

Visualización

Las técnicas de visualización de datos se utilizan fundamentalmente con dos objetivos:

- aprovechar la gran capacidad humana de extraer patrones a partir de imágenes.
- ayudar al usuario a comprender más rápidamente patrones descubiertos automáticamente por un sistema de KDD.

Visualización

Estos dos objetivos marcan dos momentos diferentes del uso de la visualización de los datos (no excluyentes):

- visualización *previa* (tb. Visual Data Mining [Wong 1999]): se utiliza para entender mejor los datos y sugerir posibles patrones o qué tipo de herramienta de KDD utilizar.
- visualización *posterior* al proceso de minería de datos: se utiliza para mostrar los patrones y entenderlos mejor.

Visualización

También marcan dos tipos de usuarios diferentes de las técnicas:

- La visualización *previa* se utiliza frecuentemente por picapedreros, para ver tendencias y resúmenes de los datos, y por exploradores, para ver ‘filones’ que investigar.
- La visualización *posterior* se utiliza frecuentemente para validar y mostrar a los expertos los resultados del KDD.

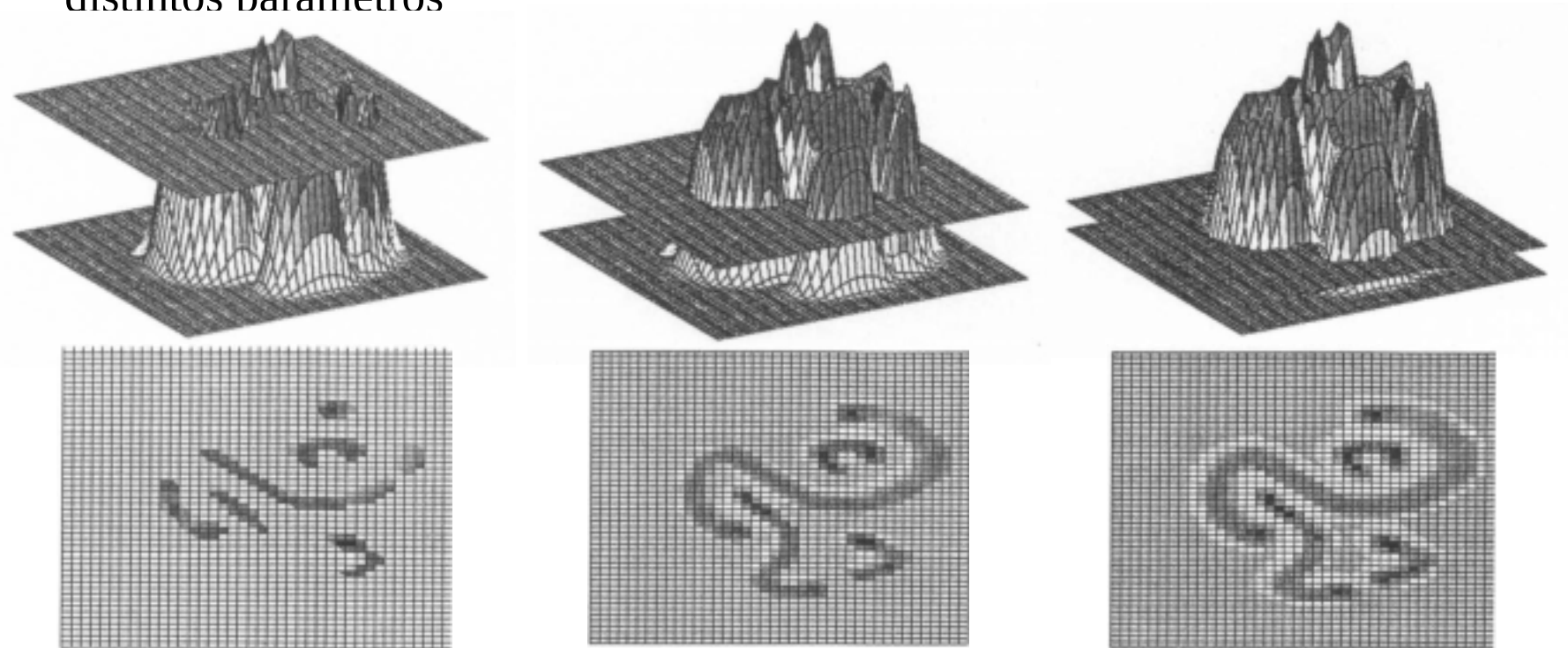
las herramientas gráficas requieren mayor experiencia para seleccionar qué gráfico nos interesa utilizar entre los cientos de gráficas que proporcionan los sistemas actuales.

Visualización

Visualización Previa:

Ejemplo: segmentación mediante funciones de densidad, generalmente representadas tridimensionalmente.

Los seres humanos ven claramente los segmentos (clusters) que aparecen con distintos parámetros



Visualización

Visualización Previa:

Mayor problema: dimensionalidad > 3 .

Objetivo: conseguir proyectar las dimensiones en una representación en 2 (ó 3 simuladas) dimensiones.

Uso de proyecciones geométricas:

- técnica de visualización de coordenadas paralelas [Inselberg & Dimsdale 1990]. Se mapea el espacio k -dimensional en dos dimensiones mediante el uso de k ejes de ordenadas (escalados linealmente) por uno de abscisas. Cada punto en el espacio k -dimensional se hace corresponder con una línea poligonal (polígono abierto), donde cada vértice de la línea poligonal intersecta los k ejes en el valor para la dimensión.
 - Cuando hay pocos datos cada línea se dibuja de un color.
 - Cuando hay muchos datos se utiliza una tercera dimensión para los casos.⁵

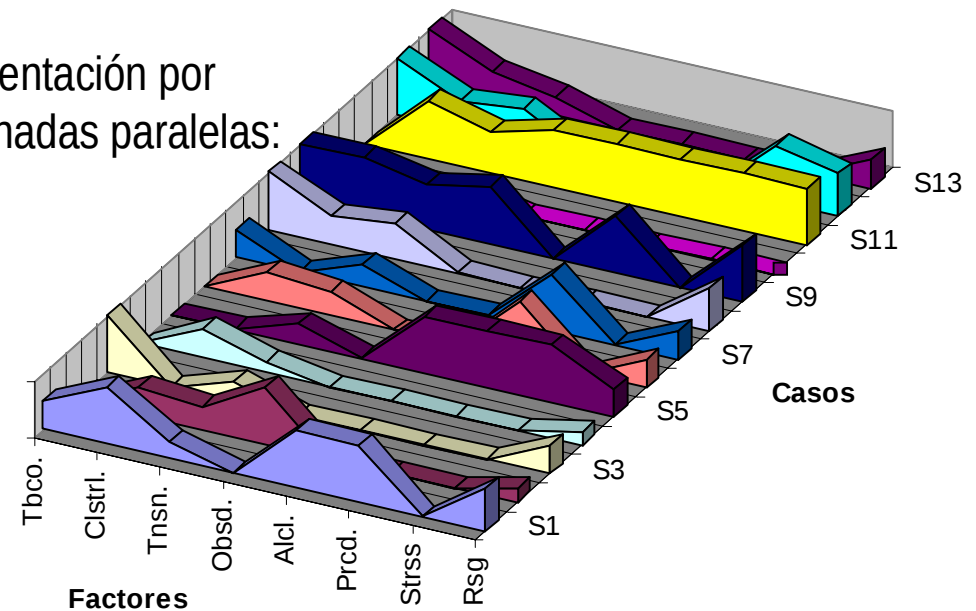
Visualización

Visualización Previa: Ejemplo:dimensionalidad...

Dados ciertos atributos de pacientes (tabaquismo, colesterol, tensión, obesidad, alcoholismo, precedentes, estrés) y su riesgo (muy bajo, bajo, medio, alto, muy alto) de enfermedades coronarias:

Tbco.	Ciترل	Tnsn	Obsd	Alcl	Prcl	Strs	Rsg
Med	Alto	8	No	Sí	Sí	No	Alto
Bajo	Med	9	Sí	No	No	No	Bajo
Alto	Bajo	8,5	No	No	No	No	Med
Bajo	Med	7	No	No	No	No	Bajo
Bajo	Bajo	8,5	No	Sí	Sí	Sí	Med
Bajo	Med	9	No	No	Sí	No	Med
Med	Bajo	9	No	No	Sí	No	Med
Alto	Med	11	No	No	No	No	Alto
Alto	Alto	13	Sí	No	Sí	No	M.A.
Bajo	Bajo	7	No	No	No	No	M.B.
Bajo	Alto	12	Sí	Sí	Sí	Sí	M.A.
Alto	Med	11	No	No	No	Sí	Alto
Alto	Med	8	No	No	No	No	Med

Representación por
coordenadas paralelas:



El mayor problema de estas representaciones (y de otras muchas) es que no acomodan bien las variables discretas.

Visualización

Visualización Previa:

- Icónicas: Existen otro tipo de técnicas que sí permiten combinar atributos continuos y discretos, mediante el uso de transformaciones menos estándar y el uso de iconos.
 - Se utilizan rasgos compatibles y diferenciados para distintas dimensiones, como son círculos, estrellas, puntos, etc., con la ventaja de que se pueden combinar más convenientemente valores discretos y continuos.
- Otras aproximaciones más sofisticadas se basan en estructuras jerárquicas, como por ejemplo, los Cone Trees [Robertson et al. 1991].

Visualización

Visualización Posterior:

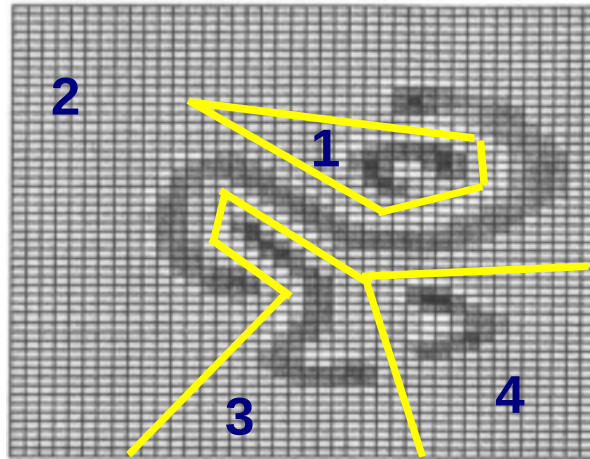
Se utiliza para mostrar los patrones y entenderlos mejor.

- Un árbol de decisión es un ejemplo de visualización posterior.
- Otros gráficos de visualización posterior de patrones:
 - muestran una determinada segmentación de los datos, una asociación, una determinada clasificación.
 - utilizan para ello gráficos de visualización *previa* en los que además se señala el patrón.
 - permiten evaluar gráficamente la calidad del modelo.

Visualización

Visualización Posterior:

EJEMPLO: se muestra una segmentación lineal para el corte del ejemplo anterior:



Visualización

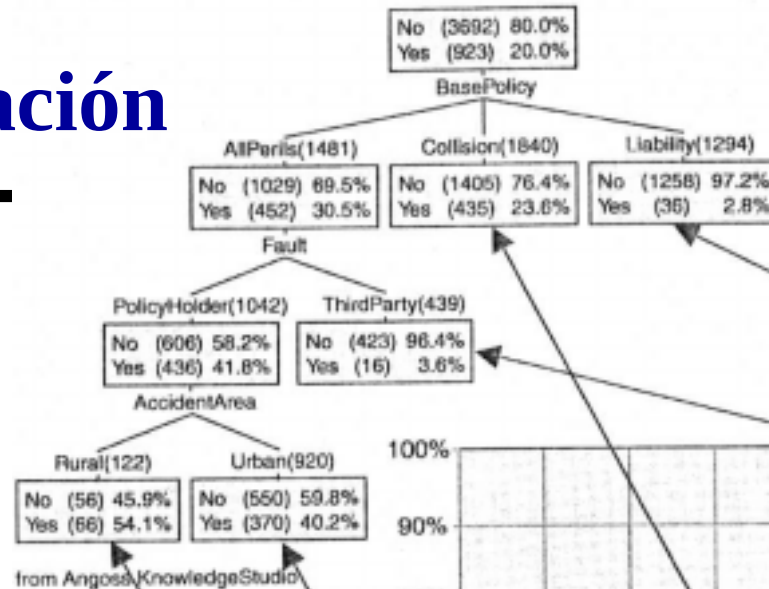
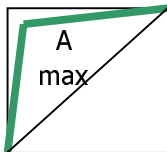
Visualización Posterior:

EJEMPLO:

representación de ganancias acumulativas de un árbol de decisión:

$$\text{lift}^\circ = \arcsen \text{No}/\text{Total}$$

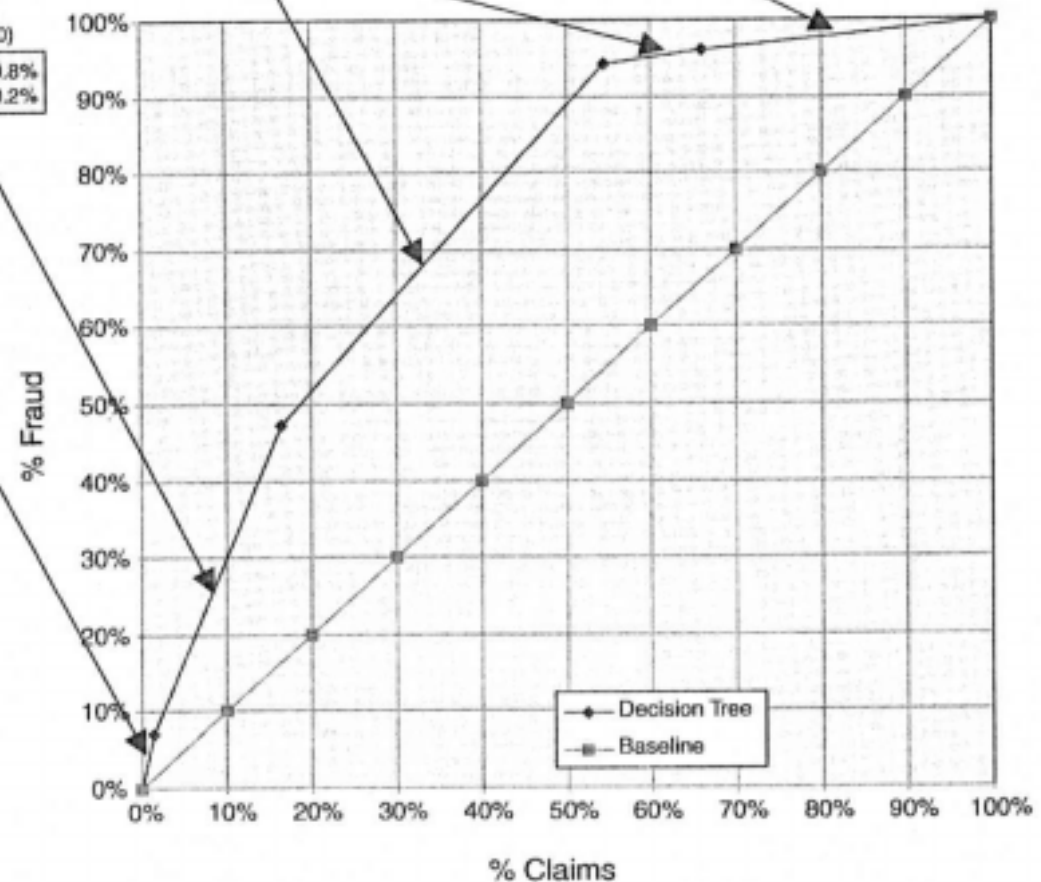
El árbol óptimo sería así:



Each segment corresponds to one of the leaves of the tree.

The slope of the line corresponds to the lift at that leaf. The length corresponds to the number of records that land there.

The steepest segments correspond to the leaves with the biggest lift (highest density of the desired outcome).



Áreas de Aplicación

Áreas de Aplicación:

Más importante
industrialmente

- Toma de Decisiones (banca-finanzas-seguros, márketing, políticas sanitarias/demográficas, ...)
- Investigación Científica (medicina, astronomía, meteorología, psicología, ...). Aquí la eficiencia no es tan importante.
- Soporte al Diseño de Bases de Datos.
- Reverse Engineering (dados una base de datos, desnormalizarla para que luego el sistema la normalice).
- Mejora de Calidad de Datos.
- Generación y Mejora de Consultas (si se descubren dependencias funcionales nuevas u otras condiciones evitables).

Áreas de Aplicación

KDD para toma de decisiones (Dilly 96)

- Comercio/Marketing:
- Identificar patrones de compra de los clientes.
 - Buscar asociaciones entre clientes y características demográficas.
 - Predecir respuesta a campañas de mailing.
 - Análisis de cestas de la compra.
- Banca:
- Detectar patrones de uso fraudulento de tarjetas de crédito.
 - Identificar clientes leales.
 - Predecir clientes con probabilidad de cambiar su afiliación.
 - Determinar gasto en tarjeta de créditos por grupos.
 - Encontrar correlaciones entre indicadores financieros.
 - Identificar reglas de mercado de valores a partir de históricos.
- Seguros y Salud Privada:
- Análisis de procedimientos médicos solicitados conjuntamente.
 - Predecir qué clientes compran nuevas pólizas.
 - Identificar patrones de comportamiento para clientes con riesgo.
 - Identificar comportamiento fraudulento.
- Transportes:
- Determinar la planificación de la distribución entre tiendas.
 - Analizar patrones de carga.

Áreas de Aplicación

KDD para toma de decisión

Medicina:

- Identificación de terapias médicas satisfactorias para diferentes enfermedades.
- Asociación de síntomas y clasificación diferencial de patologías.
- Estudio de factores (genéticos, precedentes, hábitos, alimenticios, etc.) de riesgo/salud en distintas patologías.
- Segmentación de pacientes para una atención más inteligente según su grupo.
- Predicciones temporales de los centros asistenciales para el mejor uso de recursos, consultas, salas y habitaciones.
- Estudios epidemiológicos, análisis de rendimientos de campañas de información, prevención, sustitución de fármacos, etc.

ILP para KDD

El modelo de datos más extendido es el relacional.
Una base de datos relacional se puede ver como una teoría lógica.

Terminología de Bases de Datos y Programación Lógica

Bases de Datos	Programación Lógica
Relación p	Símbolo de predicado p
Atributo, campo o columna de la relación p	Argumento del predicado p
Tupla o fila $\langle a_1, \dots, a_n \rangle$ de la relación p .	Hecho ground $p(a_1, a_2, \dots, a_n)$
Relación p como conjunto de tuplas (tabla)	Predicado p definido extensionalmente
Relación q definida como una vista	Predicado q definido intensionalmente

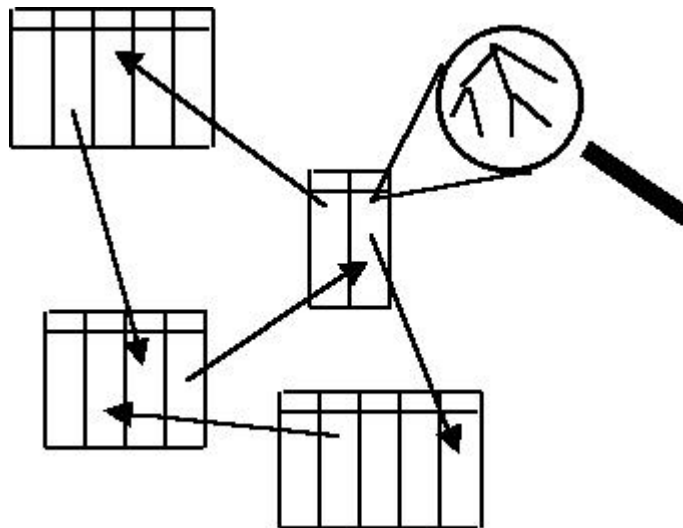
¿Por qué no utilizar inducción relacional?

Representación ILP para KDD

- Ventajas de ILP para KDD:
 - Representación de los datos: acepta múltiples tablas, esquemas complejos...
 - Representación de los patrones/hipótesis: modelos relacionales y recursivos.
 - Representación del conocimiento previo: uso de conocimiento del esquema o de expertos.
 - Mejor acoplamiento con otras tecnologías de bases de datos: bases de datos deductivas, activas, lenguajes de consulta avanzados...

Representación ILP para KDD

- Representación de los datos: acepta múltiples tablas, esquemas complejos...
 - No hay necesidad de desnormalizar --> menos cantidad de datos, menos redundancia.



Representación ILP para KDD

- Representación de las hipótesis:
 - Algunos problemas son complejos estructuralmente.
 - Necesitan conexiones entre varias tablas o entre diferentes tuplas de la misma tabla.
 - Inteligibilidad: las hipótesis en un lenguaje lógico son directamente inteligibles por los expertos.
 - La traducción de las hipótesis a SQL es directa.

Representación ILP para KDD

Representación de Datos. Ejemplo: EAST-WEST TRAINS.

Codificación del Ejemplo 1
como una base de datos:



LOAD_TABLE

LOAD	CAR	OBJECT	NUMBER
l1	c1	circle	1
l2	c2	hexagon	1
l3	c3	triangle	1
l4	c4	rectangle	3
...	...	...	...

TRAIN_TABLE

TRAIN	EASTBOUND
t1	TRUE
t2	TRUE
...	...
t6	FALSE
...	...

CAR_TABLE

CAR	TRAIN	SHAPE	LENGTH	ROOF	WHEELS
c1	t1	rectangle	short	none	2
c2	t1	rectangle	long	none	3
c3	t1	rectangle	short	peaked	2
c4	t1	rectangle	long	none	2
...	...	...	...	...	...

Representación ILP para KDD

EJEMPLO: EAST-WEST TRAINS. (datalog)

Codificación del Ejemplo 1:



E: train_table(t1, true).
 car_table(c1, t1, rectangle, short, none, 2).
 car_table(c2, t1, rectangle, long, none, 3).
 car_table(c3, t1, rectangle, short, peaked, 2).
 car_table(c4, t1, rectangle, long, none, 2).
 load_table(l1, c1, circle, 1).
 load_table(l2, c2, hexagon, 1).
 load_table(l3, c3, triangle, 1).
 load_table(l4, c4, rectangle, 3).

B: $\emptyset$

H: train_table(T, true) :- car_table(C, T, _, short, W, _), W <> none.

Representación ILP para KDD

EJEMPLO: EAST-WEST TRAINS.



H: $\text{train_table}(T, \text{true}) \text{ :- car_table}(C, T, _, \text{short}, W, _), W \neq \text{none}.$

Hipótesis en SQL:

```
SELECT DISTINCT T.TRAIN
FROM TRAIN_TABLE T, CAR_TABLE C
WHERE T.TRAIN = C.TRAIN AND
      C.LENGTH = 'short' AND
      C.ROOF <> 'none';
```

Representación ILP para KDD

Representación de Hipótesis: Ejemplo: Hospitales Alemanes

hospital(HospitalID, Name, Location, Size, Owner, Class)

patient(PatientID, Name, Age, Sex, Outcome...)

diagnosis (DiagnosisID, Name, Latin)

therapy(TherapyID, Name, Duration, StandardMedification)

patient-therapy(PatientID, TherapyID, Dosage, Date, HospitalID)

patient-diagnosis(PatientID, DiagnosisID, Date, HospitalID)

Resultado:

“Pacientes mayores de 65 años que fueron diagnosticados en un hospital pequeño”.

patient(I,N,A,S,O) & A > 65 & p_d(I,D,Dt,H) & hospital(H,_,_,small,_,_).

Representación ILP para KDD

En algunos casos se puede desnormalizar y luego aplicar técnicas proposicionales.

patient-therapy(NamePa, AgePa, SexPa, OutcomePa, NameTh, DurationTh,
StandardMedificationTh, Dosage, Date, NameHo,
LocationHo, SizeHo, OwnerHo, ClassHo)

patient-diagnosis(NamePa, AgePa, SexPa, OutcomePa, NameDi, LatinDi,
Date, NameHo, LocationHo, SizeHo, OwnerHo, ClassHo)

“Pacientes mayores que 65 años que fueron diagnosticados en un hospital pequeño”.

patient-diagnosis(N,A,S,O,ND, LD, Dt, __,small,__,) & A > 65.

Representación ILP para KDD

Pero en otros no...

Ejemplo. Esquema:

SALES(customer_name, item_name, transaction_id)

LIVES(customer_name, district, city)

ITEM(item_name, category, price)

TRANSACTION(transaction_id, day, month, day)

Reglas (relacional):

lives(C,_, "Vancouver") and

sales(C, "Census_CD", _) and sales(C, "Ms/Office97", _)

⇒ sales(C, "Ms/SQLServer",_)

Para poderlo expresar en proposicional deberíamos crear una columna para cada producto!!!

Técnicas de ILP para KDD

Uso de claves ajenas: permiten limitar (o guiar) las cláusulas para evitar literales sueltos o no unidos con los anteriores utilizando estas claves ajenas. *Formalmente:*

Dada una base de datos con relaciones R y claves ajenas F , una cláusula C **bien inspirada** por el esquema se define como.

$$C \equiv r_0(V_{01}, V_{02}, \dots, V_{0a}) :- l_1, l_2, \dots, l_n$$

- donde $r_0 \in R$ y tiene aridad a , siendo la relación objetivo.
- donde cada l_i puede ser de cualquiera de las siguientes formas:
 - $l_i \equiv r_i(V_{i1}, V_{i2}, \dots, V_{ib})$ donde $r_i \in R$ y b es la aridad de r_i .
 - $l_i \equiv V_{jk} \diamond_{comp} v$ donde V_{jk} es un argumento de otro literal, v es un valor (constante) del dominio de V_{jk} y $\diamond_{comp}$ es un predicado de comparación ($<$, $>$, $=$, $\leq$, $\geq$).
- y ninguna relación puede estar más de i enlaces de la relación objetivo.

Técnicas de ILP para KDD

EJEMPLO:

$$D = \{r_0, r_1, r_2, r_3, r_4, r_5\}$$

$$F = \{ r_0[2] \rightarrow r_1[1], r_0[3] \rightarrow r_2[1], r_1[2] \rightarrow r_3[1], r_2[2] \rightarrow r_3[2], \\ r_2[3] \rightarrow r_4[1], r_4[2] \rightarrow r_5[1] \}$$

$$i = 2$$

Ejemplos de cláusulas legales e ilegales:

$$r_0(X, Y, Z) :- r_1(Y, U), r_2(Z, R, W), r_3(U, R), X = x_0, R \geq \text{medium}, W = w_0.$$

$$r_0(X, Y, Z) :- r_1(Y, U), r_2(Z, R, W), r_4(W, S), X = x_0, U = u_0, S = s_0, R = r_0.$$

$$r_0(X, Y, Z) :- r_3(U, R).$$

$$r_0(X, Y, Z) :- r_1(Y, U), r_2(Z, R, W), r_4(W, S), r_5(S, T), U = u_0, R = r_0, T = t_0.$$

$$r_0(X, Y, Z) :- r_3(X, R).$$

Técnicas de ILP para KDD

El uso de claves ajenas mejora el rendimiento de algoritmos que usan pick & mix, especialmente los árboles de decisión:

- Algunos sistemas tienen restricción de enlace (FOIL) para el PICK & MIX. Mejoras (compatibles):
 - si el enlace es tipado
 - si el enlace es por clave ajena.
- Los criterios de GAIN y GAINRATIO nos pueden dar mucha información sobre cuando hacer PICK & MIX.

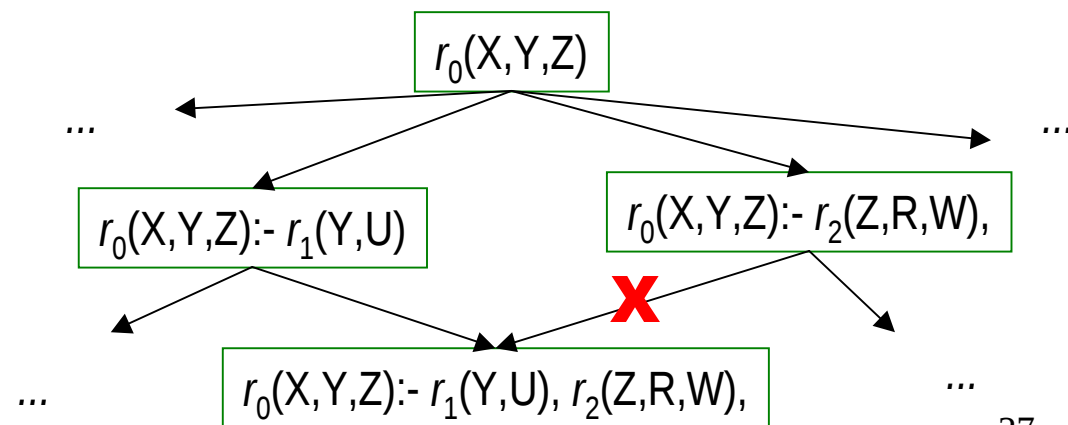
P.ej. Si una partición tiene el mejor GAIN utilizando un valor que es CAj. a una tabla S pero tiene mal GAINRATIO, es posible que exista una partición mejor utilizando alguno de los atributos de la tabla S. --> Añadirla con PICK & MIX.

Técnicas de ILP para KDD

Algoritmo MIDOS (Wrobel 1996):

Es una adaptación del FOIL para aprovechar el esquema.
De este modo se obtiene un operador de refinamiento ideal
(finito, propio, completo) y óptimo (evita caminos duplicados).
Ordena las relaciones del esquema:

Además facilita la
evaluación y la poda
posterior del árbol.



Sistemas ILP para KDD

Situación Actual:

- No hay sistemas específicos, excepto CLAUDIEN (De Raedt & Dehaspe 1997)
- Se utilizan los sistemas ILP directamente importando los datos.
- Los sistemas ILP admiten pocos formatos (el acceso a los datos suele ser a través de texto).
- Problemas de eficiencia.

Medio Plazo:

- La tecnología plug-in permite añadir tecnología ILP a sistemas de minería de datos generales.
- Los nuevos lenguajes de consulta pueden ser una vía de entrada para las técnicas ILP.

Diseño Inductivo de Bases de Datos

Cuando se diseña una base de datos (deductiva) se debe decidir si definir una relación (un predicado) de manera extensional (tabla) o de manera intensional (vista).

A veces esta decisión depende de los futuros usuarios de la base de datos, pero otras veces depende más de requerimientos de recursos (espacio, fundamentalmente).

(Blockeel & De Raedt 1996, 1998) presentan “IsIdd”, un sistema interactivo que asiste al diseñador de bases de datos.

- Se parte de una base de datos extensional.
- El sistema transforma algunas relaciones a una forma intensional, y las propone al diseñador, con el objetivo de reducir espacio.
- El sistema también propone restricciones de integridad al usuario.
 - *Detecta dependencias funcionales -> claves primarias*
 - *Detecta restricciones generales..*

Diseño Inductivo de Bases de Datos

Optimización de Consultas: Otra aplicación de la inducción en bases de datos es la optimización de consultas (Hsu & Knoblock 1996).

P.ej. supongamos el siguiente esquema:

geoloc(name, glc_cd, country, latitude, longitude).

CP:{glc_cd}

seaport(name, glc_cd, storage, silo, crane, rail).

CP:{name} CAj:{glc_cd} --> geoloc(glc_cd)

y la siguiente consulta:

Q1: answer(?name) :- geoloc(?name, ?glc_cd, 'Malta', _, _),
seaport(_, ?glc_cd, ?storage, _, _, _),
?storage > 1500.

“Localidades de Malta con puertos de capacidad mayor a 1500”³⁰

Diseño Inductivo de Bases de Datos

Optimización de Consultas:

Ejemplo (cont.).

una herramienta inductiva podría establecer las siguientes reglas a partir de la extensión de la base de datos:

R1: $\text{geoloc}(_, _, \text{'Malta'}, ?\text{latitude}, _) \Rightarrow ?\text{latitude} \geq 35.89$.

R2: $\text{geoloc}(_, ?\text{glc_cd}, \text{'Malta'}, _, _) \Rightarrow \text{seaport}(_, ?\text{glc_cd}, _, _, _)$.

R3: $\text{seaport}(_, ?\text{glc_cd}, ?\text{storage}, _, _) \wedge \text{geoloc}(_, ?\text{glc_cd}, \text{'Malta'}, _, _) \Rightarrow ?\text{storage} > 2000$.

con este conocimiento se pueden generar consultas más eficientes que Q1 y equivalentes a ella. P.ej.:

Q2: $\text{answer}(?\text{name}) \text{ :- geoloc}(?\text{name}, ?\text{glc_cd}, \text{'Malta'}, _, _)$. equivale a Q1 por R2 y R3

Q3: $\text{answer}(?\text{name}) \text{ :- geoloc}(?\text{name}, ?\text{glc_cd}, \text{'Malta'}, ?\text{latitude}, _) \wedge ?\text{latitude} \geq 35.89$.

Equivale a Q1 por R1 y R3. Mejora la anterior si el campo "latitude" está indexado¹.

Diseño Inductivo de Consultas

Query By Example: generalmente se conoce como el uso de interfaces visuales para realizar consultas. Iniciado con el QBE del QMF de IBM y hoy en día integrado en la mayoría de SGBD de gama baja (Access/Paradox) y muchos de gama alta.

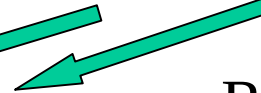
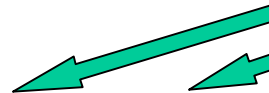
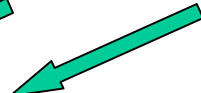
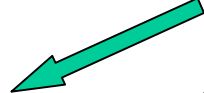
Por diseño o construcción inductiva de consultas se entiende otra cosa **diferente**.

Se trata de seleccionar el resultado de la consulta y que el sistema *induzca* la consulta.

Diseño Inductivo de Consultas

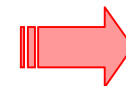
Ejemplo:

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	X
22451616	3.000.000	León	Ejecutivo	35	2	S	S	
25152516	2.000.000	Valencia	Camarero	30	0	S	S	X
23525251	1.500.000	Benidorm	Animador	25	1	N	S	X
35727238	4.500.000	Valencia	Malabarista	32	1	N	N	X
98484390	6.500.000	Castelló	Azulejero	50	4	S	S	
46572789	2.000.000	Alacant	Clavarieso	47	0	S	N	
56262462	500.000	Albacete	Astronauta	28	0	N	S	X
22622572	1.500.000	Alzira	Profesor	32	1	N	S	X



DNI	Ciudad	Profesión	Edad	Hijos
30512526	Melilla	Abogado	25	0
25152516	Valencia	Camarero	30	0
23525251	Benidorm	Animador	25	1
35727238	Valencia	Malabarista	32	1
56262462	Albacete	Astronauta	28	0
22622572	Alzira	Profesor	32	1

Regla Extraída:



Edad < 33.5

Consulta:

```
SELECT DNI, Ciudad, Profesión, Edad, Hijos
FROM PERSONAS
WHERE Edad < 33.5
```

Lenguajes de Consulta Inductivos

Nuevos Lenguajes de Consulta:

El descubrimiento en bases de datos se ve como un proceso de consulta a una base de datos (Imielinski and Manilla 1996). La situación se parece al desarrollo de lenguajes de consulta en los sesenta y setenta.

Una consulta inductiva o de búsqueda de patrones debe permitir al usuario restringir la búsqueda inductiva en los siguientes aspectos (Han et al. 1999):

- La parte de la base de datos a ser minada (también llamada la vista minable o vista relevante) (Ng et al. 1998).
- El tipo de patrón/reglas a ser minado (también llamado restricciones del conocimiento).
- Cuantificadores estadísticos: representatividad (support) $\exists\%$, precisión (confidence/accuracy) $\forall\%$.
- Otras propiedades que el patrón debería cumplir (número y forma de las reglas, interés, novedad, etc.).

Lenguajes de Consulta Inductivos

Ejemplos de consultas que se desean:

ASOCIACIÓN

-¿**Por qué (causa)** la división de "serie para torpes" es tan provechosa?

ASOCIACIÓN + CLASIFICACIÓN

-¿**Qué características comparten** los clientes que no renovaron sus pólizas y **en qué se diferencian de** las que renovaron sus pólizas?

CLUSTERING:

-**Grupos** de clientes que no pagaron su crédito.

-**Grupos** de productos que han fallado el test de calidad.

CLUSTERING + PREDICCIÓN

-**Grupos** de clientes que **es probable** que **vayan a** comprar un nuevo producto en el próximo año.

CLUSTERING + ASOCIACIÓN

-**Grupos** de pacientes cuya muerte la causó **combinaciones** (cócteles) de fármacos.

Lenguajes de Consulta Inductivos

La consultas no pueden ser en lenguaje natural...

¿Qué es exactamente lo que se busca?

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

Lenguajes de Consulta Inductivos

EJEMPLO (cont.):

Correlaciones y Asociaciones

Tipo de consulta:

```
SELECT CORRELATIONS ON NUMERICAL  
FROM Persona
```

Respuesta:

Renta Familiar y Edad correlan (0.XX).

Hijos y Edad correlan (0.YY).

Renta Familiar e Hijos correlan (0.ZZ).

Tipo de consulta:

```
SELECT APPROX. ASSOCIATIONS ON (Obeso, Casado, Hijos > 0)  
FROM Persona
```

Respuesta:

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

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

Se deben poder poner condiciones, p.ej.:

```
SUPPORT > 3, CONFIDENCE > 75%
```

Lenguajes de Consulta Inductivos

EJEMPLO (cont.):

Reglas de Dependencias de Valor

Tipo de consulta:

```
SELECT EXACT DEPENDENCY RULES ON (Obeso, Casado, Hijos > 0)
FROM Persona
```

Respuesta: (Hijos > 0) → Casado (5 casos, 100%).
Casado → Obeso (5 casos, 100%)

Tipo de consulta:

```
SELECT EXACT DEPENDENCY RULES ON *
FROM Persona
```

Respuesta: (DNI) → ...
Ciudad → ...

Tipo de consulta:

```
SELECT PROB. DEPENDENCY RULES ON *
FROM Persona
```

Respuesta: Casado → Obeso (5 casos, 100%)

Se compara $P(\text{RHS/LHS})$ con $P(\text{RHS})$ para ver si tiene SUPPORT (significación). 38

Lenguajes de Consulta Inductivos

EJEMPLO (cont.):

Clasificación

Tipo de consulta:

```
SELECT CLASSIFICATION RULES FOR (Hijos > 0) ON *  
FROM Persona
```

Respuesta:

Casado AND (Renta Familiar > 2.000.000) → (Hijos > 0)

También se deberían poder obtener reglas exactas, fuertes (sólo se permite un máx de errores) o probabilísticas.

Lenguajes de Consulta Inductivos

EJEMPLO (cont.):

Segmentación

Tipo de consulta:

```
SELECT SEGMENTATION RULES ON *  
FROM Persona
```

Respuesta: Three Classes

```
Class 1 if Casado AND (Renta Familiar > 2.000.000)  
Class 2 if Casado AND (Renta Familiar <= 2.000.000)  
Class 3 if ¬Casado
```

Tipo de consulta:

```
SELECT SEGMENTATION RULES FOR Renta Familiar ON *  
FROM Persona
```

Respuesta: Two classes on Renta Familiar

```
Renta Familiar >= 2.000.000 if Casado  
Renta Familiar < 2.000.000 if ¬Casado
```

Lenguajes de Consulta Inductivos

EJEMPLO (cont.):

Tendencias temporales, Predicción

Patrón secuencial:

¿Qué compras preceden a la compra de un microondas?

Respuesta:

Frigorífico con congelador de cuatro pisos (60%).

Predicciones:

¿Volumen total de ventas estimado para el año 2000?

Regresión lineal sobre ventas 1995-1999 para predecir ventas 2000.

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

Tipo de consulta:

SELECT PRIMARY KEYS ON (Obeso, Casado, Hijos, Profesión)

FROM Persona

Respuesta:

(Profesión, Hijos)

(Hijos, Obeso, Casado)

Lenguajes de Consulta Inductivos

EJEMPLO (cont.):

Patrones más complicados

Varias tablas:

```
SELECT RULES  
FROM Persona, Casado
```

Respuesta:

Persona(X) AND Persona (Y) AND Casado(X,Y) $\rightarrow$ Renta Familiar(X) = Renta Familiar(Y)

Esta aproximación es restrictiva a los “modelos de consulta” que permita el lenguaje.

Lenguajes de Consulta Inductivos

Propuesta M-SQL (Imielinski et al. 1996)

Basada en modelos de consulta...

Ejemplo:

```
SELECT FROM MINE(T): R
WHERE R.Consequent = { (Age = *) }
      R.Support > 1000
      R.Confidence > 0.65;
```

R es una variable de regla y se puede utilizar:

- R.Consequent
- R.Body (antecedente)
- R.Support
- R.Confidence.

Lenguajes de Consulta Inductivos

Propuesta DMQ (Data-Mining Query) language (Ng et al. 1998):

- Utiliza la sintaxis del SQL para la vista minable
- También basado en modelos de consulta.

EJEMPLO:

Esquema:

SALES(customer_name, item_name, transaction_id)

LIVES(customer_name, district, city)

ITEM(item_name, category, price)

TRANSACTION(transaction_id, day, month, day)

Consulta Inductiva (lenguaje natural):

“buscar las ventas de qué artículos baratos (con una suma de precios menor que \$100) que puede motivar las ventas de qué artículos caros (con el precio mínimo de \$500) de la misma categoría de los clientes de Vancouver en 1998”.

Lenguajes de Consulta Inductivos

Propuesta DMQ. EJEMPLO:

Ejemplo de Consulta Inductiva:

mine associations as

lives(C,_, "Vancouver") and

sales+(C, ?[I], {S}) $\Rightarrow$ sales+(C, ?[J], {T})

from sales

where S.year = 1998 and T.year = 1998 and I.category = J.category

group by C, I.category

having sum(I.price) < 100 and min(J.price) >= 500

with min_support = 0.01 and min_confidence = 0.5

+: operador regular (1 o más tuplas)

?[I] : utilizar clave ajena. I es la tupla instanciada.

Ejemplo de Respuesta:

lives(C,_, "Vancouver") and

sales(C, "Census_CD", _) and sales(C, "Ms/Office97", _)

$\Rightarrow$ sales(C, "Ms/SQLServer",_) [0.015, 0.68]

Es un patrón relacional.

Support & Confidence.

Lenguajes de Consulta Inductivos

Propuesta LDL+ (Shen et al. 1996)

Lenguaje lógico de orden superior que permite expresar gran variedad de restricciones de reglas utilizando esquemas de orden superior. Los esquemas se llaman *metaqueries*:

P.ej.: $P(X,Y) \text{ and } Q(Y,Z) \rightarrow R(X,Z)$

determina que se quieren buscar dependencias funcionales de tipo transitivo.

Podría dar como resultado:

$\text{citizen}(X,Y) \text{ and } \text{officiallanguage}(Y,Z) \rightarrow \text{speaks}(X,Z)$
with a probability 0.93.

Lenguajes de Consulta Inductivos

Propuesta LDL+ (Shen et al. 1996)

También tiene acciones de meta-consultas especiales:

P.ej.: $P(X,Y) \text{ and } Q(Y,Z) \rightarrow \text{Cluster}(Z)$

determina que se quieren buscar segmentos por la variable Z.

P.ej.: $P(X,Y) \text{ and } Q(Y,Z) \rightarrow \text{Classifier}(Z)$

determina que se clasifique por la variable Z.

Otros sistemas como RDT/DB (Brockhausen & Morik 1997)

utilizan también el concepto de metaquery.

Lenguajes de Consulta Inductivos

Propuesta RDM (De Raedt 1998):

Busca “Patrones”.

Un patrón se define como un par (Query, Key), donde Query es una consulta en Prolog (un conjunto de condiciones) y Key es una lista de variables que aparecen en la query (representa una tupla).

Ejemplo: (father(X,Y) and parent(Y,Z), [X,Z])
este patrón daría los X, Z tales que X es abuelo de Z.

Se define una relación de generalidad entre queries

Query1 $\preceq$ Query2 :- subset(Query1, Query2) similar a la θ -subsunción de Plotkin.

La cobertura se establece de la siguiente manera:

covers(Query, Key, Example) :- $\forall$ (Key = Example, Query)

Ejemplo: Dada la B.D.: human(luc), human(lieve), male(luc), female(lieve)
el patrón ((human(X), male(X), true), [X]) cubre el ejemplo [luc]

Lenguajes de Consulta Inductivos

Propuesta RDM (cont.):

A partir de aquí se puede definir la frecuencia con la cual se cumple un conjunto de ejemplos ya sean en la base de datos $\text{frequency}(\text{Conclusion}, \text{Key}, \text{Freq})$ o pasados como un argumento $\text{frequency}(\text{Conclusion}, \text{Key}, \text{ExampleList}, \text{Freq})$.

Utilizando Prolog es fácil definir qué es una regla de asociación:

$\text{Associationrule}(\text{Conclusion} \leftarrow \text{Condition}), \text{Key}, \text{Acc}, \text{Freq}) :- \text{Condition} \prec \text{Conclusion},$
 $\text{frequency}(\text{Condition}, \text{Key}, \text{Freq1}), \text{frequency}(\text{Conclusion}, \text{Key}, \text{Freq}), \text{Acc is } (\text{Freq1})/\text{Freq}.$

para expresar “consultas inductivas” no haría falta nada más que buscar patrones que fueran más específicos o más generales que otros. El problema es que algunas de estas consultas NO serían SEGURAS!

El sistema RDM obliga a que se fuerce que la Query a buscar debe ser más general que una query más específica compuesta por todas las condiciones.

$\text{Query} \preceq (\text{lit1}, \dots, \text{lit2}, \text{true})$

y con algunas construcciones más ya se pueden hacer “consultas inductivas”⁴⁹!

Lenguajes de Consulta Inductivos

Propuesta RDM (cont.):

Ejemplos de Consultas Inductivas:

- Patrones de Frecuencia: Ejemplo: Dame las combinaciones frecuentes de productos en una cesta.
?- Query $\preceq$ (Item-1(X), ..., Item-n(X), true) , frequency(Query, X, F), F > 100.
- Asociaciones: Ejemplo: Dame las asociaciones entre productos (Freq > 100, Acc > 0.5)
?- Query 2 $\preceq$ (Item-1(X), ..., Item-n(X), true) ,
associationrule(Query1 $\leftarrow$ Query2), X, Freq, Acc), Freq > 100, Acc > 0.5.
- Clasificación: Ejemplo: Dame condiciones de reglas que sean al menos 90% correctas y que cubran como mucho 10 ejemplos negativos.
?- Query $\preceq$ (Item-1(X), ..., Item-n(X), true), frequency(Query,X,Positives,P),
frequency(Query,X,Negatives,N), N < 10, Acc is P / (P+N), Acc > 0.9.

Esto supone una generalización de M-SQL o Metaqueries.

Lenguajes de Consulta Inductivos

Lenguajes de Consultas por Abducción:

Se basan en SQL estándar en el que se fuerza al que el resultado de la consulta tenga un determinado valor. En la consulta aparecen variables:

Ejemplo:

Siendo B, A1, A2 y A3 variables.

```
1= (      SELECT MAX(COUNT(DISTINCT B))  
      FROM R  
      GROUP BY A1, A2, A3;      );
```

Darían aquellas dependencias funcionales exactas del tipo:
 $A1, A2, A3 \twoheadrightarrow B$.

Lenguajes de Consulta Inductivos

Lenguajes de Consultas Inductivos por Extrapolación:

Se basan en SQL estándar en el que las condiciones (WHERE) se sustituyen por supuestos (CASE)

```
SELECT SUM(V.precio)
FROM VENTAS V
WHERE V.year = 1999;
```

Darí­a el total de ventas del año 1999.

```
SELECT SUM(V.precio)
FROM VENTAS V
CASE V.year = 2005;
```

Darí­a el total de ventas que se predice para el año 2005.

Lenguajes de Consulta Inductivos

Lenguajes de Consultas Inductivos por Extrapolación:

Otros ejemplos: cualquiera de clasificación o predicción.

P. ej. ¿cuál es el tratamiento médico adecuado para los síntomas $s_1, s_2, \dots, s_k$?

Si esa combinación de síntomas no se da en la base de datos, en vez de retornar la relación vacía, retorna una extrapolación.

Ventajas:

- permite utilizar la sintaxis del SQL.

Inconvenientes:

- el usuario debe saber qué quiere predecir (minería de datos dirigida).
- en muchos casos no existe una resolución única de la consulta.

Algunas Cuestiones Legales

Hay dos cuestiones importantes respecto a un uso indiscriminado de KDD:

- El primero es si los clientes o otros usuarios externos en general se pueden ver incomodados o amenazados por la compañía al atacar su privacidad o someterlos a márketing abusivo.
- El segundo es si estas políticas pueden ser ilegales.

Consecuencias:

- En el primer caso, la compañía o institución obtienen mala prensa y antipatía (lo cual se puede traducir en una pérdida económica).
- En el segundo caso, la compañía puede ser demandada por miles de clientes, con unos costes de millones de euros.

Algunas Cuestiones Legales

Si nos centramos sólo en las cuestiones legales del KDD:

- Uso de datos de fuentes internas a la compañía:
 - se pueden utilizar los datos como se quiera, siempre internamente.
- Uso de datos de fuentes externas a la compañía:
 - SÍ se puede si los datos son públicos (la persona ha decidido que lo sean). Ejemplo: páginas web, guía telefónica, visitas en la web.
 - SÍ se puede si los datos son agregados (i.e. no contienen individuos). Ejemplo: asociaciones entre productos, llamadas por distrito/hora, segmentaciones, etc ...
 - NO se puede si se la persona los ha cedido por transacción u operación habitual y privada con la otra compañía. Ejemplo: datos bancarios, horarios y números de llamadas, cestas de la compra, historiales clínicos, viajes, etc.

Algunas Cuestiones Legales

Cuestiones legales del KDD:

También depende del país. Hay dos tendencias:

- *Defender* los derechos de los consumidores (si éstos lo solicitan y se pagan el juicio):
 - los usuarios pueden decir que “no” a que se usen sus datos (por defecto se pueden usar). Es la filosofía dominante en EE.UU. Ver p.ej. (Volokh 2000).
- *Proteger* los derechos de los consumidores:
 - se prohíbe que las empresas hagan uso de ciertos tipos de datos (bancarios, médicos, comunicaciones) o que al utilizar el KDD implementen políticas o campañas racistas, sexistas, xenófobas o discriminatorias. Es la filosofía dominante en la Unión Europea (excepto Reino Unido).

Algunas Cuestiones Legales

Cuestiones legales del KDD. KDD y Discriminación:

*Una parte importante de los objetivos del KDD es **discriminar** poblaciones (especialmente clientes).*

No existe una línea clara entre discriminación legal/ilegal...

- ¿Enviar una campaña/oferta de compresas sólo a mujeres es legal?
- ¿Enviar una campaña/oferta de libros científicos sólo a mujeres es legal?
- ¿Enviar una campaña/oferta de libros de matemáticas sólo a clientes de raza asiática (determinado por análisis de apellidos) es legal?
- ¿Enviar una campaña/oferta de bronceadores sólo a clientes de piel blanca (determinado por análisis de las fotos) es legal?
- ¿Enviar una campaña/oferta de una colección de libros de la literatura clásica greco-romana sólo a clientes de raza blanca (determinado por análisis de apellidos) es legal?
- ¿Enviar una campaña/oferta de biblias a cristianos (determinado por los minutos de visión del Christian Channel en un paquete digital de pago)?

Algunas Cuestiones Legales

Cuestiones legales del KDD. KDD y Protección de la Intimidad:

Ley orgánica 15/1999 de 13 de diciembre de protección de datos (LOPD):

Resumen:

"el responsable del fichero o tratamiento que incluya datos de carácter personal (p.ej. direcciones de correo electrónico) deberá informar al afectado de la existencia de dicho tratamiento, de la finalidad de la recogida de sus datos, de los destinatarios de la información, de la obligación de responder a las preguntas que se les formulan, de las consecuencias de la obtención de sus datos, de la posibilidad de ejercitar los derechos de acceso, rectificación, cancelación y oposición; y de la identidad y dirección del responsable del fichero, o de su representante.

El afectado puede ejercitar su derecho de oposición, lo que supone que el responsable tiene que dar de baja el tratamiento de los datos que le conciernan, cancelando las informaciones que figuren sobre él, tras su simple solicitud. El ejercicio de este derecho es GRATUITO para el interesado".

Algunas Cuestiones Legales

Cuestiones legales del KDD. KDD y Comercio Electrónico:

- Correo no solicitado o indiscriminado (vulgarmente, 'spam'): supone para el/los destinatario/s un menoscabo de su intimidad por tratamiento de datos no consentido o recepción de mensajes publicitarios no deseados y, más importante, por el coste de la conexión y del uso de equipos informáticos para leerlos.

Según la LOPD puede ser ilegal, hasta la salida de la nueva normativa sobre comercio electrónico (DOCEC 128 de 8 de mayo de 2000), que recoge:

- obligación de identificación (persona jurídica + dirección),
- la finalidad comercial del mensaje debe aparecer claramente,
- existirá un censo promocional (extraído del censo) pero también existirán también listas de exclusión.

Sistemas

TIPOS:

- *Standalone*: Los datos se deben exportar/convertir al formato interno del sistema de data mining: Knowledge Seeker IV (Angoss International Limited, Groupe Bull).
- *On-top*: funcionan sobre un sistema propietario (microstrategy sobre Oracle).
- *Embedded* (propietarios): Oracle Discoverer, Oracle Darwin, IBM...
- Extensible (Tecnología *Plug-ins*): proporcionan unas herramientas mínimas de interfaz con los datos, estadísticas y visualización, y los algoritmos de aprendizaje se pueden ir añadiendo con plug-ins. (ej. KEPLER).

Sistemas

Producto	Compañía	Técnicas	Plataformas	Interfaz
Knowledge Seeker	Angoss http://www.angoss.com/	Decision Trees, Statistics	Win NT	ODBC
CART	Salford Systems www.salford-systems.com	Decision Trees	UNIX/NT	
Clementine	SPSS/Integral Solutions Limited (ISL) www.spss.com	Decision Trees, ANN, Statistics, Rule Induction, Association Rules, K Means, Linear Regression.	UNIX/NT	ODBC
Data Surveyor	Data Distilleries http://www.datadistilleries.com/	Amplio Abanico.	UNIX	ODBC
GainSmarts	Urban Science www.urbanscience.com	Especializado en gráficos de ganancias en campañas de clientes (sólo Decision Trees, Linear Statistics y Logistic Regression).	UNIX/NT	
Intelligent Miner	IBM http://www.ibm.com/software/data/iminer	Decision Trees, Association Rules, ANN, RBF, Time Series, K Means, Linear Regression.	UNIX (AIX)	IBM, DB2
Microstrategy	Microstrategy www.microstrategy.com	Datawarehouse sólo	Win NT	Oracle
Polyanalyst	Megaputer http://www.megaputer.com/html/polyanalyst4.0.html	Symbolic, Evolutionary	Win NT	Oracle, ODBC
Darwin	Oracle http://www.oracle.com/ip/analyze/warehouse/datamining/index.html	Amplio Abanico (Decision Trees, ANN, Nearest Neighbour)	UNIX/NT	Oracle
Enterprise Miner	SAS http://www.sas.com/software/components/miner.html	Decision Trees, Association rules, ANN, regression, clustering.	UNIX (Sun), NT, Mac	Oracle, ODBC
SGI MineSet	Silicon Graphics http://www.sgi.com/software/mineset/	association rules and classification models, used for prediction, scoring, segmentation, and profiling	UNIX (Irix)	Oracle, Sybase, Informix.
Wizsoft/Wizwhy	http://www.wizsoft.com/			

Sistemas

- Más software comercial DM:
http://www.kdcentral.com/Software/Data_Mining/
<http://www.cs.bham.ac.uk/~anp/software.html>
http://www.cs.bham.ac.uk/~anp/dm_docs/oudshoff.tools.posting
- Algunos Prototipos No Comerciales o Gratuitos:
 - Kepler: sistema de plug-ins del GMD
(<http://ais.gmd.de/KD/kepler.html>).
 - Rproject: herramienta gratuita de análisis estadístico
(<http://www.R-project.org/>)
 - Librerías WEKA (<http://www.cs.waikato.ac.nz/~ml/weka/>) (Witten & Frank 1999)

Sistemas

EJEMPLO: **Clementine** (Integral Solutions Limited (ISL))

www.spss.com

- Herramienta que incluye:
 - fuentes de datos (ASCII, Oracle, Informix, Sybase e Ingres).
 - interfaz visual.
 - distintas herramientas de minería de datos: redes neuronales y reglas.
 - manipulación de datos (pick & mix, combinación y separación).

Sistemas

EJEMPLO: **Clementine**

Ejemplo Práctico: Ensayo de Medicamentos

http://www.pcc.qub.ac.uk/tec/courses/datamining/ohp/dm-OHP-final_3.html

- Un número de pacientes hospitalarios que sufren todos la misma enfermedad se tratan con un abanico de medicamentos.
- 5 medicamentos diferentes están disponibles y los pacientes han respondido de manera diferente a los diferentes medicamentos.
- Problema:

¿qué medicamento es apropiado para un nuevo paciente?

Sistemas

EJEMPLO: **Clementine**. Ejemplo Práctico: Ensayo de Medicamentos

Primer Paso: ACCEDIENDO LOS DATOS:

- Se leen los datos. Por ejemplo de un fichero de texto con delimitadores.
- Se nombran los campos:

age	edad
sex	sexo
BP	presión sanguínea (High, Normal, Low)
Cholesterol	colesterol (Normal, High)
Na	concentración de sodio en la sangre.
K	concentración de potasio en la sangre.
drug	medicamento al cual el paciente respondió satisfactoriamente.

SE PUEDEN COMBINAR LOS DATOS:

P.ej. se puede añadir un nuevo atributo: Na/K

Sistemas

EJEMPLO: Clementine

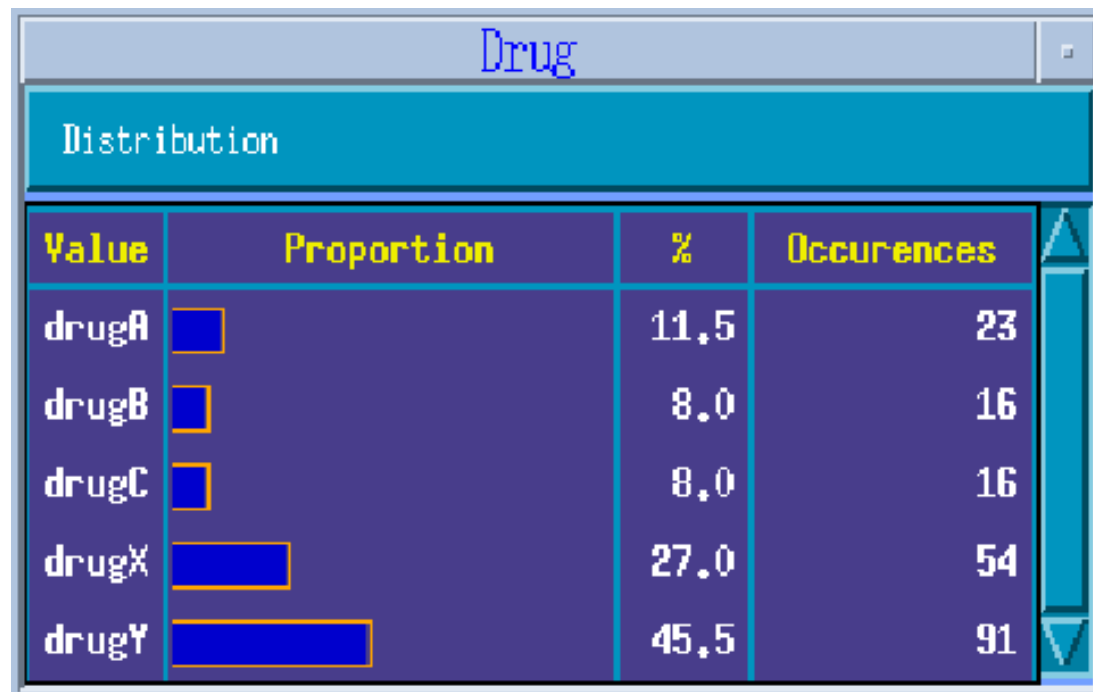
Segundo Paso: Familiarización con los Datos. Visualizamos los registros:

table1							
Table							
Age	Sex	BP	Cholestrol	Na	K	Drug	Na_to_K
23	F	HIGH	HIGH	0.79	0.03	drugY	25.35
47	M	LOW	HIGH	0.74	0.06	drugC	13.09
47	M	LOW	HIGH	0.7	0.07	drugC	10.11
28	F	NORMAL	HIGH	0.56	0.07	drugX	7.8
61	F	LOW	HIGH	0.56	0.03	drugY	18.04
22	F	NORMAL	HIGH	0.68	0.08	drugX	8.61
49	F	NORMAL	HIGH	0.79	0.05	drugY	16.28
41	M	LOW	HIGH	0.77	0.07	drugC	11.04
60	M	NORMAL	HIGH	0.78	0.05	drugY	15.17
43	M	LOW	NORMAL	0.53	0.03	drugY	19.37

Sistemas

EJEMPLO: Clementine

- Permite seleccionar campos o filtrar los datos
- Permite mostrar propiedades de los datos. Por ejemplo:
¿Qué proporción de casos respondió a cada medicamento?



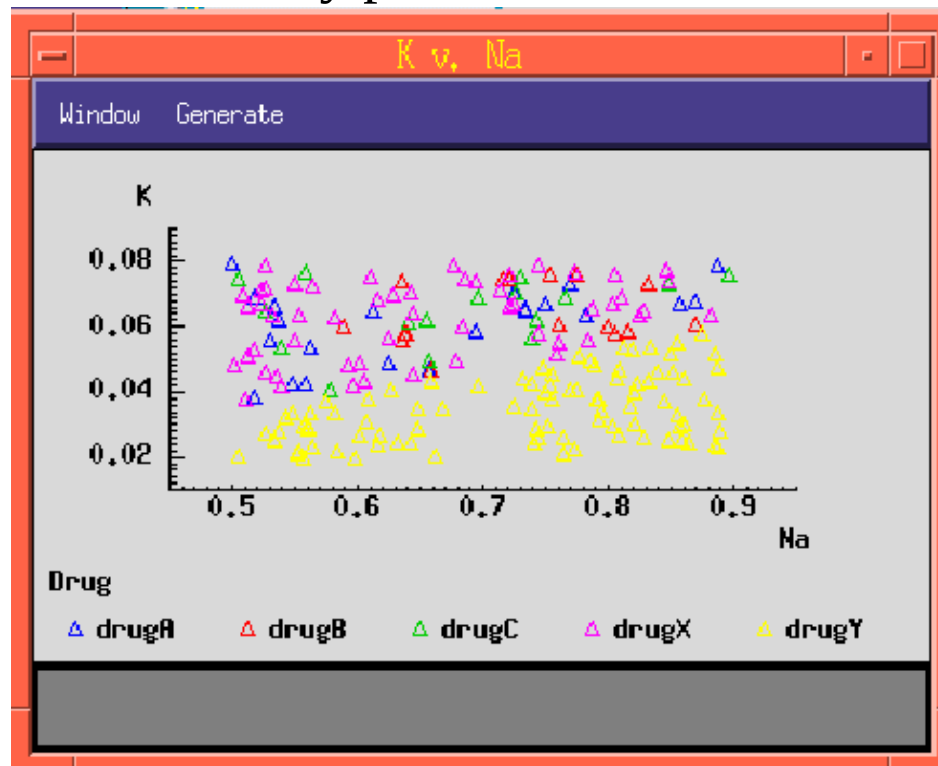
The screenshot shows a window titled 'Drug' with a 'Distribution' tab. It displays a table with four columns: 'Value', 'Proportion', '%', and 'Occurrences'. Each row represents a drug category, and the 'Proportion' column includes a horizontal bar chart. The data is as follows:

Value	Proportion	%	Occurrences
drugA		11,5	23
drugB		8,0	16
drugC		8,0	16
drugX		27,0	54
drugY		45,5	91

Sistemas

EJEMPLO: Clementine

- Permite encontrar relaciones. Por ejemplo:
La relación entre sodio y potasio se muestra en un gráfico de puntos.



Se observa una dispersión aparentemente aleatoria (excepto para el medicamento. Y)

Sistemas

EJEMPLO: Clementine

Se puede observar a simple vista que los pacientes con alto cociente Na/K responden mejor al medicamento Y.

Pero queremos una clasificación para todos los medicamentos. Es decir, nuestro problema original:

¿Cuál es el mejor medicamento para cada paciente?

Tercer Paso: Construcción del Modelo

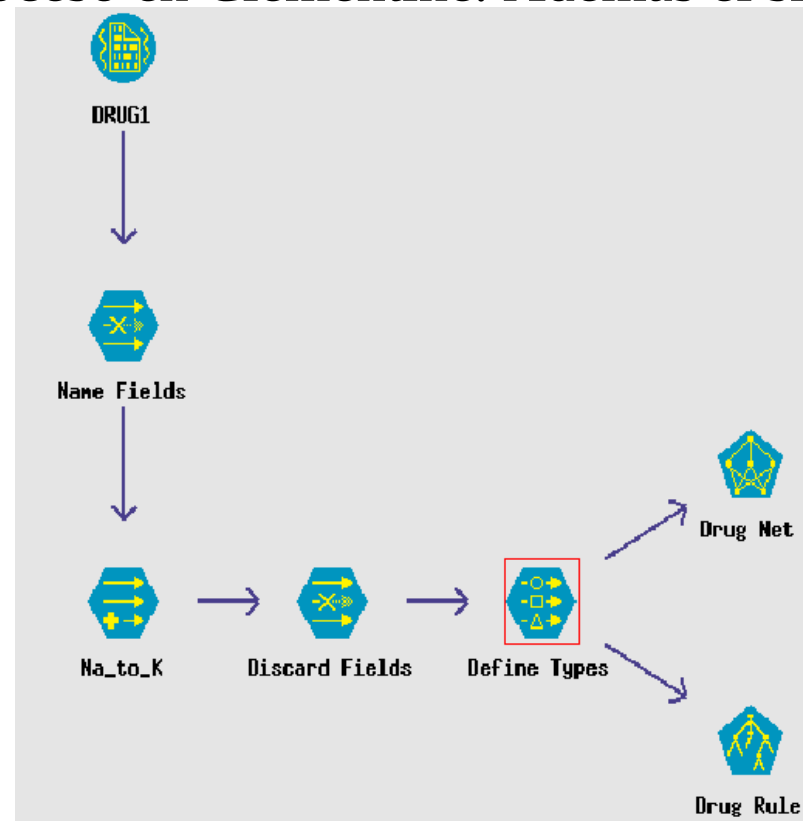
Tareas a realizar en Clementine:

- Filtrar los campos no deseados.
- Definir tipos para los campos.
- Construir modelos (reglas y redes)

Sistemas

EJEMPLO: Clementine

Se sigue este proceso en Clementine. Además el sistema lo visualiza:

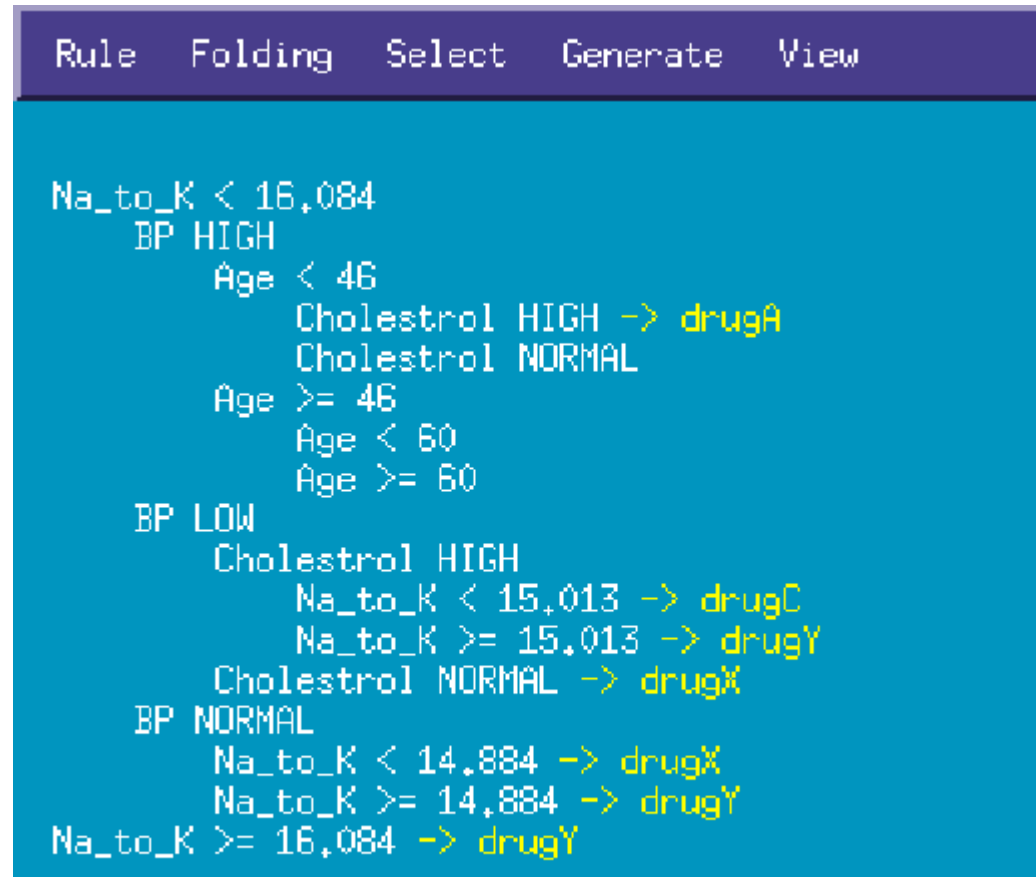


A partir de 2000 ejemplos entrena la red y construye las reglas.

Sistemas

EJEMPLO: Clementine

Permite examinar las reglas:



```
Rule  Folding  Select  Generate  View
Na_to_K < 16,084
  BP HIGH
    Age < 46
      Cholestrol HIGH -> drugA
      Cholestrol NORMAL
    Age >= 46
      Age < 60
      Age >= 60
  BP LOW
    Cholestrol HIGH
      Na_to_K < 15,013 -> drugC
      Na_to_K >= 15,013 -> drugY
    Cholestrol NORMAL -> drugX
  BP NORMAL
    Na_to_K < 14,884 -> drugX
    Na_to_K >= 14,884 -> drugY
Na_to_K >= 16,084 -> drugY
```

Las reglas extienden el mismo criterion que se había descubierto previamente: es decir, medicamento Y para los pacientes con alto cociente Na/K. Pero además añaden reglas para el resto.

Sistemas

EJEMPLO: SAS ENTERPRISE MINER (EM)

- Herramienta completa. Incluye:
 - conexión a bases de datos (a través de ODBC y SAS data sets).
 - muestreo e inclusión de variables derivadas.
 - partición de la evaluación del modelo respecto a conjuntos de entrenamiento, validación y chequeo (test)*.
 - distintas herramientas de minería de datos: varios algoritmos y tipos de árboles de decisión, redes neuronales, regresión y clustering.
 - comparación de modelos.
 - conversión de los modelos en código SAS.
 - interfaz gráfico.
- Incluye herramientas para flujo de proceso: trata en el proceso KDD como un proceso y las fases se pueden repetir, modificar y grabar.

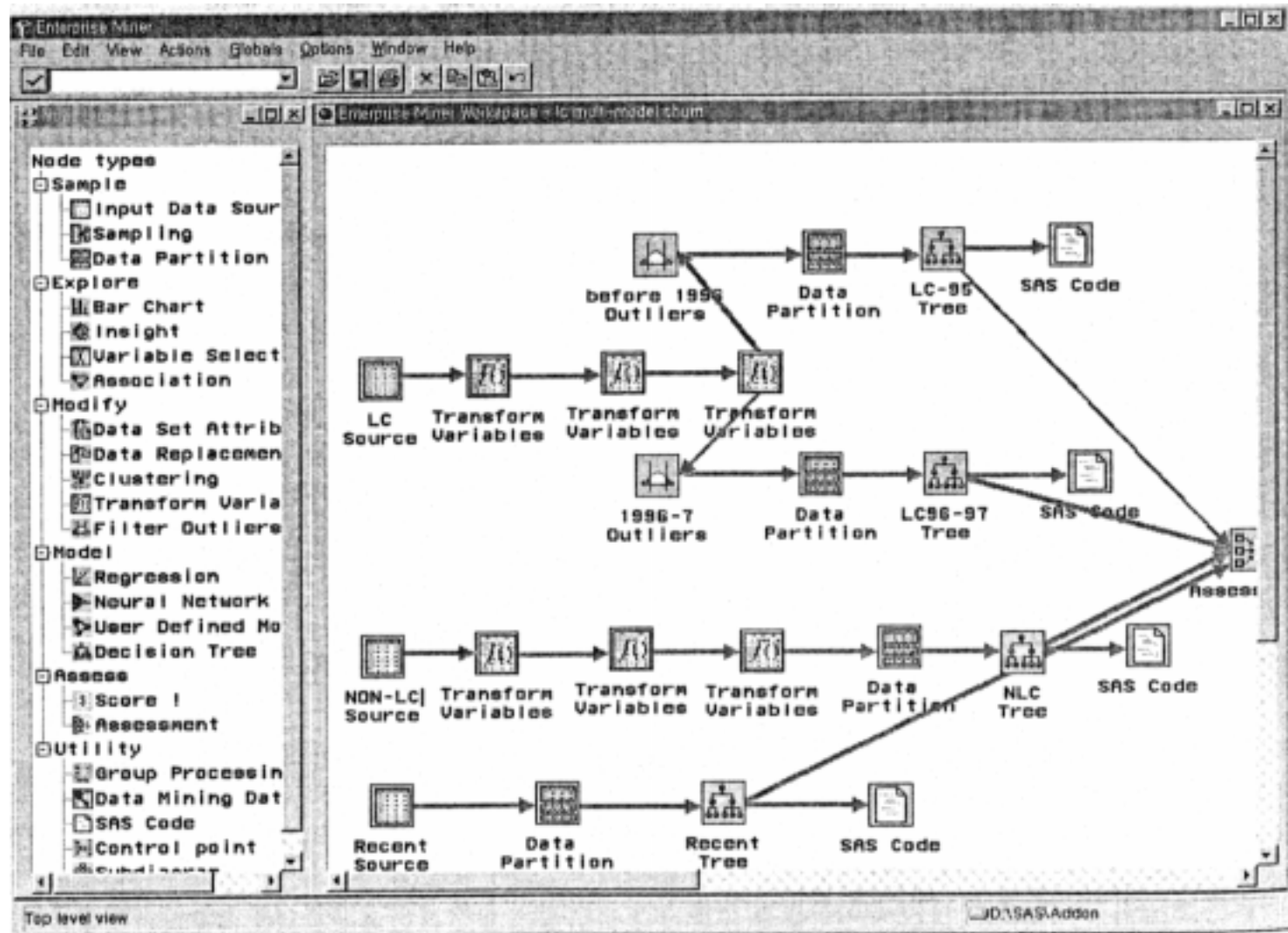
* EM llama validation a test y viceversa. En realidad son dos fases del test de validación.

Sistemas

EJEMPLO:

SAS
ENTERPRISE
MINER (EM)

(flujo del
proceso KDD)

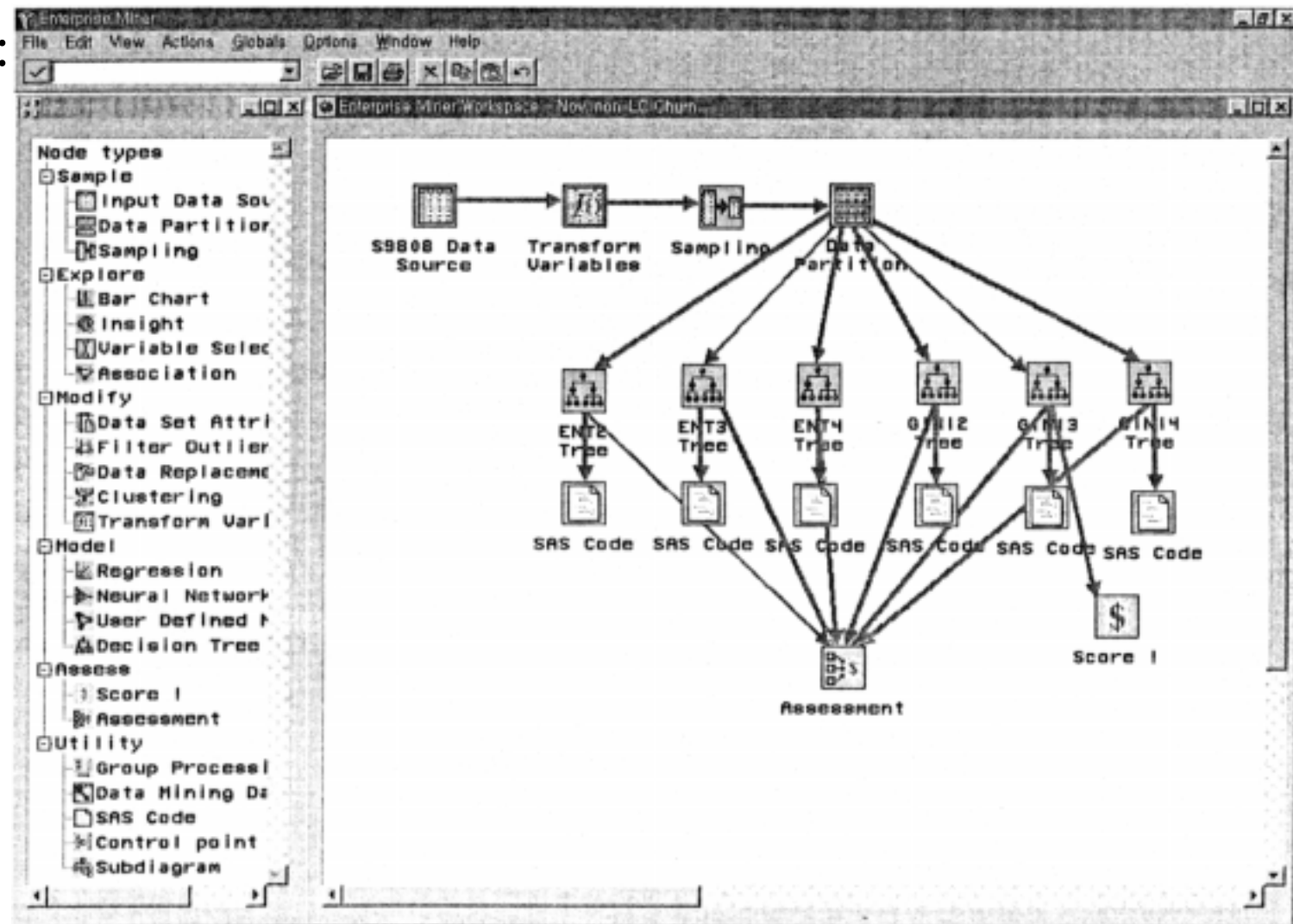


Sistemas

EJEMPLO:

SAS
ENTERPRISE
MINER (EM)

Selección
(assessment)
de modelos



Sistemas

EJEMPLO: Herramientas “Business Intelligence” Oracle

http://www.oracle.com/ip/analyze/warehouse/bus_intell/index.html

Online Analytical Processing (OLAP) provides business intelligence by analyzing data in a way that supplies insight into an enterprise, its customers, suppliers, and markets. Oracle business intelligence tools and Express OLAP products are unique in providing a single, integrated software architecture to support strategic decision making. These products include:

- Oracle Express Server, a powerful and extensible calculation engine for creating derived data values.
- Sales Analyzer and Financial Analyzer, end-user applications allowing data access, calculation, and information sharing.
- Oracle Express Objects and Oracle Express Analyzer, tools supplying a visual, point-and-click environment for building OLAP applications capable of modeling, graphical display, forecasting, statistical analysis, communication, database management, and data acquisition.
- Oracle Discoverer and Oracle Reports, query and reporting tools that integrate seamlessly with Express technology and enable users across the enterprise to deliver fast answers to hard questions.

Sistemas

EJEMPLO: Herramientas “Data Mining” Oracle

<http://www.oracle.com/ip/analyze/warehouse/datamining/index.html>

Oracle Data Mining Suite (**Oracle Darwin**) is powerful enterprise data mining software that finds meaningful patterns hidden within corporate and e-business data — patterns that can provide the insight needed for highly personalized and profitable customer relationships. Oracle Data Mining Suite enables 1:1 marketing by segmenting customers and predicting their behavior.

With Oracle Data Mining Suite, you can forge strategies to:

- Profile customers with greater accuracy
- Identify your most profitable customers
- Acquire new customers
- Prevent customer attrition
- Enable cross-selling
- Detect fraud

Oracle Data Mining Suite enhances Oracle's E-Business Suite customer relationship management software and Oracle Business Intelligence Tools by generating new customer insights. Oracle Data Mining Suite integrates with your existing enterprise systems to rapidly transform business insights into profitable action.

Oracle Data Mining Suite's key differentiators are enterprise scalability, comprehensive modeling, and ease of use.

Sistemas

EJEMPLO: Herramientas “Data Mining” Oracle

<http://www.oracle.com/ip/analyze/warehouse/datamining/index.html>

Oracle Darwin: Enterprise Scalability

- Fully Scalable and Very Large-Database Capable
Oracle Data Mining Suite imposes no limit on the amount of data to be mined, so you can exploit all available computer and data resources for maximum benefit.
- Fast Data Mining
Oracle Data Mining Suite utilizes parallel implementations of data mining algorithms to tackle the data mining process in parallel, yielding rapid information discovery. By mining more data faster, superior modeling results are delivered in record time.
- Powerful **Scripting Capabilities**
Analysts can easily record, rerun, and automate common procedures.
- Deployable Business Models
With a single click, **Oracle Data Mining Suite generates models in C, C++, or Java code** for easy integration with customer "touchpoints" across the enterprise — such as call center, campaign management, and Web-based applications.

Sistemas

EJEMPLO: Herramientas “Data Mining” Oracle

<http://www.oracle.com/ip/analyze/warehouse/datamining/index.html>

Oracle Darwin: Comprehensive Modeling

- Fast Data Access
Oracle Data Mining Suite's one-click data-import wizards access Oracle databases and data warehouses via a direct interface (OCI) for rapid import and export of data.
- Multi-Algorithmic Approach
A comprehensive array of techniques increases modeling accuracy. Oracle Data Mining Suite features **parallel implementations of classification and regression trees, neural networks, k-nearest neighbors (memory-based reasoning), regression, and clustering algorithms.**

Sistemas

EJEMPLO: Herramientas “Data Mining” Oracle

<http://www.oracle.com/ip/analyze/warehouse/datamining/index.html>

Oracle Darwin: Ease of Use

- Intuitive Windows GUI
Oracle Data Mining Suite combines familiar Windows ease of use with the power of a fully scalable, UNIX server-based solution.
- Model Seeker, Key Fields, and Modeling Wizards
Easy-to-use wizards automate the data mining process, while providing expert users with full control over all advanced options.

Sistemas

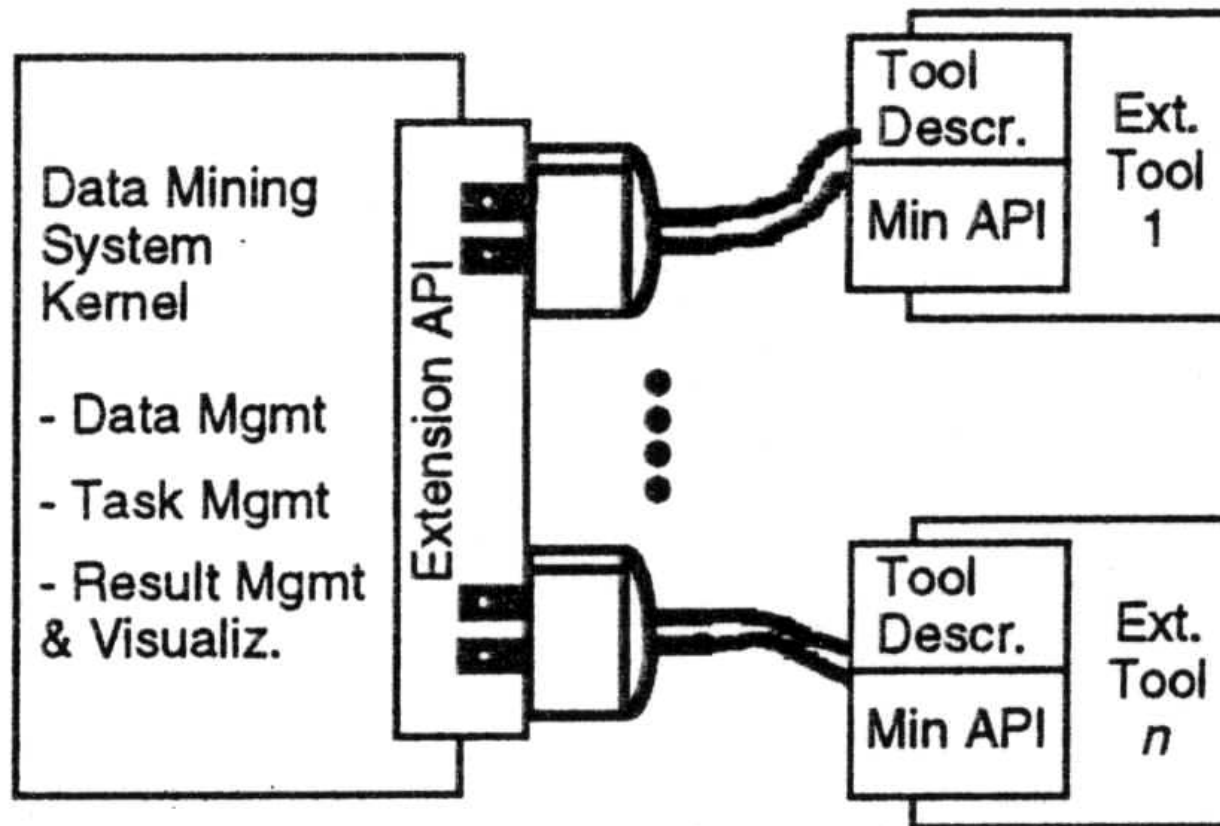
EJEMPLO: Tecnología PLUG-INS (Kepler, <http://ais.gmd.de/KD/kepler.html>)
Permite que el sistema sea extensible a nuevos algoritmos de aprendizaje.

Para ello el sistema (Wrobel et al. 1999) requiere:

- Una API de extensión bien definida y abierta a través de la cual las extensiones (plug-ins) acceden a los datos y comunican los resultados de vuelta al sistema.
- Una descripción declarativa de la herramienta de extensión que contenga información sobre los datos aceptados y necesarios por una herramienta y el tipo de salida producida.
- Un gestor de tareas y resultados para permitir un acceso uniforme (especificación, manipulación y visualización).
- Una API mínima de la herramienta, que el kernel utiliza para realizar funciones específicas de la herramienta.

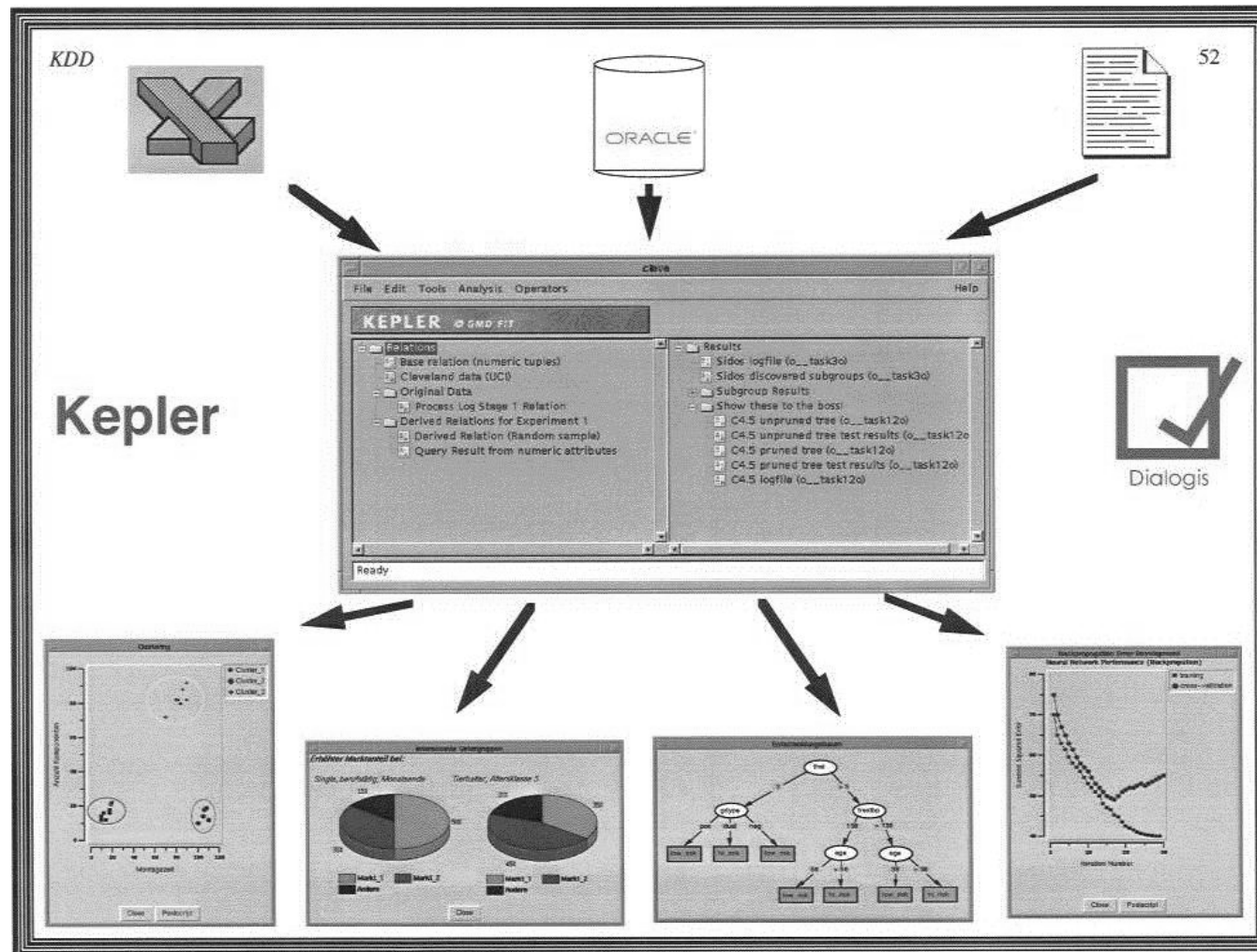
Sistemas

EJEMPLO: Kepler (cont.)



Sistemas

EJEMPLO: Kepler (cont.)



Sistemas

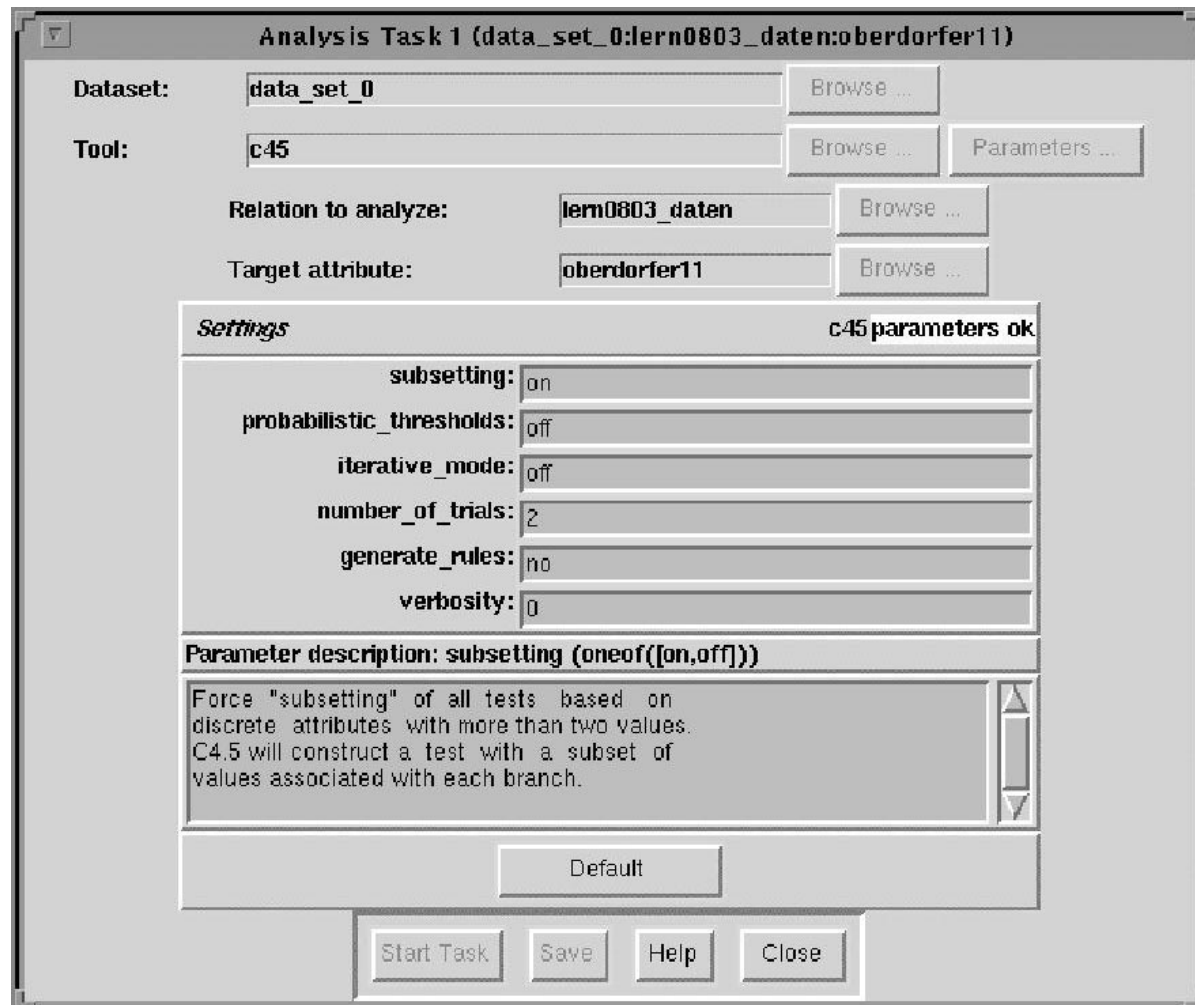
EJEMPLO: Kepler (cont.). Ejemplo de interfaz de plug-in:

Example: C4.5 DTD and TAPI entries

```
tool_description(c45,'C4.5 by J. Ross Quinlan',<HelpMessage>).
tool_result(dtree-pruned,'Decision Tree','Unpruned',<HelpMessage>).
tool_result(dtree-unpruned,'Decision Tree','Pruned',<HelpMessage>).
tool_result(logfile-c45log,'C4.5 Logfile','Log File',<HelpMessage>).
tool_result(binfile-avrules,'Attribute/Value Rules','Binary File',<HelpMessage>).
tool_result(logfile-avrules,'Attribute/Value Rules','Log File',<HelpMessage>).
tool_specific_task_data([[ 'Relation to analyze',one(target_relation)],
                        [ 'Target attribute',one(target_attribute)],
                        [ 'Attributes to be ignored',list_of(background_attribute)]]).
param(subsetting,oneof([on,off]),off,<HelpMessage>, basic).
param ...
%%% Tool API
tool_gen_task_shortcode(TaskS,Atom):- ...
learn(TaskS,ProcessType,ProcessInfo):- ...
tool_retrieve_results(TaskS,c45,ProcessInfo):- ...
```

Sistemas

EJEMPLO: Kepler (cont.). Ejemplo de interfaz de plug-in:



Sistemas

EJEMPLO: Kepler (cont.)

Algunos plug-ins integrados:

- Clasificación: el C4.5 (Quinlan 1993), un algoritmo de backpropagation utilizando SNSS (Zell 1994), un método kNN (Wettschereck 1994) y NGE (Salzberg 1991, Wettschereck & Dietterich 1995).
- Clustering: el algoritmo AUTOCLASS (Cheeseman & Stutz 1996).
- Regresión: MARS (Multiple Adaptive Regression Spline) de (Friedman 1991).
- Algoritmo de descubrimiento de patrones (reglas) EXPLORA (Klösgen 1996)

Sistemas

EJEMPLO: Kepler (cont.)

Esto está en la línea de las denominadas “Core DM APIs”, que deberían permitir una serie de llamadas de interrelación entre los algoritmos de aprendizaje y la base o almacén de datos.

- SQL (consultas de extensión), pero además...
- Comprobación de hipótesis (cálculos de cobertura).
- Carga de datos (muestreos, sobremuestreos).
- Operadores de navegación de copo de nieve (drill down, drill up, slice & dice)

Esto permitiría el uso de algoritmos no batch, es decir, algoritmos incrementales, interactivos, bajo el paradigma Query Learning.

Retos para la Minería de Datos

- Escalabilidad:
 - esquemas de muestreo eficientes y suficientes.
 - procesamiento en memoria vs. en disco.
 - combinación de recursos entre tareas involucradas.
 - interfaces con los almacenes de datos.
 - uso de metadata para optimizar el acceso.
 - cuestiones cliente/servidor (dónde hacer el procesamiento).
 - aprovechamiento de paralelismo y de computación distribuida.

Retos para la Minería de Datos

- Automatización:
 - Desarrollo de asistentes (wizards) y/o lenguajes de consulta:
 - para definir la tarea de minería, entradas, salidas, ...
 - seleccionar y utilizar el conocimiento previo.
 - Transformación de los datos y reducción de dimensionalidad.
 - Compromiso entre simplicidad y precisión de los modelos en pro de una mayor inteligibilidad.

Retos para la Minería de Datos

- Otros Retos:
 - Tratamiento de datos cambiantes: necesidad de revisión y extensión de patrones (incrementalidad).
 - Minería de datos con tipos no-estándar (no numérico o no textual, p.ej. gráficos vectoriales, índices a ficheros, hiperenlaces), multimedia u orientados a objetos.

Tendencias

- 80s y principios 90s:
 - OLAP: consultas predefinidas. El sistema OLAP como sistema para extraer gráficas y confirmar hipótesis. Técnicas fundamentalmente **estadísticas**.
 - Se usa exclusivamente información interna a la organización.
- Finales de los 90
 - Data-Mining: descubrimiento de patrones. Técnicas de **aprendizaje automático** para generar patrones novedosos.
 - El Data-Warehouse incluye Información Interna fundamentalmente.
- Principios de los 00
 - Técnicas de “scoring” y simulación: descubrimiento y uso de modelos globales. Estimación a partir de variables de entrada de variables de salida (causa-efecto) utilizando simulación sobre el modelo aprendido.
 - El Data-Warehouse incluye Información Interna y Externa (parámetros de la economía, poblacionales, geográficos, etc.).

Direcciones:

Recursos Generales:

- KDcentral (www.kdcentral.com)
- The Data Mine (<http://www.cs.bham.ac.uk/~anp/TheDataMine.html>)
- Knowledge Discovery Mine (<http://www.kdnuggets.com>)

Mailing list:

- KDD-nuggets: moderada y con poco ruido:
Para suscribirse, enviar un mensaje a “kdd-request@gte.com” con “subscribe kdnuggets” en la primera línea del mensaje (el resto en blanco).

Revistas:

- Data Mining and Knowledge Discovery. (<http://www.research.microsoft.com/>)
- Intelligent Data Analysis (<http://www.elsevier.com/locate/ida>)

Asociaciones:

- ACM SIGDD (y la revista “explorations”,
<http://www.acm.org/sigkdd/explorations/instructions.htm>)

Referencias del Tema

- (Agrawal et al. 1996) Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; Verkamo, A.I. “Fast Discovery of Association Rules”, in Fayyad, U.M.; Piatetsky-Shapiro, G.; Smyth, P. *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press, 1996, pp. 307-328.
- (Akutsu & Takasu 1994) Akutsu, T.; Takasu, A. “On PAC Learnability of Functional Dependencies” *New Generation Computing*, 12 (1994), 359-374.
- (Blockeel & De Raedt 1996) Blockeel, H. and De Raedt, L. “Inductive database design”, Proceedings of Foundations of Intelligent Systems, *Proc. of the 9th International Symposium on Methodologies for Intelligent Systems*, Lecture Notes in Artificial Intelligence, Vol. 1079, 1996, pp. 376-385.
- (Blockeel & De Raedt 1998) Blockeel, H. and De Raedt, L. “IsIdd: An interactive system for inductive database design”, *Appl. Artif. Intell.* 12 (1998), no. 5, 385-421.
- (Brockhausen & Morik 1997) Brockhausen, P.; Morik, K. “A multistrategy approach to relational knowledge discovery in databases” *Machine Learning Journal*, 1997.
- (Cheeseman & Stutz 1996) Chessemann, P.; Stutz, J. “Bayesian Classification (AutoClass). Theory and Results”, chap. 6 of Fayyad, U.M.; Piatetsky-Shapiro, G.; Smyth, P. *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press, 1996.
- (Cheung et al. 1996) Cheung, D.; Han, J.; Ng, V.; Fu, A.W.; Fu, A.Y. “A Fast and Distributed Algorithm for Mining Association Rules” in *Proc. of International Conference on Parallel and Distributed Information Systems*, PDIS 1996.
- (Flach & Savnik 1999) Flach, P.A. and Savnik, I. “Database dependency discovery: a machine learning approach”. *AI Communications*, 12(3):139--160, November 1999.

Referencias del Tema

- (Friedman 1991) Friedman, J. "Multivariate adaptive regression splines (with discussion)" *Annals of Statistics*, 19(1), 1-141.
- (Han et al. 1999) Han, J.; Lakshmanan, V.S.; Ng, R.T. "Constraint-Based, Multidimensional Data Mining" *Computer*, Vol. 32, nº8, 1999.
- (Hipp et al. 1999) Hipp, J.; Güntzer, U.; Gholamreza, N. "Algorithms for Association Rule Mining - A General Survey and Comparison" *ACM SIGKDD Explorations*, vol 2, issue 1, 2000, pp. 58-64.
- (Hsu & Knoblock 1996) Hsu, C-N.; Knoblock, C.A. "Using Inductive Learning to Generate Rules for Semantic Query Optimization" in Fayyad, U.M.; Piatetsky-Shapiro, G.; Smyth, P. *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press, 425-445, 1996
- (Imielinski and Manilla 1996) Imielinski, T.; Mannila, H. "A database perspective on knowledge discovery" *Communications of the ACM*, 39(11):58-64, 1996.
- (Imielinski et al. 1996) Imielinski, T.; Virmani, A., and Abdulghani, A. "Discovery board application programming interface and query language for database mining" in *Proceedings of Knowledge Discovery in Databases Conference (KDD'96)*, Portland, Ore, Aug, 1996, pp. 20-26.
- (Inselberg & Duimsdale 1990) Inselberg, A.; Dimsdale, B. "Parallel Coordinates: A Tool for Visualizing Multidimensional Geometry" *Proc. Visualization 1990*, IEEE CS Press, Los Alamitos, Calif. 1990, pp. 361-370
- (Kiel 2000) Kiel, J.M. "Data Mining and Modeling: Power Tools for Physician Practices", *MD Computing*, 33-34, May/June 2000.

Referencias del Tema

- (Klösgen 1996) Klösgen, W. "Explora: A Multipattern and Multistrategy Discovery Assistant" in Fayyad, U.M.; Piatetsky-Shapiro, G.; Smyth, P. *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press, 1996, pp. 249-271.
- (Lin and Dunham 1998) Lin, J.; Dunham, M.H. "Mining Association Rules" in *Proc. of the IEEE International Conference on Data Engineering*, ICDE 1998.
- (Mannila & Räihä 1994) Mannila, H. & Räihä, K. "Algorithms for inferring functional dependencies from relations" *Data Knowledge Engineering*, 12, 83-99, 1994.
- (Mehta, Agrawal and Rissanen 1996) Mehta, M.; Agrawal, R.; Rissanen, J. "SLIQ: A Fast Scalable Classifier for Data Mining". EDBT 1996: 18-32
- (Ng et al. 1998) Ng et al. "Exploratory Mining and Pruning Optimizations of Constrained Association Rules" *Proc. ACM SIGMOD Int'l Conf. Management of Data*, ACM Press, New York, 1998, pp. 13-24.
- (De Raedt 1998) De Raedt, L. "An inductive logic programming query language for database mining", *Proceedings of the 4th International Conference on Artificial Intelligence and Symbolic Computation* (J. Calmet and J. Plaza, eds.), LNAI, vol. 1476, Springer, 1998, pp. 1-13.
- (De Raedt & Dehaspe 1997) De Raedt, L.; Dehaspe, L. "Clausal discovery" *Mach. Learning* 26, 1997, 99-146.
- (Dehaspe and de Raedt 1997b) Dehaspe, L.; de Raedt, L. "Mining association rules in multiple relations" in *Proc. of the 7th International Workshop on Inductive Logic Programming*, volume 1297 of *Lecture Notes in Computer Science*, Springer, pp. 125-132, Prague, Czech Republic, 1997.
- (Robertson et al. 1991) Robertson, G.; Card, S.; Mackinlay, J. "Cone Trees: Animated 3D Visualisations of Hierarchical Information", *Proc. ACM CHI Intl. Conf. On Human Factors in Computing*, ACM Press, New York, 1991, pp. 189-194.

Referencias del Tema

- (Salzberg 1991) Salzberg, S. "A Nearest Hyperrectangle Learning Method" *Mach. Learning* 6:277-309, 1991.
- (Savasere et al 1995) Savasere, A.; Omiecinski, E.; Navathe, S. "An Efficient Algorithm for Mining Association Rules" in *Proc. of the International Conference on Very Large Databases, VLDB*, Morgan Kaufmann, 1995.
- (Shen et al. 1996) Shen, W.M.; Ong, K.; Mitbender, B.; Zaniolo, C. "Metaqueries for Data Mining" in *Advances in Knowledge Discovery and Data Mining*, U.M.Fayyad et al. (eds.) AAAI, 1996, pp. 375-398.
- (Volokh 2000) Volokh, E. "Personalization and Privacy" *Communications of the ACM*, August 2000, Vol. 43, No.8, pp. 84-88.
- (Wettscherech 1994) Wettschereck, D. *A Study of distance-Based Machine Learning Algorithms*, Ph.D. dissertation, Oregon State University, 1994.
- (Wettscherech & Dietterich 1995) Wettschereck, D.; Dietterich, T. "An experimental comparison of the nearest-neighbor and nearest hyperrectangle algorithms" *Machine Learning*, 19:5-28, 1995.
- (Wong 1999) Wong, P.C. "Visual Data Mining", Special Issue of *IEEE Computer Graphics and Applications*, Sep/Oct 1999, pp. 20-46.
- (Wrobel 1996) Wrobel, S; Wettschereck, D.; Sommer, E.; Emde, W. Extensibility in Data Mining Systems. *KDD* 1996: 214-219
- (Wrobel et al. 1999) Wrobel, S., Andrienko, G., Andrienko, N., and Savinov, A. "Kepler and Descartes" in Kloesgen, W. and Zytkow, J. (eds.) *Handbook of Data Mining and Knowledge Discovery* NY: Oxford University Press, 1999.
- (Zell 1994) Zell, A.E. "SNNS user manual 3.2", Fakultätsbericht 6/94, IPVR, Univ. Stuttgart, Germany, 1994.