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

T.3 Extracción de Conocimiento a Partir de Información No Estructurada

Mª José Ramírez-Quintana

Master: Ingeniería del Software, Métodos Formales y Sistemas de Información

Objetivos

- Conocer las características especiales de la extracción automática de conocimiento desde otras fuentes de información no estructurada (textos y *web*) y semiestructurada (XML).

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

- Conocer herramientas para intercambiar conocimiento.

- Conocer técnicas para extraer patrones de navegación y asistentes Web para personalización/recomendación.

Temario

3.1. Los Problemas de la Extracción de Conocimiento de Información No Estructurada.
3.2. Extracción de Conocimiento a partir de Documentos No Estructurados (Web Content Mining).
3.3. XML y DTDs. Extracción de Conocimiento a partir de Información Semi-Estructurada (XML).
3.4. Lenguajes de consulta e Intercambio de Conocimiento (PMML y RuleML).
3.5. Extracción de Conocimiento a partir de la Estructura (Web Structure Mining).
3.6. Extracción de Conocimiento a partir de Patrones de Uso (Web Usage Mining).
3.7. Personalización, Recomendación y Asistentes (Web) 'Inteligentes'.

Características Especiales de la Extracción de Conocimiento de Info. No Estructurada

- Información No Estructurada
  - Atributos no se conocen a priori.
  - Los datos son secuencias de símbolos. En el mejor caso, se estructuran como un árbol, y generalmente son grafos dirigidos: páginas = nodos, hiperenlaces = ejes.
- Características habituales:
  - Grandes volúmenes de datos.
  - Mucho ruido, información faltante, mucha información irrelevante.
  - El tiempo de aprendizaje admisible suele ser grande (horas o días).
  - El tiempo de respuesta de las hipótesis debe ser muy rápido (ms.)
- *La inteligibilidad de los modelos es secundaria.*

Métodos No Apropiados

- A priori (sin una profunda transformación de los datos), muchas técnicas de aprendizaje automático son inútiles para muchas aplicaciones:
  - Métodos de clasificación (árboles de decisión, fence & fill, ...): están basados en una clase dependiente de un número de atributos predeterminados (exceptuando Naive Bayes).
  - Métodos numéricos (regresión, redes neuronales, ...): los datos son simbólicos, no numéricos.
  - Métodos lazy (kNN, CBR, ...): tiempos de respuesta serían muy altos.

Métodos Apropiados

- No estructurada:
  - Métodos Bayesianos.
  - Otros métodos estadísticos: (Rochio 1971) (Salton 1991)
  - ILP: sólo si el volumen de datos es pequeño o se hace una buena recodificación.

- Semiestructurada:
  - Gramaticales (autómatas).
  - Métodos (Lógico) Funcionales (p.ej. IFLP): sólo si el volumen de datos es pequeño o se hace una buena recodificación.

Web Mining

Web Mining se refiere al proceso global de descubrir información o conocimiento potencialmente útil y previamente desconocido a partir de datos de la Web. (Etzioni 1996)

Ejemplos:

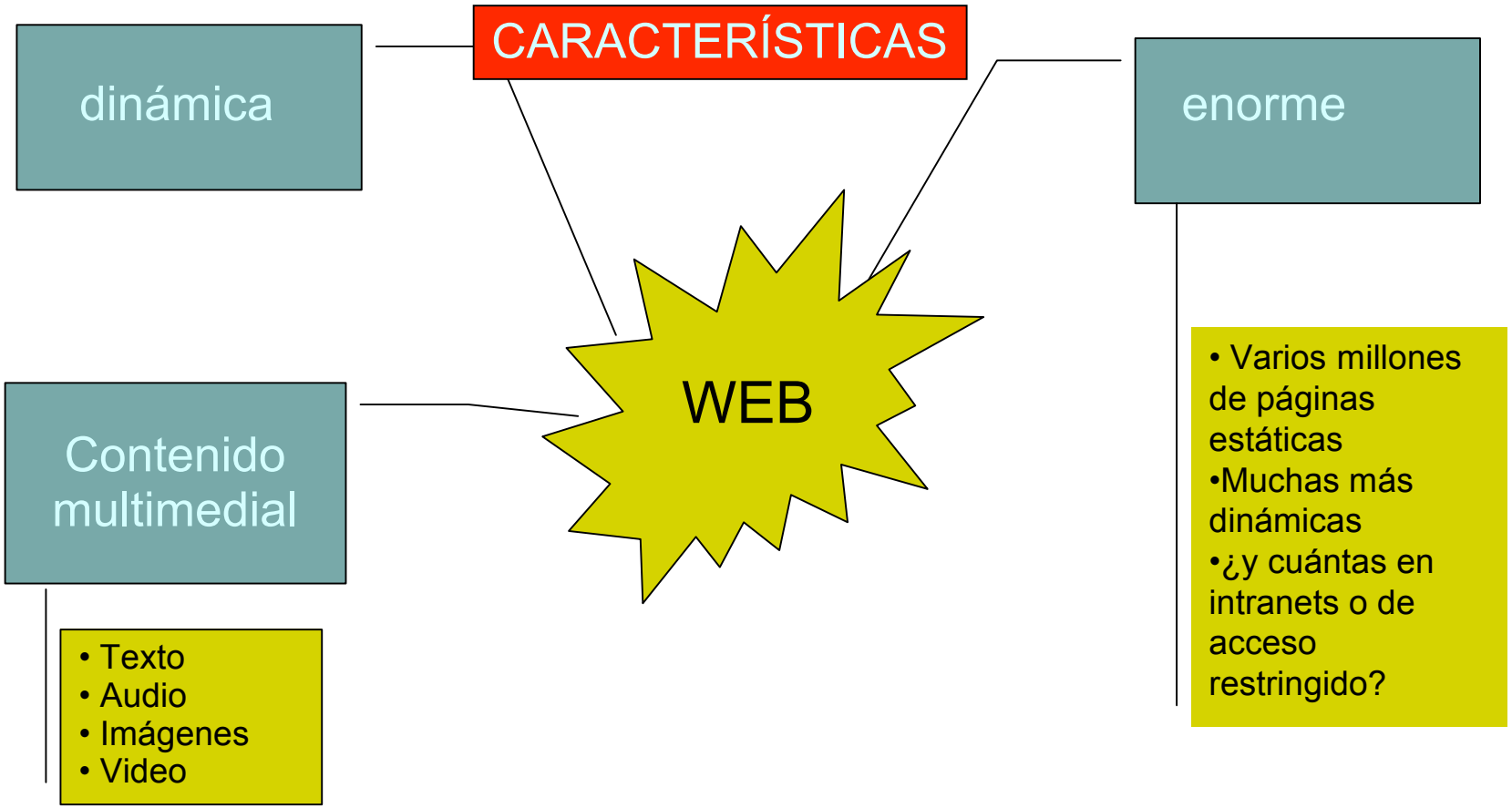
- Búsqueda en la Web, ej. Google, Yahoo, MSN, Ask, …
- Búsqueda especializada: ej. Froogle (comparativa de precios), bolsas de trabajo (Monster, Flipdog)
- eComercio: ej. MediaXpress, Amazon
- Publicidad, e.g. Google Adsense
- Mejorar el diseño y las prestaciones de los sitios Web

Web Mining

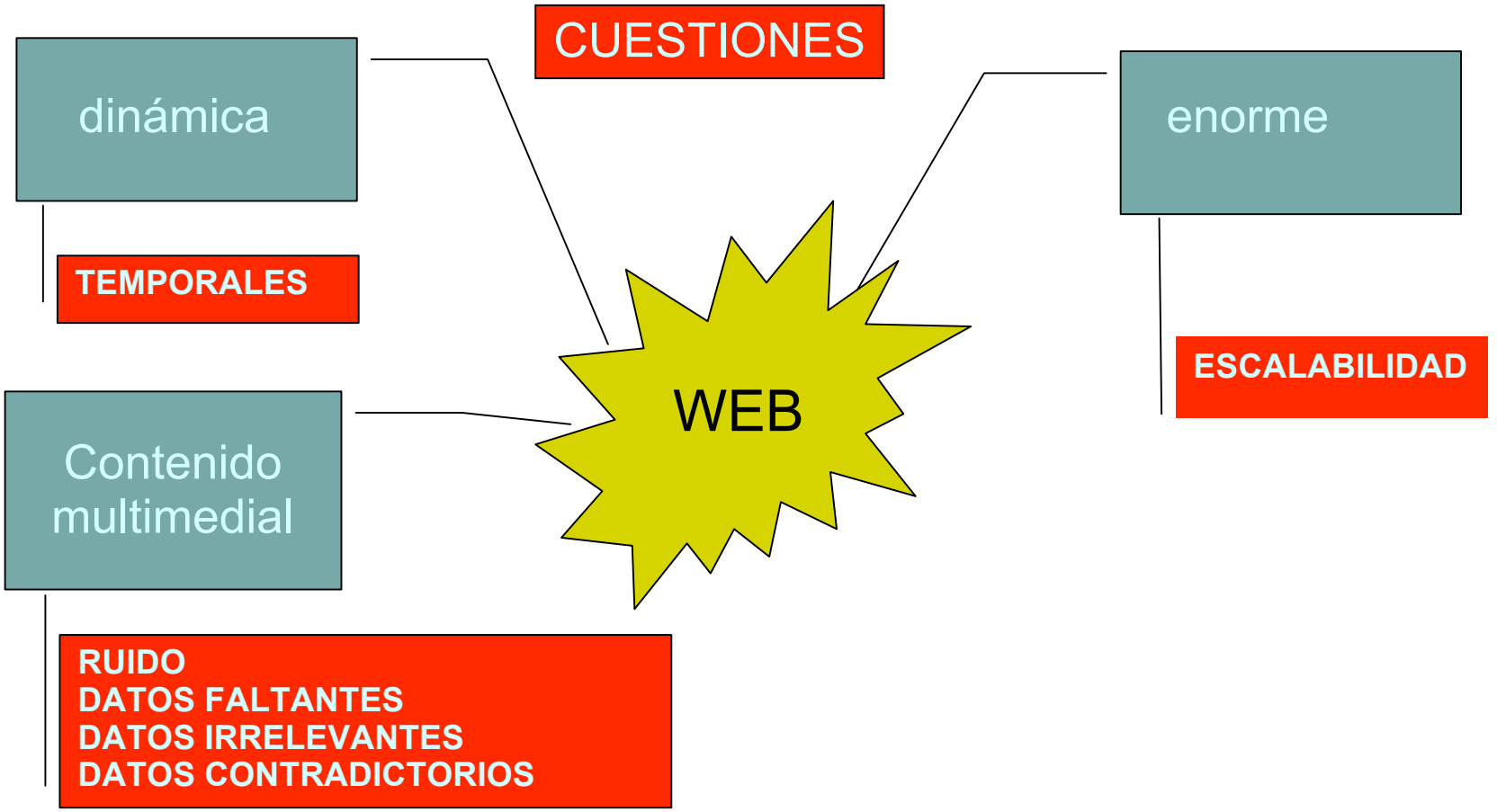
Web Mining combina objetivos y técnicas de distintas áreas:

- Information Retrieval (IR)
- Natural Language Processing (NLP)
- Data Mining (DM)
- Databases (DB)
- WWW research
- Agent Technology

Web Mining


CARACTERÍSTICAS
dinámica
enorme
WEB
Contenido multimedial
• Texto
• Audio
• Imágenes
• Video
• Varios millones de páginas estáticas
•Muchas más dinámicas
•¿y cuántas en intranets o de acceso restringido?

Web Mining


CUESTIONES
dinámica
TEMPORALES
enorme
ESCALABILIDAD
WEB
Contenido multimedial
RUIDO
DATOS FALTANTES
DATOS IRRELEVANTES
DATOS CONTRADICTORIOS

Web Mining

¿Está la Información de la Web lo suficientemente estructurada para facilitar minería de datos efectiva? (Etzioni 1996)

Web Mining se puede estructurar en fases (Kosala & Blockeel 2000):

- **Descubrimiento de recursos**: localización de documentos relevantes o no usuales en la red. Ésta es la función de índices buscadores (extraen contenido en palabras, zona del documento, idioma) e índices temáticos (clasifican los documentos).
- **Extracción de información**: extraer determinada información a partir de un documento, ya sea HTML, XML, texto, ps, PDF, LaTeX, FAQs, ...
- **Generalización**: descubrir patrones generales a partir de sitios web individuales: *clustering*, asociaciones entre documentos.
- **Análisis, validación e interpretación** de los patrones.

Web Mining

Objetivos

- Búsqueda de Información Relevante o Relacionada.

  *Web Mining como Information **Retrieval** (IR)* (query-triggered)

  - Es el objetivo de numerosas herramientas: buscadores e índices.
  - *Las herramientas son clásicas estadísticas y ad-hoc....*

- Creación de Nueva Información a partir de información existente (resúmenes, listas, ...).

  *Web Mining como Information **Extraction** (IE)* (data-triggered)

  - Es una visión próxima a la Minería de Datos.
  - *Las herramientas son más generales y de aprendizaje automático.*

IR persigue seleccionar documentos relevantes mientras que IE persigue extraer hechos relevantes a partir de los documentos. (Kosala & Blockeel 2000)

- Personalización de la Información.
- Aprendizaje a partir de los usuarios, visitantes o consumidores.

Web Mining

No sólo se requiere **información relevante** sino información de **calidad** o autorizada. Para ello es importantísimo no analizar los documentos de forma inconexa, sino analizar su red de interconexiones (sus **enlaces**)

Mucha información está en **enlaces entrantes**: muchas páginas no se autodescriben. P.ej. una página puede ser clasificada por los enlaces que le llegan (referentes), que suelen ir acompañados de una pequeña descripción de la página o junto a otros enlaces similares (*clustering*).

También (no tanta) información sobre la página se encuentra en **enlaces salientes**.

Otra fuente de información son los datos almacenados en los **logs de los servidores Web**.

Clasificación del Web Mining

Clasificación no disjunta (Kosala & Blockeel 2000):

- **Web Content Mining**: extraer información del contenido de los documentos en la web. Se puede clasificar a su vez en:
  - Text Mining: si los documentos son textuales (planos).
  - Hypertext Mining: si los documentos contienen enlaces a otros documentos o a sí mismos.
  - Markup Mining: si los docs. son semiestructurados (con marcas).
  - Multimedia Mining: para imágenes, audio, vídeo, ...
- **Web Structure Mining** (Chakrabarti et al. 1999): se intenta descubrir un modelo a partir de la topología de enlaces de la red. Este modelo puede ser útil para clasificar o agrupar documentos
- **Web Usage Mining** (Cooley et al. 1997): se intenta extraer información (hábitos, preferencias, etc. de los usuarios o contenidos y relevancia de documentos) a partir de las sesiones y comportamientos de los usuarios y navegantes.

Web Content Mining

Web Content Mining:

Hay aplicaciones diferentes: *categorización/clasificación/agrupamiento de texto*.

- Asistir, mejorar o filtrar la información que proporcionan los buscadores posiblemente usando el perfil del usuario

  (Visión de Recuperación de la Información: Minería de Textos, Hipertextos)

- Modelar e integrar los datos encontrados en la web para permitir peguntas más sofisticadas que las basadas en palabras clave. Extraer esquemas o *DataGuides*.

  (Visión de Bases de Datos: Minería de Marcado)

Web Content Mining

Web Content Mining:

Las técnicas varían dependiendo del tipo de documento:

- *Text Mining*: técnicas de recuperación de información (IR) fundamentalmente. Técnicas estadísticas y lingüísticas.
- *Hypertext Mining*: no sólo se refiere a enlaces entre documentos sino también intro-documentos (OEM). Se ha de construir un grafo de referencias...
- *Markup Mining*: La información de las marcas contiene información (HTML: secciones, tablas, negritas: relevancia, cursiva, etc., XML: mucho más...).
- *Multimedia Mining*: algunos trabajos sobre librerías de imágenes. El resto muy verde. Para una referencia (Zaiane et al. 1998).

Text Mining

Web Content Mining. Text Mining: cientos o miles de palabras...

- Existen varias aproximaciones a la representación de la información (Hearst and Hirsh 1996):
  - "Bag of Words": cada palabra constituye una posición de un vector y el valor puede referirse a frecuencias (nº de veces que ha aparecido) o puede ser booleano (si ocurre o no en el documento). **Ignora la secuencia de ocurrencia de una palabra**
  - N-gramas o frases: considera frases o partes de frases (similar al procesamiento del lenguaje natural) permite tener en cuenta el orden de las palabras. **Trata mejor frases negativas "... *excepto* ...", "... *pero no*...", que tomarían en otro caso las palabras que le siguen como relevantes.**

Text Mining

- Representación relacional (primer orden): permite detectar patrones más complejos (si la palabra *X* está a la izquierda de la palabra *Y* en la misma frase...). La subárea de ILP (*Inductive Logic Programming*) denominada LLL *(Learning Language in Logic)*. **Por ejemplo, w_i(d,p) representa que la palabra w_i aparece en el documento d en la posición p.**

- Categorías de conceptos (Indexado Semántico Latente): reducir la dimensión del vector, reduciendo las palabras a su raíz morfológica **(palabras que pueden tener la misma raíz pero no pertenecer a la misma familia: inform-al e inform-ática)**.

Casi todas las representaciones se enfrentan con el "vocabulary problem" (Furnas et al. 1987). **Tienen problemas con la sinonimia (empezar, comenzar) y quasi-sinonimia (comunicado, declaración) , la polisemia (bomba), los lemas (descubrir, descubrimiento), etc.**

Text Mining

- Generalmente, se aplican distintas técnicas para reducir la dimensionalidad (Apte et al. 1994):
  - Reducción global del conjunto de características: se construye un diccionario (global) eliminando números, nombres propios y separadores. Se eliminan las palabras que no ocurran más de n veces. Adicionalmente se aplican técnicas estadísticas de selección de carácterísticas para cada categoría.
  - Reducción local (orientada a la categoría): se crean diccionarios independientes para cada categoría y se usan como carácterísticas las n palabras más frecuentes.

Clasificando Texto con NB Classifier

Ejemplo:

- Consideremos documentos de texto o de hipertexto T, que se pueden clasificar en varias clases, p.ej. (interesante, no-interesante) o en diferentes temas.

- Definimos un atributo a_i como cada posición i de cada palabra en el texto. Por ejemplo, dado este párrafo, tendríamos 40 atributos, donde el valor t_i para el primer atributo sería “Definimos”, el valor para el segundo sería “un”, etc.

Clasificando Texto con NB Classifier

- Debemos hacer la siguiente suposición FALSA:

  la probabilidad de una palabra es independiente de sus precedentes y sus siguientes.

  Es decir, los $P(a_i|v_j)$ son *independientes entre sí.*
- Falso, por ejemplo, con las palabras "por" y "ejemplo".

- Pero este supuesto no es tan grave en general y nos permite, como vimos, utilizar el clasificador bayesiano naïve.

$$v_{NB} = \underset{v_j \in \{sí, no\}}{\arg\max} \left\{ P(v_j) ? \prod_{i=1}^{|T|} P(a_i = t_i \mid v_j) \right\}$$

Clasificando Texto con NB Classifier

- A primera vista, parece que:
  - $P(v_j)$ es fácil de determinar (la proporción de documentos de cada clase).
  - $P(a_i=w_k|v_j)$, sin embargo, requeriría millones de casos para tener una estimación.

Otra suposición (más razonable):

las probabilidades son independientes de la posición

Quitando las primeras y últimas, la probabilidad que una palabra aparezca en la posición 45 es la misma que en la 87.

Esto quiere decir que:

$$P(a_i = w_k \mid v_j) = P(a_m = w_k \mid v_j) \text{ for all } i, j, k, m$$

Lo que supone estimar únicamente las $P(w_k|v_j)$. Es decir, la probabilidad de aparición de cada palabra según la clase.

Clasificando Texto con NB Classifier

- Volvemos a adoptar un *m*-estimado:

$$P(w_k \mid v_j) = \frac{n_k + mp}{n + m} = \frac{n_k + |Voc| \frac{1}{|Voc|}}{n + |Voc|} = \frac{n_k + 1}{n + |Voc|}$$

Donde n es el número total de posiciones de palabras en los ejemplos de entrenamiento (documentos) donde la clase es v_j. En otras palabra, la suma de las longitudes (en nº de palabras) de todos los documentos de la clase v_j.

Donde n_k es el número de veces que se ha encontrado en estos documentos de la clase v_j.

|Voc| es el número de palabras del lenguaje considerado (inglés, castellano, o lenguajes informáticos). Voc puede ser un subconjunto (se pueden eliminar palabras muy usuales: preposiciones, artículos, verbos muy comunes, etc.)

Clasificando Texto con NB Classifier

- Otro problema es que:
  - Todos los documentos deberían ser de la misma longitud para que fueran comparables.
  - Solución: A la hora de calcular v_{NB} (no antes) los documentos cortos se replican hasta llegar a una longitud suficiente L, y/o los documentos largos se truncan aleatoriamente (se eligen L palabras al azar)

- También se pueden eliminar palabras que aparecen muy pocas veces en los textos.

Clasificando Texto con NB Classifier

Resumen del Algoritmo:

- Input: E (examples): conjunto de documentos de texto.
  V: conjunto de clases v_1, v_2, ...,.

  1. Voc := todas las palabras (w_1, w_2, ...) y otros signos extraídos de E.
  2. Cálculo de $P(v_j)$ y $P(w_k|v_j)$
     - Para cada clase v_j en V hacer:
       - $docs_j$:= subconjunto de docs. de E de la clase v_j.
       - $P(v_j)$:= $|\, docs_j \,| \,/\, |\, E \,|$
       - $text_j$:= concatenación de todos los elementos de $docs_j$
       - n := el número total de posiciones de palabras distintas en $text_j$
       - Para cada palabra w_k en Voc
         - n_k := número de veces la palabra w_k aparece en $text_j$
         - $P(w_k \mid v_j) = \dfrac{n_k + 1}{n + |\, Voc \,|}$

Clasificando Texto con NB Classifier

Una vez aprendidas las probabilidades, simplemente se trata de aplicar el clasificador:

CLASIFY_NAIVE_BAYES_TEXT(Doc)

Retorna la clase estimada para el documento Doc

(a_i denota la palabra encontrada en la posición *i* de *Doc*)

- positions := todas las posiciones de palabra en Doc que contienen palabras que se encuentran en Voc.
- Retorna v_{NB}, donde

$$v_{NB} = \arg\max_{v_j \in V} \left\{ P(v_j) \cdot \prod_{i=positions} P(a_i \mid v_j) \right\}$$

También sirve para categorizar (en vez de coger el máximo, se puede elegir las clases que superen un límite, o las n más probables)

Clasificando Texto con NB Classifier

clasificación/categorización: se representa el documento como una bolsa/vector de palabras y se cuenta el número de aparición de cada palabra para determinar si pertenece o no a algún tópico predefinido. Se puede usar para hacer rankings.

EJEMPLOS:

- (Joachims 1996):
  - Clasifica artículos en 20 grupos de noticias.
  - A partir de 1000 artículos (mensajes) de cada uno, la precisión (accuracy) de clasificación para nuevos mensajes era del 89%, simplemente utilizando el NB Classifier.
- (Lang 1995):
  - Clasifica artículos y noticias dependiendo del interés que han creado en el usuario. Después de una etapa de entrenamiento del usuario, del 16% de artículos que eran interesantes para el usuario, se pasó a un 59% de los que el sistema recomendaba.

Text Mining

- **Agrupamiento**: similar a la clasificación pero sin determinar a priori los tópicos.

Ejemplo, el sistema WEBSOM (Lagus et al. 1998) organiza los documentos en un mapa tal que documentos próximos sean similares.

- **Reglas de asociación para conceptos**: sirve para relacionar documentos identificando conceptos compartidos o conectados.

Clasificando Texto por Conceptos

Reglas de asociación para conceptos (Loh et al. 2000)

Procedimiento: crear conceptos, obtener la frecuencia relativa de los conceptos en las páginas, buscar asociaciones entre conceptos.

- los conceptos se definen previamente
- representación: vectores de términos (sinónimos, quasi-sinónimos, plurales, derivaciones verbales, …)

Se realiza en dos fases: En el primer paso se trata de **asociar** las **palabras con** distintos **conceptos**. Con el objetivo de no utilizar análisis sintáctico del texto (técnicas lingüísticas costosas), se utiliza razonamiento difuso (fuzzy): se asocia un peso (entre 0 y 1).

Clasificando Texto por Conceptos

deporte →

1	fútbol	0.9
2	baloncesto	0.8
3	jugador	0.7
4	ejercicio	0.5
5	partido	0.4
…	…	…

navegación →

1	navegar	0.9
2	barco	0.8
3	patrón	0.2
4	timón	0.7
5	vela	0.4
6	mar	0.5
7	proa	0.8
…	…	…

Se comparan los textos con los conceptos haciendo uso de los pesos, y se calcula la probabilidad relativa de que el concepto esté presente en el texto.

Clasificando Texto por Conceptos

- En el segundo paso, se asocian conceptos con conceptos usando reglas de asociación A → B:

  “navegación” → “deporte”

  “credito” ∧ “promocion” → “compra”

  – confianza: proporción de textos que tienen A y B en relación con el número de textos que tienen sólo A
  – soporte: proporción de textos que tienen A y B en relación con el número de textos totales
- Problemas con asociaciones positivas para frases negativas (del estilo “excepto...”, “cuando no hay...”...).

Text Mining

Otras Técnicas: proceden del procesamiento del lenguaje natural

- **extracción información**: identifica frases clave y relaciones dentro del texto. Ejemplo, el sistema Harvest (extrae el autor y el título de documentos LaTeX) (Brown et al. 1994)
- **resumen**: resumir el texto sin perder significado.
  - "extracción de sentencias": en la que se seleccionan las sentencias de un artículo a partir de su peso estadístico
  - "por posición": por ejemplo, las sentencias que siguen a la expresión "en conclusión".
- ***por preguntas***: trata sobre cómo extraer información a partir de preguntas. Ejemplo, FAQ-finder busca por emparejamiento la mejor respuesta a una pregunta en ficheros de FAQs) (Hammond et al. 1995).

Clasificando textos con el método Rocchio

- Usa el *feedback* del usuario sobre la relevancia/irrelevancia del documento.
- Consiste en obtener un clasificador de la siguiente forma:
  - construir un vector prototipo c_i para cada clase i={relevante,irrelevante}

$$\mathbf{c}_i = \frac{\alpha}{|D_i|}\sum_{\mathbf{d}\in D_i}\frac{\mathbf{d}}{\|\mathbf{d}\|} - \frac{\beta}{|D - D_i|}\sum_{\mathbf{d}\in D - D_i}\frac{\mathbf{d}}{\|\mathbf{d}\|}$$

donde: α, β son parámetros, D_i el conjunto de documentos de clase i, y cada documento d es un vector de pesos tal que d_k representa el peso ω_k de la palabra t_k en d con $t_k \in Voc$.

Clasificando textos con el método Rocchio

- Esquema TF-IDF
  - TF: **term frequency**
  - IDF: **inverse document frequency**.

  N: número total de documentos

  df_i: número de documentos en los que aparece t_i
- El peso de la palabra t_i en el documento d_j, w_{ij} es

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, \ldots, f_{|V|j}\}}$$

$$idf_i = \log \frac{N}{df_i}$$

$$w_{ij} = tf_{ij} \times idf_i .$$

Clasificando textos con el método Rocchio

Algoritmo:

para cada clase i hacer

construir su vector c_i

para cada documento del test d_t **hacer**

la clase de d_t es arg max coseno(d_t, c_i)

donde coseno(d_t, c_i) es el coseno del ángulo que forman los dos vectores

XML

El lenguaje XML (eXtensible Markup Language).
Motivación:

- Es una respuesta a:
  - cómo tratar datos semi-estructurados de la web,
  - cómo organizar colecciones de datos de distintas fuentes y formatos,
  - cómo intercambiar datos entre diferentes sitios/organizaciones.

- Permitirá integrar sistemas de información hasta ahora separados:
  - *documentos*: tienen estructura irregular, anidados profundamente, utilizan tipos de datos relativamente simples y dan gran importancia al orden.
  - *relaciones*: tienen una estructura muy regular, son relativamente planos, utilizan tipos de datos relativamente complejos y dan poca importancia al orden.

XML

El lenguaje XML (eXtensible Markup Language):

- XML es un metalenguaje, desciende de GML (Generalized Markup Language) y de su estándar SGML (Standard GML).
- La sintaxis del XML es muy sencilla. Consta exclusivamente de:
  - Marcas:
    - de apertura <coche>. Toda marca de apertura debe ir seguida por un término XML y una marca de cierre con el mismo identificador.
    - de cierre </coche>.
    - de apertura y cierre (vacías) <coche/> ≡ <coche></coche>
  - Atributos de las marcas (en la apertura de una marca):
    - <coche matricula=334125 color=verde> .... </coche>
  - Las marcas se pueden repetir a cualquier nivel. Los atributos no se pueden repetir en la misma marca. El valor de un atributo es siempre una cadena ASCII sin comillas. Todas las marcas y atributos se escriben en minúscula.
  - El orden de las marcas es relevante pero el orden de los atributos no.

XML

El lenguaje XML (eXtensible Markup Language):

- Sintaxis (cont.):
  - Comentarios: <!-- esto es un comentario -->
  - Instrucciones: <?name pidata?>
  - Comandos: <! … >
  - Secuencias de escape: <![CDATA[lo que haya aquí dentro es literal]] >
  - Macros: empiezan por & o por % y terminan con punto y coma. Hay algunas predefinidas y se pueden definir por el usuario.
    - Se definen mediante la expresión:

      <!ENTITY %pi “3.14159”>
  - Otros identificadores: empiezan por #.

XML

El lenguaje XML. Documentos Bien Construidos (o Formados):

- Un documento que sigue la sintaxis XML.
- Ejemplo de documento XML bien construido:

```
<?xml version = "1.0"?>  -- declaración
<biblioteca>             -- marca raíz
<autor> <nombre>José Luis Borges</nombre>
    <nacionalidad>Argentino</nacionalidad>
    <especialidad>cuento</especialidad> <ciego/> </autor>
<autor> <nombre>Isaac Asimov</nombre> <especialidad>ciencia
    ficción</especialidad> <nacionalidad>Estadounidense</nacionalidad>
    <especialidad>ciencia</especialidad> </autor>
<autor> <nombre>Vicent Andrés Estellés</nombre> </autor>
</biblioteca>
```

- Ejemplo de documento mal construido: en general cualquier documento HTML, hay marcas de comienzo sin fin, no hay anidamiento correcto de marcas, atributos por defecto, etc.

XML

El lenguaje XML. Las DTDs.

¿Y a qué viene tanto bombo?

- Aparte de los documentos XML existen DTD's (Document Type Definition) que pueden ir incluidos en el propio documento XML o en otro URL.
- Los documentos DTD tienen una sintaxis similar a la definición de una gramática regular:
  - existenciales: los signos ?, +, *, significando, respectivamente, 0-1, 1 ó más, 0 ó más.
  - disyunciones: la alternativa $a|b$, que significa o a o b.
  - tipos de datos: #PCDATA es una secuencia de caracteres.
  - marcas vacías (de apertura y cierre): EMPTY.

XML

El lenguaje XML. Documentos Válidos:

- Un documento XML es válido respecto a una DTD si se ajusta a ella. Para que sea válido tiene, evidentemente, que estar bien formado.
- Ejemplo de DTD:

```
<!ELEMENT biblioteca(autor+) >
<!ELEMENT autor (nombre, nacionalidad?, especialidad*, sexo?, ciego?)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT sexo (hombre|mujer)>
<!ELEMENT especialidad (#PCDATA)>
<!ELEMENT nacionalidad (#PCDATA)>
<!ELEMENT ciego EMPTY>
```

Una biblioteca es un conjunto de 1 ó más autores, cada uno con un nombre y, opcionalmente, una nacionalidad, un sexo, si son ciegos o no, y 0 ó más especialidades.

El documento XML anterior es válido respecto a esta gramática.

XML

El lenguaje XML y las DTDs

- Las DTDs son en realidad gramáticas libres de contexto.
- Las DTDS se pueden almacenar en el mismo documento de la siguiente manera:

```
<?xml version = "1.0"?>
<!DOCTYPE tipo_biblioteca [
    <!ELEMENT biblioteca(autor+) >
    <!ELEMENT autor (nombre, ....
    ...
 ] >
<biblioteca>
<autor> <nombre>José Luis Borges</nombre>
...
</biblioteca>
```

- O, generalmente, en otro documento aparte, al cual se hace referencia.

XML

El lenguaje XML y las DTDs

- Aunque las DTDS se almacenan generalmente en otro documento aparte, al cual se hace referencia.

```
<?xml version = "1.0"?>
<!DOCTYPE tipo_biblioteca SYSTEM "http://www.site.es/bib.dtd">
<biblioteca>
<autor> <nombre>José Luis Borges</nombre>
...
</biblioteca>
```

- y el `bib.dtd` comenzaría directamente de la siguiente manera:

```
<!ELEMENT biblioteca(autor+) >
<!ELEMENT autor (nombre, ....
    ...
```

XML

El lenguaje XML y las DTDs

- Ejemplo de DTD para representar una base de datos relacional:

```
<!DOCTYPE db [
        <!ELEMENT db ((r1|r2)*)>
        <!ELEMENT r1 (a,b,c)>
        <!ELEMENT r2 (c,d)>
        <!ELEMENT a (#PCDATA)>
        <!ELEMENT b (#PCDATA)>
        <!ELEMENT c (#PCDATA)>
        <!ELEMENT d (#PCDATA)> ] >
```

ejemplo de documento XML válido respecto a la DTD.

```
<db><r1><a> a1 </a> <b> b1 </b> <c> c1 </c> </r1>
    <r1><a> a2 </a> <b> b2 </b> <c> c2 </c> </r1>
    <r2><c> c3 </c> <d> d3 </d> </r2>
    <r2><c> c4 </c> <d> d4 </d> </r2>
    <r2><c> c5 </c> <d> d5 </d> </r2>   </db>
```

El lenguaje XML y las DTDs

- También una DTD puede especificar los atributos de las marcas:

```
<!DOCTYPE product [
<!ELEMENT product (name, price)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ATTLIST name language CDATA #REQUIRED
                        department CDATA #IMPLIED>
<!ATTLIST price currency CDATA #IMPLIED>] >
```

#IMPLIED significa opcional

ejemplo de documento XML válido respecto a la DTD.

```
<product>
        <name language="French" department="music">
                trompette six trous </name>
        <price currency="Euro"> 420.12 </price>
</product>
```

XML

DTDs y Referencias

- Existen tres tipos especiales en XML para hacer referencias (ID, IDREF, IDREFS). Por ejemplo, la definición de DTD:

```
<!DOCTYPE family [      <!ELEMENT family (person)*>
                        <!ELEMENT person (name)>
                        <!ELEMENT name (#PCDATA)>
                        <!ATTLIST person  id ID #REQUIRED
                                  mother IDREF #IMPLIED
                                  father IDREF #IMPLIED
                                  children IDREFS #IMPLIED>] >
```

ejemplo de documento XML válido respecto a la DTD.

```
<family>
<person id="jane" mother="mary" father="john"><name> Jane Doe </name></person>
<person id="john" children="jane jack"><name> John Doe </name></person>
<person id="mary" children="jane jack"><name> Mary Smith </name></person>
<person id="jack" mother="mary" father="john"><name> Jack Doe </name></person>
</family>
```

XML

Uso de las DTDs:

- Aparecen DTDs para casi todo: clientes, proveedores, personal, bibliotecas, mapas, diccionarios, ...

- Además de las gramáticas
  - Han aparecido innumerables especializaciones o familias de lenguajes de marcas, como SMIL (lenguaje de integración multimedia sincronizada) o XSL (lenguaje de hojas de estilo extensible).

- Aparte aparece XHTML...

XHTML

El lenguaje XHTML:

- Es una versión del HTML conforme a XML.
- No es más que una DTD para XML.
- Características:
  - el código debe estar en minúsculas
  - todos los valores de atributos deben ir entre comillas dobles.
  - todas las etiquetas deben tener principio y fin.
  - el anidamiento de etiquetas se debe respetar.
  - existen elementos obligatorios (html, head, title, body, etc.)
- Existen conversores y plantillas en www.w3.org.
- Si los documentos estuvieran en XHTML en vez de HTML, el análisis de los mismos sería más sencillo.

XML Mining (Markup Mining)

XML Mining: Extracción de Información a partir de docs. XML.

Distintos Objetivos:

- SCHEMA EXTRACTION: Por esquema entendemos algo similar a un DTD, aunque representado con otros formalismos (programas lógicos, grafos, ...) ((Wang 1999), (Toivonen 1999))
- DATAGUIDES: Especie de resumen estructurado de datos semiestructurados, a veces aproximado. (Grumbach and Mecha 1999), (Nestorov et al. 1997), (Goldman and Widom 1999)
- MULTI-LAYER-DATABASES (MLDB): Hay distintos niveles de granularidad en el esquema extraído. Se pueden construir lo que se denominan MLDBs (Grumbach and Mecha 1999) , (Nestorov et al. 1997), (Zaïane et al. 1998)), en la que cada capa se obtiene por generalizaciones de capas inferiores
- CLASIFICACIÓN.

XML Mining

Representación: Un documento XML (sin OIDs) es un árbol...
pero con OIDs la estructura del documento se debe representar como un grafo y no como un árbol (extensible a HTML).

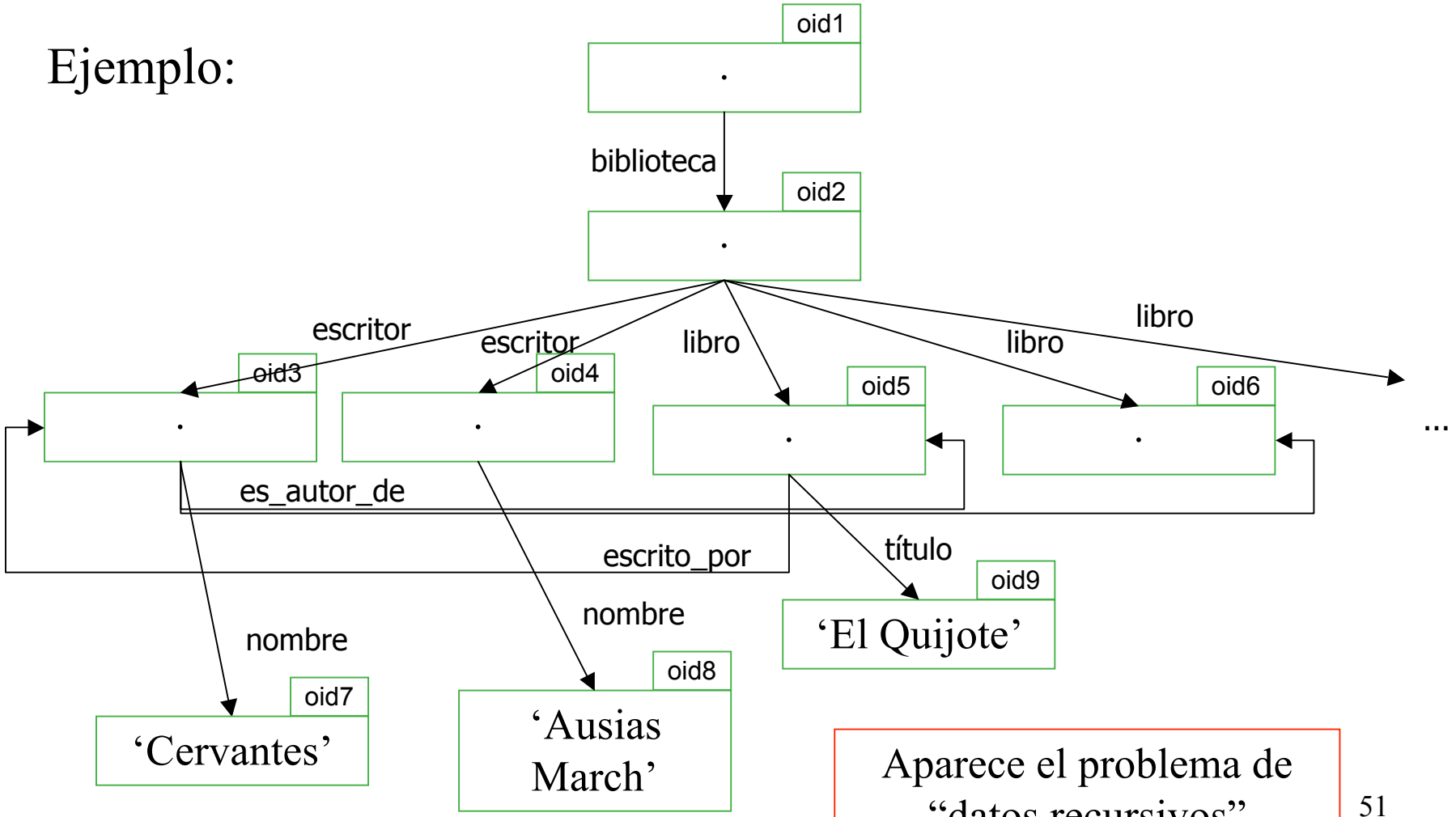
Los datos semi-estructurados se representan como grafos etiquetados (OEM, *Object Exchange Model* (Abiteboul et al. 1999))

GRAFO:

- Los nodos del grafo son los objetos, que están formados de un identificador (oid) y un valor que puede ser atómico (entero, cadena, gif, html, ...) o referencia, denotado por un conjunto de pares (etiquetas, oid).
- Las aristas del grafo están etiquetadas.
- Las hojas contienen valores atómicos.

XML Mining

Ejemplo:



Aparece el problema de “datos recursivos”

XML Mining

Equivalencia documento XML con una estructura relacional. El XML del ejemplo anterior:

```
<biblioteca oid= “&oid2” escritor= “&oid3 &oid4” libro= “&oid5 &oid6”>
<escritor oid=“&oid3” es_autor_de=“&oid5 &oid6” nombre= “&oid7”> </escritor>
<escritor oid=“&oid4” es_autor_de=“&oid15” nombre= “&oid8”> </escritor>
<libro oid=“&oid5” escrito_por=“&oid3” titulo= “&oid9”> </libro>
<libro oid=“&oid6” .... >
<nombre oid=“&oid7” > Cervantes </nombre>
<nombre oid=“&oid8” > Ausias March </nombre>
<titulo oid=“&oid9” > El Quijote </titulo>
...
</biblioteca>
```

Expresado relacionalmente:

obj	value
&oid7	“Cervantes”
&oid8	“Ausias March”
&oid9	“El Quijote”
...	...

source	label	dest
&oid1	“biblioteca”	&oid2
&oid2	“escritor”	&oid3
&oid2	“escritor”	&oid4
&oid2	“libro”	&oid5
&oid2	“libro”	&oid6
&oid3	“es_autor_de”	&oid5
&oid3	“es_autor_de”	&oid6
&oid3	“nombre”	&oid7
&oid4	“es_autor_de”	&oid15
&oid4	“nombre”	&oid8
&oid5	“escrito_por”	&oid3
&oid5	“titulo”	&oid9
...	...	...

XML Mining

XML Mining y Técnicas Relacionales:

Sea con transformación o sin ella, los datos con oids tienen referencias (grafo) y son relacionales.

Sólo ILP o técnicas AD-HOC pueden tratar con estos tipos de problema relacionales o de grafos.

Algunos ejemplos de la literatura:

- SCHEMA EXTRACTION (Wang 1999), (Toivonen 1999).
- DATAGUIDES (Grumbach and Mecha 1999), (Nestorov et al. 1997), (Goldman and Widom 1999) :

Antes de definir cualquiera de las aplicaciones anteriores, hay que esclarecer una noción de generalidad...

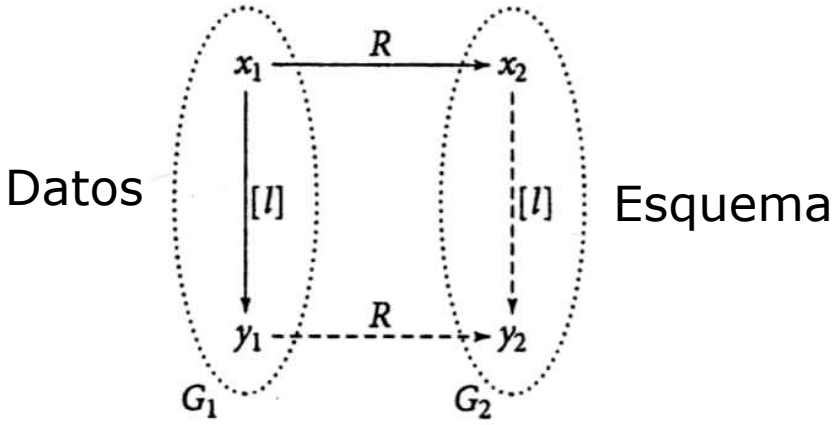
XML Mining

Comprobación de validez de unos datos respecto a un esquema:
Existe un método clásico llamado "simulación".

Dados dos grafos $G_1 = (V_1, E_1)$ y $G_2=(V_2, E_2)$, una relación R sobre V_1, V_2 es una simulación si satisface que:

$$\forall l \in L \forall x_1, y_1 \in V_1 \forall x_2 \in V_2 (x_1[l]y_1 \wedge x_1 R x_2 \Rightarrow \exists y_2 \in V_2 (y_1 R y_2 \wedge x_2[l]y_2))$$

es decir, cada arista del grafo G_1 tiene una arista correspondiente en G_2.

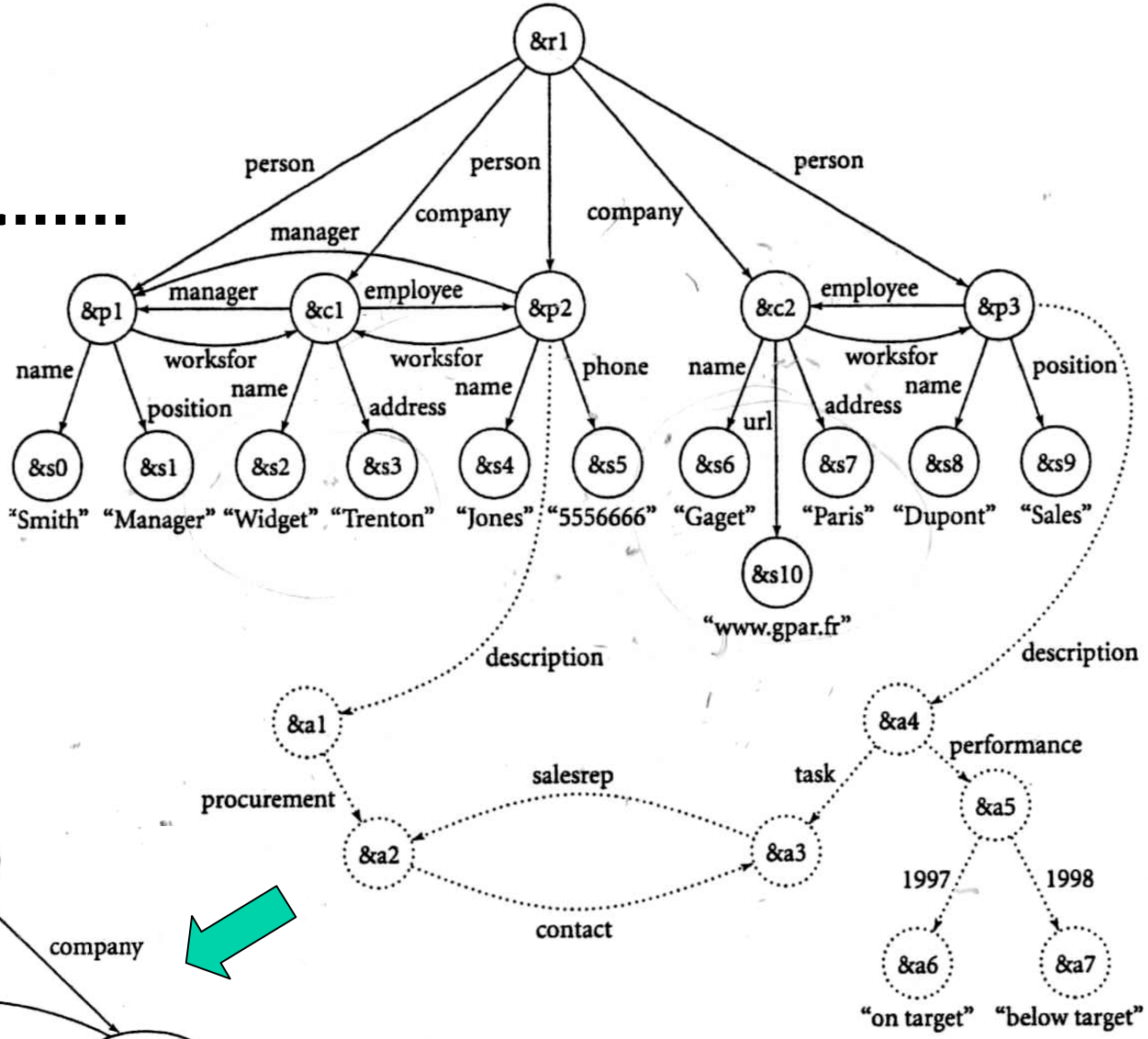


Podemos extender la definición anterior utilizando comodines. '_' representa cualquier etiqueta.

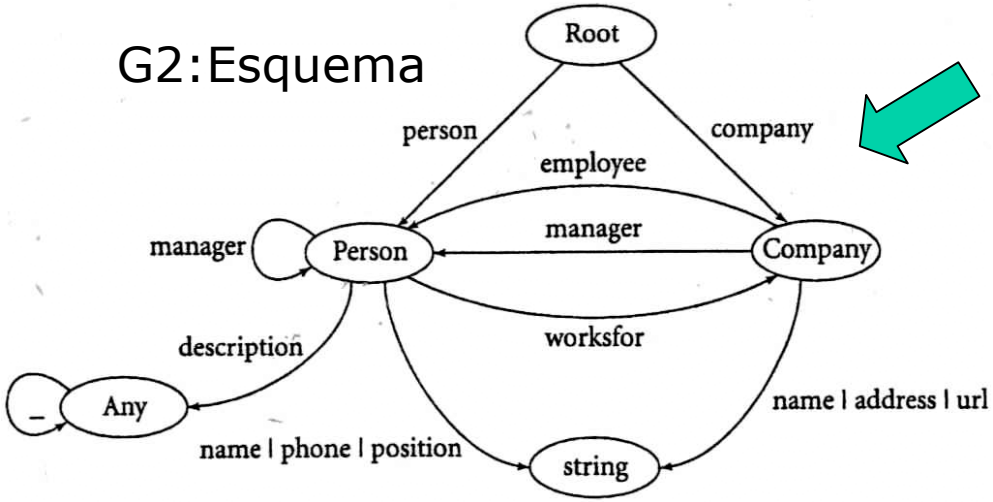
XML Mining

Ejemplo:

G1:Datos



G2:Esquema



Se interpreta como que **pueden** haber personas o compañías, las personas **pueden** ser jefes de otras, empleadas de empresas, ...

XML Mining

Ejemplo (cont.):

Un esquema clasifica (o tipa) los objetos de un documento de datos (induce una clasificación).

El grafo 2 es un esquema del grafo 1 porque hay una simulación entre ellos, a saber:

Data node	Schema node
&r1	Root
&c1,&c2	Company
&p1,&p2,&p3	Person
&s0,&s1,&s2,&s3,&s4,&s5, &s6,&s7,&s8,&s9,&s10	string
&a1,&a2,&a3,&a4, &a5,&a6,&a7	Any

La simulación muestra claramente qué objetos son de qué clases (tipos). (Aunque puede haber esquemas con ambigüedades).

XML Mining

Relación de Generalidad respecto a simulación:

Un grafo G_1 es más específico que un grafo G_2, denotado por $G_1 \ll_R G_2$ si existe una simulación R de G_1 a G_2.

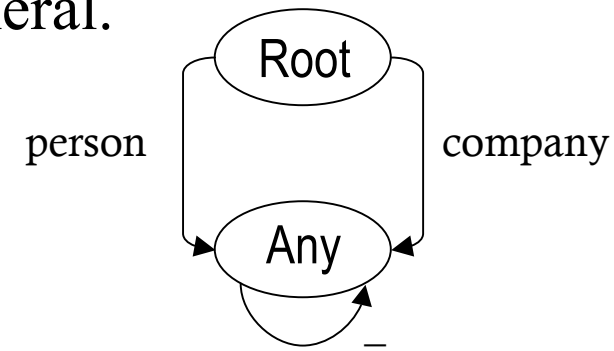
Obviamente,

si $G_1 \ll_R G_2$ y $G_2 \ll_S G_1$ entonces $G_1 \equiv G_2$

Además, si $G_1 \ll_R G_2$ y $G_2 \ll_S G_3$ entonces $G_1 \ll_{R \bullet S} G_3$

si $G_1 \ll_R G_2$ y $G_1 \ll_S G_2$ entonces $G_1 \ll_{R \cup S} G_2$

P.ej. para el caso anterior, el esquema siguiente es todavía más general.



Pero es inútil de lo general que es...

XML Mining

Relación de Generalidad respecto a simulación

Forma un retículo, en el cual el más específico (son los datos) y el más general es un único nodo con un arco comodín (_).

Este retículo es el que se explota en las bases de datos multicapa (MLDB).

Aparte de esto, las generalizaciones interesantes de los datos están entre medio (se podrían utilizar criterios MDL...).

XML Mining

Relación de Generalidad respecto a simulación

El proceso de simulación entiende los arcos del grafo esquema como “opcionales”.

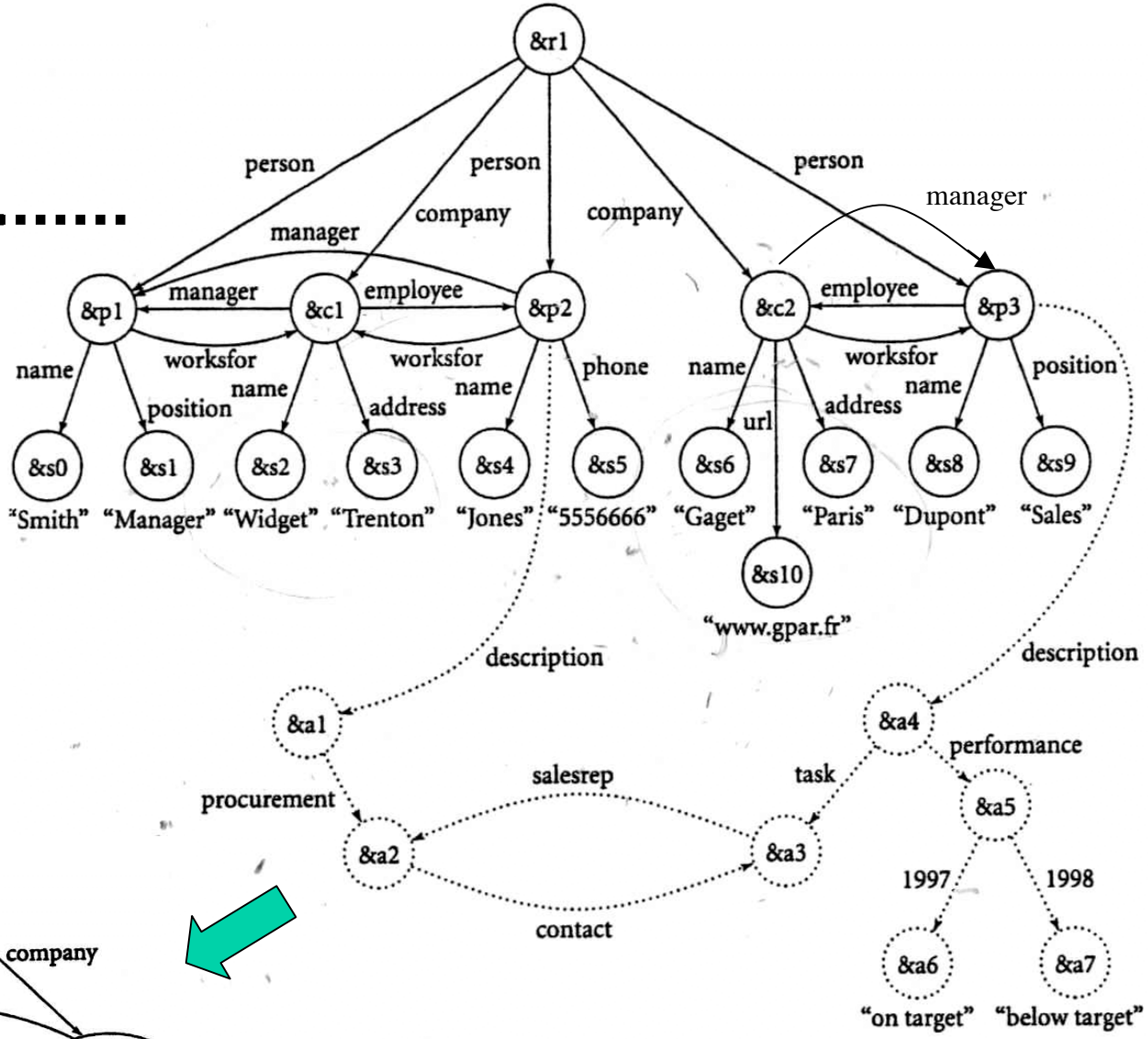
¿Qué pasa si los entendemos como “obligatorios”?

Tenemos una interpretación dual de los grafos.

XML Mining

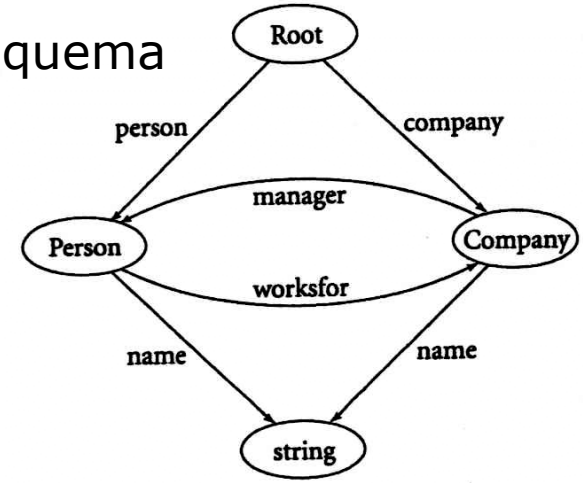
Ejemplo:

G1:Datos



ESQUEMA INTERPRETADO DUALMENTE

G2:Esquema



Se interpreta como que un objeto es de tipo persona si se ha llegado a través de la etiqueta 'person' y además **debe** trabajar para una compañía, y **debe** tener un nombre.

XML Mining

Relación de Generalidad (dual)

Lo interesante es que la relación de generalidad dual se calcula simplemente dando la vuelta (invirtiendo) los arcos en la simulación.

Y ésta corresponde con reglas DATALOG del estilo:

```
pers(X) :- ref(X, worksfor, U), comp(U), ref(X, name, Y), string(Y).
```

Y por tanto una simulación y su dual se puede computar como puntos fijos!!

Vamos a ver las aplicaciones de esta relación...

XML Mining

SCHEMA EXTRACTION:

Dado un documento XML (o su representación) para el cuál no hay DTD definido (o éste es muy genérico), ¿sigue algún esquema?

Sin el uso de métricas de evaluación, sólo podemos buscar el grafo más específico que cumpla ciertas condiciones.

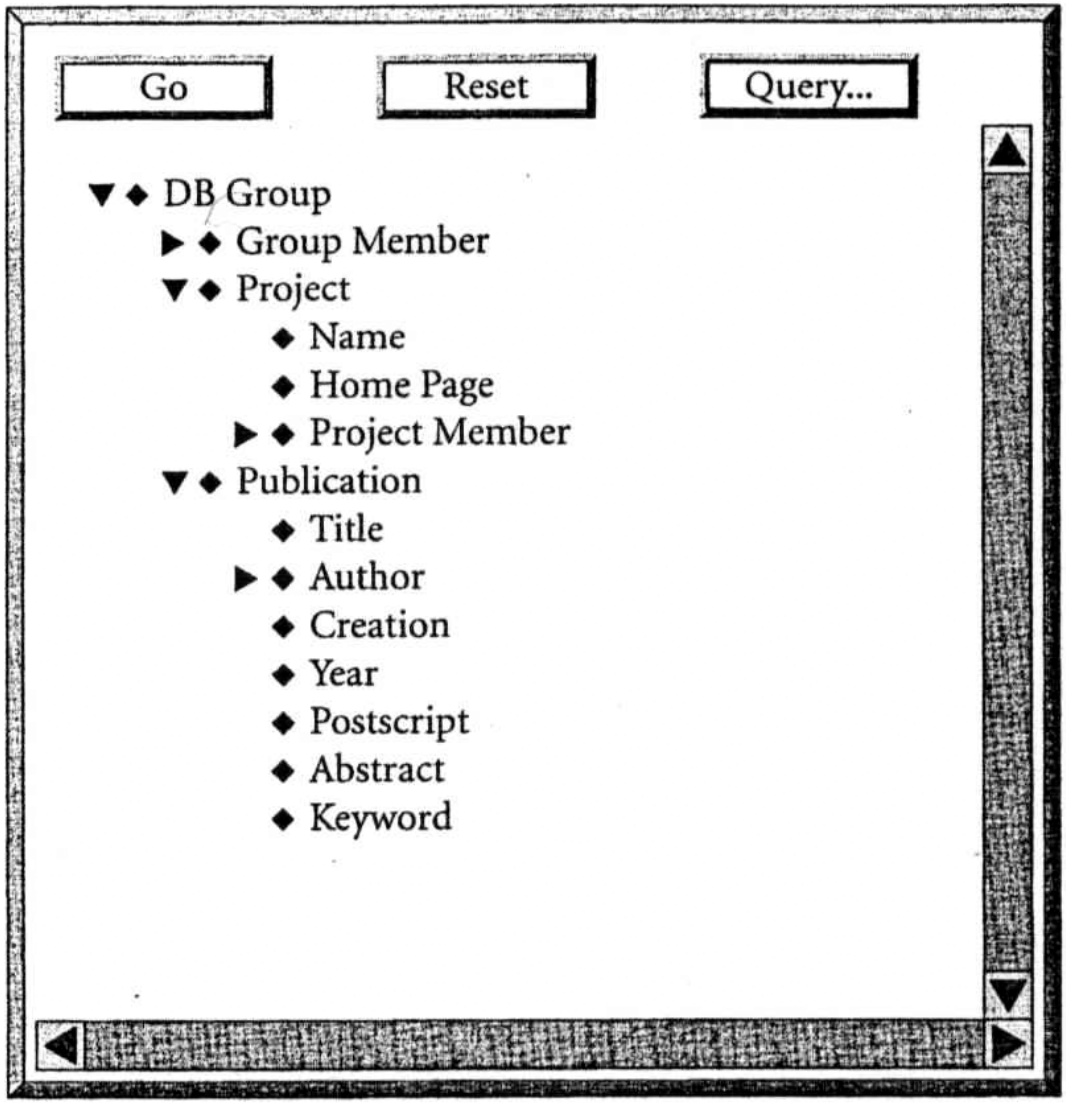
En concreto nos interesan:

- El grafo determinista más específico (DATAGUIDES).
- El grafo determinista dualmente más específico.

XML Mining

DATAGUIDES:

Son
site maps
de navegación...



XML Mining

DATAGUIDES:

Una *dataguide* de un grafo *G* es un grafo *D* tal que

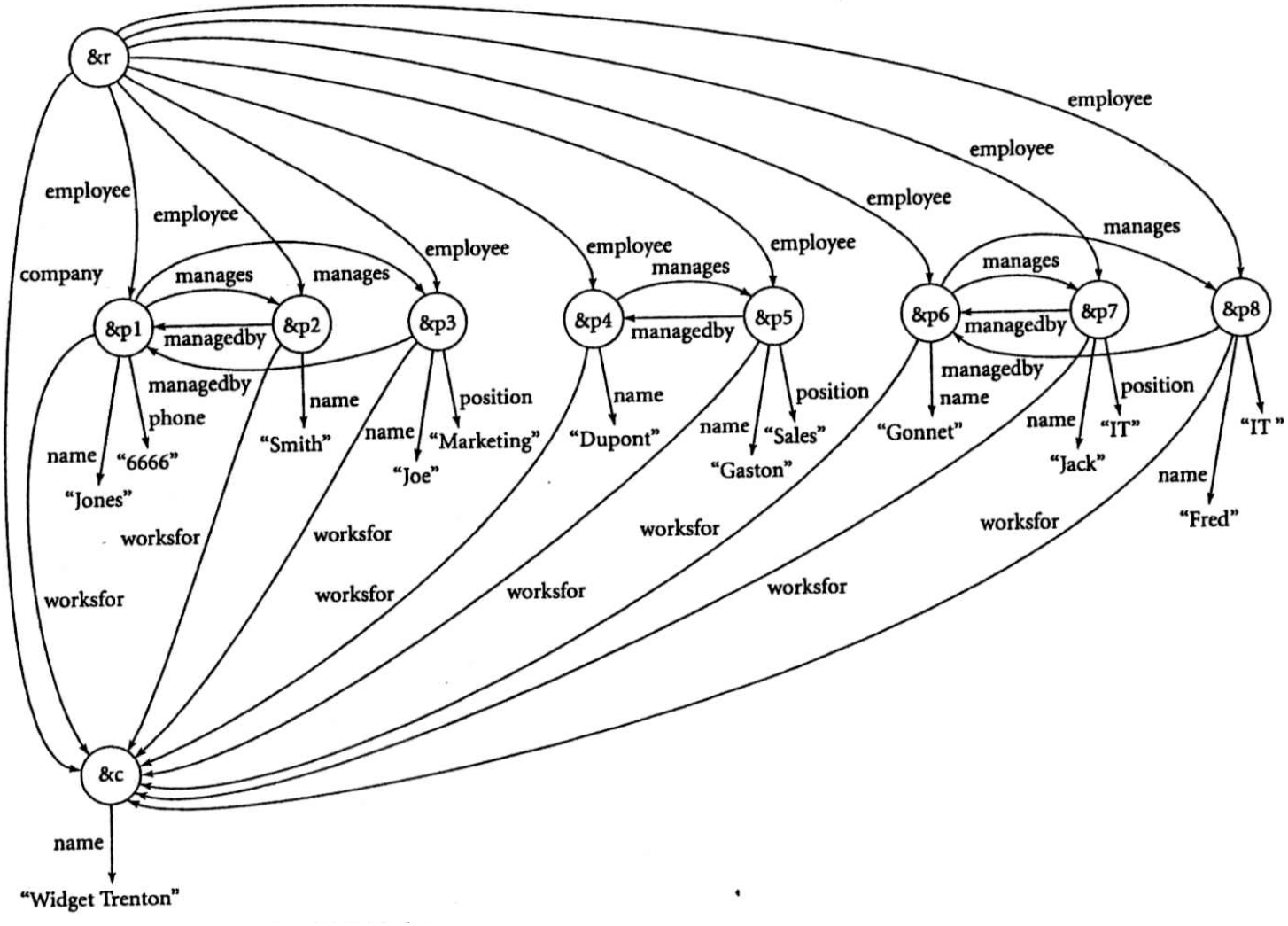
- *D* es un grafo de esquema determinista (i.e., de cada nodo no pueden salir dos arcos con la misma etiqueta).
- Cualquier otro grafo determinista conforme con G subsume a D.

Una manera de extraer la 'dataguide' es recorrer todos los caminos desde la raíz creando los nodos y los arcos necesarios para que todos tengan una representación en el esquema

Vamos a ver un ejemplo...

XML Mining

Ejemplo: extraer la Dataguide del siguiente grafo



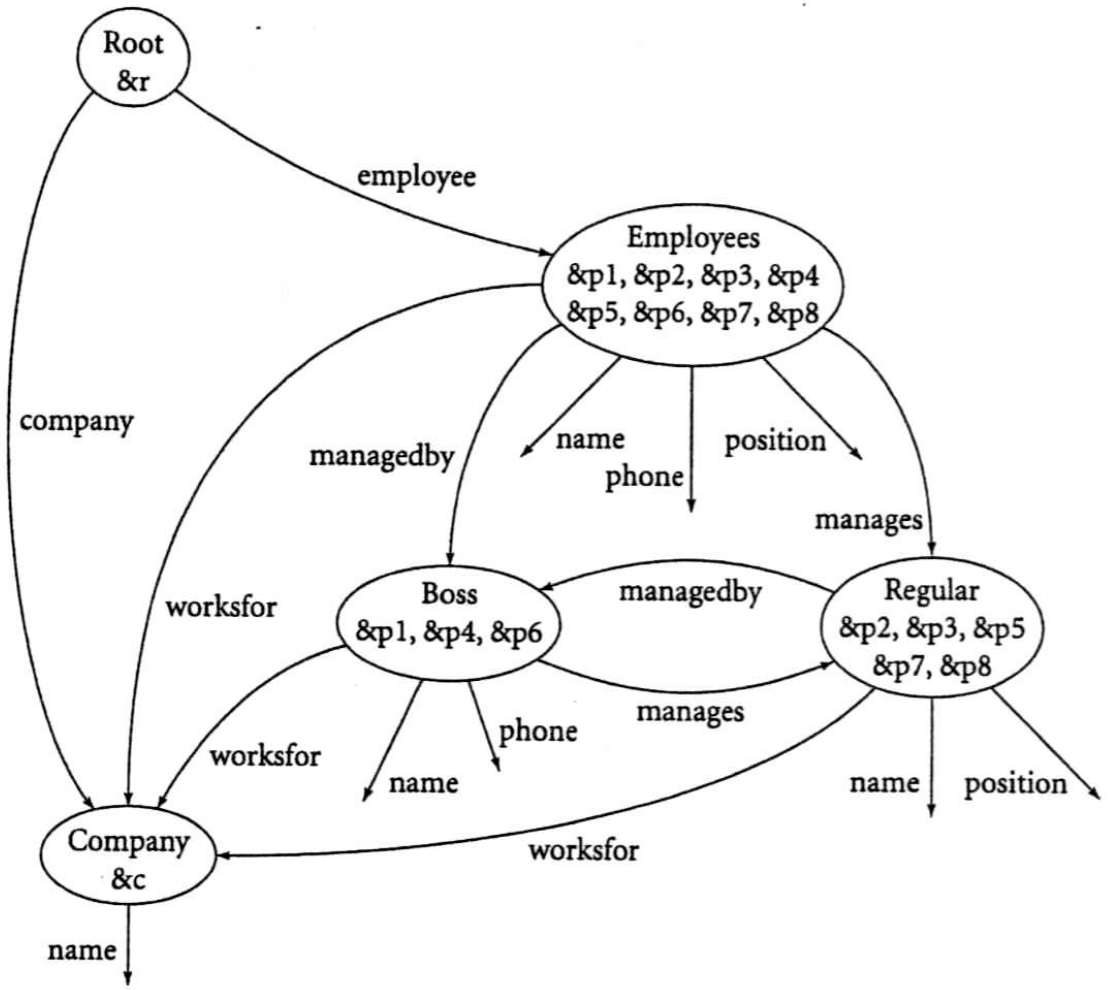
XML Mining

DATAGUIDES. Una manera de extraer la 'dataguide' es así:

- Creamos un nodo raiz "Root". Después se examinan los posibles caminos desde él.
  - camino "employee": Hay ocho en los datos, que llegan a &p1, &p2, &p3, &p4, &p5, &p6, &p7, &p8. Creamos un nuevo nodo, que lo llamamos "Employees", que se enlaza a "Root" por un arco "employee".
  - camino "employee.name": Todos estos caminos llevan a valores atómicos de tipo "string". Creamos un nuevo nodo, que llamamos "Strings" y conectamos "Employees" con él con un arco etiquetado "name".
  - camino "employee.manages": Hay cinco en los datos, que llegan a &p2, &p3, &p5, &p7, &p8. Creamos un nuevo nodo, que lo llamamos "Regular" y conectamos "Employees" con él por un arco etiquetado "manages".
  - camino "employee.manages.managedby": Hay tres en los datos, que llegan a &p1, &p4, &p6. Creamos un nuevo nodo, que lo llamamos "Boss" y conectamos "Regular" con él por un arco etiquetado "managedby".
  - camino "employee.manages.managedby.manages": Hay cinco en los datos, que llegan a &p2, &p3, &p5, &p7, &p8. Hay un nodo que ya creamos al que llegaban estos cinco caminos. Por tanto sólo enlazamos "Boss" y "Regular" con un arco etiquetado "manages".
  - camino "employee.manages.managedby.manages.managedby": Hay tres en los datos, que llegan a &p1, &p4, &p6. Tanto el nodo como los arcos ya existen. No se hace nada.
  - camino "company": Hay uno en los datos, que llega a &c. Creamos un nuevo nodo, que lo llamamos "Company", que se enlaza a "Root" por un arco "company".
  - ...
- Seguimos con otros caminos "company.name", "employee.worksfor", "employee.worksfor.name", hasta que no se creen nuevos nodos ni arcos.

XML Mining

DATAGUIDES. Su dataguide sería:



XML Mining

Extrayendo esquemas a partir de consultas:

Si el documento de partida ha sido generado por una consulta que conocemos, es mucho más inteligente aprovechar esa consulta para extraer el esquema de de los datos.

Ejemplo: Dada la siguiente consulta en StruQL:

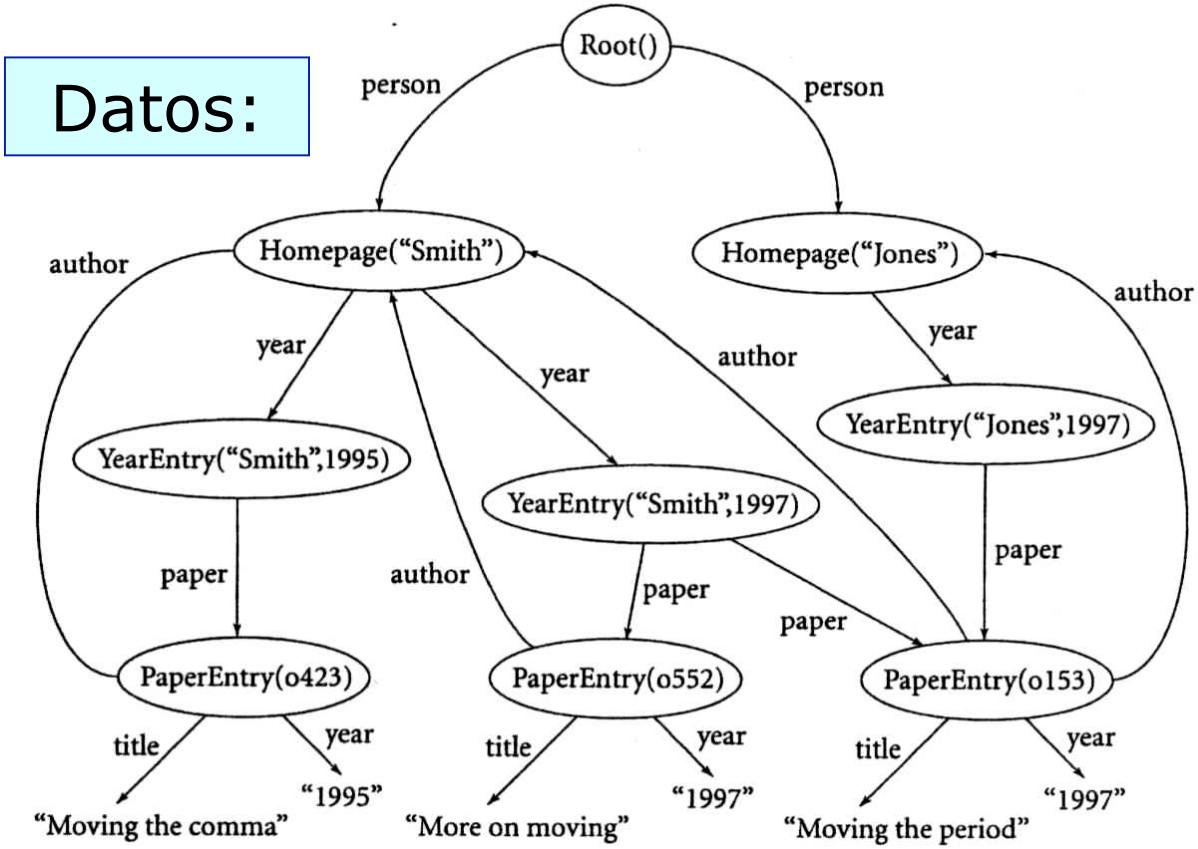
```
where    bib -> L -> X, X -> "author" -> A, X -> "title" -> T, X -> "year" -> Y
create   Root(), HomePage(A), YearEntry(A,Y), PaperEntry(X)
link     Root() -> "person" -> HomePage(A),
         HomePage(A) -> "year" -> YearEntry(A,Y),
         YearEntry(A,Y) -> "paper" -> PaperEntry(X),
         PaperEntry(X) -> "title" -> T,
         PaperEntry(X) -> "author" -> HomePage(A),
         PaperEntry(X) -> "year" -> Y
```

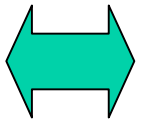
XML Mining

Extrayendo esquemas a partir de consultas:
A partir de los datos siguientes y la consulta...

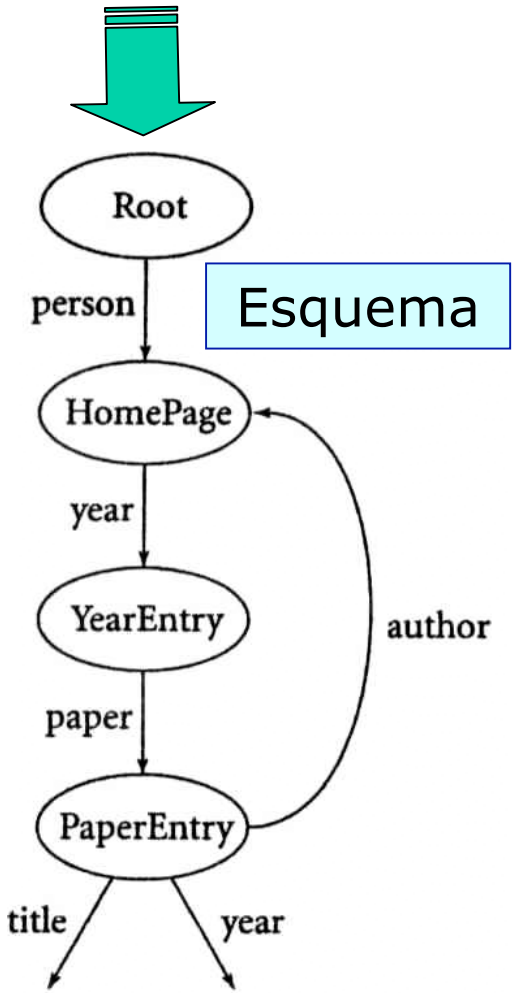
Consulta

Datos:





Esquema



XML Mining

Clasificación: problema más tradicional, se pueden aplicar directamente las técnicas ILP o IFLP.

EJEMPLO (con FLIP).

Documento XML original:

<gratification value=30><name>john></name><has_children/></gratification>
<gratification value=30><married/><teacher/><has_cellular></gratification>
<gratification value=30><sex>male</sex><teacher/><name>jimmy</name></gratification>
<gratification value=20><name>john></name><tall/></gratification>
<gratification value=10><married/><police/></gratification>
<gratification value=10><married/><politician/></gratification>
<gratification value=20><sex>male</sex><boxer/></gratification>

Clientes con diferente grado de gratificación (30%, 20% or 10%)

XML Mining

EJEMPLO (cont.). Conversión directa a notación lógico-funcional.

E:
```
gratification(: (: ( [],name ( john ), has_children)) = 30
gratification(: (: (: ( [],married), teacher) has_cellurar)) = 30
gratification(: (: (: ( [],sex(male)), teacher), name ( jimmy))) = 30
gratification(: (: ( [],name ( john ), tall)) = 20
gratification(: (: ([],married), police)) = 10
gratification(: (: ([],married), politician)) = 10
gratification(: (: ( [],sex(male)), boxer)) = 20
```

Solución FLIP.

H:
```
gratification(p(X0,X1)) = gratification(X0)
gratification(p(X0,has_children)) = 30
gratification(p(X0,sex(female))) = 30
gratification(p(X0,teacher) = 30
gratification(p(X0,politician) = 10
gratification(p(X0,boxer) = 20
...
```

Web Query Languages

WebSQL-University of Toronto
http://www.cs.toronto.edu/~websql/

Ejemplo: Encontrar documentos que hablan del aluminio

SELECT d.url, d.title
FROM Document d SUCH THAT d CONTAINS “aluminium”

Modern Information Retrieval
http://sunsite.dcc.uchile.cl/irbook/

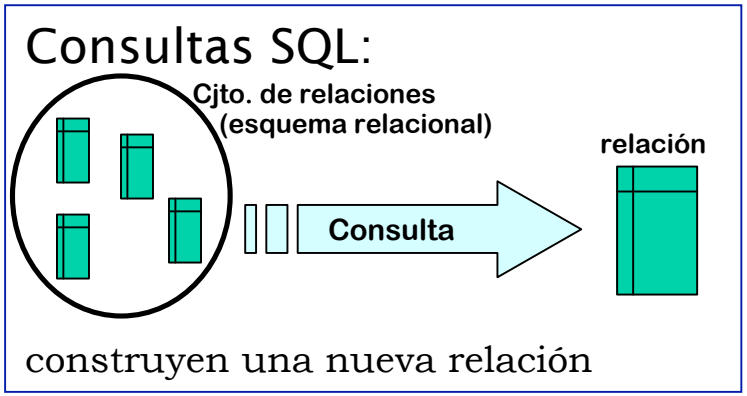
XML. Lenguajes de Consulta

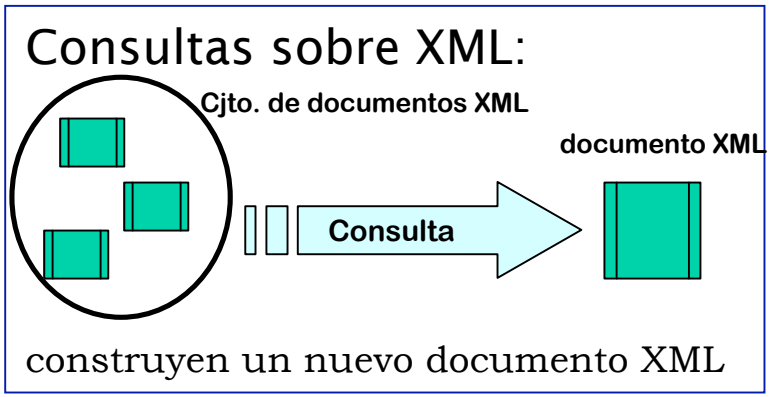
Mezclan características de SQL con sintaxis de XML.

Todavía no existe un estándar, aunque existen numerosas propuestas (Abiteboul et al. 1999):

- XML-QL (Deutsch et al. 1998-1999)
- XPath (W3C 1999)
- XQL (Robie et al 1998)
- Quilt (Chamberlin et al. 2000)

Al igual que SQL la fuente y el resultado de las consultas son compatibles, permitiendo el encadenamiento (subconsultas).





XML. Lenguajes de Consulta

Lenguaje XML-QL (Deutsch et al. 1998-1999) :

- utiliza expresiones de camino y patrones para extraer los datos de los documentos XML de entrada.
- utiliza variables a las cuales se ligan subpartes del documento.
- tiene plantillas para construir la salida.
- es completo relacionalmente (tan expresivo al menos como el Álgebra o el Cálculo Relacionales).

Ejemplo del uso de patrones:

```
where <book>
    <publisher><name>Morgan Kaufmann</name></publisher>
    <title> $T </title>
    <author> $A </author> </book>
in "www.dsic.upv.es/bib.xml"
construct $A
```

patrón, corresponde a la selección (where del SQL)

fuente de datos, corresponde al from del SQL

construcción, corresponde a la proyección (select del SQL)

que extrae todos los autores de los libros con título de la editorial 'Morgan Kaufmann'.

XML. Lenguajes de Consulta

Lenguaje XML-QL (Deutsch et al. 1998-1999) :

Ejemplo de enlace de variables (construcción de datos XML):

```
where <book> <author> $A </> </>
                    content_as $B1 in "www.dsic.upv.es/bib.xml"
      <book> <author> $A </> </>
                    content_as $B2 in "www.dsic.upv.es/bib.xml"
      $B1 != $B2
construct <result> $A </result>
```

son abreviaturas de </book> y </author>

extrae todos los autores que al menos han publicado dos libros.

Ejemplo de expresiones regulares (*, +, ., ?, |) :

```
where <part*> <name> $R </> <brand> Ford </> </>
                    in "www.dsic.upv.es/piezas.xml"
construct <result> $R </result>
```

extrae todos los elementos tales que contengan alguna subpieza (a algún nivel de profundidad) que sea de Ford.

XML. Lenguajes de Consulta

Lenguaje Quilt (Chamberlin et al. 2000) : a partir de las propuestas XML-QL, XPath and XQL.

Quilt es un lenguaje funcional cuya construcción fundamental es la expresión FLWR que puede ligar variables en un FOR así como en un LET, aplicar un predicado en el WHERE y finalmente construir un resultado en la cláusula RETURN.

El acceso a los datos se basa en XPath:

- . denota el nodo actual
- / denota el hijo del nodo actual
- // denota los descendientes del nodo actual a cualquier nivel (cierre de /).
- @ representa atributos del nodo actual
- [ ] incluyen expresiones booleanas (condiciones de nivel).
- [*n*] caso especial. Si *n* es un entero, retorna el elemento número *n* de todos los accedidos por el camino.

XML. Lenguajes de Consulta

Lenguaje Quilt (Chamberlin et al. 2000).

Ejemplo de camino:

```
document("zoo.html")/chapter[2]
    //figure[caption = "Tree Frogs"]
```

extrae la figura en el segundo capítulo de "zoo.html" que tenga como *caption* "Tree Frogs".

Ejemplo de consulta completa:

```
FOR $b IN document("bib.xml")/book
WHERE $b/publisher = "Morgan Kaufmann" AND $b/year = "1998"
RETURN $b/title
```

extrae los títulos de los libros publicados por "Morgan Kaufmann" en 1998.

También tiene funciones agregadas como AVG() y COUNT().

XML. Lenguajes de Consulta

Lenguaje Quilt (Chamberlin et al. 2000).

Ejemplo de consulta con expresiones condicionales:

```
FOR $b IN //holding
RETURN <holding> $h/title, IF $h/@type='Journal' THEN $h/editor   ELSE $h/author
    </holding> SORTBY (title)
```

hace una lista de "holdings", ordenados por títulos. Para las revistas, se incluye el editor, y para el resto de holdings, incluir el autor.

Ejemplos de cuantificadores:

```
FOR $b IN //book
WHERE SOME $p IN $b//para SATISFIES contains($p, "sail") AND contains($p, "surf")
RETURN $b/title
```

busca títulos de libros en los cuales "sail" y "surf" salen en el mismo párrafo.

```
FOR $b IN //book
WHERE EVERY $p IN $b//para SATISFIES contains($p, "sailing") RETURN $b/title
```

busca títulos de libros en los cuales "sailing" sale en todos los párrafos.

XML. Lenguajes de Consulta

XQuery 1.0: An XML Query Language

W3C Working Draft 20 December 2001

(http://www.w3.org/TR/xquery/)

Usa el estándar XPath (también del W3C).

Intercambio de Conocimiento

HTML y XML permiten intercambiar **información**...
A partir de esta (y de otra) información se puede extraer conocimiento...

¿Cómo se puede compartir/publicar/intercambiar **conocimiento**?

P.ej. Un comercio *A* (p.ej. una pizzería) puede haber recogido información sobre los patrones de compra de los clientes durante los últimos cinco años. Esta información se muestra muy importante a la hora de gestionar recursos personales y laborales. Un nuevo comercio *B* de la misma rama no se puede permitir esperar unos años para recoger esa información y refinar ese conocimiento.
Una solución es comprar ese conocimiento al comercio A.

Intercambio de Conocimiento

El área de intercambio de conocimiento es incipiente.

Se basa en las siguientes áreas (Sarawagi & Nagaralu 2000):

- minería de datos distribuida (Kargupta et al. 1998).
- aprendizaje multi-agente (Weis 1996).
- ontologías.

Existen algunas aproximaciones parciales: E-bizness XML (www.ebxml.com), OLE DB (www.microsoft.com/data/oledb/db.htm), "Cpexchange" (www.idealliance.org/cpexchange/index.htm).

La idea es desarrollar estándares más generales que permitan:

- estandarizar entradas y salidas de datos.
- estandarizar semántica de los datos: tipos y significado de los atributos según área de negocio/aplicación.
- estandarizar la representación de reglas de distintos modelos de aprendizaje automático / minería de datos (estructura del modelo).

Intercambio de Conocimiento (PMML)

La iniciativa PMML (*Predictive Model Markup Language*):

- basado en XML. Define una DTD para los documentos válidos.
- permite codificar datos (*data dictionary*) y el esquema de minería, así como algunas mínimas características estadísticas.
- Entre los modelos que soporta la versión 3.2 están:
  - Árboles de Decisión.
  - Regresión polinomial.
  - Reglas de asociación.
  - Redes neuronales.
  - Clustering basado en centros o basado en distribuciones.
  - Naive Bayes
  - Modelos Secuenciales
  - Conjuntos de reglas
  - Máquinas de vectores soporte
- Más información en "DataMining Group": www.dmg.org
- Algunos sistemas permiten usarlo desde SGBD o herramientas de DM. P.ej., el Ms-OLESQL tiene la instrucción "CREATE MINING MODEL <identifier> FROM PMML"

Intercambio de Conocimiento (PMML)

Ejemplo de la parte de la DTD correspondiente a un árbol de decisión.

```
<!ELEMENT TreeModel (Extension*, MiningSchema, ModelStats?, Node)>
      <!ATTLIST TreeModel
        modelName     CDATA    #IMPLIED      >
<!ELEMENT Node ( Extension*, (%PREDICATES;), Node*, ScoreDistribution* )>
       <!ATTLIST Node
         score        CDATA    #REQUIRED
         recordCount  %NUMBER;  #IMPLIED      >
<!ELEMENT ScoreDistribution EMPTY>
      <!ATTLIST ScoreDistribution
        value      CDATA        #REQUIRED
        recordCount %NUMBER;     #REQUIRED     >
      <!ATTLIST TreeModel
        x-splitCharacteristic (binarySplit | multiSplit) #REQUIRED      >
```

Intercambio de Conocimiento (PMML)

Ejemplo de la parte de la DTD corr. a un árbol de decisión (cont.).

```
<!ENTITY % PREDICATES "( Predicate | CompoundPredicate | True | False  ) " >
<!ELEMENT Predicate EMPTY>
<!ATTLIST Predicate
     field        %FIELD-NAME;   #REQUIRED
     operator ( equal | notEqual | lessThan | lessOrEqual | greaterThan |
    greaterOrEqual )  #REQUIRED
     value        CDATA          #REQUIRED     >
<!ELEMENT CompoundPredicate ( %PREDICATES; , (%PREDICATES;)+ >
       <!ATTLIST CompoundPredicate
          booleanOperator (or | and | xor | cascade) #REQUIRED>
<!ELEMENT True EMPTY>
<!ELEMENT False EMPTY>
```

http://www.dmg.org/v2-0/TreeModel.html

Intercambio de Conocimiento (PMML)

Ej. de árbol de decisión válido respecto a la DTD anterior.

```
<?xml version="1.0" ?>
      <PMML version="1.1" >
       <Header description="A very small binary tree model to show structure."/>
       <DataDictionary numberOfFields="5" >
          <DataField name="temperature" optype="continuous"/>
          <DataField name="humidity" optype="continuous"/>
          <DataField name="windy" optype="categorical" > <Value value="true"/> <Value value="false"/>
    </DataField>
          <DataField name="outlook" optype="categorical" > <Value value="sunny"/> <Value value="overcast"/>
    <Value value="rain"/> </DataField>
          <DataField name="whatIdo" optype="categorical" > <Value value="play"/> <Value value="no_play"/>
    </DataField>
       </DataDictionary>
       <TreeModel modelName="golfing">
          <MiningSchema>
                 <MiningField name="temperature"/>
                <MiningField name="humidity"/>
                <MiningField name="windy"/>
                <MiningField name="outlook"/>
                <MiningField name="whatIdo" usageType="predicted"/>
        </MiningSchema>
```

Def. de los datos

Definición del problema

Intercambio de Conocimiento (PMML)

```
<Node score="play">
if → <Predicate field="outlook" operator="equal" value="sunny"/>
then → <Node score="play">
        <CompoundPredicate booleanOperator="and" >
if →      <Predicate field="temperature" operator="lessThan" value="90F" />
          <Predicate field="temperature" operator="greaterThan" value="50F" />
          <Predicate field="humidity" operator="lessThan" value="70" /> </CompoundPredicate>
  then → <Node score="play"> <True/> </Node>
  else → <Node score="no_play"> <True/> </Node>          </Node>
else → <Node score="play">
  if → <Predicate field="outlook" operator="equal" value="rain"/>
  then → <Node score="no_play"> <True/> </Node>
  else → <Node score="play">
    if → <Predicate field="windy" operator="equal" value="true" />
    then → <Node score="no_play"> <True/> </Node>
    else → <Node score="play">
            <CompoundPredicate booleanOperator="and" >
      if →    <Predicate field="temperature" operator="lessThan" value="100F" />
              <Predicate field="humidity" operator="lessThan" value="60" /> </CompoundPredicate>
      then → <Node score="play"> <True/> </Node>
      else → <Node score="no_play"> <True/> </Node>  </Node>  </Node>  </Node> </Node>
    </TreeModel>
  </PMML>
```

Definición de la solución (modelo)

Intercambio de Conocimiento (RuleML)

La iniciativa RuleML (http://www.ruleml.org):

El objetivo es desarrollar un estándar de intercambio de reglas basado en XML que sirva para diferentes áreas (del URL):

- *Engineering: Diagnosis rules (also model-based approaches appreciate and combine with rules, as described by Adnan Darwiche in Model-based diagnosis under real-world constraints, AI Magazine, Summer 2000)*
- *Commerce: Business rules (including XML versions such as the Business Rules Markup Language (BRML) of IBM's Business Rules for Electronic Commerce project)*
- *Law: Legal reasoning (Robert Kowalski and Marek Sergot have been formalizing legal rules in an Imperial College group)*
- *Internet: Access authentication*

Ya sean éstas obtenidas manualmente o utilizando técnicas de minería de datos.

Intercambio de Conocimiento (RDF)

Resource Description Framework
(RDF) Model and Syntax Specification
W3C Proposed Recommendation 05 January 1999

http://www.w3.org/TR/PR-rdf-syntax/#glossary

Web Structure Mining

Web Structure Mining:

Consiste en estudiar la estructura de enlaces entre e intra documentos.

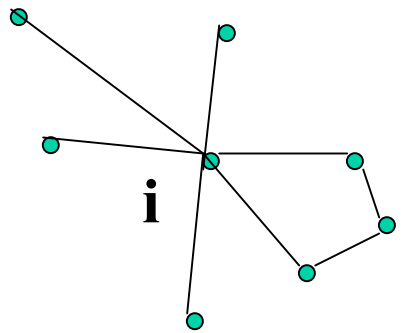
Las técnicas se inspiran en el estudio de redes sociales y análisis de citaciones (Chakrabarti 2000):

una página (persona, artículo) se ve *reforzado* por la cantidad de referencias (amistades, citas) que tiene.

Web Structure Mining

REDES SOCIALES: centralidad y prestigio

CENTRALIDAD: un actor es importante (*central*) si está ampliamente enlazado a otros actores (muchos enlaces).



Dependiendo del tipo de enlace se distinguen varios tipos de centralidad en grafos dirigidos y no dirigidos.

Web Structure Mining

A. CENTRALIDAD POR GRADO (nº enlaces)

i. Grafo no dirigido: $C_D(i) = \frac{d(i)}{n-1}$ → **grado nodo i**; → **n=nº nodos**

ii. Grafo dirigido: $C_D(i) = \frac{d_o(i)}{n-1}$ → **grado salida nodo i**

B. CENTRALIDAD POR PROXIMIDAD (distancia, longitud camino)

i. Grafo no dirigido: $C_C(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}$ → **nº enlaces en el camino más corto de i a j**

ii. Grafo dirigido: se tiene en cuenta la dirección de los ejes

Web Structure Mining

C. CENTRALIDAD POR ESTAR EN MEDIO (estar en muchos caminos)

i. Grafo no dirigido: $C_B(i) = \sum_{j<k} \frac{p_{jk}(i)}{p_{jk}}$

$p_{jk}(i)$ → nº caminos cortos entre j y k que pasan por i

p_{jk} → nº caminos cortos entre j y k

$$C_{Bnorm}(i) = \frac{2 \cdot \sum_{j<k} \frac{p_{jk}(i)}{p_{jk}}}{(n-1)\cdot(n-2)}$$

ii. Grafo dirigido: idem pero multiplicado por 2

Web Structure Mining

PRESTIGIO: Un actor de prestigio es aquel que recibe muchos ejes (grafos dirigidos).

A. PRESTIGIO POR GRADO: $P_D(i) = \frac{d_I(i)}{n-1}$ → nº enlaces entrada nodo i

B. PRESTIGIO POR PROXIMIDAD: considera actores que directa o indirectamente se enlazan a i.

distancia media ← $d_m(i) = \frac{\sum_{j \in I_i} d(j,i)}{|I_i|}$ → distancia camino mas corto de j a i

→ actores que alcanzan i

Web Structure Mining

$$P_P(i) = \frac{|I_i| / (n-1)}{\sum_{j \in I_i} d(j,i) / |I_i|}$$

→ proporción actores que alcanzan i

→ distancia media

C. PRESTIGIO POR ORDENAMIENTO: tiene en cuenta los actores que eligen o votan a otros.

$$P_R(i) = A_{1i} \cdot P_R(1) + A_{2i} \cdot P_R(2) + \cdots + A_{ni} \cdot P_R(n)$$

con A_{ji}=1 si j apunta a i y 0 en caso contrario.

Web Structure Mining

- Si $P = (P_R(1), \cdots, P_R(n))^T$

y A es la matriz de adyacencia entonces

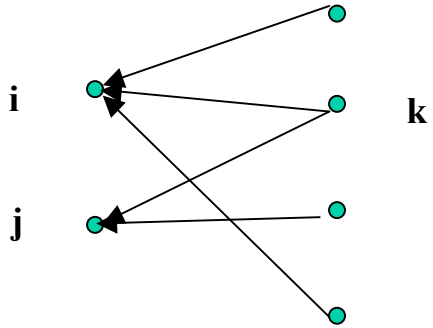
$$P = A^T P$$

álgebra lineal: P es un eigenvector de A^T

Web Structure Mining

ANÁLISIS DE CITACIONES: es un área de la investigación bibliométrica.

A. CO-CITACION: mide la similitud de 2 documentos. Si i y j están citados por k están relacionados. Cuantos más k's, más fuerte es la relación.



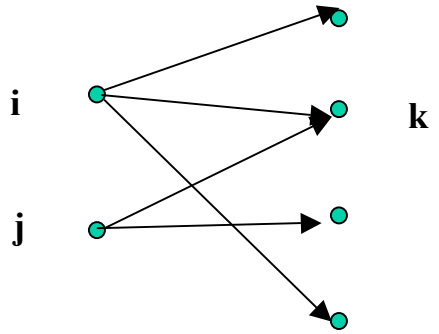
- Matriz de citaciones: L_{ij}=1 si i cita a j, 0 en otro caso.
- Co-citación C_{ij} es el nº de documentos que citan a i y j

$$C_{ij} = \sum_{k=1}^{n} L_{ki} \cdot L_{kj}$$

- C_{ii}=documentos que citan a i

Web Structure Mining

B. ENLAZADO BIBLIOGRÁFICO: enlaza documentos que citan a los mismos documentos, incluso aunque no se citen entre sí (la imagen espejada de la co-citación)



$$B_{ij} = \sum_{k=1}^{n} L_{ik} \cdot L_{jk}$$ → **documentos citados por i y j**

- B_{ii}= documentos citados por i

Web Structure Mining

Web Structure Mining: GRAFO DE ENLACES:

Cada página es un nodo y cada hipervínculo de página a página, constituye un arco dirigido.

- Los enlaces duplicados se ignoran.
- Los enlaces entre páginas del mismo dominio se ignoran (no son autorizativos y suelen ser de navegación (back, ...)).

Web Structure Mining

Web Structure Mining: HITS (análisis de citaciones)

Ejemplo: El sistema *Clever* (Chakrabarti et al. 1999). Analiza los hiperenlaces para descubrir:

- **autoridades**, que proporcionan la mejor fuente sobre un determinado tema.
- "**hubs**", que proporcionan colecciones de enlaces a autoridades.

Construcción de un grafo ponderado de enlaces:

Siendo: x_p = peso de autoridad del nodo p.

y_p = peso de hub del nodo p.

Los valores se ajustan siguiendo unas simples fórmulas de propagación de refuerzo:

Web Structure Mining

Construcción de un grafo ponderado de enlaces (Clever):

Se inicializan todos los x_p y y_p a constantes positivas y se ajustan *iterativamente* de la siguiente manera hasta que $\Delta x<\varepsilon$ y $\Delta y<\varepsilon$:

$$x_p = \Sigma_{q \text{ such that } q \to p} y_q$$

$$y_p = \Sigma_{q \text{ such that } p \to q} x_q$$

Construyendo una matriz adyacente a partir del grafo $A_{ij} = 1$ si el nodo *i* hace referencia al nodo *j*, se puede hacer un análisis de álgebra lineal y ver que converge (eigenvalues):

$$\vec{x} \leftarrow A^T \vec{y} \leftarrow A^T A\vec{x} = (A^T A)\vec{x}$$

$$\vec{y} \leftarrow A\vec{x} \leftarrow AA^T \vec{y} = (AA^T)\vec{y}$$

Web Structure Mining

Web Structure Mining: PageRank (prestigio en redes sociales)

- Usando la matriz de adyacencia A_{ij}=1 si i→j, =0 en otro caso
  - grado de salida del nodo i: $d_o(i) = \sum_j A_{ij}$
  - como la web no está fuertemente conectada, en cada nodo se hace una elección
    - con probabilidad d (0.1≤d≤0.2)se pasa a una página aleatoria
    - con probabilidad (1-d) se pasa a una página vecina a través de un enlace
  - Las páginas se visitan con distintos ratios, siendo las más populares (con muchos enlaces) las que se visitan más frecuentemente

$$PageRank(i) = \frac{d}{n} + (1-d)\sum_{j \to i} \frac{PageRank(j)}{d_o(j)}$$

Web Structure Mining

Aproximaciones Relacionales:

Al trabajar con un grafo:

- no se pueden aplicar técnicas proposicionales.
- las técnicas ad-hoc no pueden combinar web structure mining con web content mining, porque dicha información es difícil de expresar en el grafo (hacen falta otros grafos o subgrafos).

Solución:

- Problema Relacional ⇒ Técnicas de ILP
- Se añaden predicados en el background para representar:
  - documentos enlazan con otros documentos.
  - documentos contienen palabras u otros objetos.
  - tipo de documento (.html, .doc, .txt, ...)

Web Structure Mining

Aproximaciones Relacionales (ILP):

Tras una buena codificación relacional, las reglas son bastante sencillas de extraer.

Ejemplo: Hipótesis sobre página web con papeles on-line:

```
webwithpapers(A, Topic) :-  has-word(A, Topic),
                            link-from(A, B),
                            (URL-has-word(B, 'ps');
                             URL-has-word(B, 'pdf')),
                            not is-html(B).
```

Web Usage Mining

El **Web Usage Mining** (Minería de Utilización de la Web) se centra en técnicas que puedan predecir el comportamiento del usuario cuando interacciona con la web (aunque otra información sobre la topología y relevancia de los enlaces también se puede extraer de aquí).

Esta información puede residir en:

- Clientes Web: p.ej. cookies
- Servidores.
- Proxies.
- Servidores de banner: doubleclick.com...

Previa a la minería, esta información debe ser preprocesada (eliminar reintentos, separar distintos usuarios, unir diferentes sesiones, juntar páginas con marcos, filtrar por tiempos, cribar páginas irrelevantes, etc..). Ver (Mobasher et al. 2000) o (Cooley et al. 1999).

Web Usage Mining

El resultado del preprocesado puede ser:

- datos específicos para métodos específicos.
- datos relacionales (una b.d. corriente).
- datos en XML (p.ej. Cingil et al. 2000)

Sea como sea la representación, muchas técnicas proposicionales no son útiles:

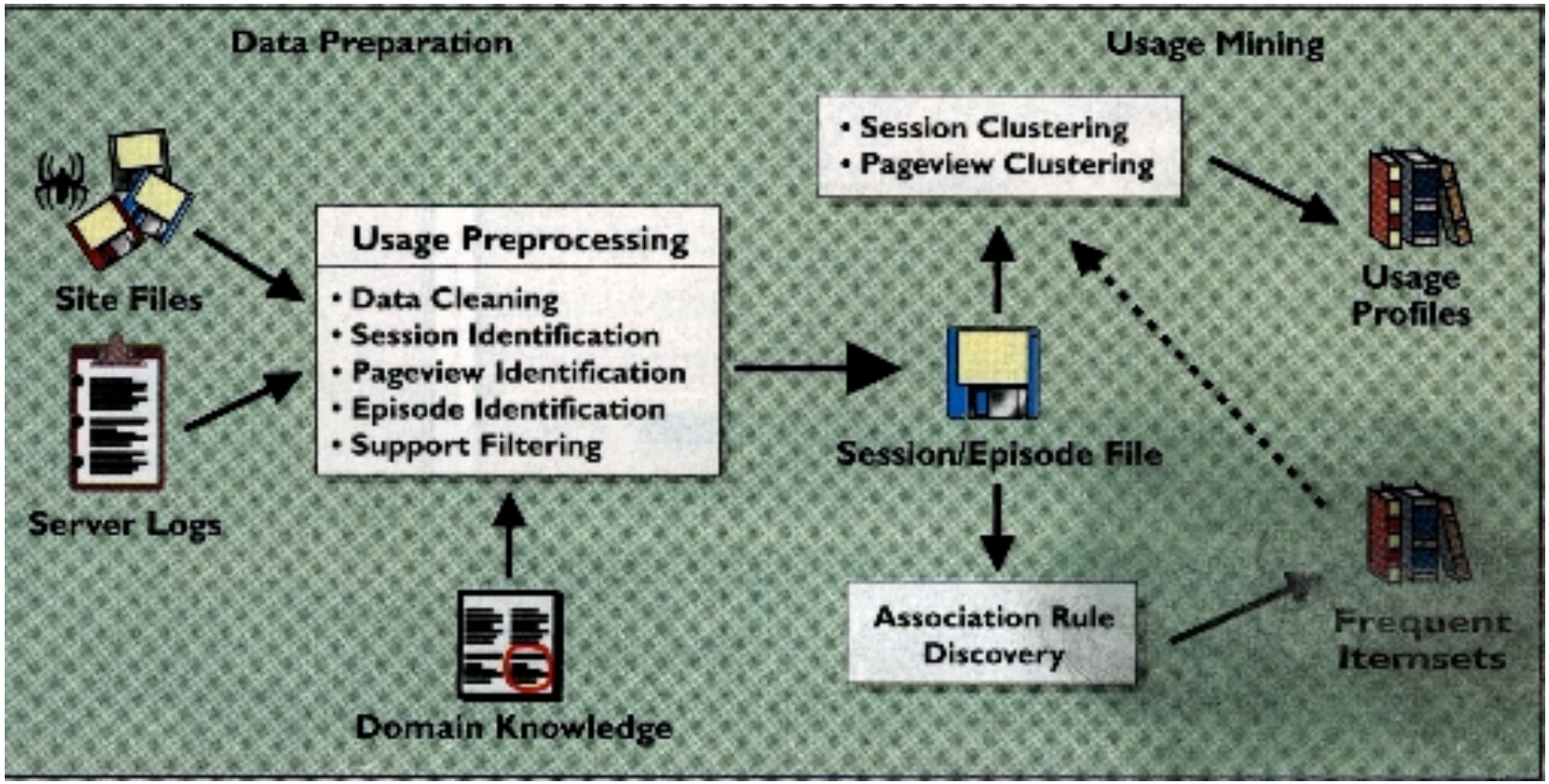
- los patrones de navegación suelen ser grafos
- se requiere de nuevo expresividad relacional.

P.ej. un predicado sencillo como 'reach' vimos que no podía ser aprendido por métodos proposicionales y sí por relacionales recursivos (p.ej. FOIL). Y este predicado es fundamental para este problema.

Además, la importancia del conocimiento previo es fundamental: estos comportamientos dependen de la topología de la red, del contenido de las páginas y de categorías de conceptos.

Web Usage Mining

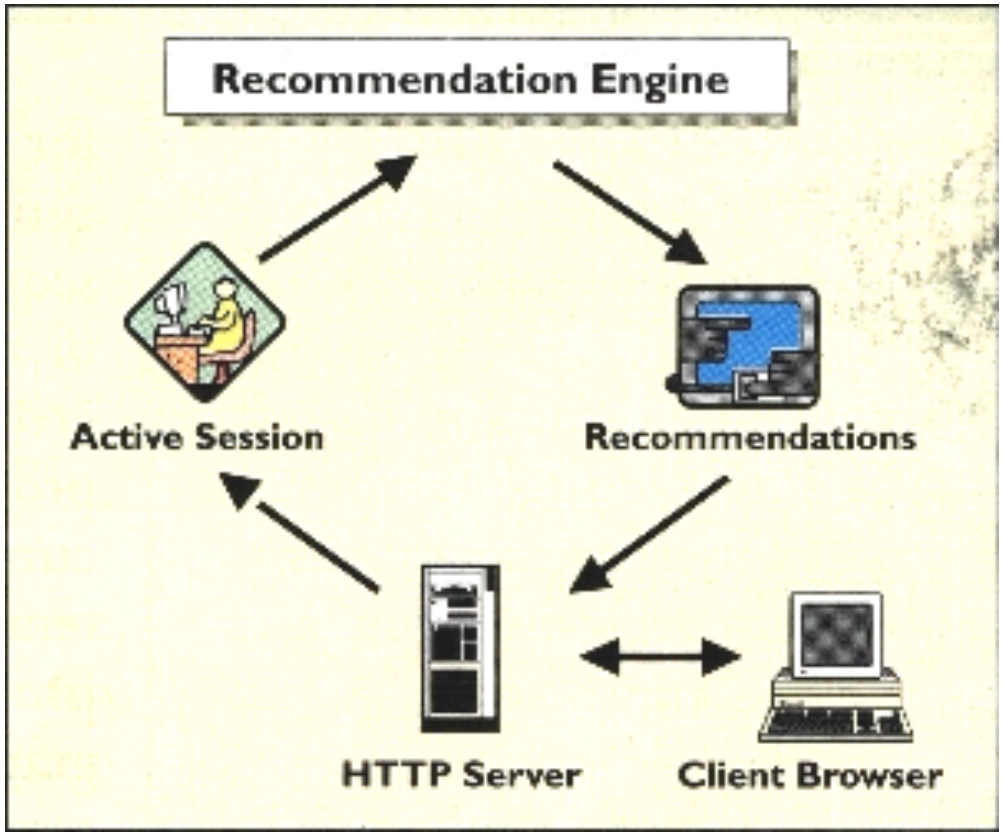
Web Usage Mining:



Batch Process (learning process)

Web Usage Mining

Web Usage Mining (una aplicación, recomendación de visitas):



On-line Process

Web Usage Mining

Buscando Patrones de Navegación:

Las sesiones o log files de navegación toman la forma de secuencias de enlaces recorridos por un usuario, conocidos como *navigation trails* o sesiones.

Las distintas sesiones de un mismo usuario se separan cuando entre la visita de un enlace y otro existe más de 30 minutos de diferencia.

- Este valor se determina como 1.5 desviación estándar de la media de tiempo entre visitas de enlaces (Borges and Levene 2000)

(También se puede utilizar porciones más pequeñas, llamadas *episodios*).

A partir de ahora consideraremos enlace como sinónimo de página, documento, URL o visita.

Web Usage Mining

Buscando Patrones de Navegación mediante HPGs (Borges and Levene 2000):

Los ‘navigation trails’ se utilizan para construir una Hypertext Probabilistic Grammar (HPG).

Una HPG es una tupla <V, Σ, S, P>. No es más que un tipo especial de gramáticas probabilísticas regulares, con la característica especial que tienen el mismo número de terminales Σ que no terminales V (con lo que se hace una correspondencia 1 a 1 entre ellos).

Se construye el grafo de transiciones de la gramática de la siguiente manera:

- Se añade un único nodo inicial S y un nodo final F, que no corresponden con ningún URL.
- Se añaden tantos nodos como URLs distintos haya en los distintos trails.

Web Usage Mining

¿Qué valores probabilísticos ponemos en las flechas?
(para saber las reglas de producción probabilística P)

Existen dos parámetros para construir esta HPG:

- α: importancia de inicio.
  - Si α=0 sólo habrá flechas de S a los nodos que han sido alguna vez inicio de sesión, y el valor de la flecha dependerá de cuántas veces lo han sido.
  - Si α=1 el peso de las flechas dependerá de la probabilidad de visitas a cada nodo, independientemente de que fueran iniciales.
  - Si α>0 habrá flechas con peso > 0 de S a todos los nodos.
- *N* (donde $N \geq 1$): valor de *N*-grama. Determina la memoria cuando se navega la red, es decir el número de URLs anteriores que pueden influir en la elección del próximo URL. Si *N*=1 el resultado será una cadena de Markov.

Web Usage Mining

Ejemplo: Supongamos la siguiente tabla de navigation trails:

ID	Trail
1	$A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_4$
2	$A_1 \rightarrow A_5 \rightarrow A_3 \rightarrow A_4 \rightarrow A_1$
3	$A_5 \rightarrow A_2 \rightarrow A_4 \rightarrow A_6$
4	$A_5 \rightarrow A_2 \rightarrow A_3$
5	$A_5 \rightarrow A_2 \rightarrow A_3 \rightarrow A_6$
6	$A_4 \rightarrow A_1 \rightarrow A_5 \rightarrow A_3$

De aquí extraemos los no terminales y los terminales correspondientes:

$$V = \{S, A_1, A_2, A_3, A_4, A_5, A_6, F\}$$

$$\Sigma = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

Tenemos 6 trails y 24 visitas, donde A_1, p.ej., fue visitada 4 veces, 2 de las cuales como página de inicio.

Por tanto, tomando p.ej. α=0.5 y N=1, podemos calcular la probabilidad de la producción $p(S \rightarrow a_1 A_1)$ que corresponde con la flecha de S a A_1 en el grafo de transiciones de la siguiente manera:

$$p(S \rightarrow a_1 A_1) = (0.5 \cdot 4)/24 + (0.5 \cdot 2)/6 = 0.25$$

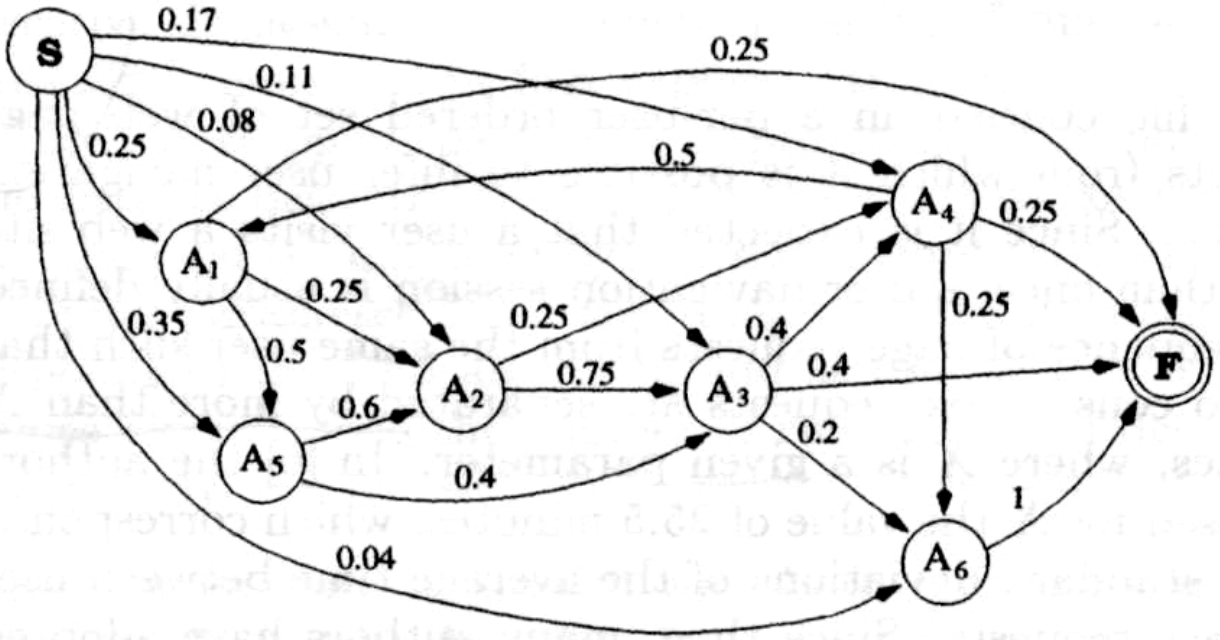
Web Usage Mining

Las flechas interiores se calculan de manera similar.

P.ej. si A_4 se ha visitado 4 veces, 1 justo antes del final, otra antes de A_6 y dos antes de A_1 tenemos:

$p(A_4 \rightarrow a_1 A_1) = 2/4$ $\quad$ $p(A_4 \rightarrow a_6 A_6) = 1/4$ $\quad$ $p(A_4 \rightarrow F) = 1/4$

Siguiendo así para el resto tenemos:



Web Usage Mining

En forma tabular podemos expresar el conjunto de producciones probabilísticas *P* derivadas del grafo anterior (para α=0.5 y *N*=1):

start prod.		transitive prod.		final prod.	
$S \rightarrow a_1 A_1$	0.25	$A_1 \rightarrow a_2 A_2$	0.25	$A_1 \rightarrow F$	0.25
$S \rightarrow a_2 A_2$	0.08	$A_1 \rightarrow a_5 A_5$	0.5	$A_3 \rightarrow F$	0.4
$S \rightarrow a_3 A_3$	0.11	$A_2 \rightarrow a_3 A_3$	0.75	$A_4 \rightarrow F$	0.25
$S \rightarrow a_4 A_4$	0.17	$A_2 \rightarrow a_4 A_4$	0.25	$A_6 \rightarrow F$	1.0
$S \rightarrow a_5 A_5$	0.35	$A_3 \rightarrow a_4 A_4$	0.4		
$S \rightarrow a_6 A_6$	0.04	$A_3 \rightarrow a_6 A_6$	0.2		
		$A_4 \rightarrow a_1 A_1$	0.5		
		$A_4 \rightarrow a_6 A_6$	0.25		
		$A_5 \rightarrow a_2 A_2$	0.6		
		$A_5 \rightarrow a_3 A_3$	0.4		

- Bueno, ¿y ahora esto para qué sirve?

Web Usage Mining

En primer lugar, permite estimar la probabilidad de cualquier 'navigation trail' todavía no producido.

Esto es útil para:

- calcular la probabilidad de llegar a una cierta página si el usuario está en una página dada.
- la prueba de aplicaciones con los trails más comunes.
- el diseño ajustado a estos trails más comunes.
- la detección de usuarios anómalos (aquéllos que realizan trails con muy baja probabilidad).

Web Usage Mining

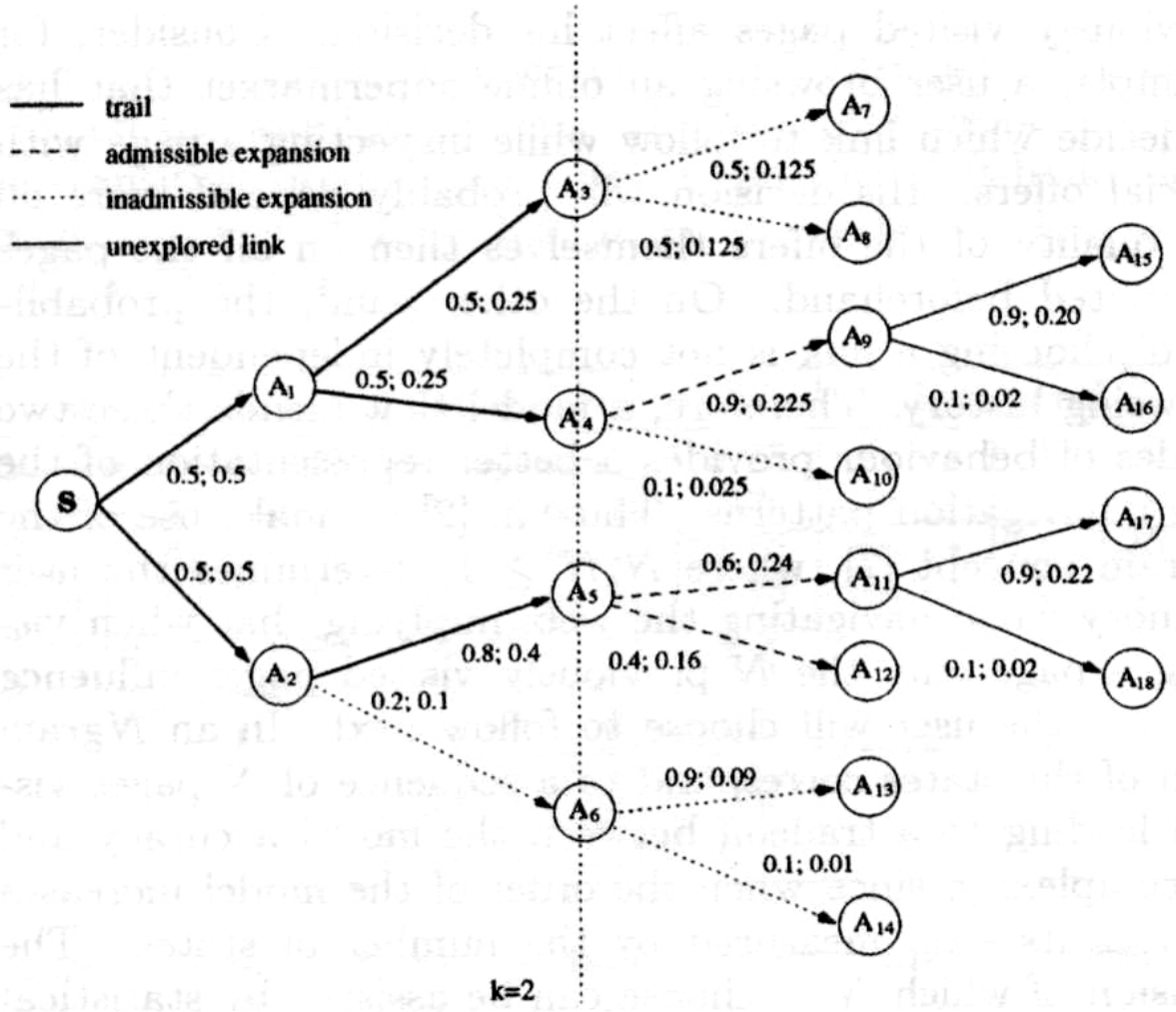
En segundo lugar, y más importante, nos interesa ver aquellos "patrones de navegación":

- Subcadenas (LARGE SUBSTRINGS) del lenguaje generado por la gramática con los siguiente parámetros:
  - θ ($0 \leq \theta \leq 1$): "*support threshold*": corresponde a que la subcadena debe comenzar al menos con cierta probabilidad. Al ser una cadena, se hace corresponder con la primera probabilidad de transacción.
  - δ ($0 \leq \theta \leq 1$): "*confidence threshold*": corresponde a que la subcadena debe tener cierta probabilidad en conjunto.
- Ambos parámetros se pueden combinar en un único punto de corte $\lambda = \theta \cdot \delta$.
- También se puede limitar la longitud de los patrones: factor *k*. Esto también influye en la eficiencia del algoritmo de búsqueda de patrones.

Web Usage Mining

Esto permite hacer una búsqueda más controlada de las expansiones admisibles:

Ejemplo (para otros trails) con k=2 y λ=0.15



Existen algoritmos más refinados y que optimizan los recursos para no buscar demasiadas subcadenas (cuando λ es alto) o muy pocas (cuando λ es bajo). (Borges and Levene 2000)

Web Usage Mining

También existen lenguajes de consulta para seleccionar patrones relativos a uso de páginas web:

P.ej. En el sistema WUM (Web Utilization Miner) (Berendt & Spiliopoulou 2000), basado también en un grafo de secuencias de visitas, se puede utilizar el lenguaje MINT para hacer consultas del estilo:

```
SELECT t
FROM NODE AS a b,
TEMPLATE a * b AS t
WHERE a.support > 7
AND (b.support / a.support) >= 0.4
AND b.url != “G.html”
```

Seleccionaría pares de páginas visitadas consecutivamente en la que la primera se ha visitado al menos 7 veces y de éstas, al menos el 40% han llegado a la segunda. Además la segunda no puede ser “G.html”.[117]

Web Usage Mining

Otros Métodos:

- Uso de métodos colaborativos:
  - Los patrones (navigation trails) de otros usuarios se utilizan para *recomendar* nuevos URLs a un usuario que visita previamente una página.
  - WebWatcher (Joachims et al. 1997) combina este método junto con métodos basados en contenido para recomendar la página siguiente a visitar o el enlace a pinchar entre los disponibles.

- Los objetivos de estos sistemas es evaluar sitios web y ayudar a rediseñarlos para captar más visitantes a las páginas deseadas (las que contengan más propaganda o la orden de compra).

Web Usage Mining

Otros Métodos:

- Uso de clustering:
  - WebPersonalizer (Mobasher et al. 2000) utiliza dos métodos:
    1. Genera clusters a partir de sesiones y después calcula la distancia de una nueva sesión a los clusters para ver el cluster más cercano y predecir la siguiente página.
    2. Utiliza el método ARHP (Association Rule Hypergraph Partitioning) (Han et al. 1997). Es un método de clustering que funciona eficientemente para un número grande de dimensiones.

- Los objetivos de este sistema es aconsejar o sugerir ...

Web Usage Mining

Perspectivas del Web Usage Mining:

- La navegación es mucho más que un recorrido de enlaces.
- Las páginas web son cada día más interactivas, existen búsquedas, formularios, ... (Mena 1999).

Por ejemplo:

IF search-keywords are “reliable software”
AND age 24-29
THEN download = ‘Linux’
WITH accuracy = 20% AND support = 1%

Tendencia hacia “Adaptive Web Sites” (Perkowitz & Ertzioni 2000).

Nos acercamos al “Application Usage Mining”, o la extracción de conocimiento automático sobre cómo un usuario utiliza una determinada aplicación software, sea distribuida o no.

Personalización

Personalización

Internet y la globalización de mercados (especialmente el informático) han hecho que gran parte de las aplicaciones que utilizan los usuarios se desarrollen genéricamente para toda la humanidad (sólo existen versiones diferentes para zonas geográficas o comunidades culturales, o diferentes tipos de instalaciones (básica, profesional, ...)).

El usuario requiere aplicaciones que se adapten a sus características. Quiere aplicaciones personalizadas.

Esto no es nada nuevo: antes de la revolución industrial todo producto y comercio era “personalizado”.

Personalización

No se trata del “software a medida”, en la que cada aplicación se hace para un solo usuario. (P.ej. los cientos de programas de contabilidad diferentes que hay). Menos aún en el caso de aplicaciones C/S o para Internet, donde no tiene sentido tener muchas versiones diferentes en el servidor.

- Tampoco se puede permitir el gran coste en personal que se requeriría para tener una atención personalizada de usuarios y clientes a través de las múltiples “ventanillas virtuales” (P.ej. telebanca).

Hay que hacer que el software sea *adaptable para la masa*!!! Es lo que se conoce como “mass customization” (Mobasher et al. 2000)

Personalización

- “El conocimiento de las preferencias del usuario y sus características (contexto, historia, etc.) ayuda a que...”

(Estrictamente Técnicas)

  - “la aplicación funcione de una manera más eficaz y agradable para el usuario”.

(Empresariales)

  - “se le ofrezca al cliente los servicios que puede requerir y de la manera y en el tiempo en que los puede ir requiriendo”
    (más relacionado con data-mining)

Personalización

NIVEL DE PERSONALIZACIÓN:

- Una aplicación se puede personalizar para un único usuario:
  - Se refuerzan o sugieren las acciones de la aplicación que agradan al usuario.
  - Se evitan o corrigen las acciones de la aplicación que no gustan al usuario.
  - Técnicas a utilizar: Reinforcement Learning, Modelado de Cadenas de Acciones (Hirsh et al. 2000).

- Una aplicación se puede personalizar para grupos de usuarios:
  - La información de personalización obtenida de algunos usuarios puede utilizarse para personalizar la aplicación (sobre todo al principio) de otros usuarios del mismo *grupo*.
  - Esto se conoce como *Métodos Colaborativos* (Hirsh et al. 2000) o 'word of mouth' (Shardanand & Maes 1995).
  - Técnicas a utilizar: Clustering para descubrir grupos (*eager*), probabilidades de pares (usuario, item) (*lazy*).

Personalización

MOMENTO DE LA PERSONALIZACIÓN:

- Durante el desarrollo del producto. El producto final sale personalizado.
  - Diseño más parecido al software a medida.

- Durante el uso del producto. Se conoce por "self-customizing sw":
  - El software puede venir con un mero menú de preferencias para activar o desactivar opciones, submenús, tareas, etc., manualmente. Ejemplos: aplicaciones ofimáticas, My Yahoo, My CNN, etc.
  - El software debe diseñarse para aprender patrones de conducta y adaptarse a ellos. Uso de aprendizaje automático. Ejemplos:
    - predicción de comandos de Unix. (Davison & Hirsh 1998)
    - predicción de consultas de recuperación en la web. (Padmanabhan & Mogul 1996)
    - predicción de teclas en calculadoras. (Darragh et al. 1990)
    - predicción de películas o canciones interesantes. (Basu et al. 1998) (Shardanand & Maes 1995)

Personalización

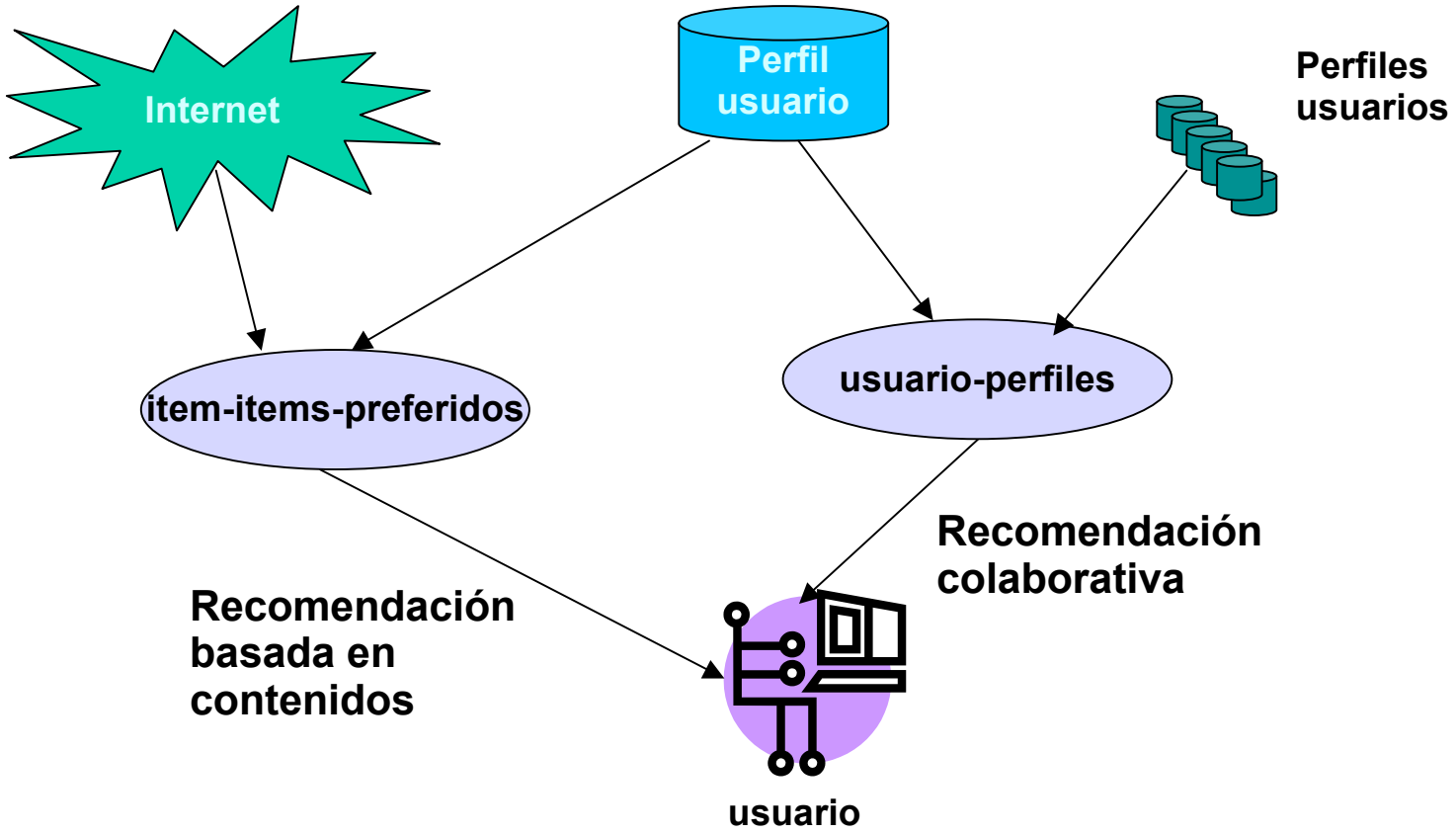
Sistemas de Recomendación: básicamente consisten en estimar una puntuación para los items no vistos por un usuario basada en las puntuaciones dadas por el usuario o por otros usuarios para otros items o para el item en cuestión, y después recomendar los items con mayor puntuación.

Elementos fundamentales

- Creación y mantenimiento del perfil del usuario
- El filtrado de la información
  - Basado en contenidos
  - Colaborativos
  - Híbridos

Personalización

Sistemas de Recomendación: Esquema general (Montaner et al. 2003)



Personalización

Perfil del usuario: Información relevante sobre sus intereses.

Creación del perfil de usuario

- **Vacío**: inicialmente es vacío y se completa a medida que el usuario interactúa con el sistema (ejemplo, Amazon.com)
- **Manual**: el sistema pregunta al usuario (palabras clave, aficiones, …)
- **Estereotipos**: son grupos de usuarios a partir de datos demográficos. El usuario rellena una hoja de registro aportando sus propios datos (nombre, dirección, ciudad, distrito, país ,edad, sexo, estilo de vida, profesión,…)
- **Conjunto de entrenamiento**: se le pide al usuario que califique (interesante/no interesante) algunos ejemplos concretos (recomendación de música, películas,..).

Personalización

Técnicas de aprendizaje de perfiles a partir de conjuntos de entrenamiento:

- **No necesarios**: usar los datos adquiridos por el sistema como perfil del usuario. Por ejemplo, sistemas de comercio electrónico (amazon.com) que usan la lista de compras como perfil.
- **Agrupamiento**: crear grupos a partir de la información de los usuarios. Por ejemplo, ACR News (Mobaster et al. 2000), las transacciones de los usuarios se representan como vectores de referencias URL. Luego se forman grupos de transacciones similares basados en la presencia de patrones de referencias URL. Finalmente, se usan para recomendar URL's interesantes en sesiones activas.
- **Clasificación**: usan los datos del item y el perfil del usuario como datos de entrada, y la salida es si el item es recomendado o no. Por ejemplo, clasificar los items en interesante/no interesante.

Personalización

Métodos de filtrado: **Filtrado basado en contenidos**.

Recomienda items basándose en la descripción de los items previamente recomendados. Sólo aquellos items con alta similitud con los preferidos por el usuario son recomendados.

Ejemplo: WebWatcher (Joachims et al. 1997) observa los enlaces a páginas web preferidos por un usuario para recomendar la página siguiente a visitar o el enlace a pinchar entre los disponibles.

Personalización

Ejemplo:
Uso de *Incremental Probabilistic Action Modeling (IPAM)* .
Reconoce patrones de comportamiento (Hirsh et al. 2000).

- Simplemente registra los comandos observados y mantiene una distribución de probabilidad para los comandos que podrían seguir. Es decir $p(\text{Comando1}_t \mid \text{Comando2}_{t-1})$.
- Los nunca elegidos también se incluyen utilizando m-estimados u otras ponderaciones para no tener probabilidades 0 y que todas sumen 1.

Resultados:

- *44% de aciertos en el siguiente comando. 75% de los casos está entre la lista sugerida de los 5 con mayor probabilidad.*
- *80% y 90% de acierto respectivamente si además se sabe la primera letra del comando.*

Personalización

Ejemplo: ***News Dude*** se ajusta a las preferencias de mensajes de news (Billsus & Pazzani 1999).

Se basa en cuatro valores de refuerzo:

- No apropiado por no interesante.
- No apropiado por redundante (es el mismo contenido que algún mensaje anterior).
- Apropiado.
- Muy apropiado. *"Consígueme más artículos de ese tema"*.

Cada vez que el usuario ojea un artículo lo califica.

El sistema mantiene dos perfiles:

- Un perfil del *interés reciente* del mensaje. Basado en co-ocurrencias de palabras. Pero sin pasarse para no ser redundante.
- Un perfil del *interés a largo plazo* del mensaje. Basado en un clasificador probabilístico.

Personalización

Métodos de filtrado: **Filtrado colaborativo.**

Hace recomendaciones sobre la base de grupos de usuarios con intereses similares.

El perfil del usuario consta de los datos especificados por el mismo. Entonces se comparan con los datos de los otros usuarios para encontrar similitudes entre los intereses. Generalmente, para cada usuario se crea un conjunto con los "usuarios (vecinos) más próximos".

Ejemplo: GroupLens (Konstan et al. 1997), computa correlaciones entre lectores de grupos de noticias Usenet comparando sus puntuaciones de nuevos artículos. Las puntuaciones de un usuario se usan para encontrar otros usuarios con puntuaciones similares para predecir el interes del usuario en nuevas noticias.

Personalización

Ejemplo: El sistema **Firefly** (Shardanand & Maes 1995) se basa en métodos colaborativos para recomendar música a los usuarios.

- *Obtiene puntuaciones de una muestra de artistas y grabaciones del usuario X.*
- *Busca otros usuarios Ys que tenga un patrón de gustos similar a X.*
- *Recomienda a X los artistas y grabaciones preferidos de los Ys.*

Personalización

Métodos de filtrado: **Filtrado híbrido**.

Métodos de combinación:

- Implementar los métodos por separado y combinarlos.
  DailyLearner, seleccciona el recomendador con mayor confianza
- Construir un modelo integrado con características de los dos tipos.
  (perfil conjunto usuario-item)
- Incorporar en el método colaborativo alguna característica basada en contenidos/ Incorporar alguna característica colaborativa en el método basado en contenidos.
  - usar perfiles de usuario basados en el análisis de contenidos en métodos colaborativos
  - reducción de la dimensionalidad sobre una colección de perfiles representados como vectores de términos

Personalización

Ejemplo: El sistema de (Basu et al. 1998) combina métodos colaborativos con métodos de contenido para recomendar películas.

Un sistema similar es el que realiza "**amazon.com**" que suele mostrar sugerencias del tipo:

"*30% of users who selected/bought this book also looked for these other topics/books".*

Personalización

Ejemplo de Refuerzo en Lenguajes Relacionales:

Se pueden utilizar medidas de refuerzo constructivo:

Ejemplo:

- Un usuario está inicialmente satisfecho porque el agente le encuentra muchos teléfonos con el siguiente modelo:

```
get-phone(Person, Phone) :- agenda(Person, Phone).

get-phone(Person, Phone) :- agenda(Person2, EmailAddress),
                            ask-for-phone(EmailAddress, Person2, Phone).
```

- Pero poco más tarde se enfada porque se da cuenta que se ha incordiado a toda la gente de su agenda.

Personalización

Ejemplo de Refuerzo en Lenguajes Relacionales (cont.):

- El usuario penaliza el agente.

El agente puede desechar la regla o refinarla:

```
get-phone(Person, Phone) :- knows(Person, Person2),
                            agenda(Person2, EmailAddress),
                            ask-for-phone(EmailAddress, Person2, Phone).
```

Utilizando información de mensajes enviados en los que ve que ciertas personas conocen a otras (por los destinatarios y los CC).

Personalización

Y si hablamos de personalización en la web y el mundo Internet...

> ¿Por qué no aplicar lo mismo a aplicaciones Intranet, Cliente/Servidor y aplicaciones clásicas de toda la vida?

Pero ya no sólo en el uso, sino en su concepción, en sus **requisitos**...

Más allá aún:

> ¿Por qué no aplicamos las técnicas de **aprendizaje** y de descubrimiento de patrones a todo el **ciclo de vida** del software?

Ése es el asunto del siguiente tema...

Más Información

Revistas (números especiales):

- **Número Especial sobre *Web Mining***, *June 2000, Vol. 2, nº1 de la revista ACM SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*
- **Número Especial sobre *Personalization***, *Communications of the ACM*, Vol. 43, no. 8, 2000.

Páginas web:

- ***Información sobre XML*** *y lenguajes de consulta:* http://www.w3.org/
- ***Información sobre Web Mining:*** http://www.webminer.com/
- ***Información sobre Intercambio de Conocimiento:*** Standards and Industry Associations for Data Mining and Decision Support: http://www.kdnuggets.com/websites/standards.html

Referencias del Tema

(Apte et al. 1994) Apte, C.; Damerau, F.; Weiss, S.M. “Automated Learning of Decision Rules for Text Categorization” ACM Transactions on Information Systems", 12 (3), pp. 233-251, 1994.

(Basu et al. 1998) Basu, C.; Haym, H.; Cohen, W.W. “Recommendation as classification: Using social and content-based information in recommendation” in *Proc. of the Fifteenth National Conference on Artificial Intelligence*, 1998.

(Berendt & Spiliopoulou 2000) Berendt, B.; Spilopoulou, M. “Analyzing navigation behavior in Web sites integrating multiple information systems” *VLDB Journal*, Special Issue on Databases and the Web 9, 1, 2000, 56-75.

(Billsus & Pazzani 1999) Billsus, D; Pazzani, M. “A hybrid user model for news story classification”, in *Proc. of the Seventh International Conference on User Modeling*, June 1999.

(Borges and Levene 2000) Borges, J.; Levene, M. “A Fine Grained Heuristic to Capture Web Navigation Patterns” *SIGKDD Explorations*, Vol. 2, Issue 1, pp. 40-50.

(Brown et al. 1994) Brown, C.M.; Danzig, P.B.; Hardy, D.; Manber, U.; Schwartz, M.F. “The harvest information discovery and access system” in Proc. of the 2nd International World Wide Web Conference, 1994, pp. 763-771.

(Chakrabarti et al. 1999) Chakrabarti, S. et al. “Mining the Web’s Link Structure” Computer, August 1999, pp. 60-67.

(Chakrabarti 2000) Chakrabarti, S. “Data Mining for hypertext: A tutorial survey” ACM SIGKDD Explorations, 1(2):1-11, 2000.

(Chamberlin et al. 2000) Don Chamberlin, Jonathan Robbie, Daniela Florescu, "Quilt; An XML Query Language for Heterogeneous Data Source" , Proc. of the workshop on Web and databases (WebDb), in conj. with SIGMOD'00 , Addison-Wesley ,Dallas, Texas , May , 2000.

Referencias del Tema

(Cingil et al. 2000) Cingil, I.; Dogac, A.; Azgin, A. "A Broader Approach to Personalization" *Communications of the ACM*, Vol. 43, no. 8, 2000.

(Cooley et al. 1997) Cooley, R.; Mobasher, B.; Srivastava, J. "Web Mining: Information and pattern discovery on the world wide web" in Proc. of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), 1997.

(Cooley et al. 1999) Cooley, R.; Mobasher, B.; Srivastava, J. "Data preparation for mining World Wide Web browsing patterns" *Journal of Knowledge and Information Systems* 1, 1, 1999.

(Darragh et al. 1990) Darragh, J.J.; Witten, I.H.; James, M.L. "The reactive keyboard: A predictive typing aid" *IEEE Computer* 23, 11, Nov. 1990, 41-49.

(Davison & Hirsh 1998) Davison, B.D.; Hirsh, H. "Predicting sequences of user actions" *Predicting the Future: AI Approaches to Time-Series Problems*. Tech. Report WS-98-07, AAAI Press.

(Deutsch et al. 1998-1999) Deutsch, A.; Fernández, M.; Florescu, D.; Levy, A.; Suciu, D. "A Query Language for XML" en http://www.research.att.com/~mff/files/final.html.

(Etzioni 1996) Etzioni, O. "The World-Wide Web. Quagmire or Gold Mine" Communications of the ACM, November 1996, Vol. 39, no.11.

(Feldman & Hirsh 1997) Feldman, R.; Hirsh, H. "Exploiting Background Information in Knowledge Discovery from Text" *Journal of Intelligent Information Systems* 9, 83-97, 1997.

(Furnas et al. 1987) Furnas, G.W. et al. "The vocabulary problem in human system communication" *Communications of the ACM*, 30, n.11, Nov. 1987.

(Goldman and Widom 1999) Goldman, R.; Widom, J. "Approximate dataguides" in *Proc. of the Workshop on Query Processing for Semistructured Data and Non-Standard Data Formats*, 1999.

Referencias del Tema

(Grumbach and Mecha 1999) Grumbach, S.; Mecca, G. “In Search of the Lost Schema” in *Database Theory, ICDT’99, 7th International Conference,* pages 314-331, 1999.

(Hammond et al. 1995) Hammond, K.; Burke, R.; Martin, C; Lytinen, S. “FAQ finder: A case-based approach to knowledge navigation” in Working Notes of the AAAI Spring Symposium: Information gathering from Heterogeneous, Distributed Environments, 1995, AAAI Press, Standford Unviersity, pp. 69-73.

(Han et al. 1997) Han, E.; Karypis, G.; Kumar, V.; Mobasher, B. “Clustering based on association rule hypergraphs” in *Proc. of SIGMOD’97 Workshop on Research Issues in Data Mining and Knowledge Discovery* (DMKD’97), may 1997.

(Hearst and Hirsh 1996) Hearst, M.; Hirsh, H. (eds.) *Papers from the AAAI Spring Symposium on Machine Learning in Information Access, Stanford, March 25-27,* http://www.parc.xerox.com/istl/projects/mlia/

(Hirsh et al. 2000) Hirsh, H.; Basu, C.; Davison, B.D. “Learning to Personalize” *Communications of the ACM*, Vol. 43, no. 8, 2000.

(Joachims 1996) Joachims, T. “A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization” Computer Science Technical Report CMU-CS-96-118, Carnegi Mellon University, 1996.

(Kargupta et al. 1998) Kargupta H. et al. (eds.) *Workshop on Distributed Data Mining, The Fourth International Conference on Knowledge Discovery and Data Mining* 1998, http://www.eecc.wsu.edu/~hillo1/kdd98ws.html

Referencias del Tema

(Kosala & Blockeel 2000) Kosala, R.; Blockeel, H. "Web Mining Research: A Survey" ACM SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining, June 2000, Vol. 2, nº1, pp. 1-15.(Lang 1995) Lag, K. "Newsweeder: Learning to Filter netnews" in Preditis and Russell (eds.). *Proc. of the 12th International Conference on Machine Learning*, pp. 331-339, San Francisco, Morgan Kaufmann Publishers, 1995.

(Loh et al. 2000) Loh, S.; Wives, L.K.; Palazzo, J. "Concept-Based Knowledge Discovery in Texts Extracted from the Web" SIGKDD Explorations, Vol. 2, Issue 1, pp. 29-39.

(Mena 1999) Mena, Jesús *Data Mining Your Website*, Digital Press, July 1999, ISBN: 1-55558-2222

(Mobasher et al. 2000) Mobasher, B.; Cooley, R.; Srivastava, J. "Automatic Personalization Based on Web Usage Mining", *Communications of the ACM*, Vol. 43, no. 8, 2000.

(Nestorov et al. 1997) Nestorov, S.; Abiteboul, S.; Motwani, R. "Infering STructure in semistructured data" SIGMOD Record, 26(4), 1997.

(Padmanabhan & Mogul 1996) Padmanabhan, V.N.; Mogul, J.C. "Using predictive prefetching to improve World Wide Web latency" *Comput. Commun. Rev.* 26, 3, July 1996, 22-36.

(Perkowitz & Ertzioni 2000) Perkowitz, M.; Ertzioni, O. "Adaptive Web Sites", *Communications of the ACM*, Vol. 43, no. 8, 2000.

(Robie et al 1998) Robie, J.; Lapp, J. Schach, D. "XML Query Language (XQL)" http://www.w3.org/TandS/QL/QL98/pp/xql.html

(Rochio 1971) Rocchio, J. "Relevance feedback in information retrieval" *in The SMART retrieval system: Experiments in automatic document processing*, chap. 14, pp. 313-323, Englewood Cliffs, NJ. Prentice Hall, 1971.

Referencias del Tema

(Salton 1991) Salton, G. “Developments in Automatic Text Retrieval” *Science*, 253, 944-979, 1991.

(Sarawagi and Nagaralu 2000) Sarawagi, S. and Nagaralu, S.H. “Data Mining Models as Services on the Internet” *SIGKDD Explorations*, Vol. 2, Issue 1, pp. 24-28.

(Shardanand & Maes 1995) Shardanand, U.; Maes, P. “Social Information Filtering: Algorithms for Automating ‘word of mouth’” in *Proceedings of CHI’95 Conference on Human Factors in Computing Systems,* ACM Press, 1995.

(Toivonen 1999) Toivonen, H. “On knowledge discovery in graph-structured data” in Workshop on Knowledge Discovery from Advanced Databases (KDAD’99), pages 26-31, 1999.

(W3C 1999) World Wide Web Consortium “XML Path Language (XPath) Version 1.0. W3C Recommendation, Nov. 16, 1999. http://www.w3.org/TR/xpath.html

(Wang 1999) Wang, H.L.K. “Discovering association of structure from semistructured objects” IEEE Transactions on Knowledge and Data Engineering, 1999.

(Weis 1996) Weis, G. (ed.) *Distributed artificial intelligence meets machine learning: learning in multi-agente environments*, Springer Verlag, 1996.

(Zaïane et al. 1998) Zaïane, O.R.; Han, J.; Li, Z.-N.; Chee, S.H.; Chitang, J. “Multimediaminer: a system prototype for multimedia data mining” in Proc. ACM SIGMOD Intl. Conf. on Management of Data, pages 581-583, 1998.