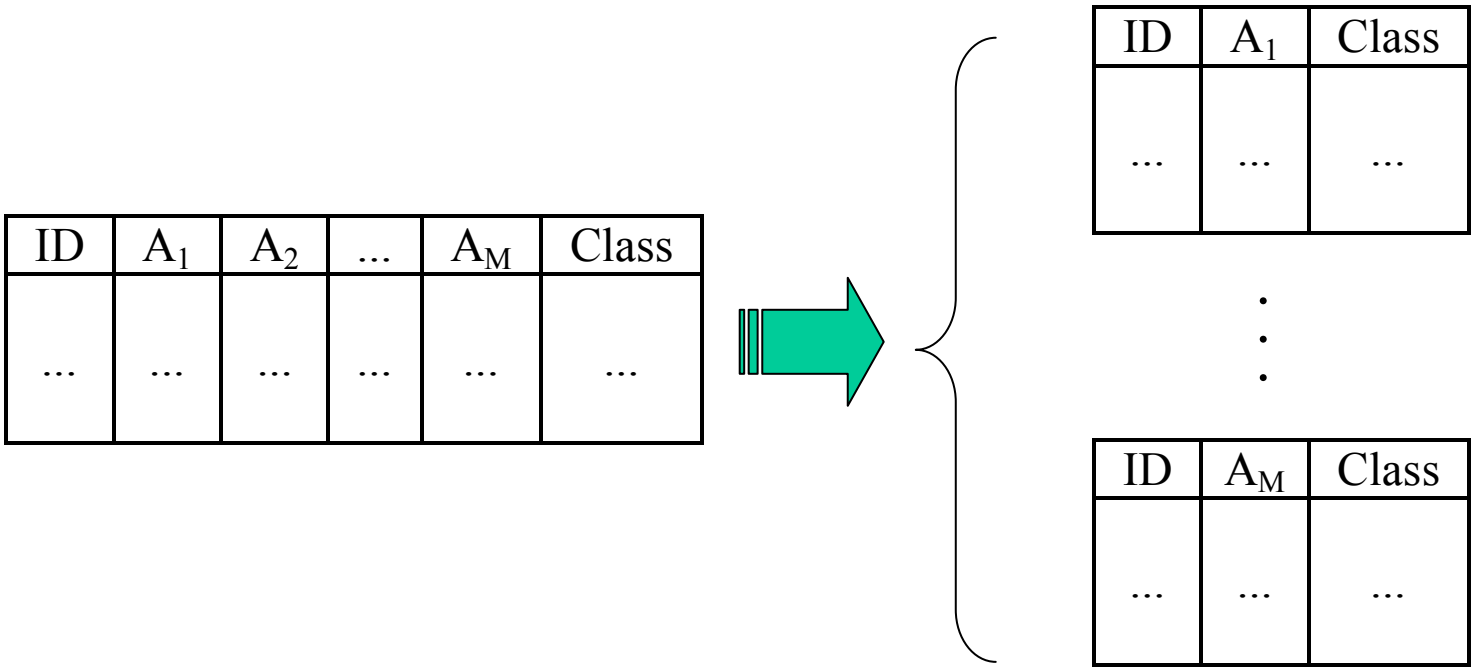
Attr-Val	$Class_1$	$Class_2$	...	$Class_K$
$A_i=a_{i1}$	$Count_{i1,1}$	$Count_{i1,2}$	...	$Count_{i1,k}$
$A_i=a_{i2}$	$Count_{i2,1}$	$Count_{i2,2}$	...	$Count_{i2,k}$
...	...	...	...	...
$A_i=a_{iri}$	$Count_{iri,1}$	$Count_{iri,2}$	...	$Count_{iri,k}$

- Suelen caber en memoria, se utilizan para decidir en cada split cuál es la mejor rama (sólo sirven para condiciones vble=cte).
- El problema es que para crearlas se deben utilizar consultas SQL costosas.

Métodos Específicos de Minería de Datos.

Algoritmos de Aprendizaje de Árboles de Decisión Escalables:

- También se puede hacer una partición vertical de la tabla:



- Escanear cada tabla es mucho menos costoso y para evaluar algunos splits sólo es necesario evaluar una o dos subtablas.

Métodos Específicos de Minería de Datos.

Reglas de Asociación y Dependencia:

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

Asociaciones:

Se buscan asociaciones de la siguiente forma:

$$(X_1 = a) \leftrightarrow (X_4 = b)$$

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

Un parámetro T_c (confidence):

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

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

Métodos Específicos de Minería de Datos.

Reglas de Asociación y Dependencia de Valor:

Dependencias de Valor:

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

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

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

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

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

T_s = *mínimo nº de casos o porcentaje en los que se aplica satisfactoriamente* (r_c *o* r_c/n *respectivamente).*

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

Métodos Específicos de Minería de Datos.

Reglas de Asociación y Dependencia de Valor. Ejemplo:

DNI	Renta Familiar	Ciudad	Profesión	Edad	Hijos	Obeso	Casado
11251545	5.000.000	Barcelona	Ejecutivo	45	3	S	S
30512526	1.000.000	Melilla	Abogado	25	0	S	N
22451616	3.000.000	León	Ejecutivo	35	2	S	S
25152516	2.000.000	Valencia	Camarero	30	0	S	S
23525251	1.500.000	Benidorm	Animador Parque Temático	30	0	N	N

Asociaciones:

Casado e (Hijos > 0) están asociados (80%, 4 casos).

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

Dependencias:

(Hijos > 0) → Casado (100%, 2 casos).

Casado → Obeso (100%, 3 casos)

Métodos Específicos de Minería de Datos.

Algoritmos de búsqueda de asociaciones y dependencias.
La mayoría se basa en descomponer el problema en dos fases:

- FASE A: BÚSQUEDA DE "LARGE ITEMSETS". Se buscan conjuntos de atributos con 'support' >= al support deseado, llamados 'large itemsets' (conjuntos de atributos grandes). De momento no se busca separarlos en parte izquierda y parte derecha.

- FASE B: ESCLARECIMIENTO DE DEPENDENCIAS (REGLAS). Se hacen particiones binarias y disjuntas de los itemsets y se calcula la confianza de cada uno. Se retienen aquellas reglas que tienen confianza >= a la confianza deseada.

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

Métodos Específicos de Minería de Datos.

Algoritmos de búsqueda de asociaciones.

FASE A:

Método genérico de búsqueda de "LARGE ITEMSETS"

Dado un support mínimo s_{min}:

1. i=1 (tamaño de los conjuntos)
2. Generar un conjunto unitario para cada atributo en S_i.
3. Comprobar el support de todos los conjuntos en S_i. Eliminar aquellos cuyo support < 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 $S_2 \cup S_3 \cup ... \cup S_i$

Hay refinamientos (Agrawal et al. 1996) (Cheung et al. 1996) (Lin and Dunham 1998) (Savarese et al 1995) que permiten una mejor paralelización (dividen en subproblemas con menos tuplas y luego comprueban para todo el problema). Para una comparativa ver (Hipp et al. 1999).

Métodos Específicos de Minería de Datos.

Algoritmo *Apriori*:

- El funcionamiento de este algoritmo se basa en la búsqueda de los conjuntos de ítems con determinada cobertura. Esta búsqueda se realiza de manera incremental.

- La siguiente fase consiste en la creación de reglas a partir de los conjuntos de ítems frecuentes.

- Si sólo buscamos reglas de asociación con un ítem en la parte derecha: de un conjunto de ítems de tamaño *i*, se crean *i* reglas colocando siempre un único ítem diferente en la parte derecha.

Métodos Específicos de Minería de Datos.

Ejemplo:

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

Métodos Específicos de Minería de Datos.

Ejemplo:

Si definimos la cobertura mínima igual a dos:

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

Métodos Específicos de Minería de Datos.

Ejemplo:

La siguiente fase consiste en la creación de reglas a partir de los conjuntos de ítems frecuentes

Por ejemplo, si tenemos el conjunto de items *horchata "Xufer"* Y *bizcochos "Goloso"* Y *galletas "Trigo"* se construyen las reglas:

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

Métodos Específicos de Minería de Datos.

Extensiones de las reglas de Asociación:

- **Valores utilizados en las reglas:** Sistemas que trabajan con atributos con más de dos valores (*país=Alemania*)
- **Dimensiones de los datos:** Podemos incrementar las dimensiones de una regla incluyendo por ejemplo, la dimensión cliente. **SI** *Comprar(vino "El cabezón"), Cliente(Juan),* **ENTONCES** " *Comprar(agua "Bendita")*.
- **Niveles de abstracción**: algunos sistemas permiten incorporar a las reglas diferentes niveles de abstracción que aglutinan ítems. **SI** *Comprar(vino)* **ENTONCES** *Comprar(gaseosa)*.
- **Reglas secuenciales**: consideran relaciones en una secuencia o serie temporal (varias compras o visitas a una página web).

Métodos Específicos de Minería de Datos.

Otros tipos de asociaciones:

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

Métodos Específicos de Minería de Datos.

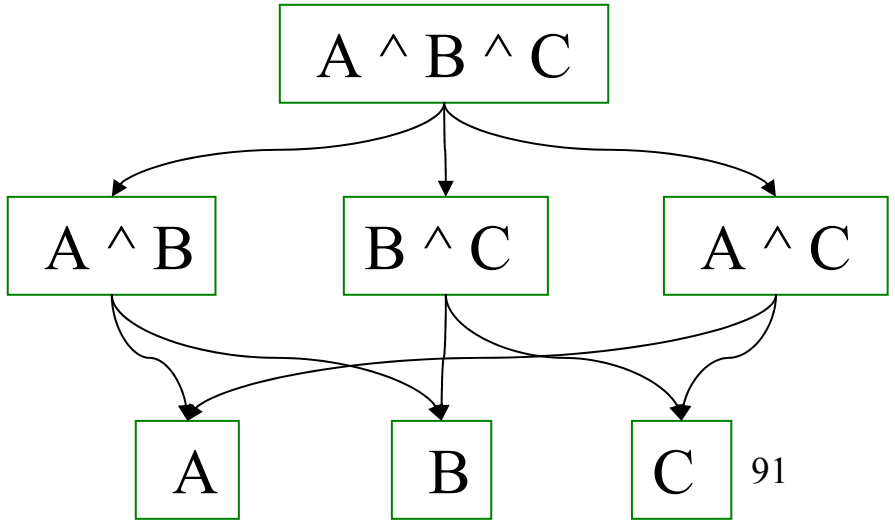
Dependencias Funcionales:

$$A \wedge B \wedge C \rightarrow D$$

Significa: para los mismos valores de A, B y C tenemos un solo valor de D. Es decir D es función de A, B y C.

Si representamos la parte izquierda como un conjunto de condiciones, podemos establecer una relación de orden entre las dependencias funcionales.

Esto genera un semi-retículo.



La búsqueda se realiza en este retículo.
(Mannila & Räihä 1994)
coste exponencial
FDEP (Flach & Savnik 1999) incluye:
a) simple top-down algorithm,
b) bottom-up algorithm, and
c) bi-directional algorithm.

Métodos Específicos de Minería de Datos.

Correlaciones y Estudios Factoriales:

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

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

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

Métodos Específicos de Minería de Datos.

Correlaciones y Estudios Factoriales. Ejemplo (cont.):

Matriz de correlaciones:

	Health	Need	Transp'tion	Child Care	Sick Time	Satisfaction	Ease	No-Show
Health	1							
Need	-0.7378	1						
Transportation	0.3116	-01041	1					
Child Care	0.3116	-01041	1	1				
Sick Time	0.2771	0.0602	0.6228	0.6228	1			
Satisfaction	0.22008	-0.1337	0.6538	0.6538	0.6257	1		
Ease	0.3887	-0.0334	0.6504	0.6504	0.6588	0.8964	1	
No-Show	0.3955	-0.5416	-0.5031	-0.5031	-0.7249	-0.3988	-0.3278	1

Coeficientes de Regresión:

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

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

Estimadores de Probabilidad

• La mayoría de los modelos, dado un nuevo caso, estiman la probabilidad de pertenencia a cada clase, asignándole la clase con mayor probabilidad (clasificadores suaves).

• Sin embargo, algunos ámbitos requieren que esta asignación venga acompañada con alguna información sobre la fiabilidad de la clasificación.

•Los clasificadores suaves son también útiles en las aplicaciones donde nos interesa ordenar ejemplos: Mailings, predicciones de apuestas…

Estimadores de Probabilidad

•Un árbol de decisión, lo podemos convertir en un clasificador suave o PET (*Probabilistic Estimator Tree*) si utilizamos la distribución de los ejemplos en la hojas

•Si una hoja tiene las siguientes frecuencias absolutas n_1, n_2, ..., n_c (obtenidas a partir del conjunto de entrenamiento) las probabilidades estimadas pueden calcularse como: $p_i=n_i/\sum n$

•Podemos mejorar las estimaciones aplicando correcciones a la estimación de probabilidad: Laplace, M-estimate..

$$p_i = \frac{n_i + 1}{\left(\sum_{i \in C} n_i\right) + c}$$

Estimadores de Probabilidad

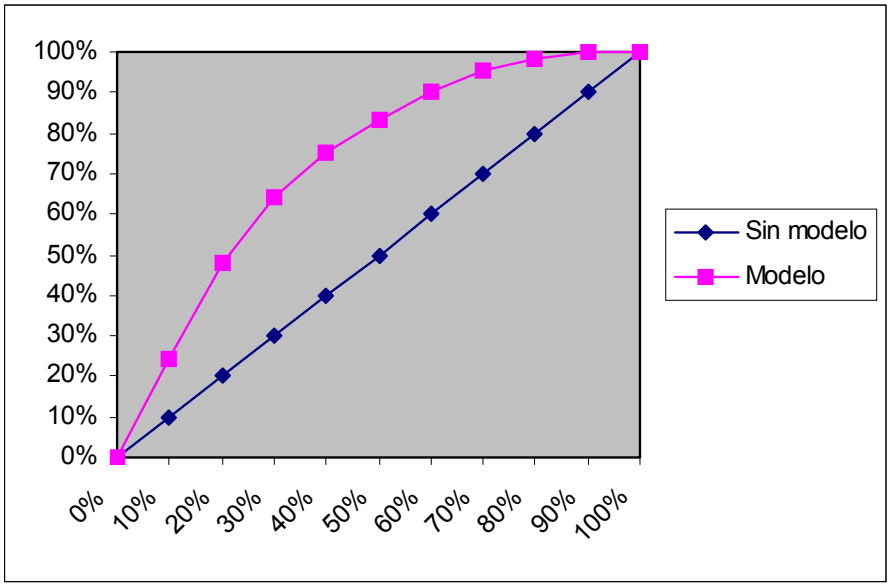
- Mailings:
  - Existen técnicas específicas para evaluar la conveniencia de campañas de 'mailings' (propaganda por correo selectiva):
    - EJEMPLO: Supongamos que una empresa de venta de productos informáticos por catálogo posee una base de datos de clientes. Esta empresa desea promocionar la venta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

Estimadores de Probabilidad

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

Estimadores de Probabilidad

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

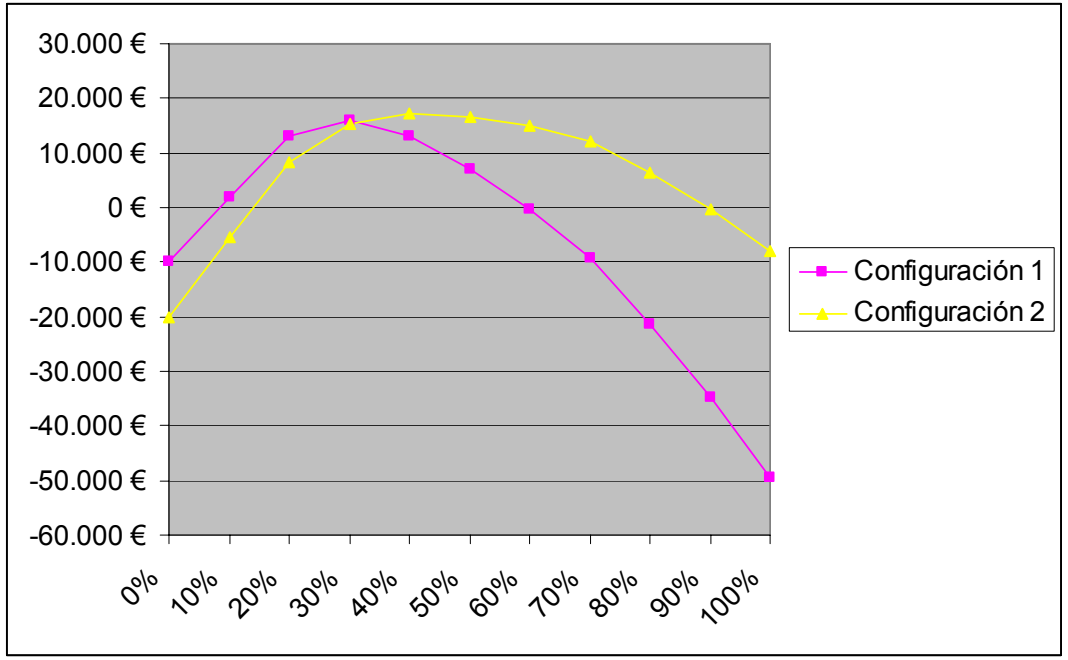


  - Nos indican qué porcentaje de las posibles respuestas vamos a obtener dependiendo del porcentaje de envíos que realicemos sobre la población total

Estimadores de Probabilidad

Además si estimamos la matriz de coste, podemos conocer la configuración optima mediante los gráficos de beneficio

- Configuración 1: Coste inicial de la campaña 10.000€, coste de envío de cada folleto 1,5€. Por cada producto vendido ganamos 3€
- Configuración 2: Coste inicial de la campaña 20.000€, coste de envío de cada folleto 0,8€. Por cada producto vendido ganamos 2,5€



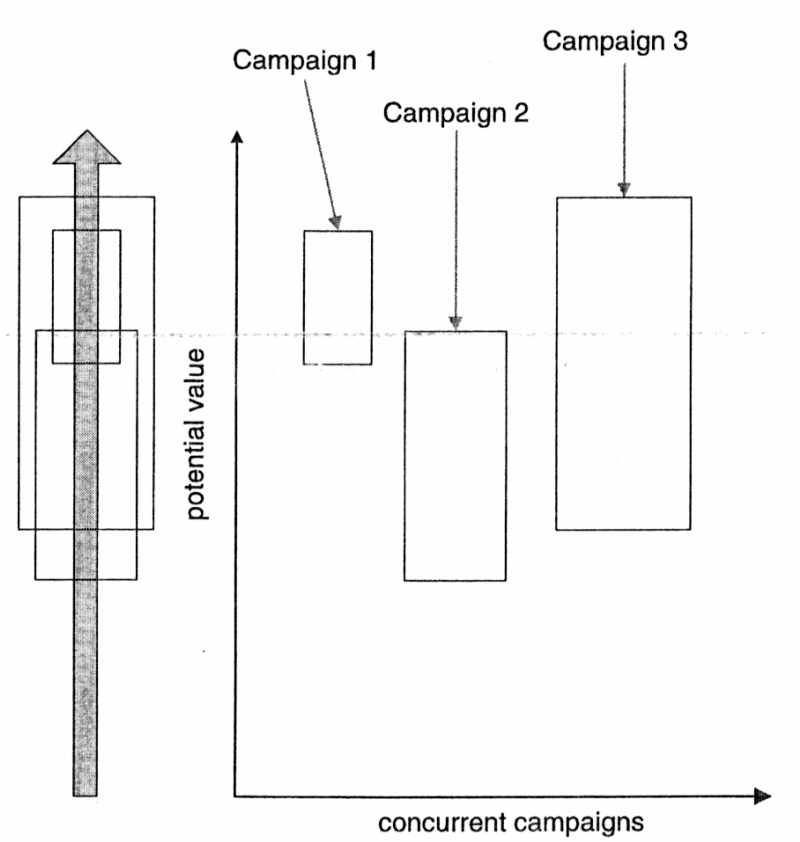
Estimadores de Probabilidad

Secuenciación de Mailings:

No sobrecargar los clientes con demasiados mensajes de márketing...

O bien acabarán ignorándolos o bien se cambiarán de compañía.

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

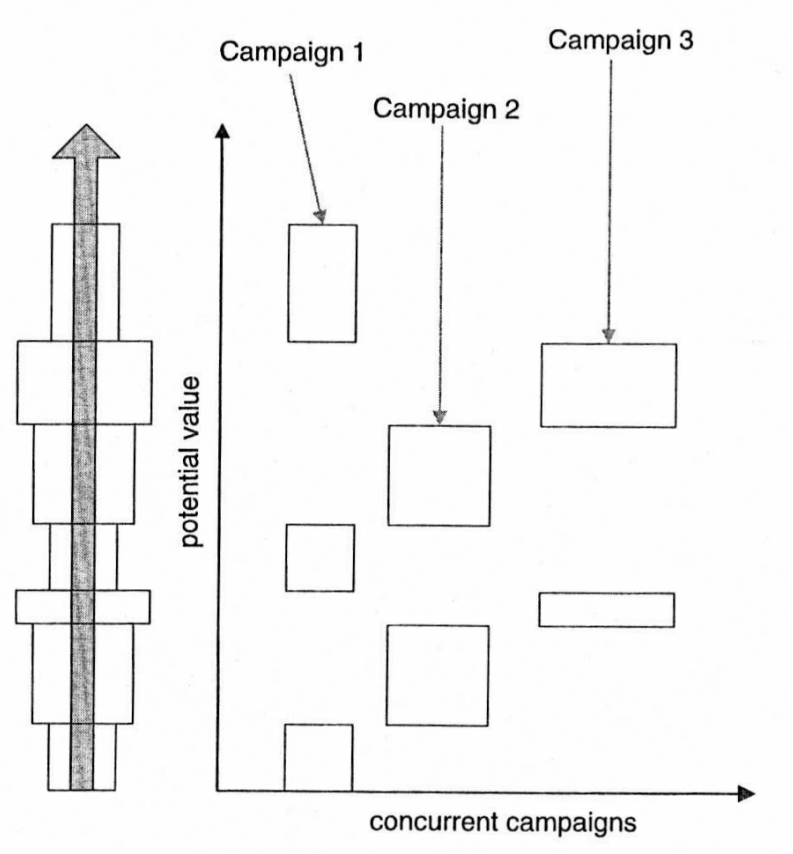


Estimadores de Probabilidad

Secuenciación de Mailings:

- Hay que intentar abarcar mejor los clientes:

Ahora todos los clientes participan en una campaña.



Aprendizaje Sensible al Coste

• **Contexto**: Una manera sencilla de definir un contexto es mediante dos aspectos:

- La distribución del valor de salida:
  - o Clasificación: distribución de las clases.
  - o Regresión: distribución de la salida.

- El coste de cada error:
  - o Clasificación: matriz de costes.
  - o Regresión: función de coste.

Aprendizaje Sensible al Coste

•Clasificación: matriz de costes.

•Ejemplo: Dejar cerrada una válvula en una central nuclear cuando es necesario abrirla, puede provocar una explosión, mientras que abrir una válvula cuando puede mantenerse cerrada, puede provocar una parada.

•Matriz de costes:

Predicho \ Real	abrir	cerrar
Abrir	0	100€
cerrar	2000€	0

•Lo importante no es obtener un "clasificador" que yerre lo menos posible sino que tenga un coste menor.

•A partir de la matriz se calcula el coste de un clasificador.

•Los clasificadores se evalúan con dichos costes.

•Se selecciona el clasificador de menos coste.

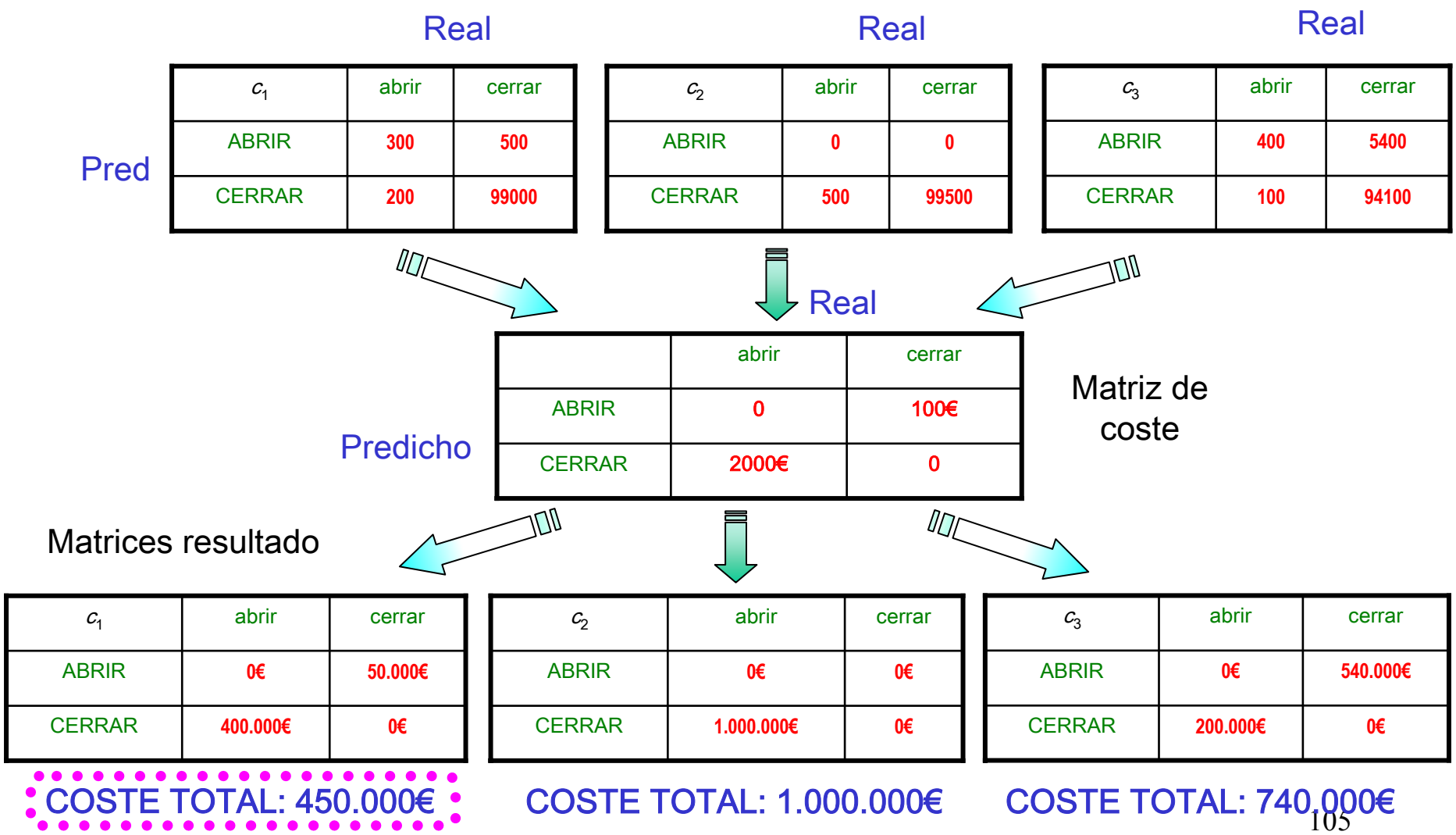
Aprendizaje Sensible al Coste

- Regresión: función de costes.
  - Ejemplo: un modelo de predicción de stocks debe penalizar más un error por exceso (al predecir mucho stock nos hemos quedado sin algún producto) que por defecto (el almacén estará más lleno pero se servirá el producto).
  - El modelo que esté "centrado" no será un buen modelo.
- Función de coste:

$$Coste = 1 - e^{\alpha \cdot (\hat{y} - y)}$$

  - Con un $\alpha = 0{,}01$:
    - Si el error es -200 el Coste= 0,86
    - Si el error es +200 el Coste= 6,3
  - De modo similar, se elige el modelo que minimice la función de coste.

Aprendizaje Sensible al Coste



Aprendizaje Sensible al Coste

- ¿De qué depende el coste final?
  - Para dos clases. Depende de un **contexto**:
    - El **coste** de los falsos positivos y falsos negativos: FPcost y FNcost
    - **Distribución de clases**: % de ejemplos de la clase negativa respecto de los de la clase positiva. (*Neg / Pos*).
    - Se calcula: (para el ejemplo anterior)

$$\frac{FPcost}{FNcost} = \frac{100}{2000} = \frac{1}{20} \qquad \frac{Neg}{Pos} = \frac{99500}{500} = 199 \qquad slope = \frac{1}{20} \cdot 199 = 9{,}95$$

- Para dos clases, el valor "slope" es suficiente para determinar qué clasificador será mejor.

Clasifi. 1: FNR= 40%, FPR= 0,5%
Coste Unitario =
1 x 0,40 + 9,95 x 0,005 = 0,45

Clasifi. 2: FNR= 100%, FPR= 0%
Coste Unitario =
1 x 1 + 9,95 x 0 = 1

Clasifi. 3: FNR= 20%, FPR= 5,4%
Coste Unitario =
1 x 0,20 + 9,95 x 0,054 = 0,74

Aprendizaje Sensible al Coste

- Adaptación de métodos para contextos con coste
  - ❑ Muchos métodos devuelven la probabilidad de pertenencia a la clase para cada ejemplo.
  - ❑ En estos casos en vez de asignar la clase con mayor probabilidad, se asigna la clase que minimice el coste.
  - ❑Ejemplo: un árbol de decisión retorna {0.4,0.6} a una instancia *t* con la siguiente matriz de coste:

		Real	
	c_1	+	-
Predicho	+	-20	200
	-	500	-10

  - ❑Teóricamente deberíamos asignar la clase - a *t*, sin embargo, dada la matriz de costes, es más sensato asignar +, ya que

Coste(+)=0.6*200+0.4*(-20)=112

Coste(-)=0.4*500+0.6*(-10)=194

Aprendizaje Sensible al Coste

- Adaptación de métodos para contextos con coste

  - Otra opción es modificar la distribución de los ejemplos de acuerdo a la matriz de costes (*Stratification*):
    - *Undersampling*: Eliminar instancias de las clases a reducir
    - *Oversampling*: Duplicar instancias de las clases a significar

  - Una solución más elegante es modificar los pesos los ejemplos de cada clase de acuerdo a la matriz de coste, siempre que los métodos lo permitan

Análisis ROC

- Problema

  - En muchas aplicaciones, hasta el momento de aplicación, no se conoce la distribución de clases y/o es difícil estimar la matriz de costes. P.ej. un clasificador de spam.

  - Pero los modelos se aprenden antes generalmente.

  - SOLUCIÓN:

Análisis ROC

(*Receiver Operating Characteristic*)

Análisis ROC

- El espacio ROC
  - Se normaliza la matriz de confusión por columnas: TPR, FNR TNR, FPR.

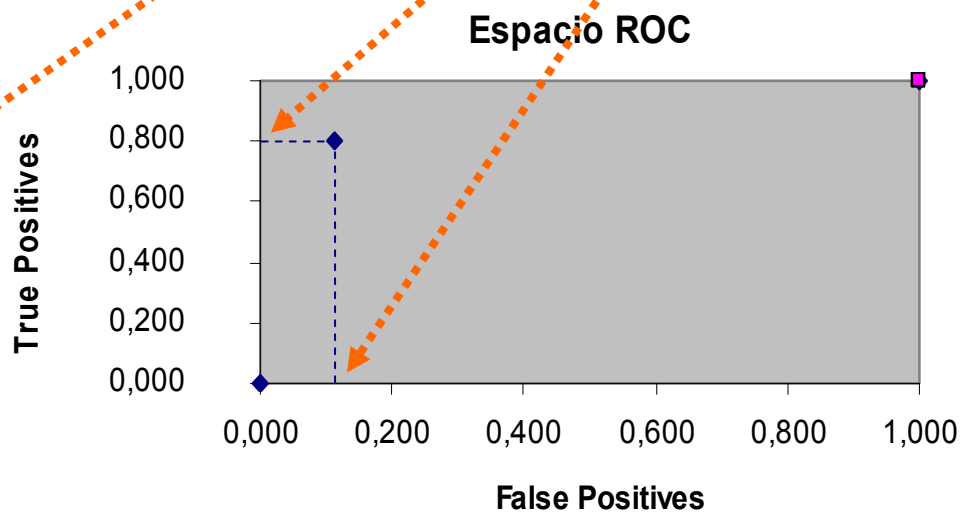
Real

Pred	abrir	cerrar
ABRIR	400	12000
CERRAR	100	87500

Real

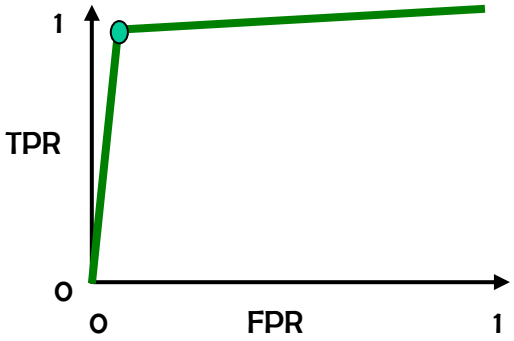
Pred	abrir	cerrar
ABRIR	0,8	0,121
CERRAR	0,2	0,879

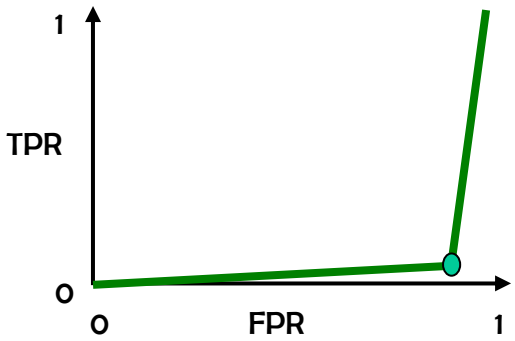
TPR= 400 / 500 = 80%
FNR= 100 / 500 = 20%
TNR= 87500 / 99500 = 87,9%
FPR= 12000 / 99500 = 12,1%

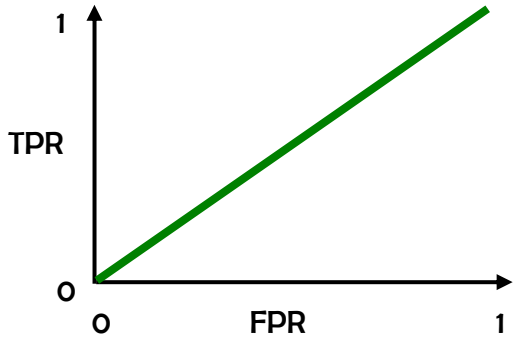


Análisis ROC

- Espacio ROC: buenos y malos clasificadores.



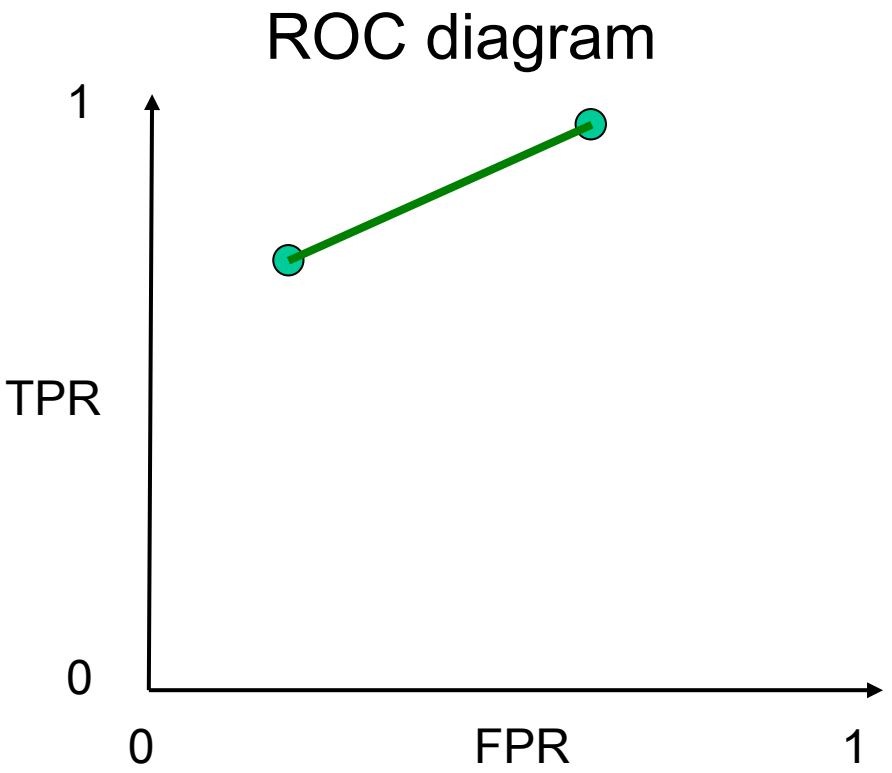




- Buen clasificador.
  - Alto TPR.
  - Bajo FPR.
- Mal clasificador.
  - Bajo TPR.
  - Alto FPR.
- Mal clasificador (en realidad).

Análisis ROC

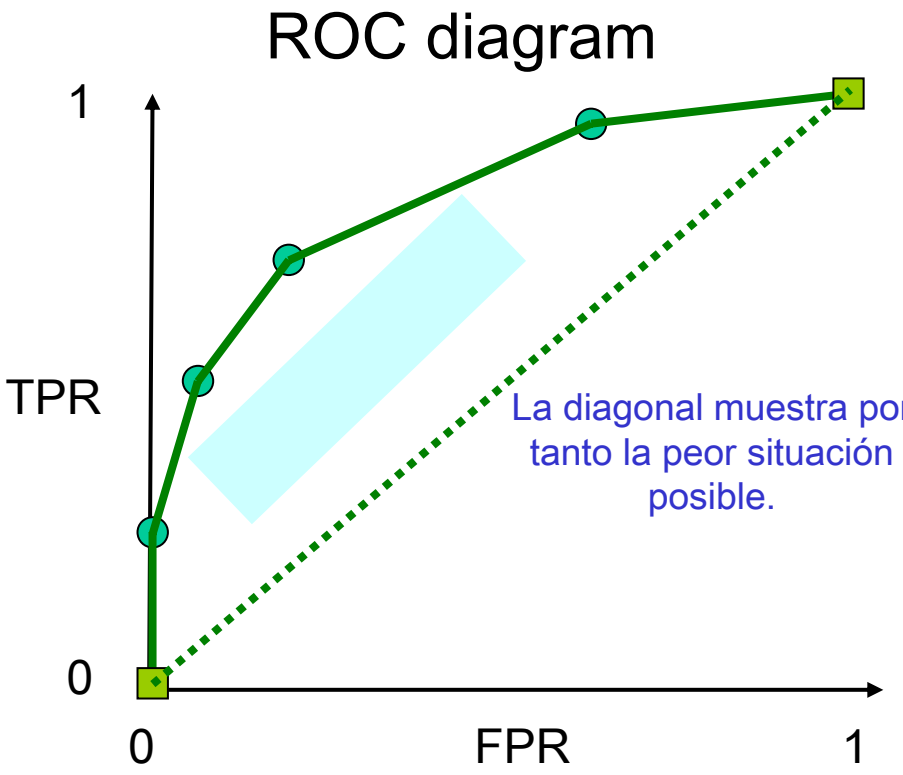
- La Curva ROC. “Continuidad”.



- Podemos construir cualquier clasificador “intermedio” ponderando aleatoramiente los dos clasificadores (con más peso a uno u otro).
- Esto en realidad crea un “continuo” de clasificadores entre cualesquiera dos clasificadores.

Análisis ROC

- La Curva ROC. Construcción.

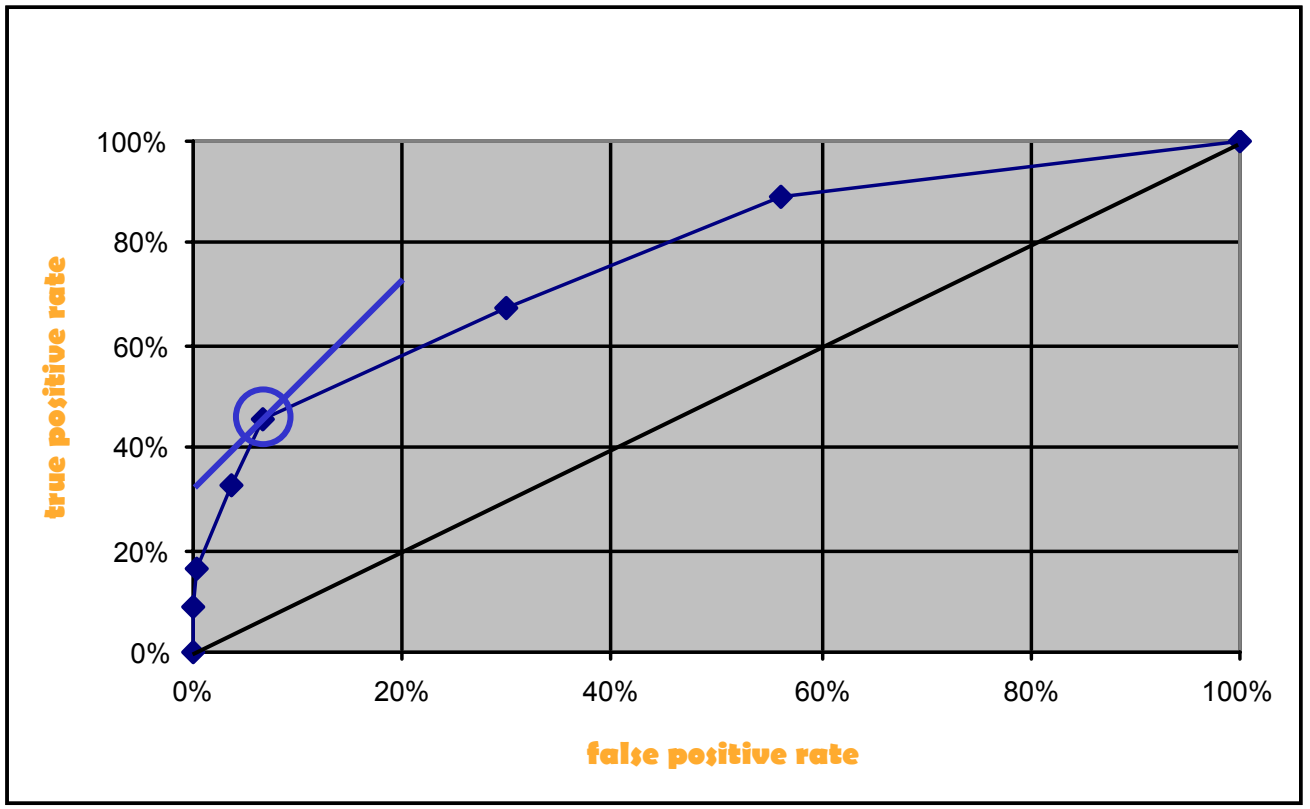


- Dados varios clasificadores:
  - Construimos el "casco convexo" (convex hull) de sus puntos (FPR,TPR) además de los dos clasificadores triviales (0,0) y (1,1).
  - Los clasificadores que caen debajo de la curva ROC se descartan.
  - El mejor clasificador de los que quedan se seleccionará en el momento de aplicación…

Podemos descartar los que están por debajo porque no hay ninguna combinación de distribución de clases / matriz de costes para la cual puedan ser óptimos.

Análisis ROC

- En el **contexto de aplicación**, elegimos el clasificador óptimo entre los mantenidos. Ejemplo 1:



Contexto:

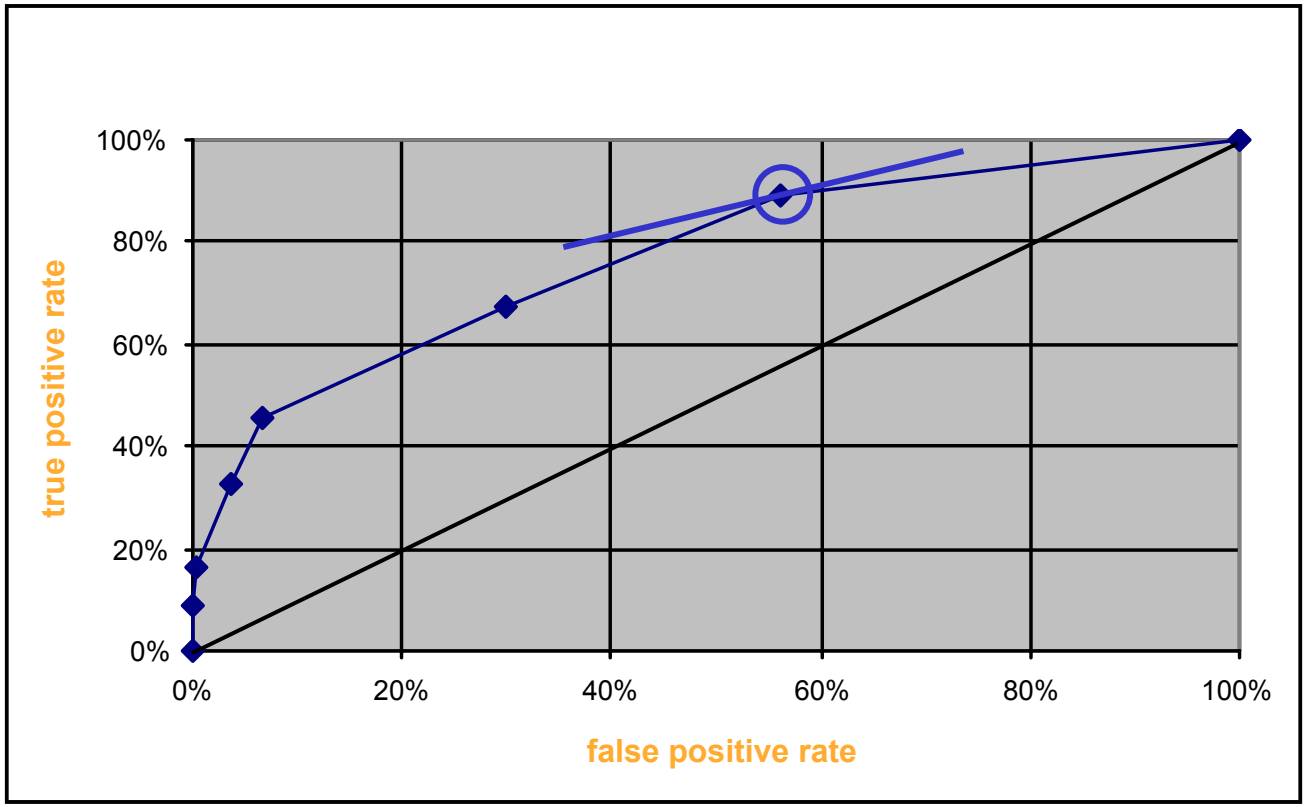
$$\frac{FPcost}{FNcost} = 1/2$$

$$\frac{Neg}{Pos} = 4$$

$$slope = 4/2 = 2$$

Análisis ROC

- En el **contexto de aplicación**, elegimos el clasificador óptimo entre los mantenidos. Ejemplo 2:



Contexto:

$$\frac{FPcost}{FNcost} = 1/8$$

$$\frac{Neg}{Pos} = 4$$

$$slope = 4/8 = .5$$

Análisis ROC

- ¿Qué hemos aprendido?
  - La optimalidad de un clasificador depende de la distribución de clases y de los costes de los errores.
  - A partir de este **contexto** se puede calcular una inclinación (o "skew") característica del contexto.
    - Si sabemos este contexto, podemos seleccionar el mejor clasificador, multiplicando la matriz de confusión por la matriz de coste.
    - Si desconocemos el contexto de aplicación en el momento de generación, usando el análisis ROC podemos elegir un subconjunto de clasificadores, entre los cuales seguro estará el clasificador óptimo para cualquier contexto posible, cuando éste se conozca.

¿Podemos ir más allá?

Análisis ROC

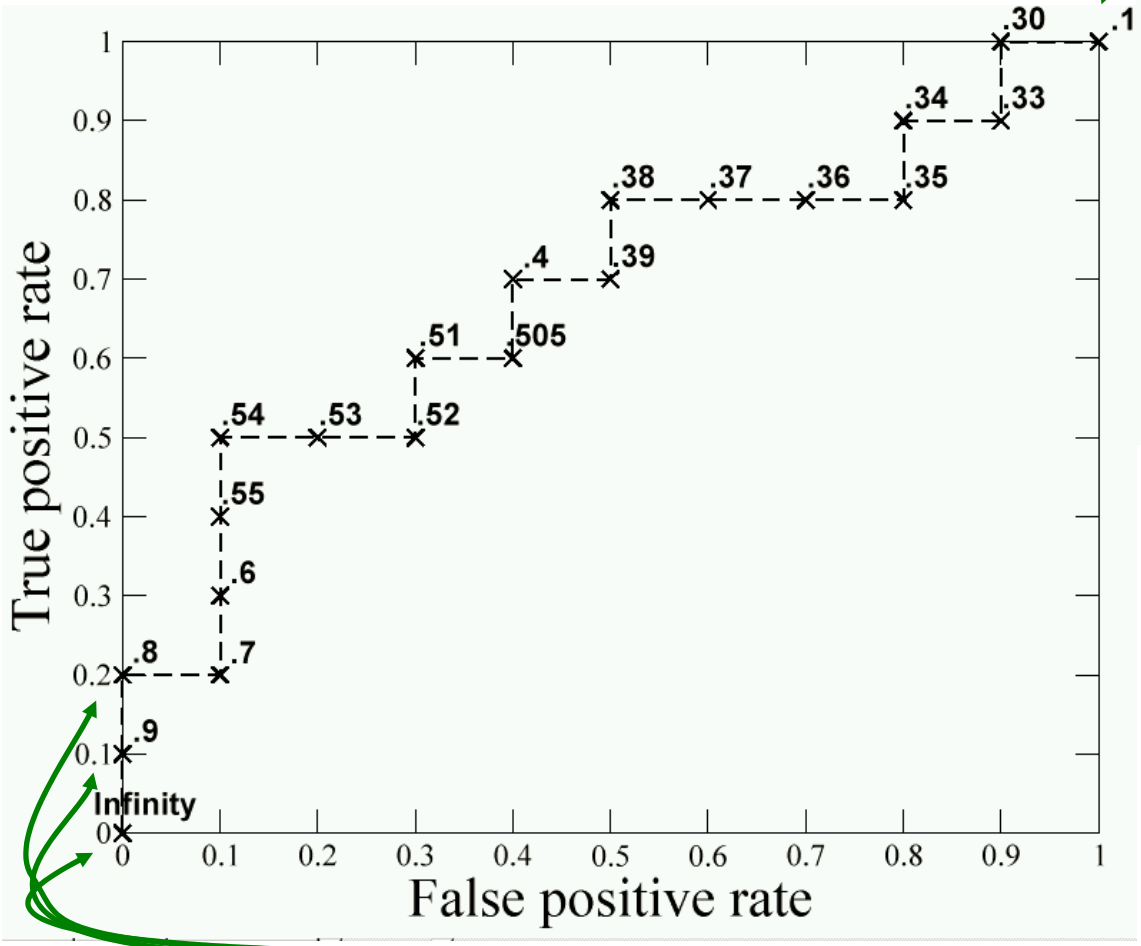
- Curva ROC de un Clasificador Probabilístico:
  - Un clasificador probabilístico (soft) se puede convertir en un clasificador discreto con un umbral.
    - Ejemplo: “si score > 0.7 entonces clase A, si no clase B”.
    - Con distintos umbrales, tenemos distintos clasificadores, que les dan más o menos importancia a cada una de las clases (sin necesidad de sobremuestreo o submuestreo).
  - Podemos considerar cada umbral como un clasificador diferente y dibujarlos en el espacio ROC. Esto genera una curva...

Tenemos una “curva” para un solo clasificador “soft”

- Esta curva es escalonada (no se suele realizar el “convex hull”).

Análisis ROC

- Curva ROC de un Clasificador “soft”:
  - Ejemplo:



Clase Real

Clase Predicha

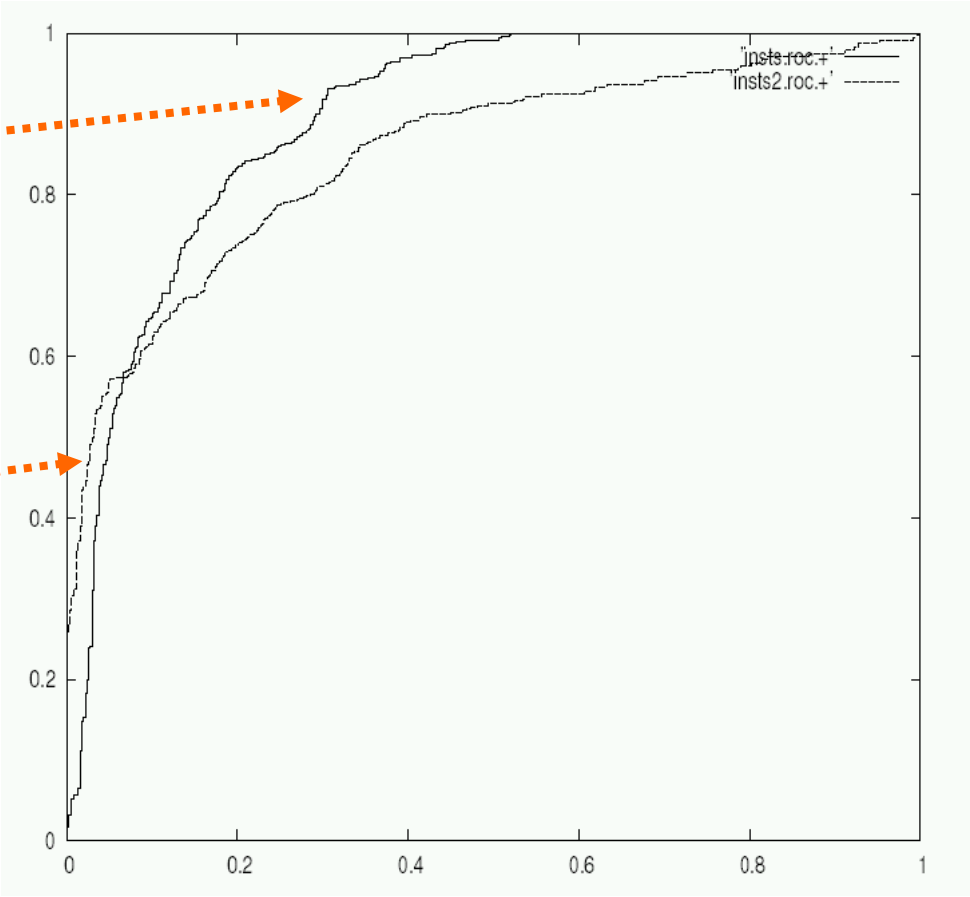
Inst#	Class	Score				
1	p	.9	n	p	p	p
2	p	.8	n	n	p	p
3	n	.7	n	n	n	p
4	p	.6	n	n	n	p
5	p	.55	n	n	n	p
6	p	.54	n	n	n	p
7	n	.53	n	n	n	p
8	n	.52	n	n	n	p
9	p	.51	n	n	n	p
10	n	.505	n	n	n	... p
11	p	.4	n	n	n	p
12	n	.39	n	n	n	p
13	p	.38	n	n	n	p
14	n	.37	n	n	n	p
15	n	.36	n	n	n	p
16	n	.35	n	n	n	p
17	p	.34	n	n	n	p
18	n	.33	n	n	n	p
19	p	.30	n	n	n	p
20	n	.1	n	n	n	p

Análisis ROC

- Análisis ROC de varios clasificadores “soft”:

En esta zona es mejor el clasificador “insts”

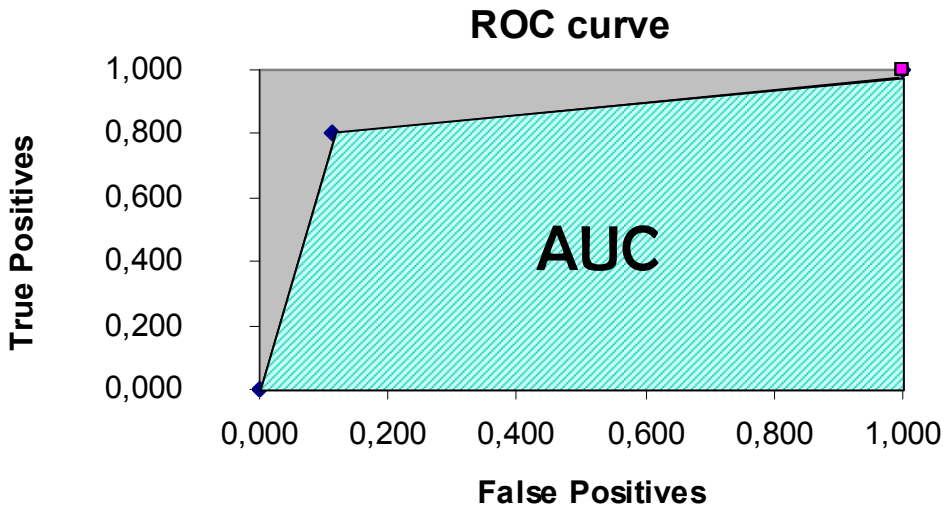
En esta zona es mejor el clasificador “insts2”



- Debemos mantener los clasificadores que tengan al menos una “zona mejor” y después actuar igual que en el caso de los clasificadores discretos.

Análisis ROC

- ¿Para seleccionar un solo clasificador discreto?
  - Se selecciona el que tiene mayor área bajo la curva ROC (AUC, *Area Under the ROC Curve*).

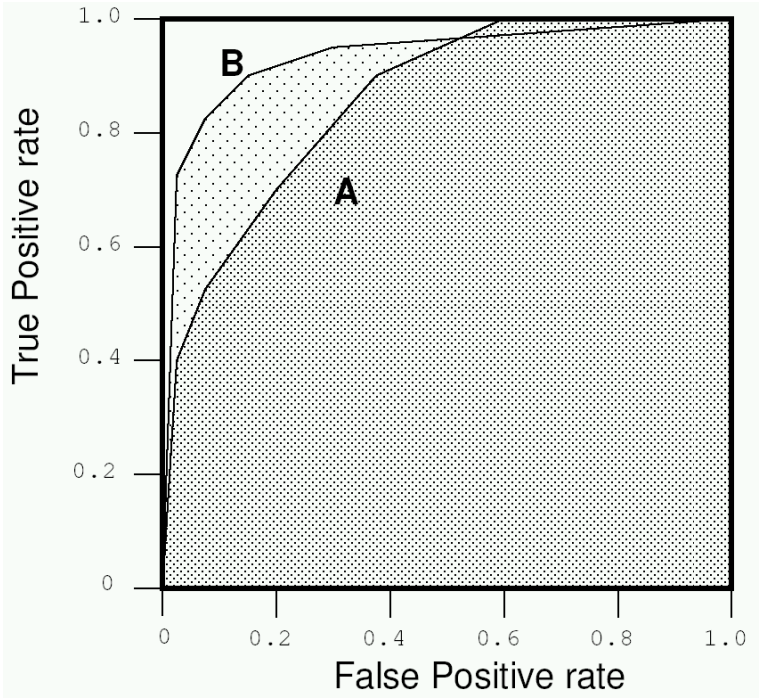


Alternativa al error para evaluar clasificadores

- Un método de aprendizaje / MD será mejor si genera clasificadores con alta AUC.

Análisis ROC

- ¿Para seleccionar un solo clasificador probabilístico?
  - Se selecciona el que tiene mayor área bajo la curva ROC (AUC, *Area Under the ROC Curve*).

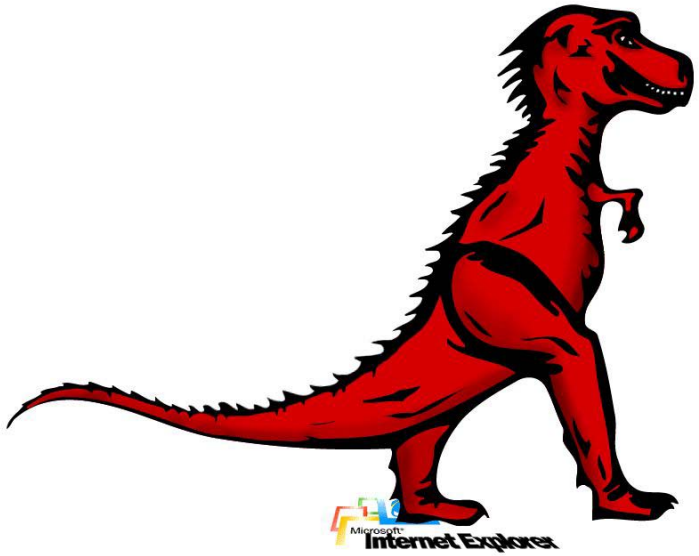


En este caso seleccionamos el B.

  - Evalúa cuán bien un clasificador realiza un ránking de sus predicciones.

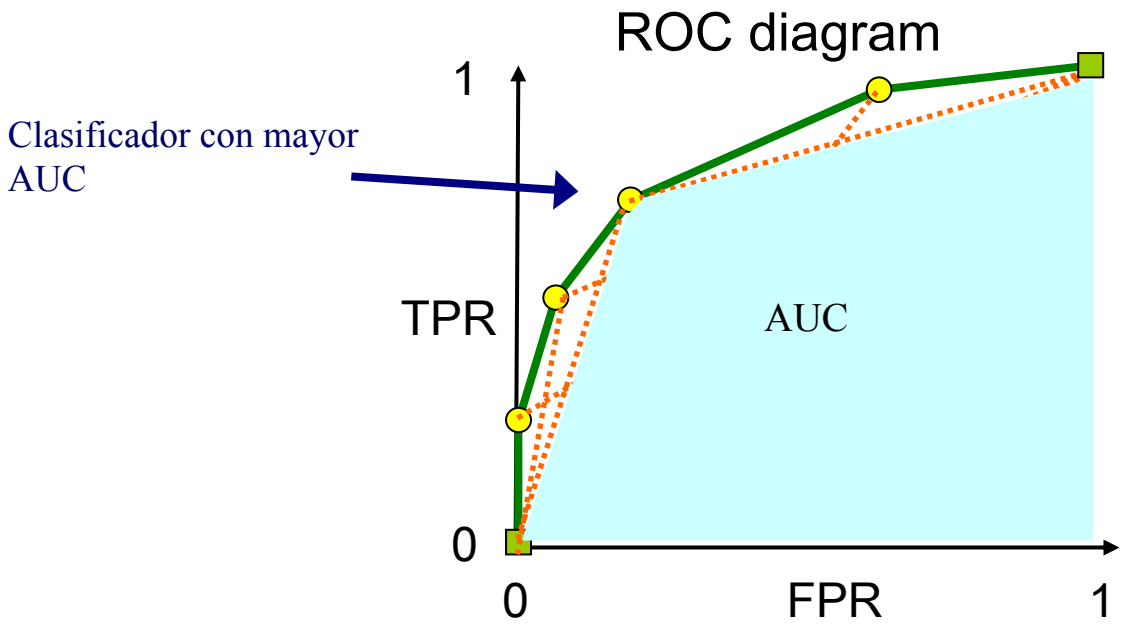
Análisis ROC

•Ejemplo: Detección Spam



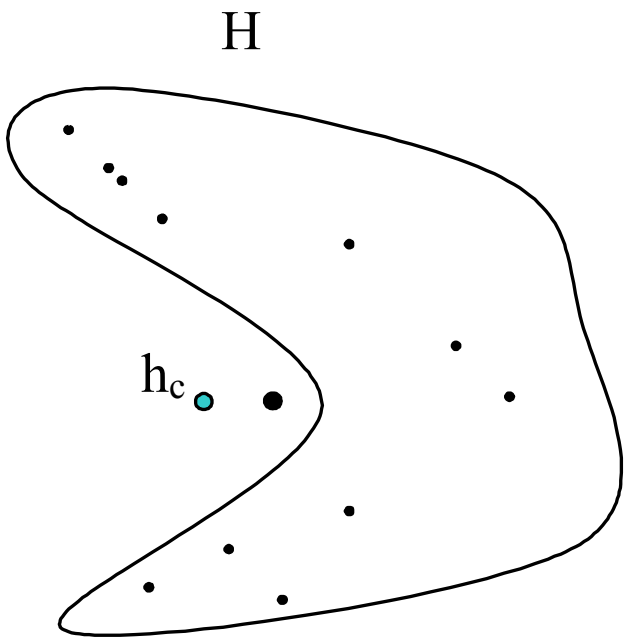
Análisis ROC

•Ejemplo: Detección Spam



Multi-clasificadores

•Una manera de mejorar las predicciones es combinar varios modelos



Multi-clasificadores

• Para obtener buenos resultados en la clasificación es necesario tener un conjunto de modelos (*ensemble*):

- Precisión alta
- Diferentes

•Dados 3 modelos {h1, h2, h3}, considere un nuevo dato x a ser clasificado:

- Si los tres clasificadores son similares, entonces cuando h1(x) sea erróneo, probablemente h2(x) y h3(x) también lo serán.

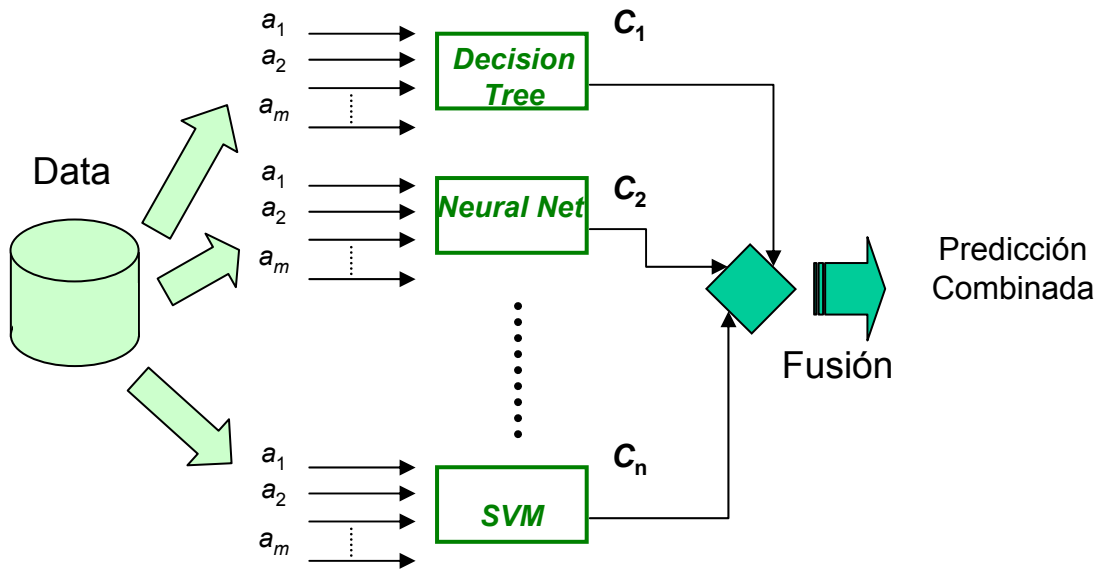
- Si los clasificadores son lo bastante diversos, cuando h1(x) sea erróneo, h2(x) y h3(x) podrían ser correctos, y entonces el conjunto combinado clasificaría correctamente el dato x.

Multi-clasificadores

- Métodos para generar *ensembles*:
  - Manipulación de los datos de entrenamiento
  - Manipulación de los atributos
  - Manipulación de las clases
  - Métodos aleatorios

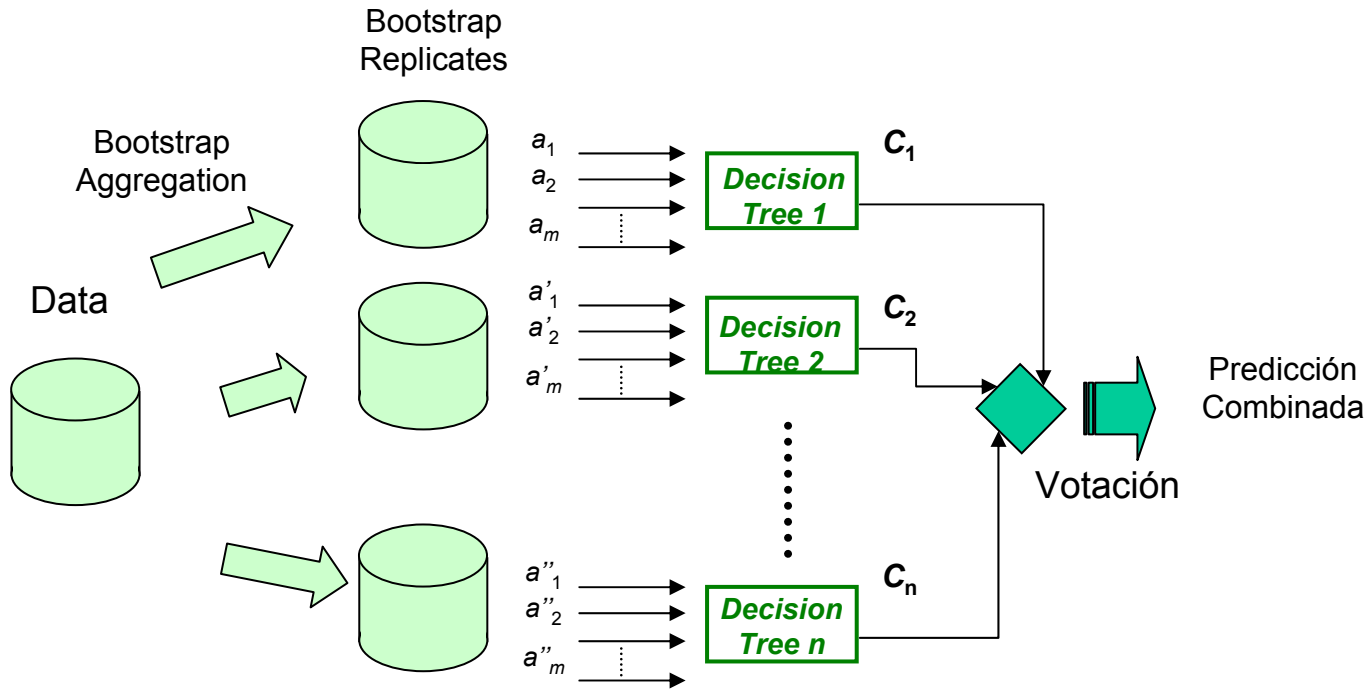
Multi-clasificadores

- Combinación simple (*voting*):



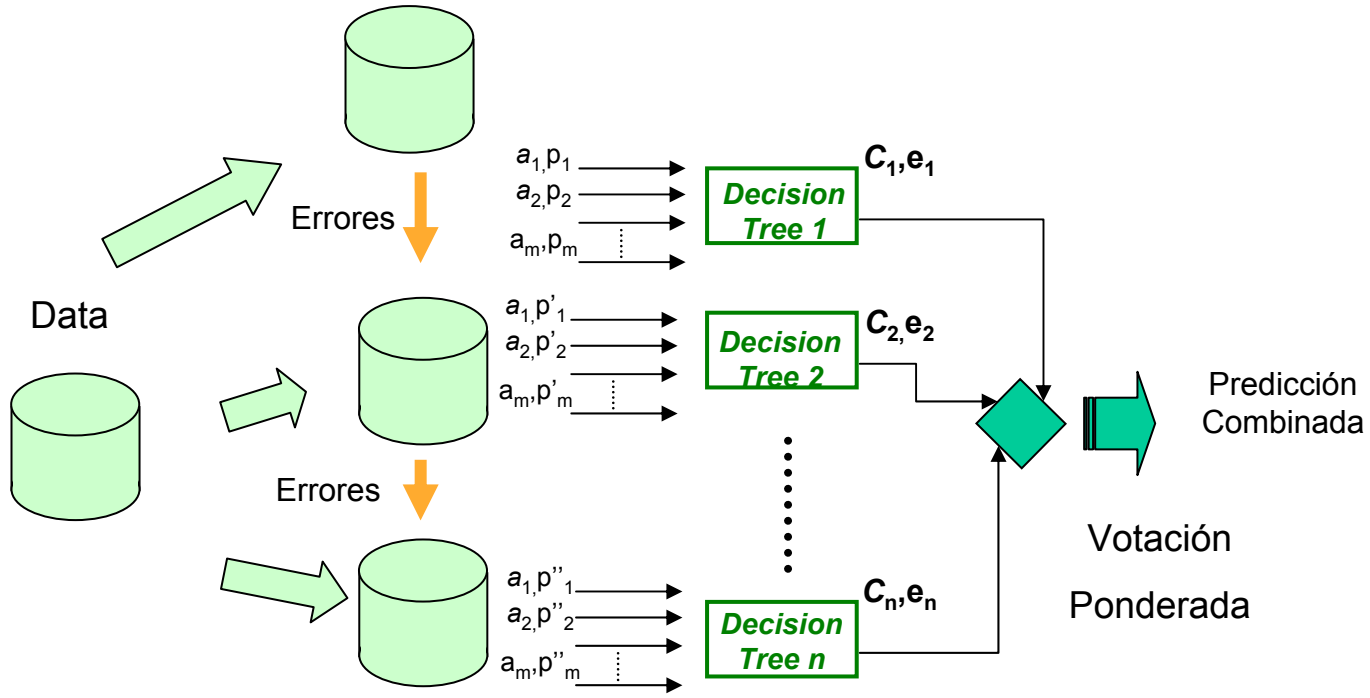
Multi-clasificadores

- Bagging (*Bootstrapt Aggregation*):



Multi-clasificadores

- Boosting



Multi-clasificadores

- Varios trabajos han comparado Boosting y Bagging
  - Boosting obtiene mejor precisión en general
  - En problemas con ruido Bagging es más robusto

Multi-clasificadores

- Stacking