



REGLAS DE ASOCIACIÓN: INTRODUCCIÓN

Jesús Alcalá Fernández

Research Group: Soft Computing and Intelligent Information Systems

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Contenidos

- Aprendizaje no supervisado - contexto
- Definición de regla de asociación
- Medidas clásicas de las reglas de asociación
- Métodos clásicos de extracción de reglas
- Conjuntos maximales y cerrados
- Generación de reglas
- Problemas abiertos
- Aplicaciones

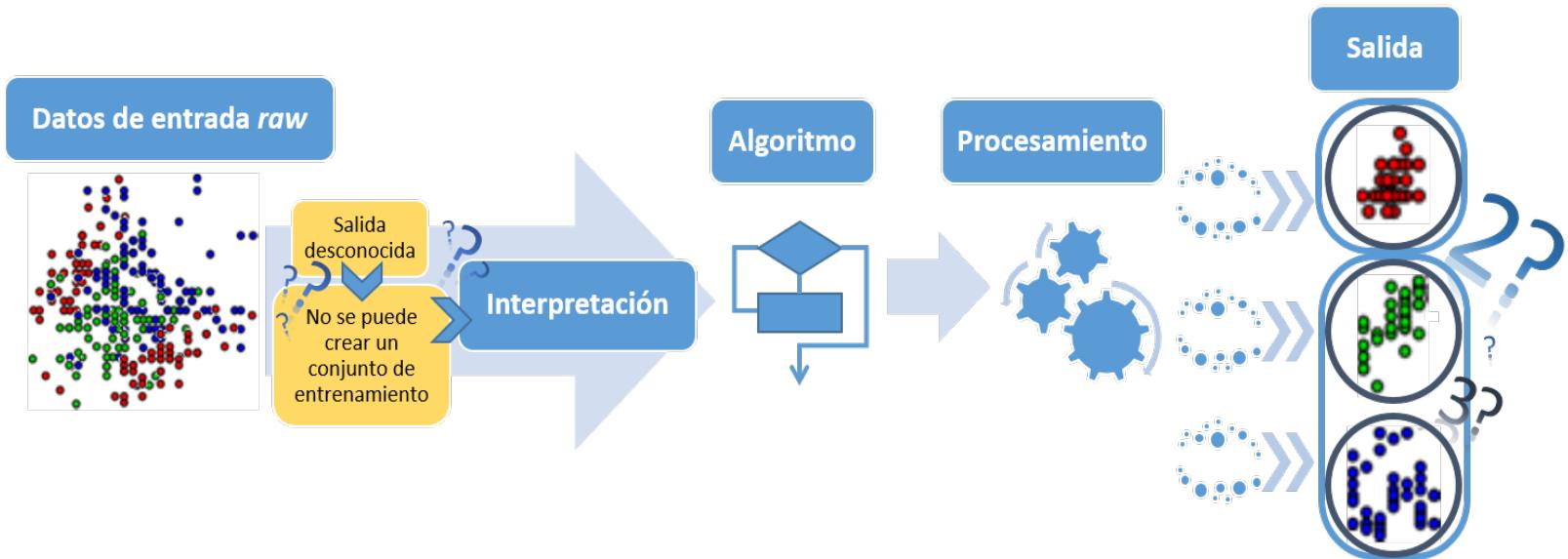
Contenidos

- Aprendizaje no supervisado - contexto
- Definición de regla de asociación
- Medidas clásicas de las reglas de asociación
- Métodos clásicos de extracción de reglas
- Conjuntos maximales y cerrados
- Generación de reglas
- Problemas abiertos
- Aplicaciones

Aprendizaje no supervisado - Contexto

Existen casos donde:

- No se dispone de una variable de salida asociada.
- Se tiene interés en descubrir otro tipo de asociaciones.



El término *no supervisado* hace referencia a que este aprendizaje **no se basa en la existencia de una respuesta previamente conocida**

Aprendizaje no supervisado - Contexto

No estamos interesados en la predicción, porque no tenemos una variable de salida asociada Y . Más bien, el objetivo es **descubrir cosas interesantes** sobre las distintas variables X_1, X_2, \dots, X_p que ayude a los expertos a conocer mejor el problema y a tomar buenas decisiones:

- ¿Hay alguna manera informativa de representar los datos?
- ¿Podemos descubrir subgrupos entre las instancias?
- ¿Sería posible encontrar relaciones de interés entre las propias variables?



Aprendizaje no supervisado - Contexto

- El **aprendizaje no supervisado suele ser mucho más desafiante.**
- Su aplicación tiende a ser más subjetiva y no existe un único objetivo claro para el análisis, como la predicción de una variable de salida.
- A menudo se realiza como parte de un análisis exploratorio de datos.
- Resulta complejo evaluar los resultados ya que no existe un mecanismo universalmente aceptado para llevar a cabo una “validación independiente”:
 - Un modelo predictivo supervisado se puede verificar con respecto la variable de salida Y en instancias no utilizadas para ajustar el modelo.
 - Para un modelo no supervisado nunca sabemos la respuesta verdadera.



Aprendizaje no supervisado - Contexto

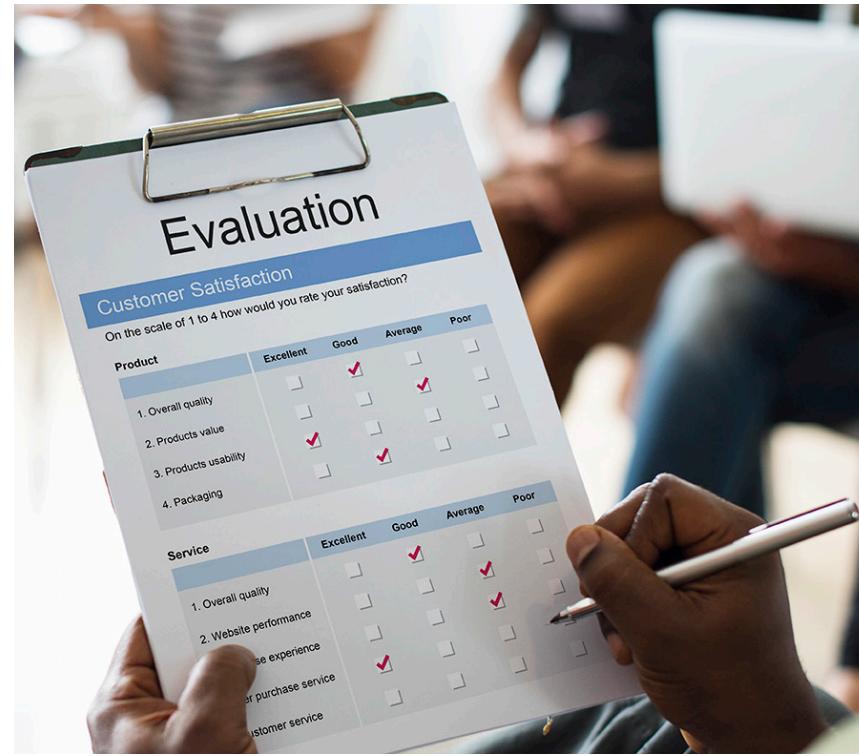
Presenta los mismos problemas que en el aprendizaje supervisado, incluso más agravados:

- **Número de instancias:** Necesitamos un mínimo de instancias que sean representativas de lo que sucede en el problema.
- **Variables de entrada:** Todas las variables son de entrada y pueden existir variables “confounding”. [variables que confunden al método](#)
- **Sesgo de los datos:** Se ve gravemente afectado por los desequilibrios en la distribución de cualquiera de las variables.
- **Heterogeneidad de las variables:** Muchas de estas técnicas se basan en medidas de distancia y similitud, por lo que algunas técnicas no pueden utilizar variables categóricas y necesitan que todas tomen valores en el mismo dominio.
- **Preprocesamiento de los datos:** Es fundamental conocer bien las variables del problema, permitiendo introducir variables nuevas que resuman la información proporcionada por otras variables o complementen la información que aportan por separado.



Aprendizaje no supervisado - Contexto

- No tiene sentido aplicar ningún tipo de validación basada en el conocimiento de ningún tipo de valores de salida.
- La pregunta a responder en este caso sería: ¿es la información que hemos descubierto fiable o interesante?
- Por ejemplo, en bioinformática sería: ¿está la información que hemos descubierto soportada por los datos y a su vez tiene fundamentación biológica?
- Existen en la literatura muchas medidas para valorar el interés del conocimiento extraído.
- No hay una medida que sea la mejor:
 - a) utilizar un conjunto de medidas para valorar el interés desde distintos puntos de vista.
 - b) comprobación/validación realizada por los expertos en el área correspondiente.



Aprendizaje no supervisado - Contexto

Reglas de asociación



Clustering



Contenidos

- Aprendizaje no supervisado - contexto
- **Definición de regla de asociación**
- Medidas clásicas de las reglas de asociación
- Métodos clásicos de extracción de reglas
- Conjuntos maximales y cerrados
- Generación de reglas
- Problemas abiertos
- Aplicaciones

Definición: Reglas de Asociación

- Las **reglas de asociación han sido una de las técnicas de minería de datos** más utilizada para extraer conocimiento interesante a partir de bases de datos grandes.
- Son utilizadas para identificar y representar dependencias entre elementos (**items**) de una base de datos en la que no conocemos la clase a la que pertenecen los datos (**aprendizaje no supervisado**).
- La reglas de asociación son reglas que tienen el siguiente $X \rightarrow Y$, donde X e Y son conjunto de items (**itemset**) que cumplen $X \cap Y = \emptyset$
- Ejemplo clásico:

Pizza → Coca Cola



(Toda persona que compra Pizza compra Coca Cola)

Definición: Reglas de Asociación

- Inicialmente, la extracción de reglas de asociación se aplicó al análisis de base de datos con información sobre los productos comprados por clientes de un supermercado. Este es un caso particular donde:
 - I: Artículos de un supermercado, que son los **items**
 - T: Conjunto de cestas de compra o ventas, que son el conjunto de transacciones. Cada transacción es un subconjunto de items (**itemset**).
- Ejemplos de reglas:

Leche y Pan → Mantequilla

Leche y Pan → Pan y Mantequilla

No pueden estar los mismos items en el antecedente y en el consecuente de la regla. En este ejemplo, el item "Pan" solo puede estar en uno de los dos sitios.



Definición: Reglas de Asociación

- Primer paso en la aplicación de reglas de asociación: determinar sobre nuestra base de datos qué son los items y cuáles son las transacciones (puede haber diversas posibilidades).
- Los items son los elementos a asociar. Las transacciones definen casos particulares de una relación entre items (por ejemplo, compra conjunta).
- Tipos de items:
 - Cada registro es un listados elementos, no hay variables. En este caso un item es uno de los posibles elementos.
 - La BD contiene un número fijo de variables (también llamados atributos) y cada registro de la BD contiene un valor para cada variable. En este caso un item es una par (atributo, valor).

Definición: Reglas de Asociación

Ejemplo de base de datos: Datos de Clientes

- **Items:** Parejas (atributo,valor)
- **Transacciones:** Registros
- **Regla:**
 $(\text{Sueldo,alto}) \rightarrow (\text{Estudios,Superiores})$

DNI	Puesto	Sueldo	Estudios
1111111111	Administrativo	Bajo	Medios
2222222222	Programador	Medio	Medios
3333333333	Analista	Medio	Superiores
4444444444	Gerente	Alto	Superiores

	Transacción
1	(Puesto,Administrativo), (Sueldo,Bajo), (Estudios,Medios)
2	(Puesto,Programador), (Sueldo, Medio), (Estudios,Medios)
3	(Puesto,Analista), (Sueldo,Medio), (Estudios,Superiores)
4	(Puesto,Gerente), (Sueldo,Alto), (Estudios,Superiores)

- Todo el que tiene un salario alto tiene estudios superiores
- Salario alto implica estudios superiores

no se incluye DNI porque es diferente para cada uno, no aporta nada

Contenidos

- Aprendizaje no supervisado - contexto
- Definición de regla de asociación
- **Medidas clásicas de las reglas de asociación**
- Métodos clásicos de extracción de reglas
- Conjuntos maximales y cerrados
- Generación de reglas
- Problemas abiertos
- Aplicaciones

Medidas Clásicas: Soporte y Confianza

- Un itemset puede estar formado por 1, 2, 3, etc, items. Para determinar cuantos items componen el itemset se suelen indican como **k-itemset**, donde k es el número de items que contienen.
 - Medidas clásicas (Soporte y Confianza)
-
- **Soporte:** Medida clásica de importancia.
 - De un Itemset (X): Es la frecuencia con la que el itemset ocurre en la base de datos.
 - a) Soporte (X) = n° de ocurrencias de X / total de transacciones en la BD
 - b) Soporte (X) = Probabilidad del itemset X en el conjunto de transacciones ($p(X)$)
 - De una Regla de Asociación ($X \rightarrow Y$): Es la frecuencia con la que ocurre el itemset $X \cup Y$.
 - a) Soporte ($X \rightarrow Y$) = Soporte ($X \cup Y$) = n° de ocurrencias de $X \cup Y$ / total de transacciones en la BD
 - b) Soporte ($X \rightarrow Y$) = Probabilidad del itemset $X \cup Y$ en el conjunto de transacciones ($p(X \wedge Y)$)

Esta medida toma valores en el rango [0.0, 1.0]. Soporte 1.0 indica que aparece en todas las transacciones de la BD, y 0 que no aparece en ninguna.

Medidas Clásicas: Soporte y Confianza

Registro de 6 ventas en un supermercado con 4 productos/items (i_1, i_2, i_3, i_4). Cada fila representa una venta, donde un 1 indica que el producto ha sido comprado y un 0 que no está incluido en la venta.

Base de Datos

i_1	i_2	i_3	i_4
1	0	1	0
0	0	0	0
0	1	1	0
0	1	1	1
1	1	1	1
1	1	1	1

Soportes:

$Sop(i_1): 3/6; Sop(i_2): 4/6; Sop(i_3): 5/6; Sop(i_4): 3/6;$
 $Sop(i_1i_2): 2/6; Sop(i_1i_3): 3/6; Sop(i_1i_4): 2/6;$
 $Sop(i_2i_3): 4/6; Sop(i_2i_4): 3/6; Sop(i_3i_4): 3/6;$
 $Sop(i_1i_2i_3): 2/6; Sop(i_1i_2i_4): 2/6; Sop(i_2i_3i_4): 3/6;$
 $Sop(i_1i_2i_3i_4): 2/6;$

Medidas Clásicas: Soporte y Confianza

□ Confianza de una Regla de Asociación ($X \rightarrow Y$) : Medida clásica de cumplimiento.

- a) Confianza ($X \rightarrow Y$) = Soporte ($X \rightarrow Y$) / Soporte (X)
- b) Confianza ($X \rightarrow Y$) = $p(X | Y) = p(X \wedge Y) / p(X)$

Esta medida toma valores en el rango [0.0, 1.0]. Confianza 1.0 indica siempre que ocurre X también ocurre Y, 0 que cuando ocurre X no ocurre Y.

Medidas Clásicas: Soporte y Confianza

Ejemplo para la confianza:

BD:

i_1	i_2	i_3	i_4
1	0	1	0
0	0	0	0
0	1	1	0
0	1	1	1
1	1	1	1
1	1	1	1

Confianza de algunas reglas:

$$\text{Confianza } (i_1 \rightarrow i_2) = \text{Sop}(i_1i_2) / \text{Sop}(i_1) = (2/6) / (3/6) = 2 / 3 = 0.67$$

$$\text{Confianza } (i_2 \rightarrow i_3) = \text{Sop}(i_2i_3) / \text{Sop}(i_2) = (4/6) / (4/6) = 1$$

$$\text{Confianza } (i_3 \rightarrow i_4) = \text{Sop}(i_3i_4) / \text{Sop}(i_3) = (3/6) / (5/6) = 3 / 5 = 0.6$$

$$\text{Confianza } (i_4 \rightarrow i_3) = \text{Sop}(i_4i_3) / \text{Sop}(i_4) = (3/6) / (3/6) = 1$$

$$\text{Confianza } (i_1i_4 \rightarrow i_2) = \text{Sop}(i_1i_4i_2) / \text{Sop}(i_1i_4) = (2/6) / (2/6) = 1$$

¿Debemos darle importancia a los casos con un soporte alto?

Soporte alto son cosas obvias (evidentes al observar el dataset), cosas con soporte bajo son cosas nada obvias (Anomalías)

Por tanto las reglas generales están bien (soporte alto) y las bajas tambien, son excepciones y cosas raras, te puede interesar

Soportes:

Dependiendo del problema se buscan unas cosas u otras

$$\text{Sop}(i_1): 3/6; \text{Sop}(i_2): 4/6; \text{Sop}(i_3): 5/6; \text{Sop}(i_4): 3/6;$$

$$\text{Sop}(i_1i_2): 2/6; \text{Sop}(i_1i_3): 3/6; \text{Sop}(i_1i_4): 2/6;$$

$$\text{Sop}(i_2i_3): 4/6; \text{Sop}(i_2i_4): 3/6; \text{Sop}(i_3i_4): 3/6;$$

$$\text{Sop}(i_1i_2i_3): 2/6; \text{Sop}(i_1i_2i_4): 2/6; \text{Sop}(i_2i_3i_4): 3/6;$$

$$\text{Sop}(i_1i_2i_3i_4): 2/6;$$

la confianza no es la misma si se intercambian antecedentes y consecuentes

Las reglas que se obtienen de un mismo itemset pueden tener igual soporte pero pueden tener diferente confianza

Contenidos

- Aprendizaje no supervisado - contexto
- Definición de regla de asociación
- Medidas clásicas de las reglas de asociación
- **Métodos clásicos de extracción de reglas**
- Conjuntos maximales y cerrados
- Generación de reglas
- Problemas abiertos
- Aplicaciones

Extracción de Reglas de Asociación

- Dado un conjunto de transacciones T , el objetivo del proceso de extracción es encontrar todas las reglas que tengan
 - Soporte \geq mínimo soporte ($minSup$)
 - Confianza \geq mínima confianza ($minConf$)

Los umbrales de $minSup$ y $minConf$ son dados por el experto del problema.

- Enfoque fuerza bruta:
 - Listar todas las reglas de asociación posibles
 - Calcular el soporte y la confianza para cada regla
 - Eliminar las reglas que no superen los umbrales de $minSup$ y $minConf$



**Computacionalmente
Prohibitivo!**

Extracción de Reglas de Asociación

- Enfoque basado en 2-pasos:

1. Generar todos los itemset frecuente

- Los itemsets frecuentes son aquellos que *suporte* es mayor o igual al umbral *minSup*.

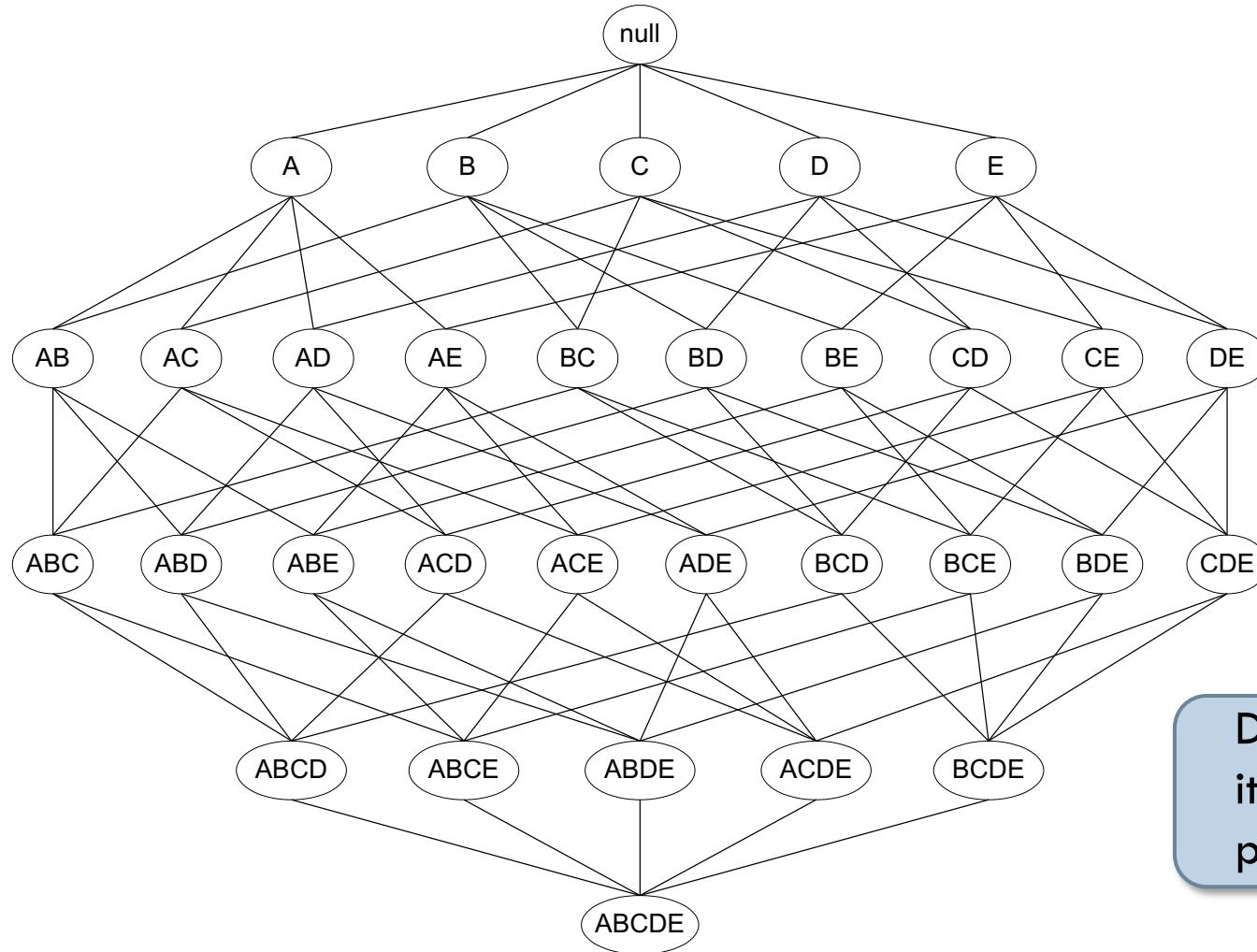
2. Generar las reglas

- Generar reglas con alta confianza a partir de los itemset frecuentes, donde cada regla es una partición binaria de un itemset frecuente.

Problema: La generación de los itemset frecuentes tiene un coste computacional muy elevado.



Generación de itemsets frecuentes

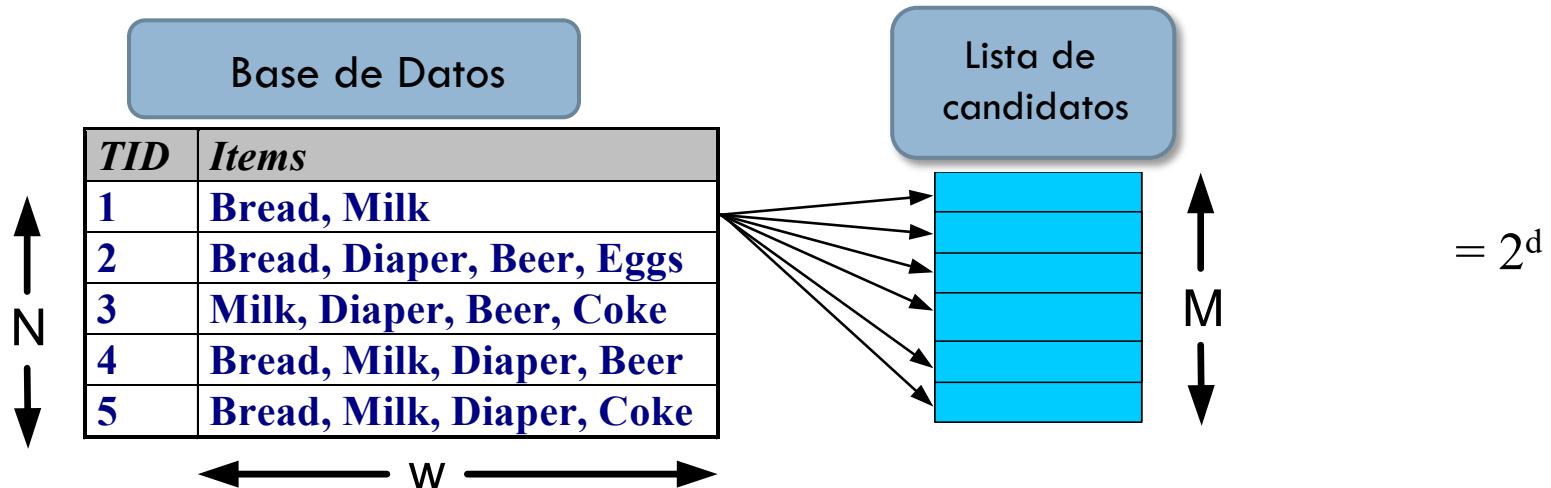


Dados d items, hay 2^d itemsets candidatos posibles

Generación de itemsets frecuentes

□ Enfoque fuerza bruta:

- Cada itemset que podamos formar es un itemset frecuente candidato
- Contamos el soporte de cada candidato recorriendo la base de datos



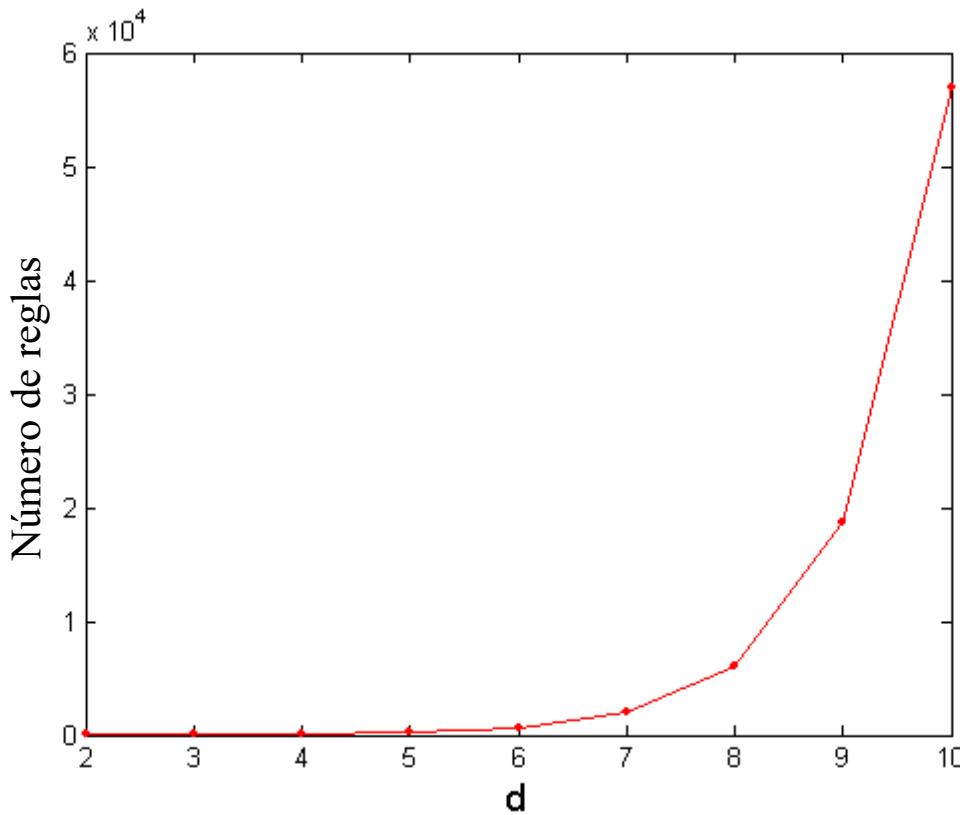
- Complejidad $\sim O(NMw)$ → **Muy costoso puesto que $M = 2^d$!!!**

Complejidad Computacional

crece demasiado rápido, con muchos casos es inviable

□ Dado d ítems:

- Número total de itemsets = 2^d
- Número total de Reglas de Asociación posibles



$$\begin{aligned} R &= \sum_{k=1}^{d-1} \left[\binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right] \\ &= 3^d - 2^{d+1} + 1 \end{aligned}$$

Si $d=6$, $R = 602$ reglas

Estrategias para generar itemsets frecuentes

- Reducir el número de candidatos (M)
 - Búsqueda completa: $M=2^d$
 - Usar técnicas de poda para reducir M
- Reducir el número de transacciones (N)
 - Reducir el tamaño de N cuando aumenta el tamaño del itemset
 - Usado por métodos de extracción basados en un enfoque vertical
- Reducir el número de comparaciones (NM)
 - Usar estructuras de datos eficientes para almacenar los candidatos o las transacciones
 - No es necesario comprobar todas las transacciones para cada candidato

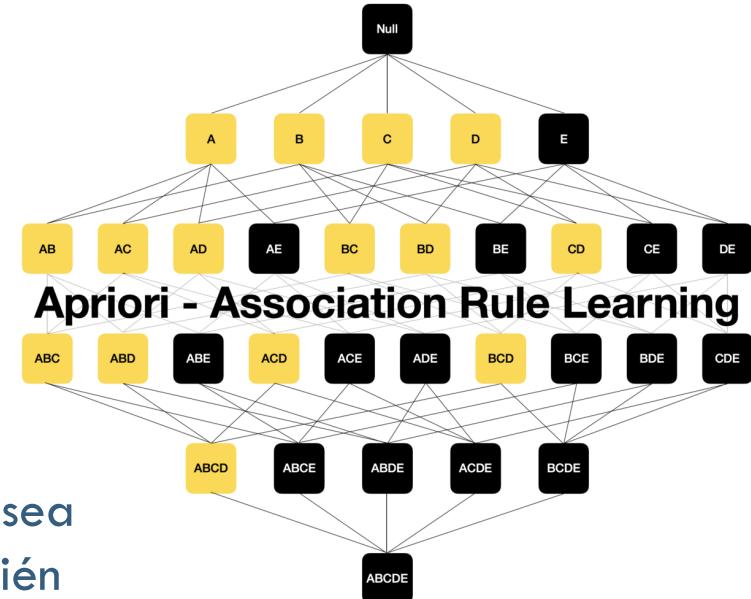
Métodos clásicos: Apriori

□ Principio de Apriori:

- Si un itemset es frecuente, entonces todos sus subconjuntos también lo son.
- Este principio es mantenido gracias a la propiedad **anti-monótona** de la medida de soporte:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- Destacar que entonces, para que un itemset sea frecuente todos sus subconjuntos tienen también que serlos.
- Además, ordena todos los itemsets para evitar generar itemsets candidatos repetidos.



Algoritmo Apriori

- **Tablas:**

L_k = Conjunto de k-itemsets que son frecuentes (**Largos**)

C_k = Conjunto de k-itemsets que pueden ser frecuentes (**Candidatos a ser frecuentes**)

- **Método:**

Supongamos k=1

Generar L₁ (itemsets frecuentes de longitud 1)

Repetir hasta que no se identifiquen itemsets frecuentes nuevos

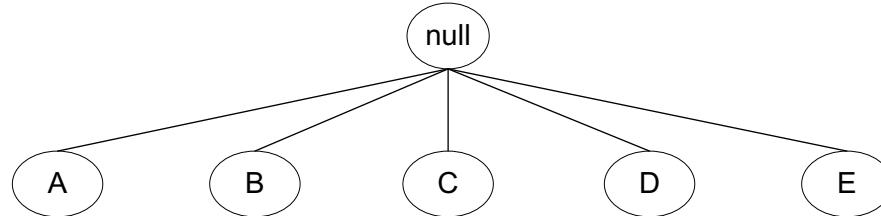
a) Generar el conjunto C(k+1) de itemsets candidatos a partir del conjunto L_k de itemsets frecuentes (combinando los itemsets frecuentes que solo se diferencien en el último ítem)

b) Calcular el soporte de cada candidato recorriendo la BD

c) Eliminar los candidatos que son infrecuentes e incluirlos en el conjunto L_{k+1}

d) k = k+1

Ilustración de Apriori

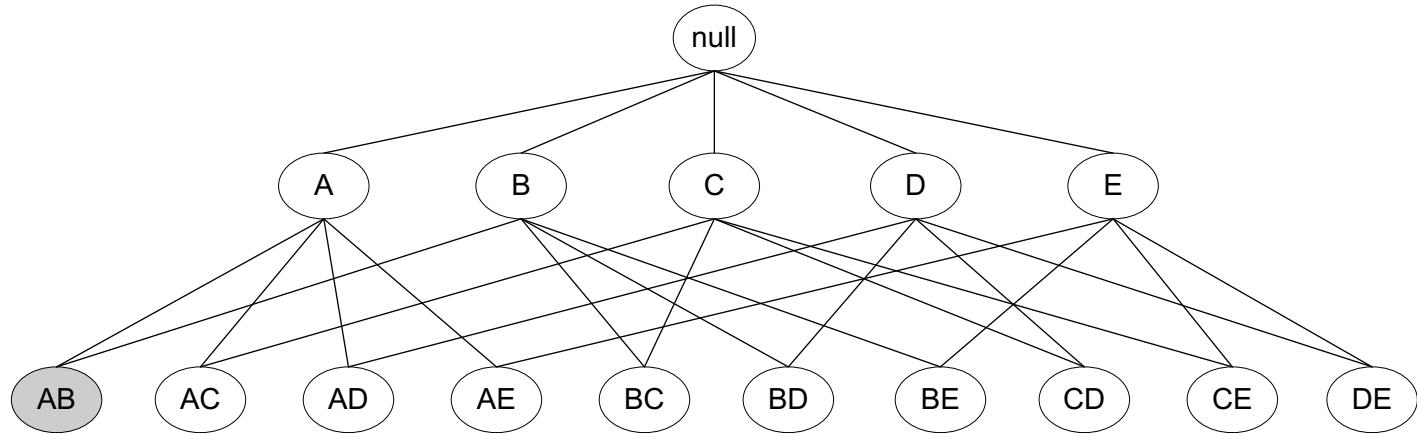


supongo que todos superan el minimo suporte dado por el jefe

Items: A, B, C, D y E

L1: A, B, C, D y E

Ilustración de Apriori



C_{k+1}: AB, AC, AD, AE, BC, BD, BE, CD, CE, DE

L_{k+1} : AC, AD, AE, BC, BD, BE, CD, CE, DE

Obtenemos que todos superan el mínimo soporte dado menos el AB

Por lo tanto no lo añadimos, ya que aunque le añadamos más cosas ya no superara ese soporte mínimo, siempre es menor o igual

Ilustración de Apriori

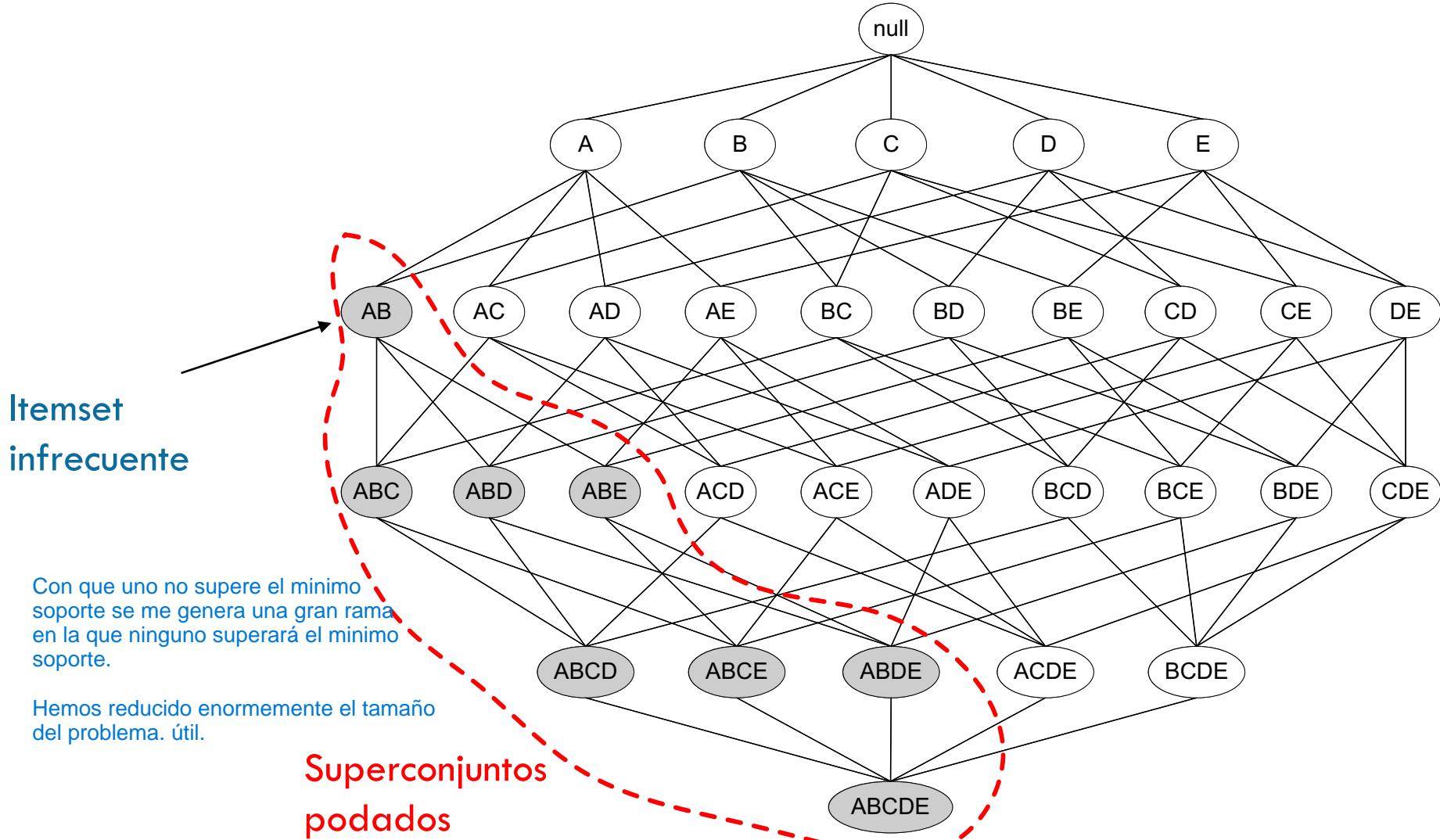


Ilustración de Apriori

Item	Cantidad
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Mínimo soporte = 3



No superan el mínimo, no interesan
No se siguen combinando

b) Recontar

Itemset	Cantidad
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk, Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

c) Eliminamos los infrecuentes

Itemset	Cantidad
{Bread,Milk,Diaper}	3

a) No generamos candidatos con {Bread, Beer} o {Milk, Beer}

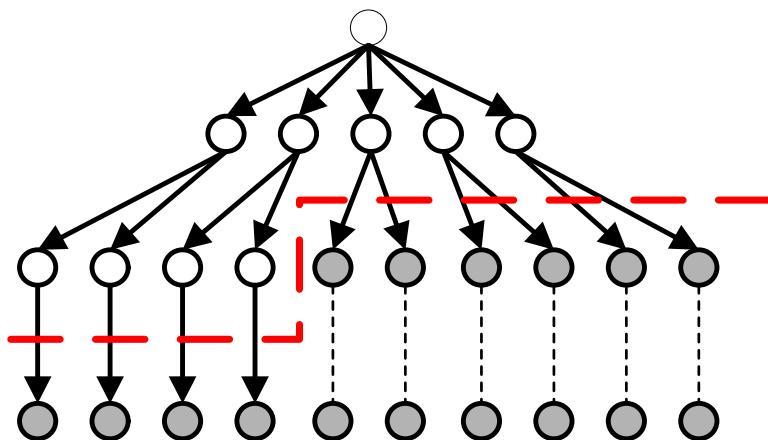
b) Recontar
c) Filtrar

Itemset
{Bread,Milk,Diaper}

Itemset	Cantidad
{Bread,Milk}	3
{Bread,Diaper}	3
{Milk,Diaper}	3
{Beer,Diaper}	3

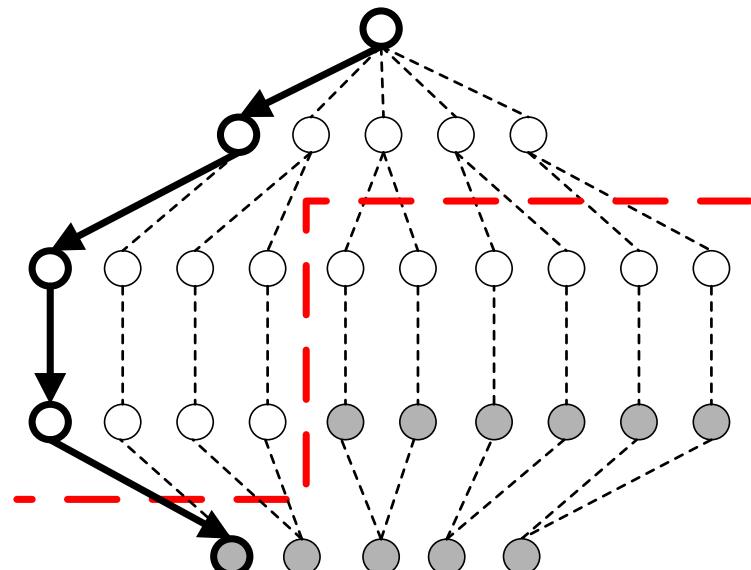
Alternativas para generar los itemsets frecuentes

□ Primero-Anchura vs. Primero-Profundidad



(a) Primero anchura

Vamos por nivel, por capas



(b) Primero profundidad

Aca se hace siguiendo una rama hasta el final si es frecuente, si no lo es subo al siguiente nodo

Son similares, se tiende a usar en profundidad por ahorro de memoria

Factores que afectan a la eficiencia de Apriori

- Elección del umbral de mínimo soporte
 - Umbrales demasiado bajos darán lugar a muchos más itemsets frecuentes, y permitirá obtener itemsets frecuentes de mucha mayor longitud.
 - Esto lleva a una mayor necesidad de espacio y de tiempo de ejecución.
- Número de items en la base de datos
 - Necesitamos más espacio para almacenar el soporte de cada itemsets.
 - Si el número de items aumenta, también aumentará el coste computacional y el tiempo necesario para las I/O.
- Tamaño de la base de datos
 - Apriori hace múltiples pasadas por toda la base de datos, por lo que el tiempo de ejecución puede incrementar con el número de transacciones.
- Longitud medida de las transacciones
 - Esto puede aumentar la longitud de los itemsets frecuentes y que por lo tanto necesitemos más espacio para almacenarlos.

Métodos clásicos: Eclat

- Mismo proceso que Apriori pero para cada itemset almacena una lista (tid-list) con la posición en la BD de las transacciones que lo contien
Para cada item se guarda en que transacciones aparece cada uno

Base de Datos

TID	Items
1	A,B,E
2	B,C,D
3	C,E
4	A,C,D
5	A,B,C,D
6	A,E
7	A,B
8	A,B,C
9	A,C,D
10	B

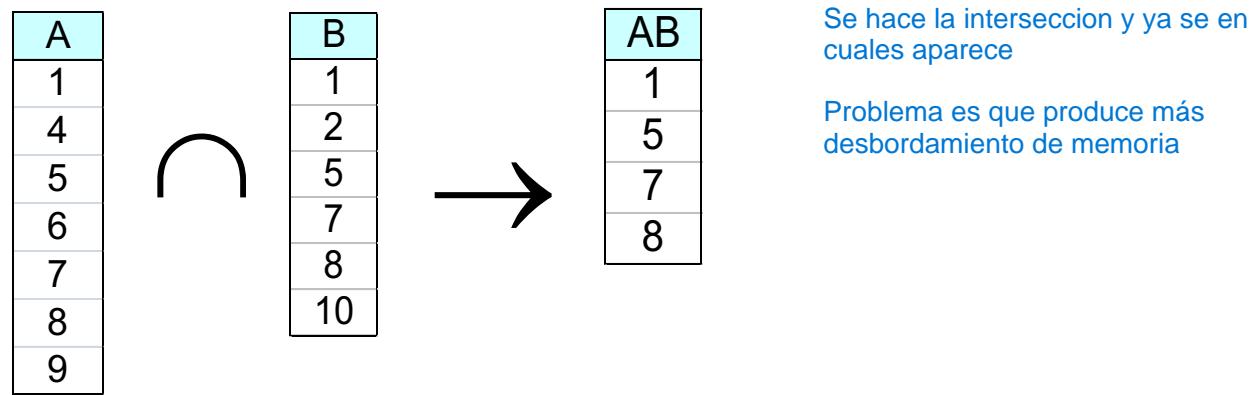
Lista para cada 1-itemset

A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

TID-list

Métodos clásicos: Eclat

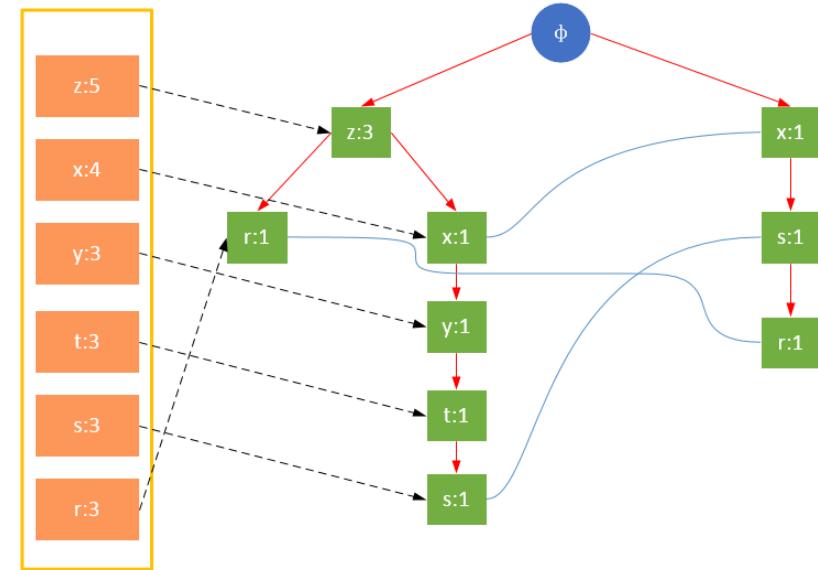
- Calcula el soporte de un k-itemset haciendo la intersección de las tid-lists de los dos de sus (k-1) subconjuntos.



- Ventaja:** El cálculo del soporte es muy rápido.
- Desventaja:** Las tid-lists intermedias pueden llegar a ser demasiado grandes para la memoria, sobre todo si la BD contiene muchas transacciones.

Métodos clásicos: FP-growth

- Crea una representación comprimida de la base de datos utilizando una estructura de árbol **FP-tree**. Esta estructura consta de:
 - ▣ Tabla cabecera: Tabla de listas. Para cada ítem de la BD hay una lista que enlaza todos los nodos del grafo donde aparece.
 - ▣ Grafo de transacciones: Describe de forma abreviada todas las transacciones de la BD, indicando en cada nodo el soporte del itemset que se forma siguiendo el camino que va desde la raíz hasta dicho nodo.
- Una vez que el árbol FP-tree ha sido construido, este utiliza un enfoque recursivo basado en la técnica divide y vencerás para extraer los ítemsets frecuentes



Solución más interesante, una de las versiones más utilizadas actualmente

Construye una estructura inteligente

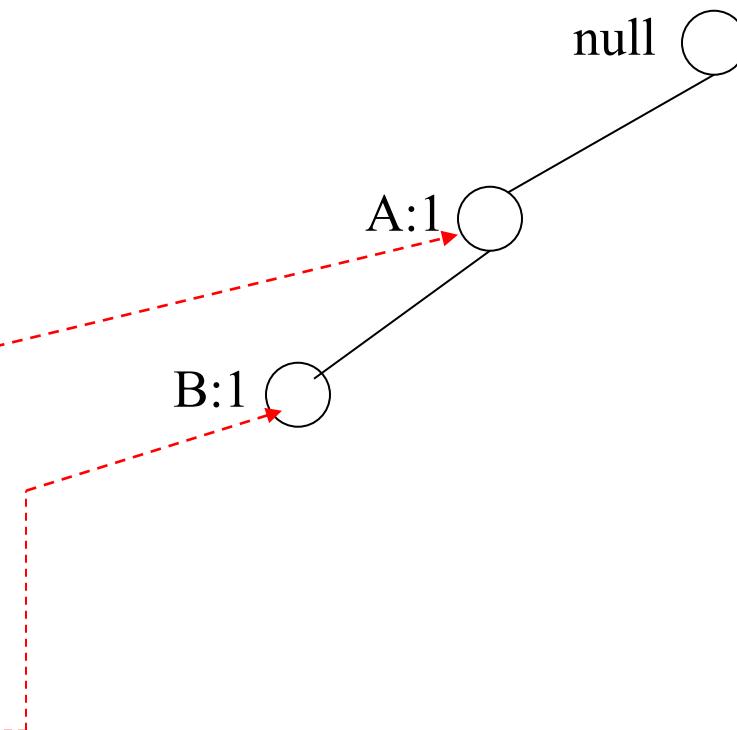
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



Requisito para aplicar este metodo es que previamente se debe establecer un orden.

Por ejemplo primero va el A, luego B...

Ese orden lo eliges tu según te interese, pero se debe mantener para un correcto resultado

Cada vez que aparece un item se añade una lista en su tabla Cabecera.

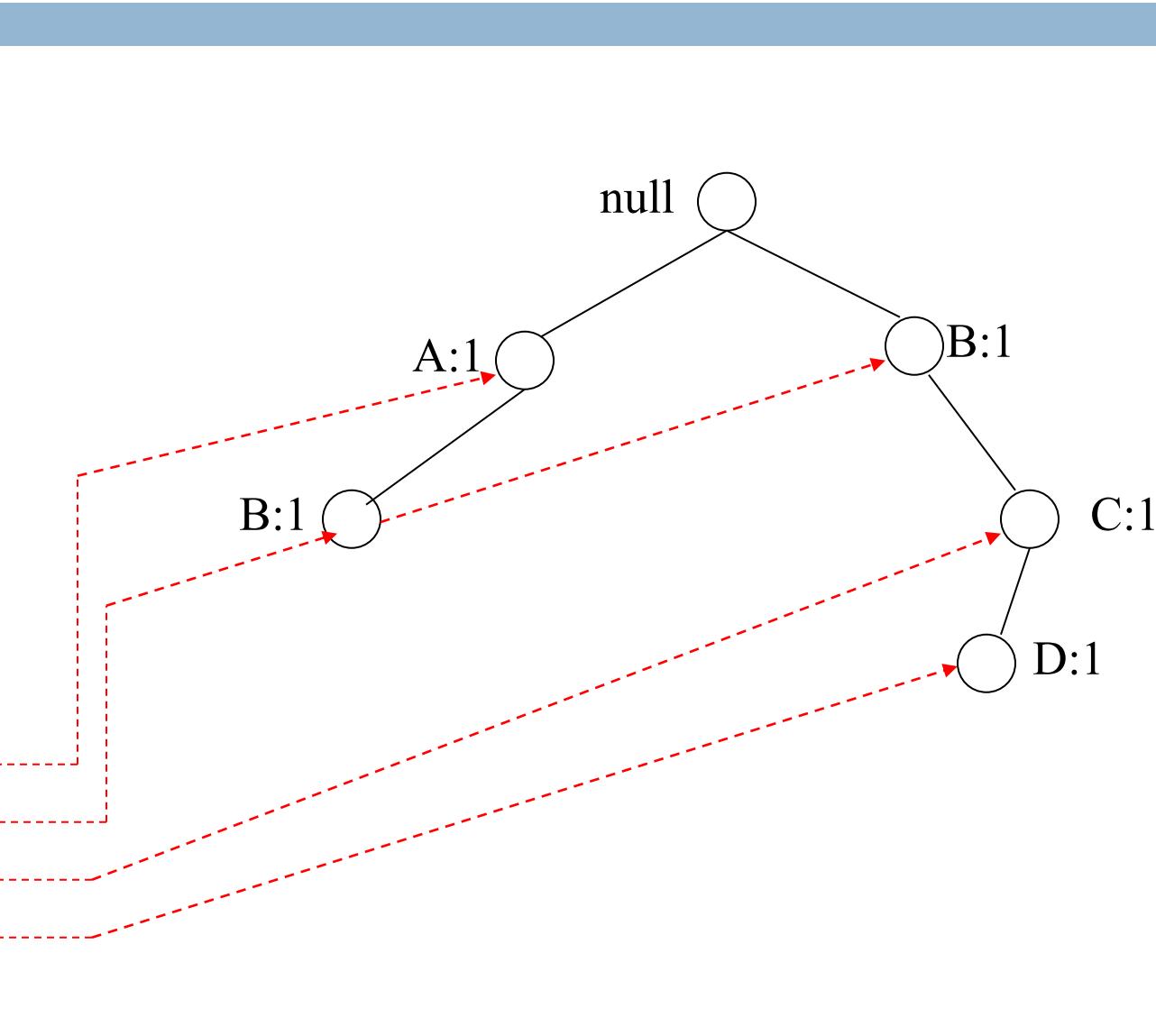
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



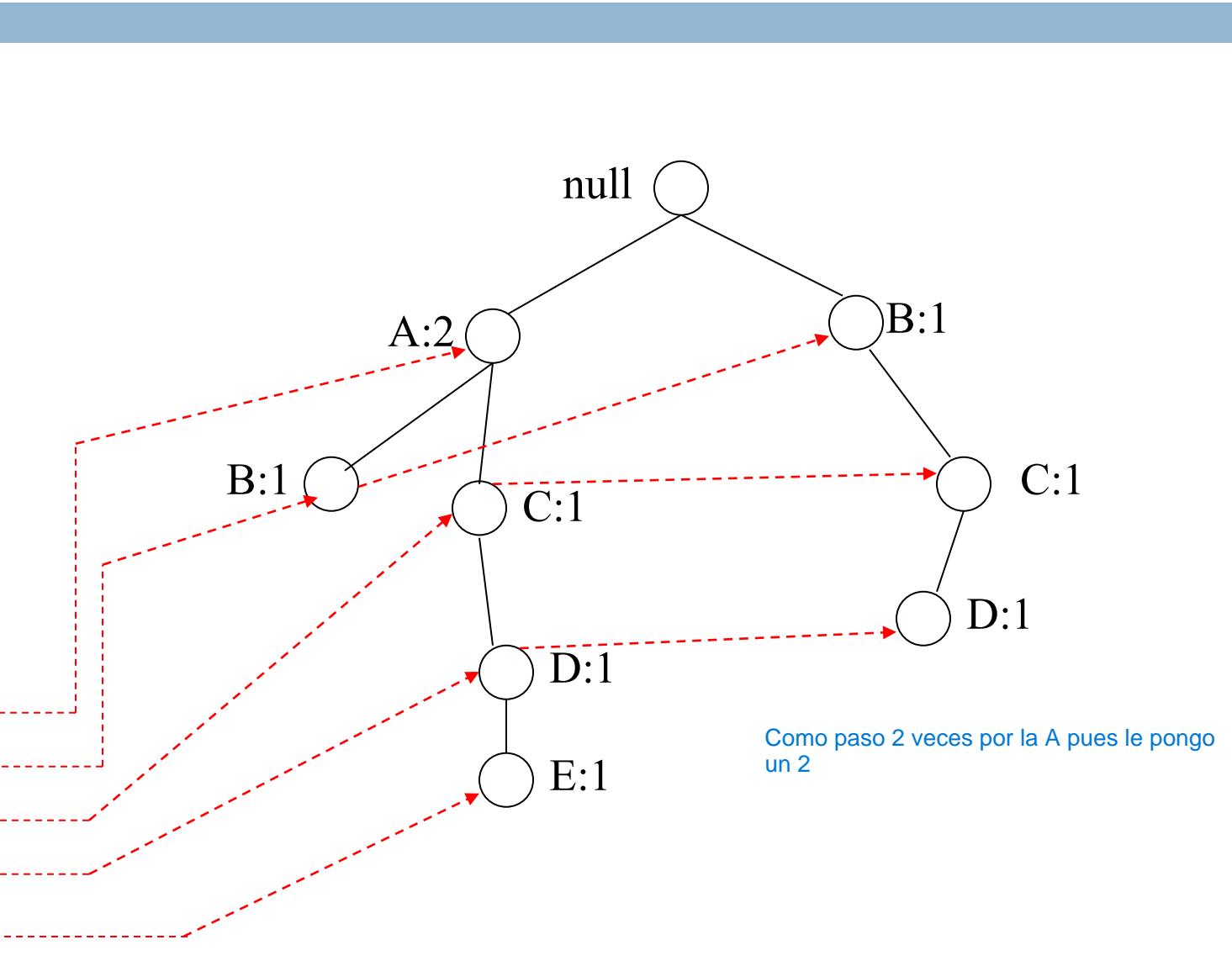
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



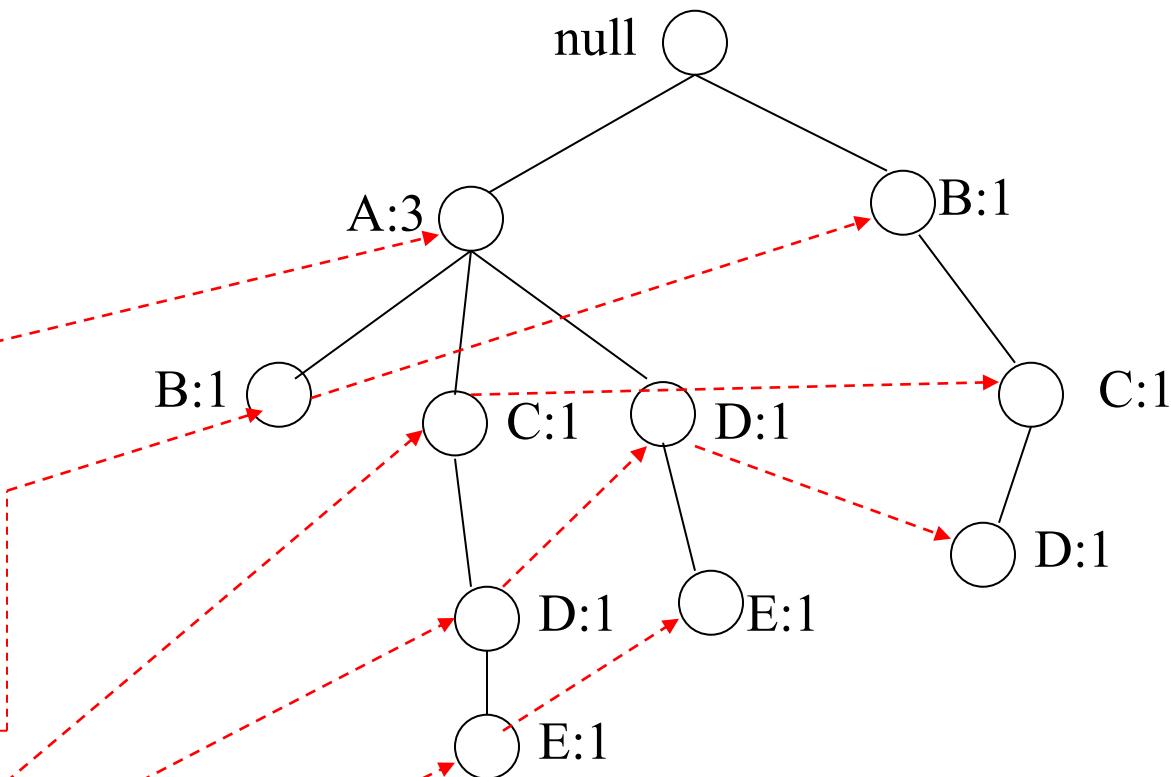
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	-
B	-
C	-
D	-
E	-



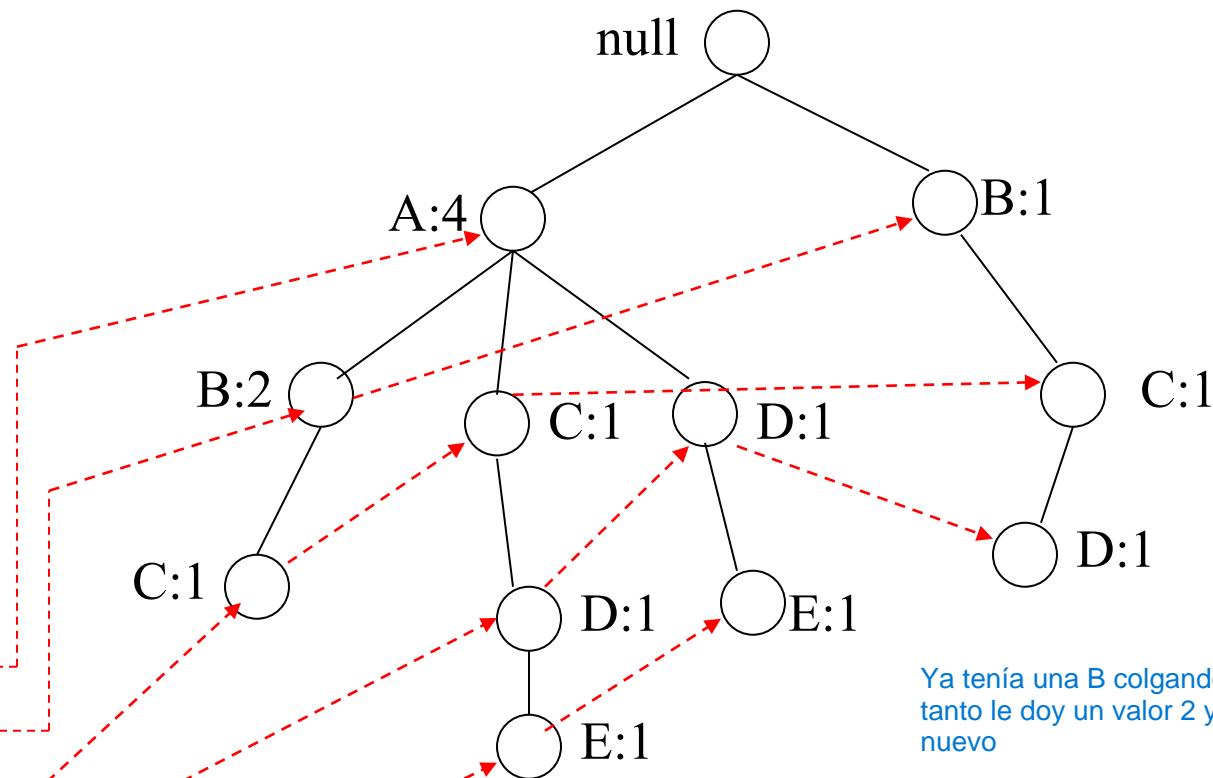
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



Ya tenía una B colgando de A, por lo tanto le doy un valor 2 y añado lo nuevo

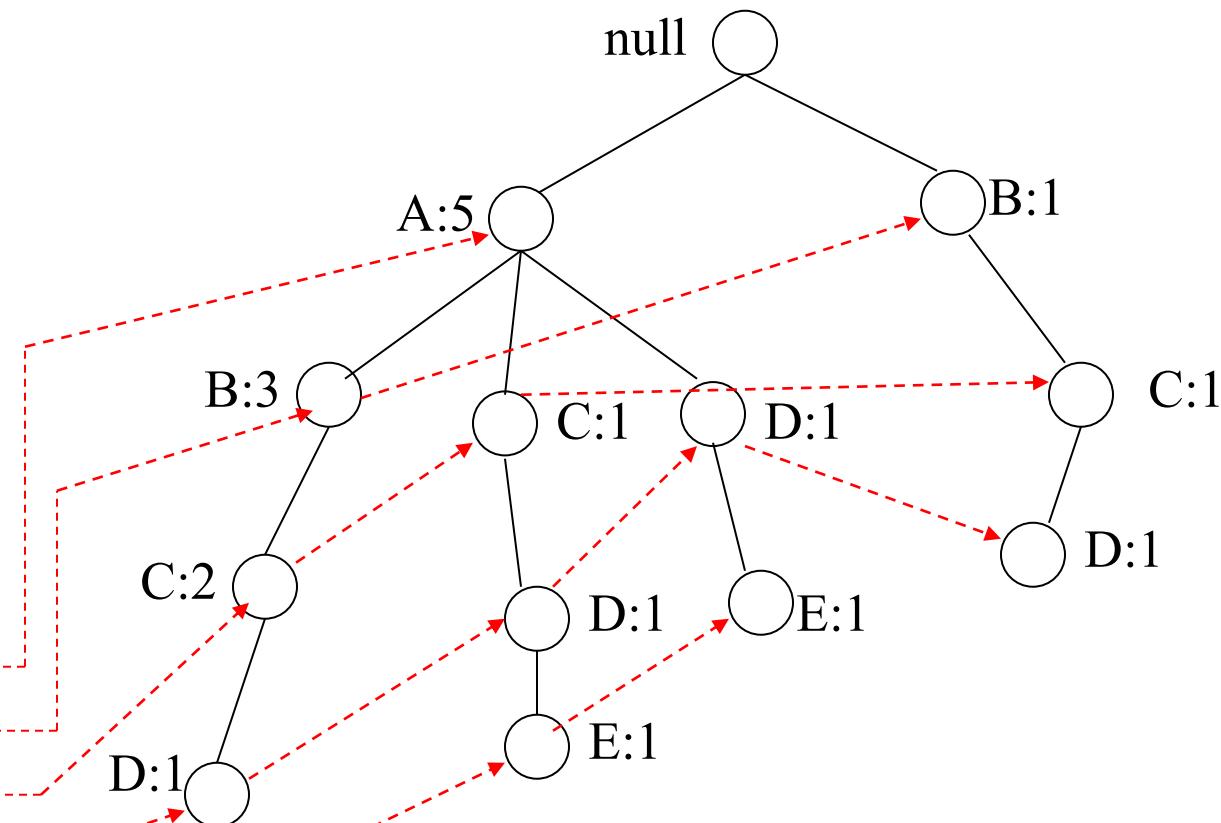
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



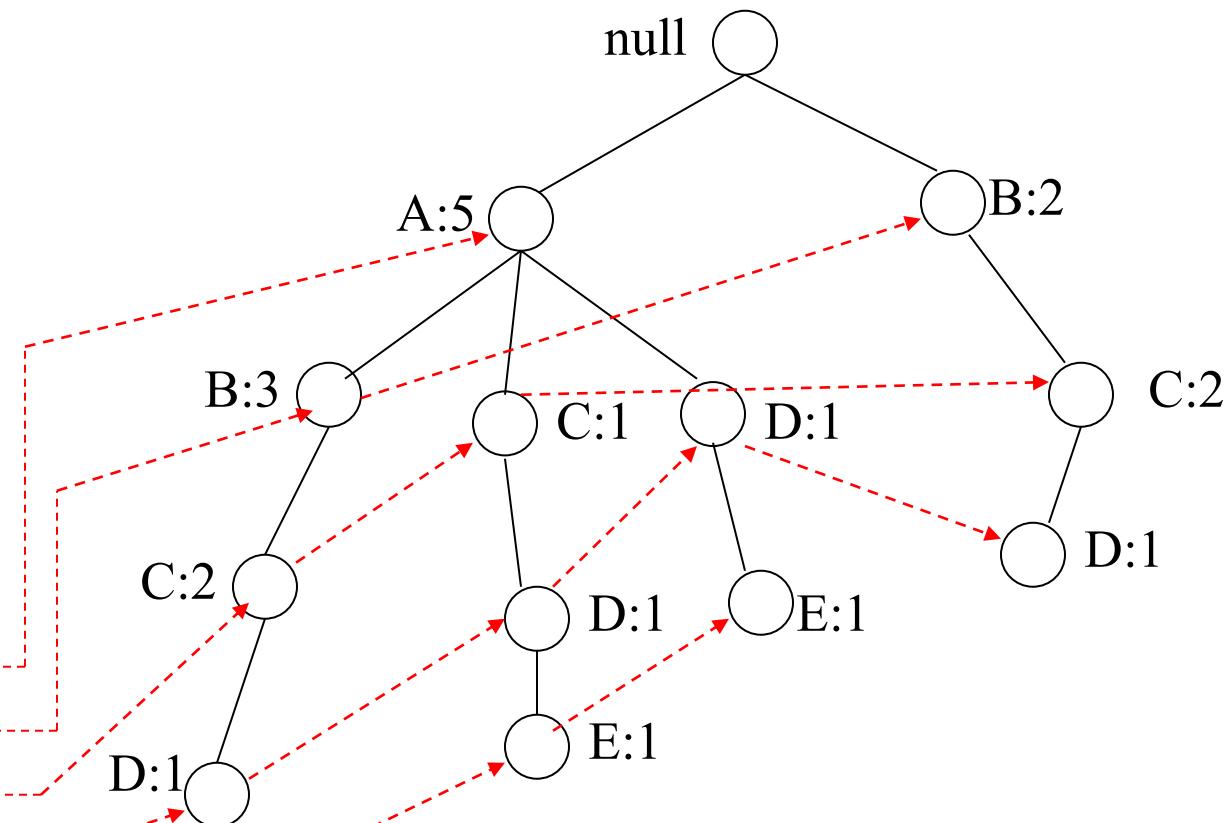
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



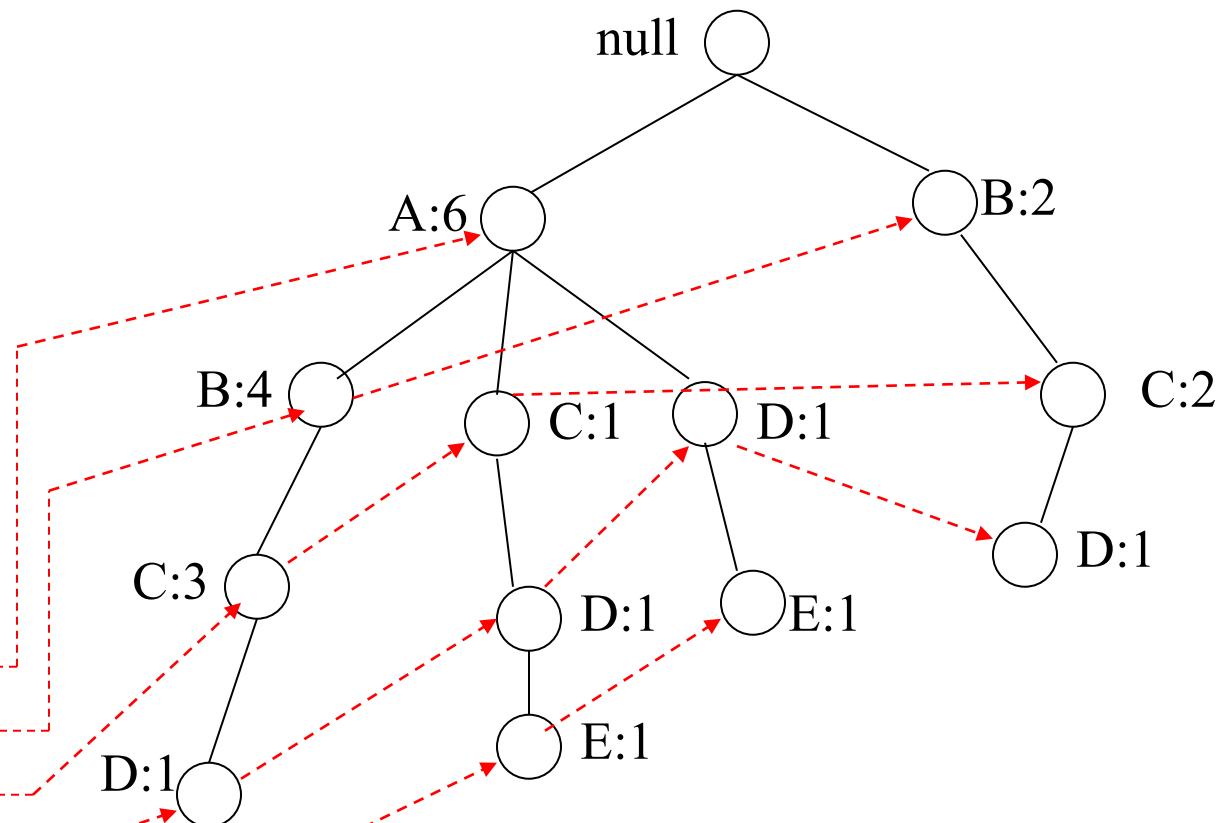
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



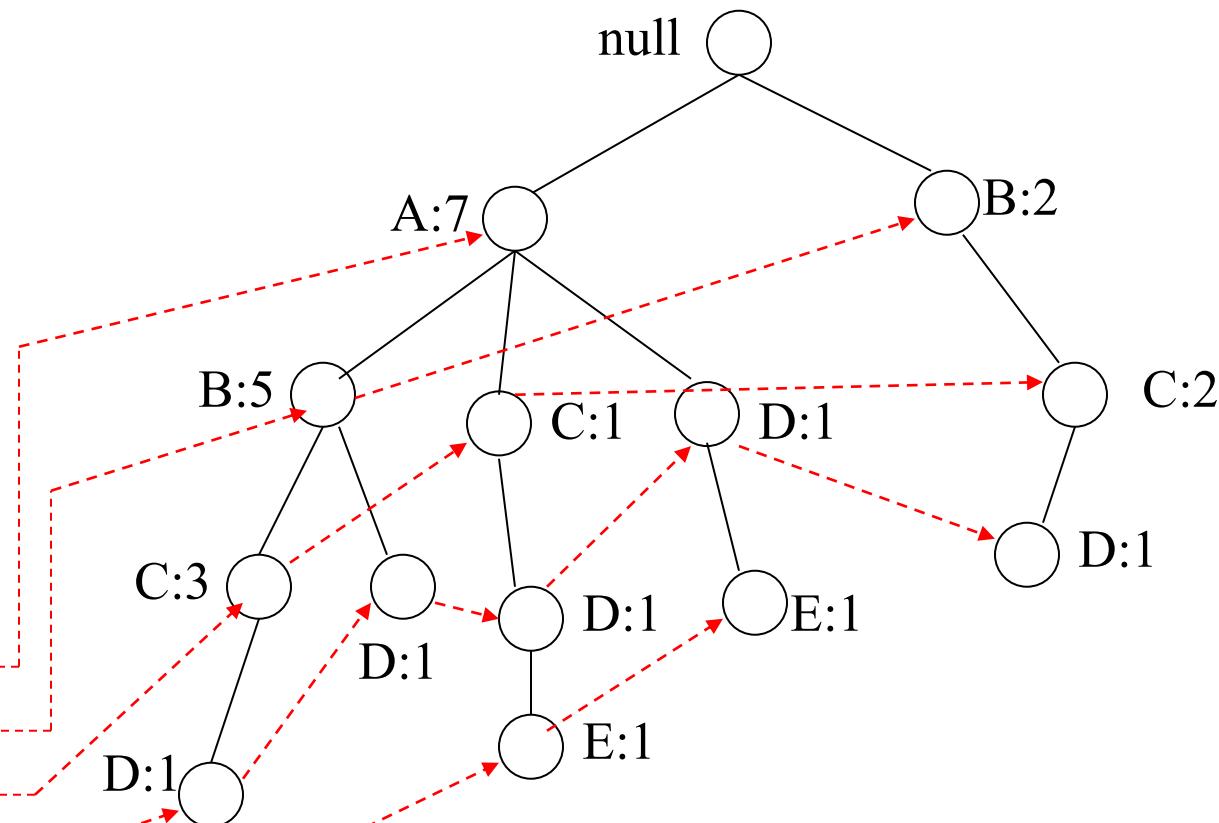
Construcción del FP-Tree

Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



Construcción del FP-Tree

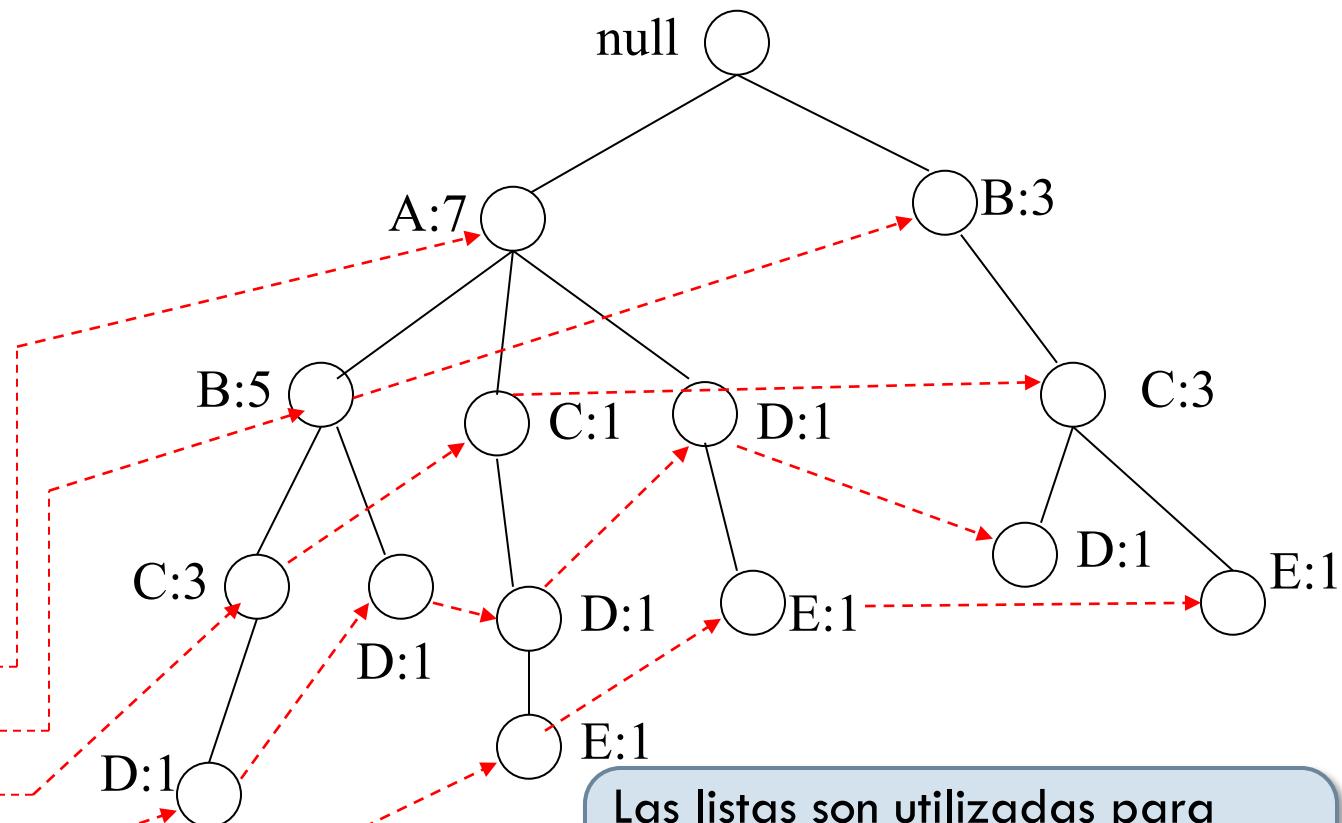
Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

MUY IMPORTANTE QUE ESTE ORDENADO EL CONJUNTO AL PRINCIPIO

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



Las listas son utilizadas para facilitar el proceso de generación de itemsets frecuentes

FP-growth: Extracción de itemsets frecuentes

- Se realiza en dos pasos:
 - Calcular el soporte de cada uno de los items que aparecen en el problema recorriendo su lista correspondiente de la tabla cabecera.
 - Para cada item que supera el soporte mínimo:
 1. Extraer las ramas en las que aparecen el item, reajustando el soporte de los items que aparecen en las ramas en función del soporte de item en esa rama.
 2. Generar FP-tree condicionado usando las ramas extraídas.
 3. Extraer los item frecuentes de la table de listas cuyo soporte supere el mínimo soporte
 4. A partir del nuevo FP-tree, generar para cada item frecuente en la table un FP-tree condicionado. Para ello saltamos al paso 1.

FP-growth: Extracción de itemsets frecuentes

Soportes

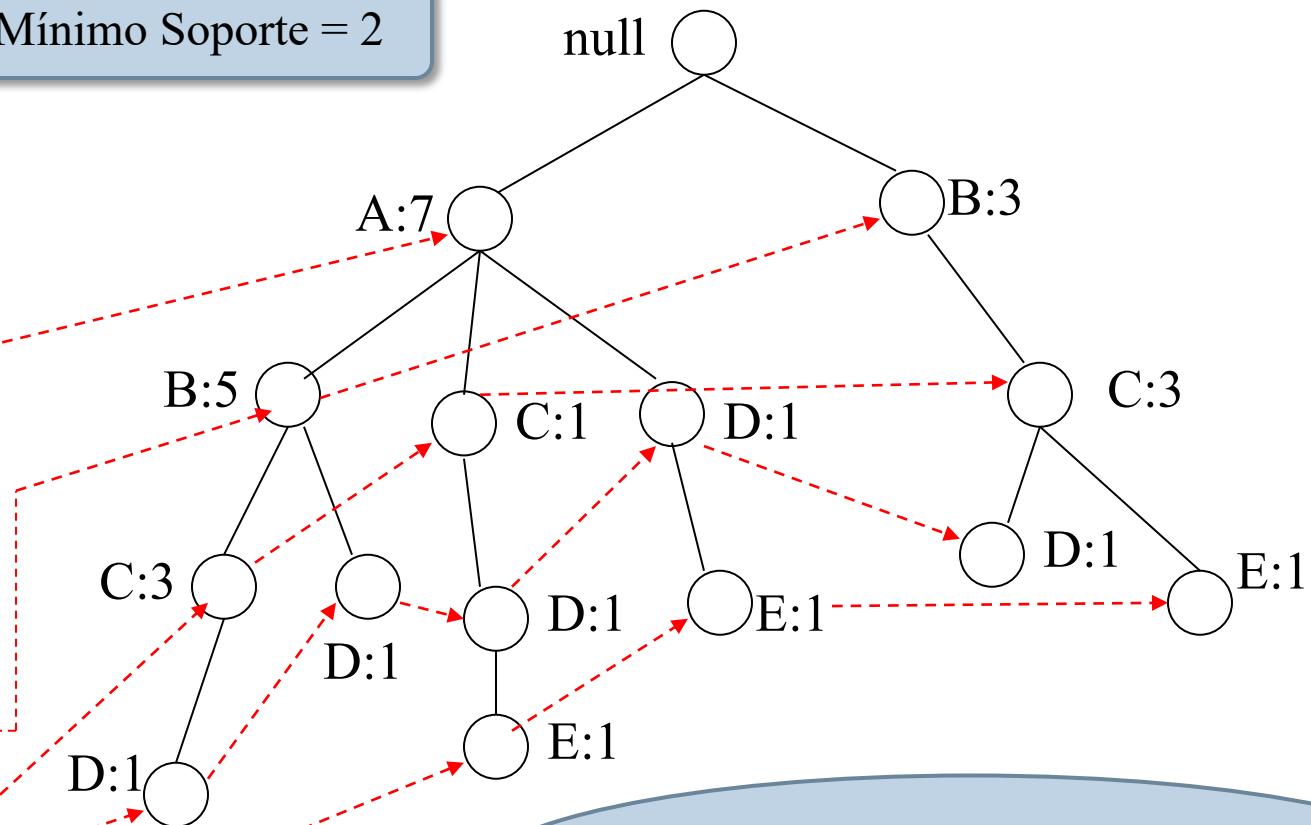
Item	Soporte
A	7
B	8
C	7
D	5
E	3

Itemsets Frecuentes: $\{A7, B8, C7, D5, E3\}$

Mínimo Soporte = 2

Tabla Cabecera

Item	Listas
A	
B	
C	
D	
E	



Todos los items son frecuentes!

FP-growth: Extracción de itemsets frecuentes

FP-tree para el item A

Mínimo Soporte = 2

null ○

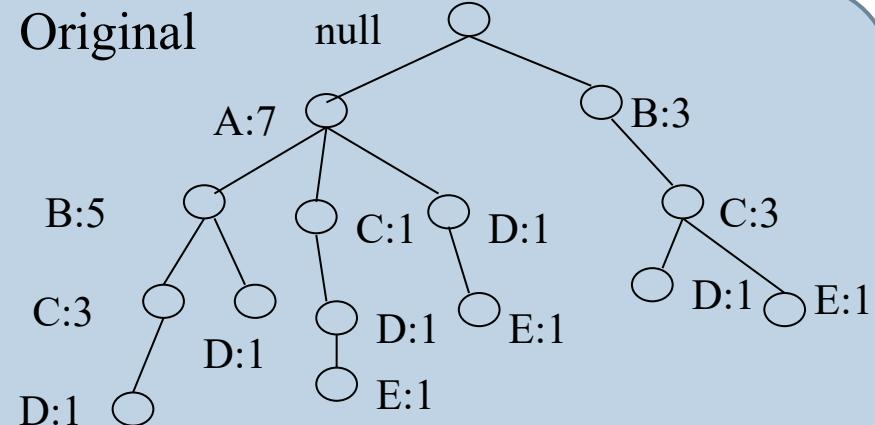
Ramas con A: {}

Itemsets Frecuentes:

{}
Miro las ramas que estén encima de cada caso
Si tomo la A, por encima no hay ningún nodo

Tabla Cabecera

Item	Listas
------	--------



FP-growth: Extracción de itemsets frecuentes

FP-tree para el item B

Ramas con B: {A5}

Itemsets Frecuentes:
 $\{AB5\}$

Guardo el Item Frecuente AB5
 y pongo B porque lo he creado para B

Con B si tengo una rama por encima, y
 esta rama es un caso de A por el que paso 5 veces (A5)

Mínimo Soporte = 2

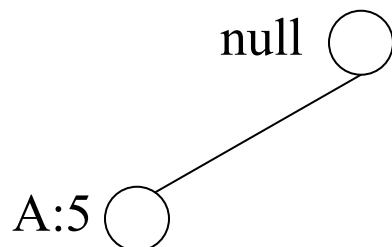
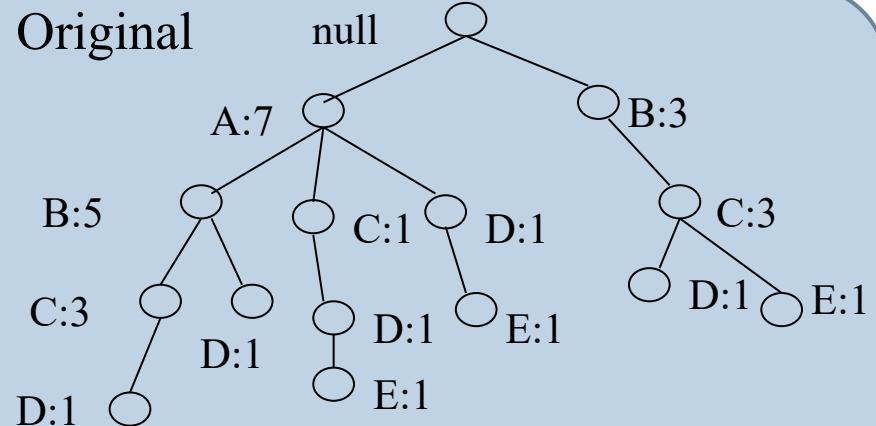


Tabla Cabecera

Item	Listas
A	



FP-growth: Extracción de itemsets frecuentes

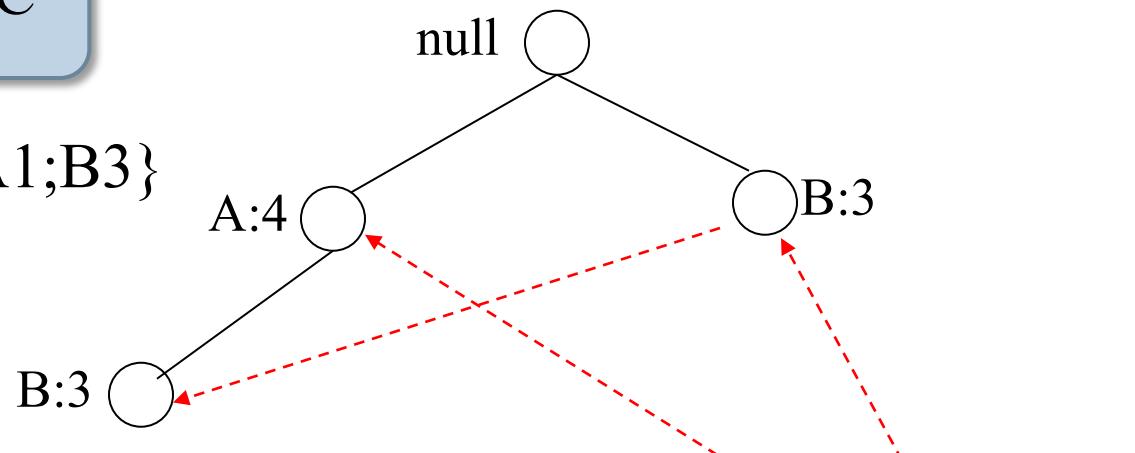
FP-tree para el item C

Ramas con C: {AB3;A1;B3}

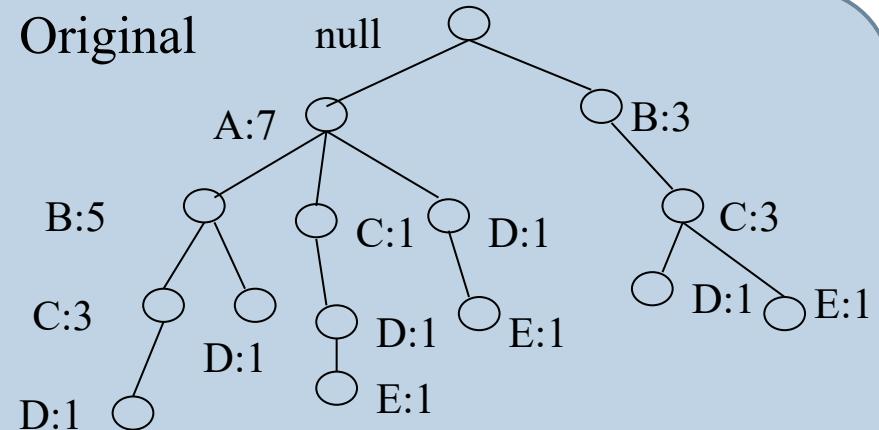
Itemsets Frecuentes:

{AC4;BC6}

Añado los que superen el minimo soporte



Original

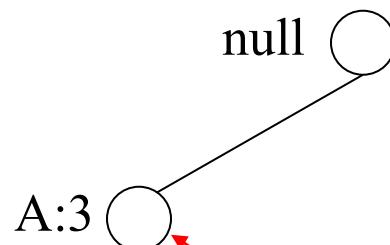


FP-growth: Extracción de itemsets frecuentes

FP-tree para el ítem CB

Ramas con CB: {A3}

Itemsets Frecuentes:
 $\{ABC3\}$



Mínimo Soporte = 2

FP-tree C

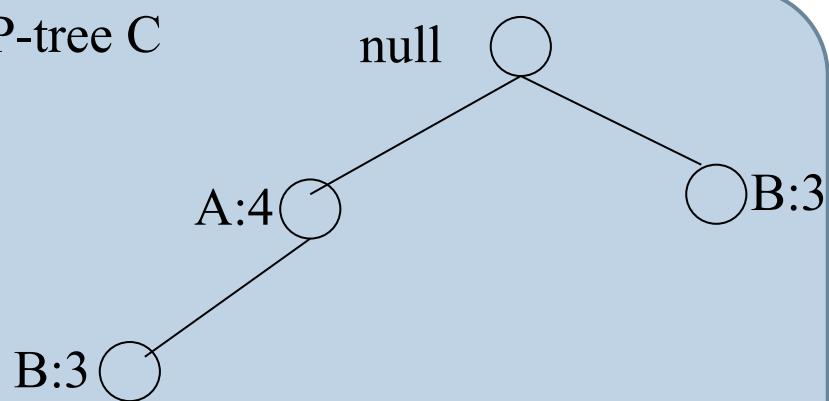


Tabla Cabecera

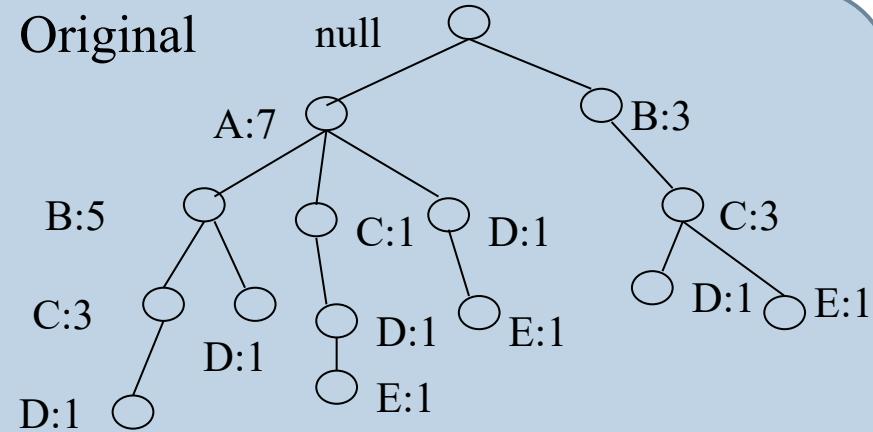
Ítem	Listas
A	

FP-growth: Extracción de itemsets frecuentes

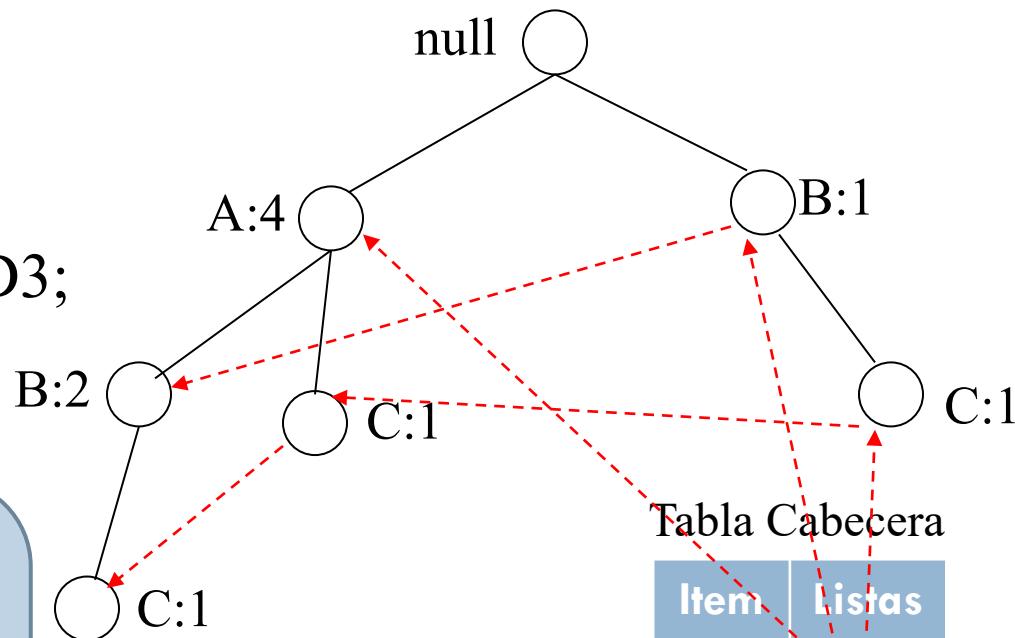
FP-tree para el item D

Ramas con D: {ABC1;AB1;
AC1;A1;BC1}

Itemsets Frecuentes: {AD4;BD3;
CD3}



Mínimo Soporte = 2



Item	Listas
A	
B	
C	

FP-growth: Extracción de itemsets frecuentes

FP-tree para el ítem DA

Mínimo Soporte = 2

null ○

Ramas con DA: {}

Itemsets Frecuentes: {}

FP-tree D

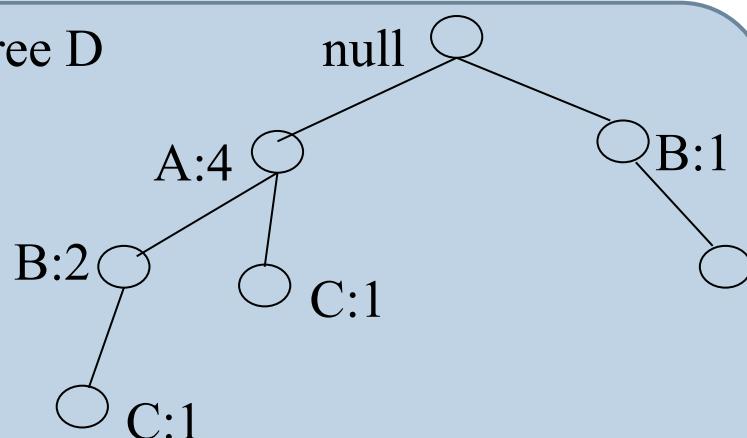


Tabla Cabecera

Item	Listas

FP-growth: Extracción de itemsets frecuentes

FP-tree para el ítem DB

Ramas con DB: {A2}

Itemsets Frecuentes: {ABD2}

Mínimo Soporte = 2

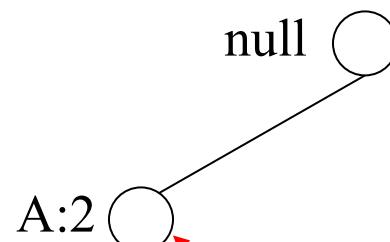
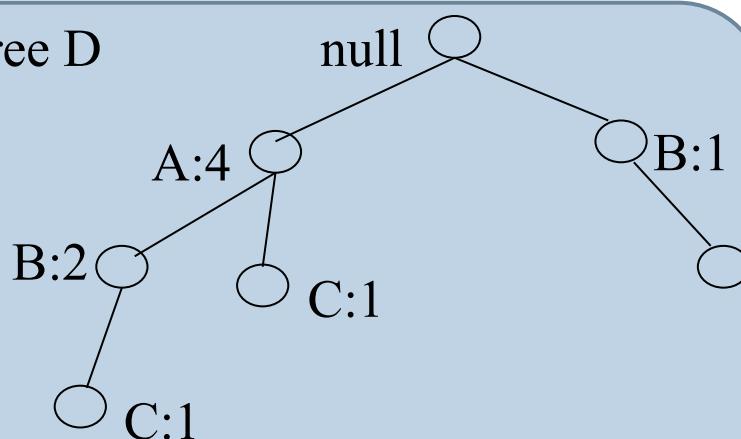


Tabla Cabecera

Item	Listas
A	

FP-tree D



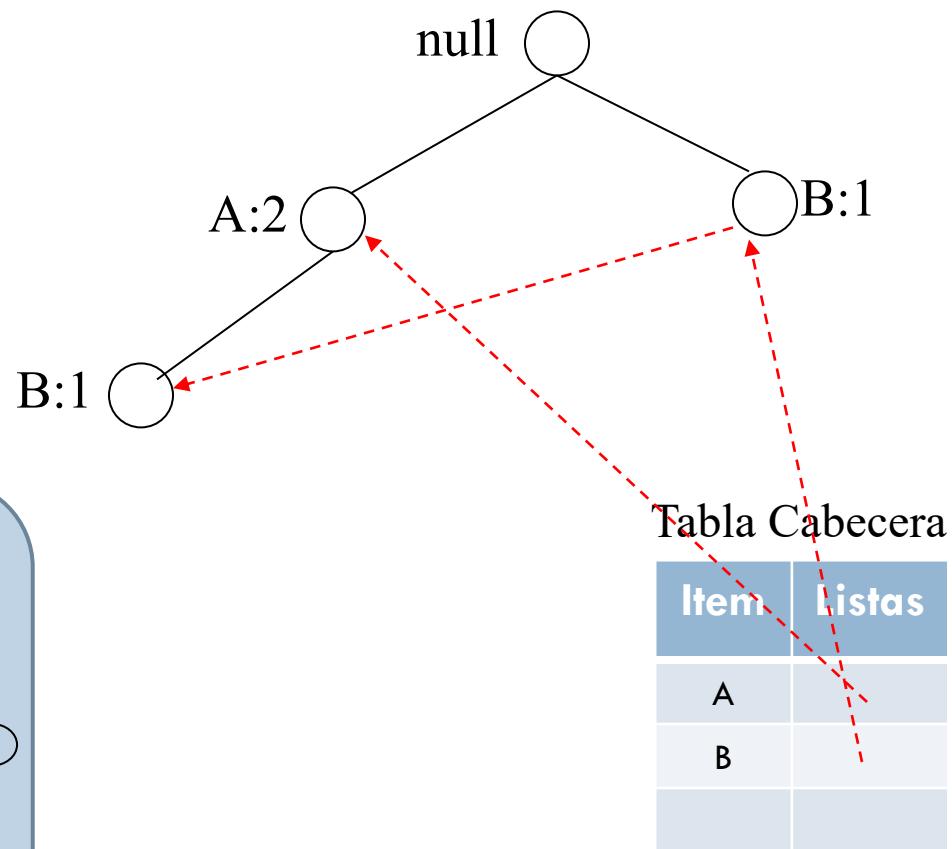
FP-growth: Extracción de itemsets frecuentes

FP-tree para el item DC

Ramas con DC: {AB1;A1;
B1}

Itemsets Frecuentes: {ACD2;
BCD2}

Mínimo Soporte = 2



FP-growth: Extracción de itemsets frecuentes

FP-tree para el item DCB

Ramas con DCB: {A1}

Itemsets Frecuentes: {}

Mínimo Soporte = 2

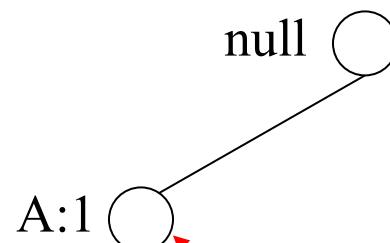
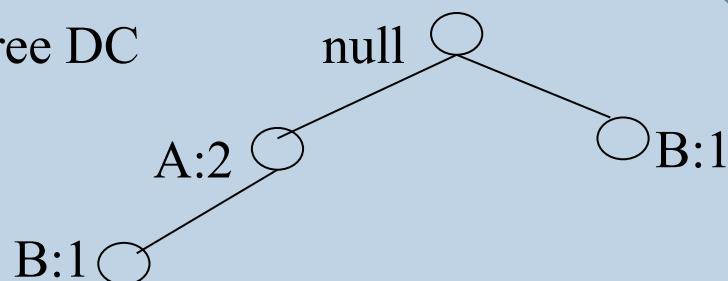


Tabla Cabecera

Item	Listas
A	

FP-tree DC

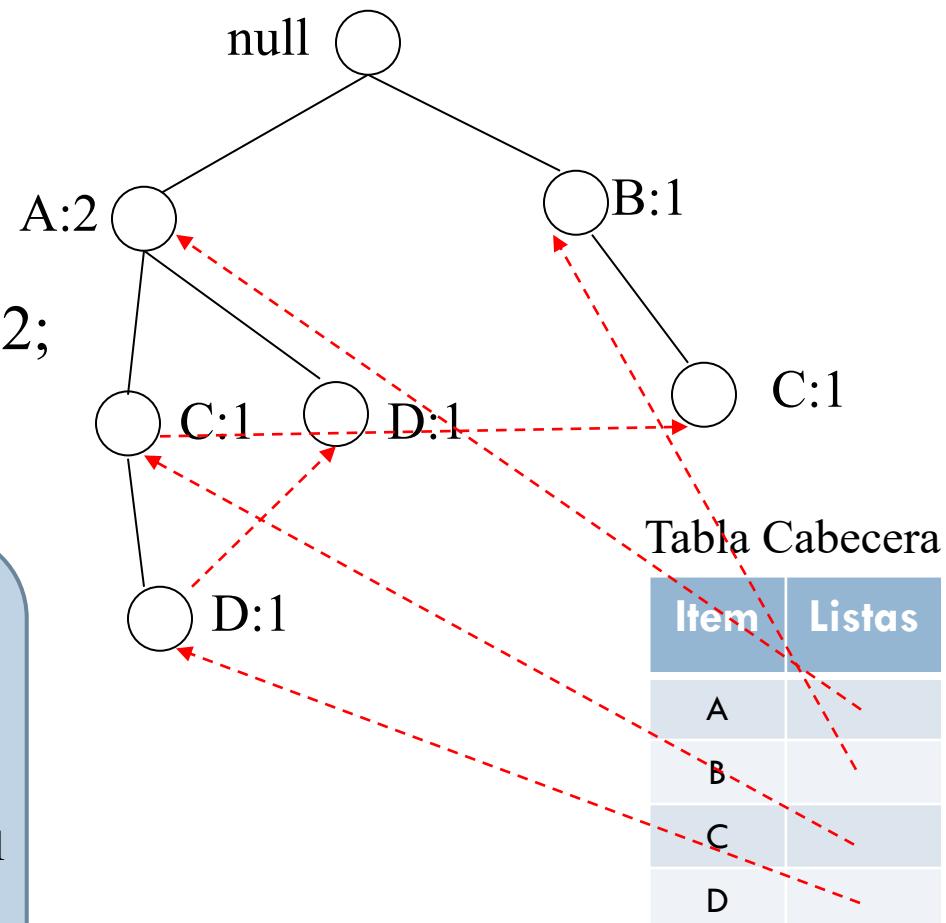
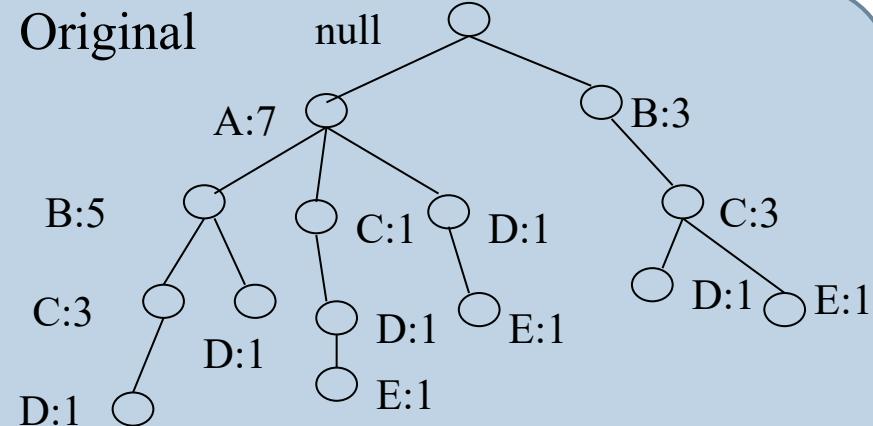


FP-growth: Extracción de itemsets frecuentes

FP-tree para el item E

Ramas con E: {ACD1;AD1;
BC1}

Itemsets Frecuentes: {AE2;CE2;
DE2}



FP-growth: Extracción de itemsets frecuentes

FP-tree para el item EA

Mínimo Soporte = 2

null ○

Ramas con EA: {}

Itemsets Frecuentes: {}

FP-tree E

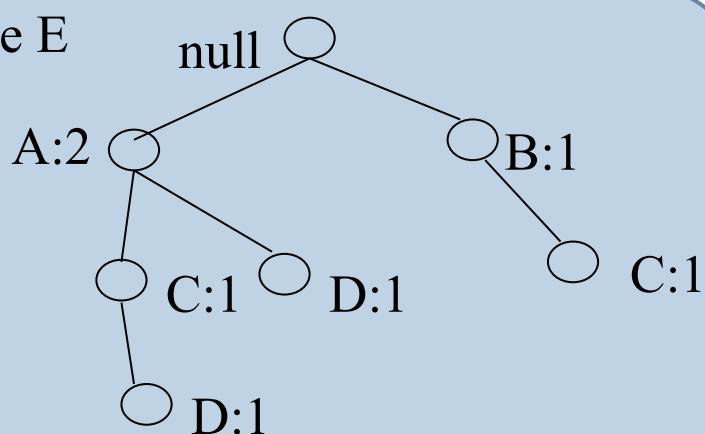


Tabla Cabecera

Item	Listas

FP-growth: Extracción de itemsets frecuentes

FP-tree para el item EB

Mínimo Soporte = 2

null ○

Ramas con EB: {}

Itemsets Frecuentes: {}

FP-tree E

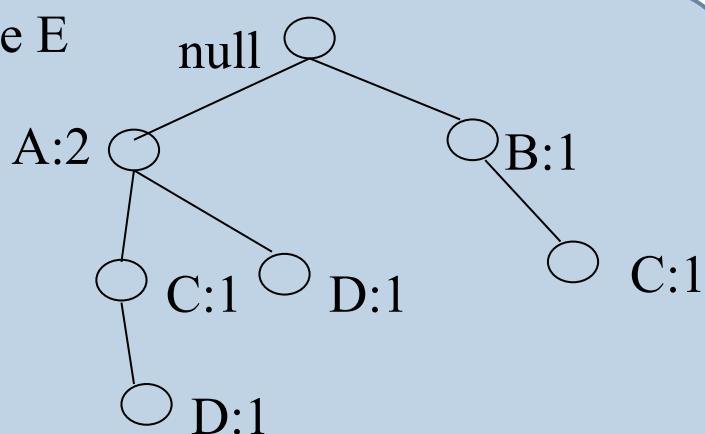


Tabla Cabecera

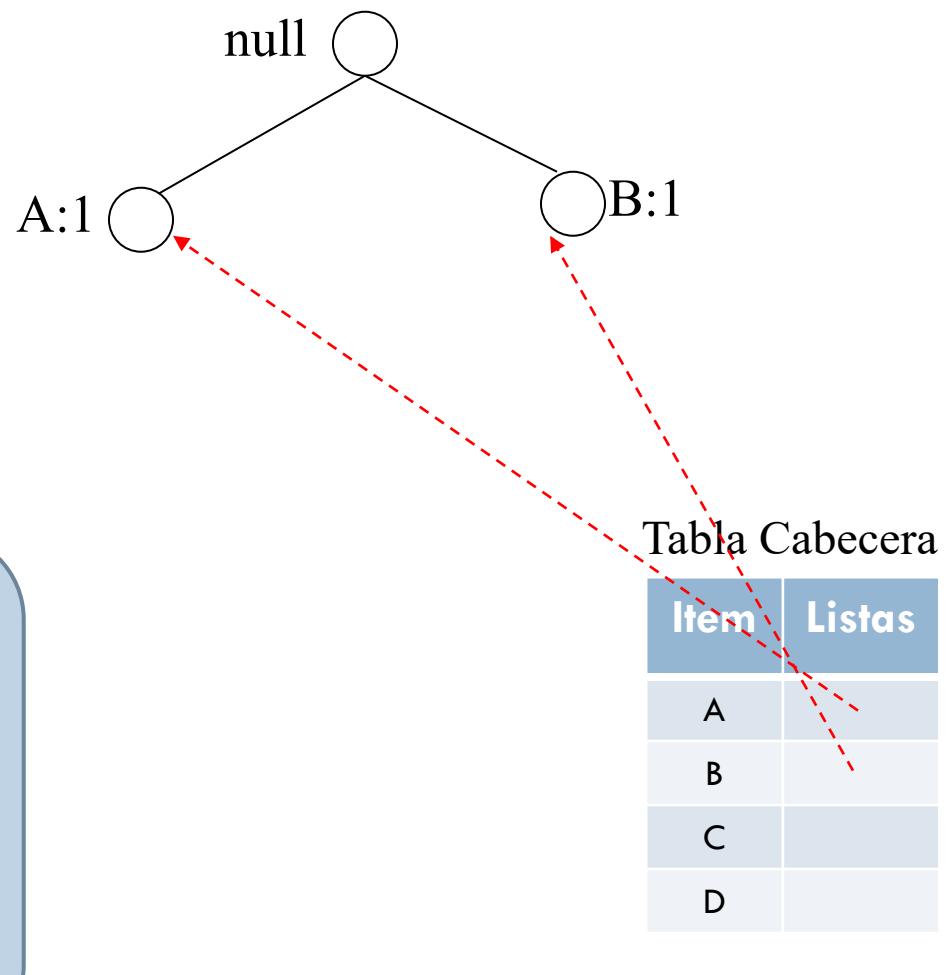
Item	Listas

FP-growth: Extracción de itemsets frecuentes

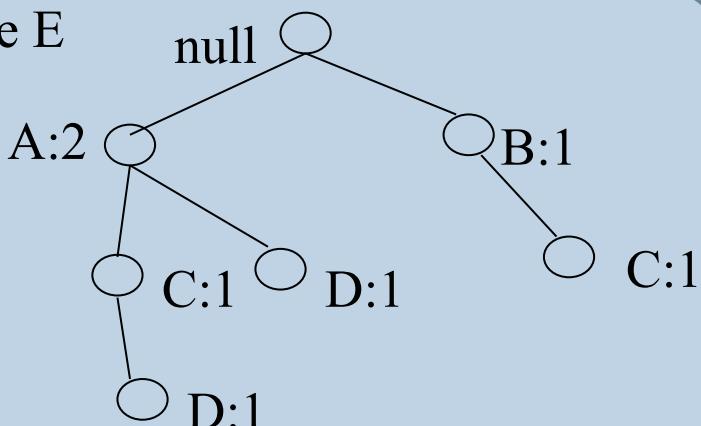
FP-tree para el item EC

Ramas con EC: {A1;B1}

Itemsets Frecuentes: {}



FP-tree E



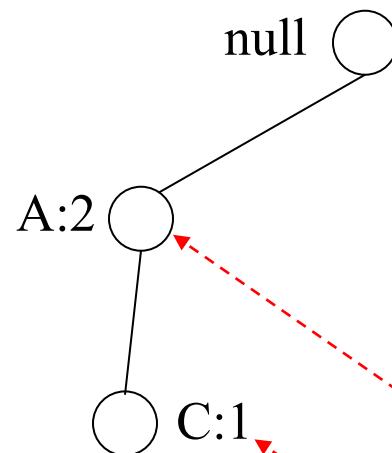
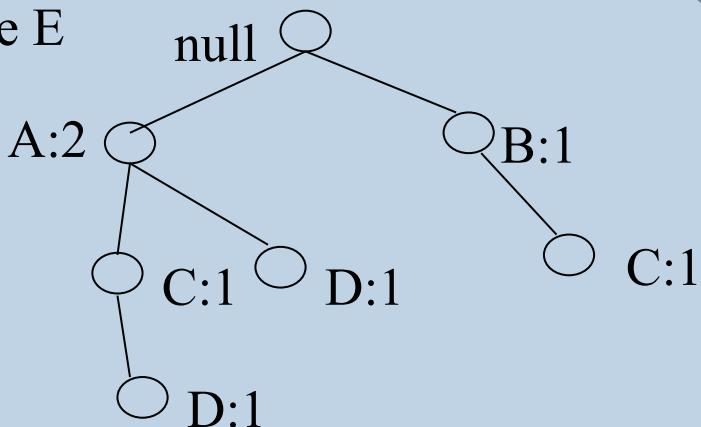
FP-growth: Extracción de itemsets frecuentes

FP-tree para el item ED

Ramas con ED: {AC1;A1}

Itemsets Frecuentes: {ADE2}

FP-tree E



Mínimo Soporte = 2

Tabla Cabecera

Item	Listas
A	
C	

FP-growth: Extracción de itemsets frecuentes

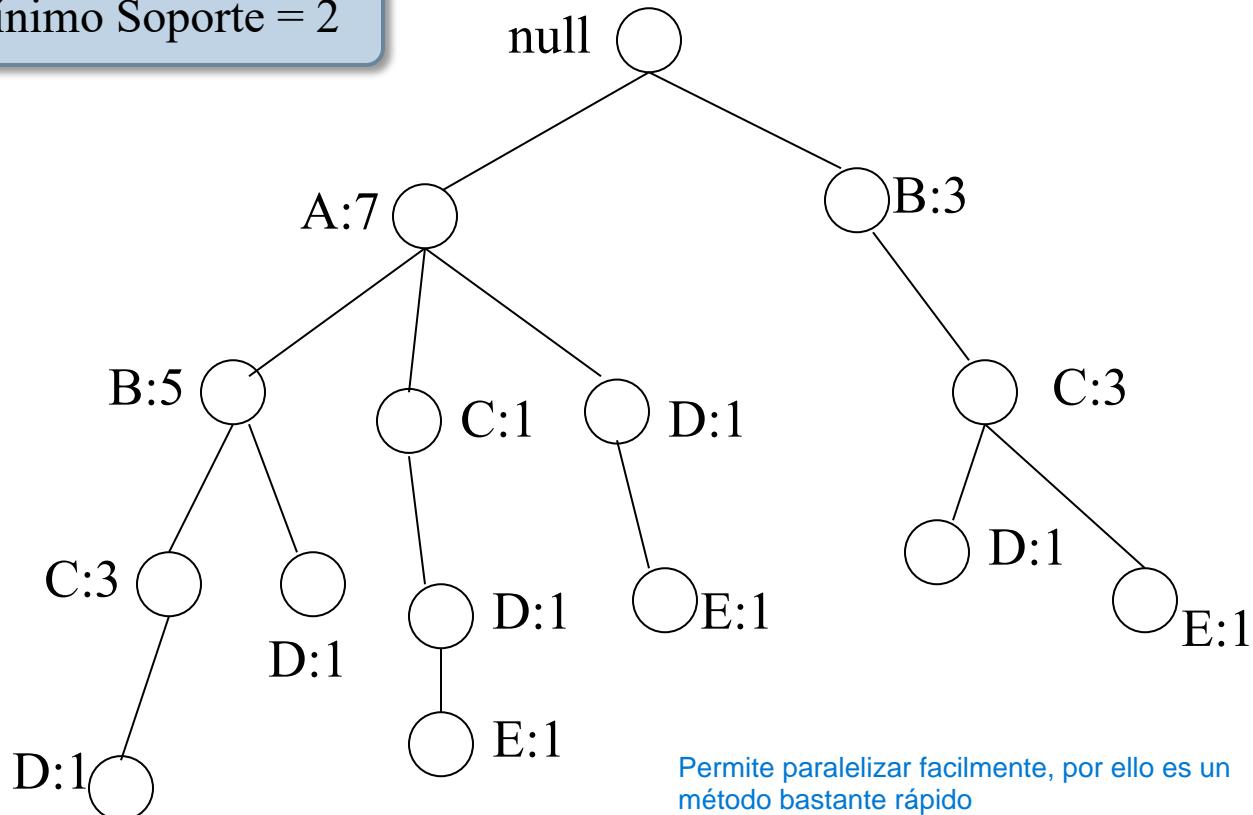
Base de Datos

TID	Items
1	A,B
2	B,C,D
3	A,C,D,E
4	A,D,E
5	A,B,C
6	A,B,C,D
7	B,C
8	A,B,C
9	A,B,D
10	B,C,E

Mínimo Soporte = 2

Itemsets frecuentes

- A7; B8; C7; D5; E3
- AB5
- AC4; BC6; ABC3;
- AD4; BD3; CD3; ABD2; ACD2; BCD2;
- AE2; CE2; DE2; ADE2



Permite paralelizar fácilmente, por ello es un método bastante rápido

Contenidos

- Aprendizaje no supervisado - contexto
- Definición de regla de asociación
- Medidas clásicas de las reglas de asociación
- Métodos clásicos de extracción de reglas
- **Conjuntos maximales y cerrados**
- Generación de reglas
- Problemas abiertos
- Aplicaciones

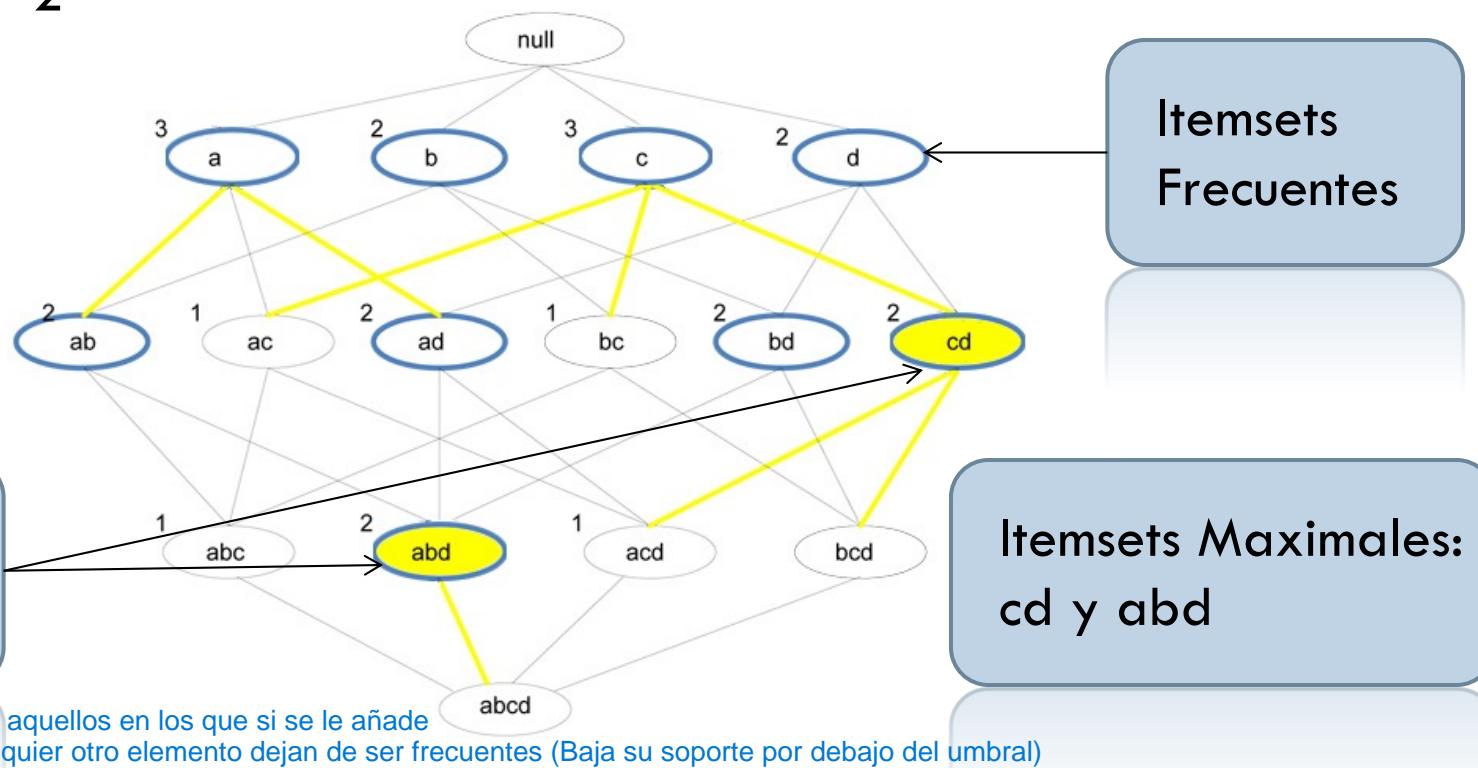
Itemsets Maximales y Cerrados

¿Qué ocurre si tenemos una BD con cientos de items?

- El número de conjuntos de itemsets frecuentes crece de forma exponencial y esto a su vez crea un problema con el almacenamiento
- **Solución:** Representaciones alternativas que permitan reducir el conjunto inicial, pero que nos dejen generar todos los itemsets frecuentes a partir de ellas.
- Dos de esas representaciones:
 - **Itemsets Maximales**
 - **Itemsets Cerrados**

Itemsets Maximales

- **Definición:** Son aquellos itemsets frecuentes para los que ningunos de sus superconjuntos inmediatos son frecuentes.
- $\text{MinSup} = 2$



Itemsets Maximales

- **Ventajas:** A partir de los itemsets frecuentes maximales podemos sacar todos los itemsets frecuentes, ya que serán todos los subconjuntos de items que podamos formar a partir de ellos.

- **Desventaja:** No sabemos el soporte de los itemsets frecuentes y tendríamos que volver a calcularlo.

Computacionalmente se debe calcular de nuevo el soporte.

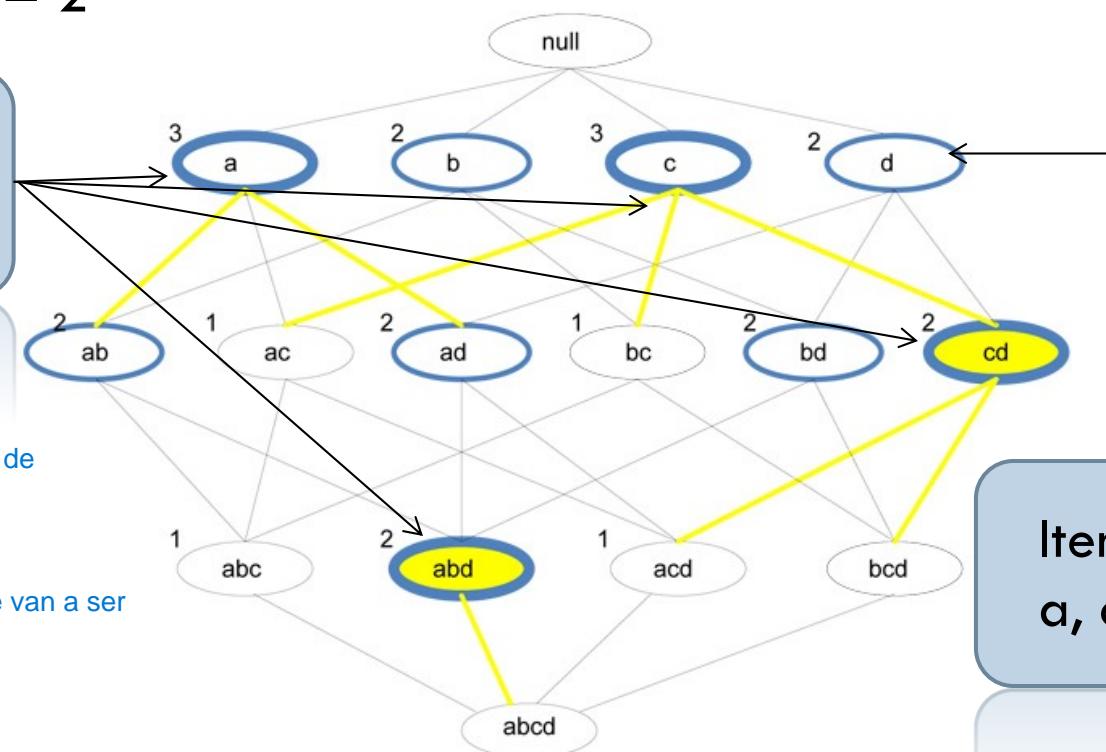
Itemsets Cerrados

- **Definición:** Son aquellos itemsets frecuentes para los que ninguno de sus superconjuntos inmediatos tienen un soporte igual al de ellos.
- $\text{MinSop} = 2$

**Itemsets
Cerrados**

Son aquellos en los que si se le añade cualquier otro elemento su soporte baja (pero no significa que dejen de ser Frecuentes)

Los maximales siempre van a ser itemsets cerrados



**Itemsets
Frecuentes**

**Itemsets Cerrados:
a, c, cd y abd**

Todo itemset maximal también es cerrado!!!

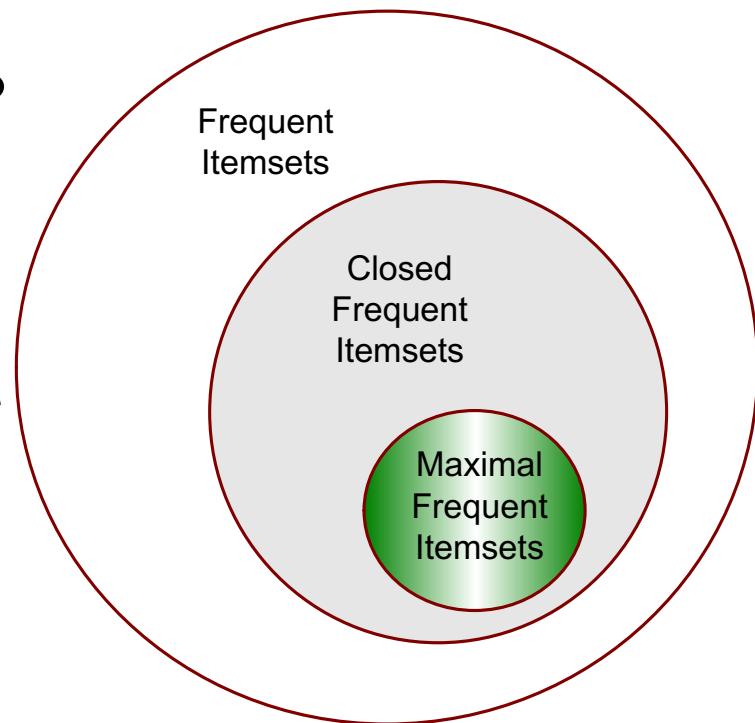
Itemsets Cerrados

- **Ventajas:** A partir de los itemsets frecuentes cerrados podemos sacar todos los itemsets frecuentes y, además, cualquier subconjunto de ellos que no sea otro itemsets cerrado tiene el mismo soporte que ellos, por lo que no tendremos que volver a calcular los soportes.
En ese caso si se guarda el soporte
- **Desventaja:** Son más numerosos que los itemsets maximales y necesitamos más espacio para almacenarlos.

Relación: Maximales & Cerrados

- Los itemsets maximales y cerrados son subconjuntos de los itemsets frecuentes, pero los maximales son una representación más compacta que los cerrados.

- Los itemsets cerrados son más utilizados que los maximales cuando la eficiencia es más importante que el espacio en disco, ya que nos proporcionan el soporte de los sub-itemsets sin necesidad de volver a recorrer la BD.



Dependiendo del problema interesarán unos u otros
Se toma el que más interese

Contenidos

- Aprendizaje no supervisado - contexto
- Definición de regla de asociación
- Medidas clásicas de las reglas de asociación
- Métodos clásicos de extracción de reglas
- Conjuntos maximales y cerrados
- **Generación de reglas**
- Problemas abiertos
- Aplicaciones

Generación de Reglas

- Dado un itemset frecuente L generamos las reglas haciendo todas las posibles combinaciones y nos quedamos con las que tengan una confianza mayor o igual que el umbral minConf. Hay dos opciones:
 - Generar reglas con un solo atributo en el consecuente (**más usadas**).
 - Generar reglas con más de un atributo en el consecuente (si k es el número de items en el itemset, hay $2^k - 2$ reglas de asociación candidatas, ignorando las reglas $L \rightarrow \emptyset$ y $\emptyset \rightarrow L$).
- Ejemplo: Si $\{A,B,C,D\}$ es un itemset frecuente, las reglas candidatas son:
 - Reglas con solo atributo en el consecuente:

Estas primeras suelen ser las más utilizadas por simplicidad, son más fáciles de entender

$$ABC \rightarrow D, \quad ABD \rightarrow C, \quad ACD \rightarrow B, \quad BCD \rightarrow A,$$

- Todas las posibles reglas:

$$\begin{array}{cccc} ABC \rightarrow D, & ABD \rightarrow C, & ACD \rightarrow B, & BCD \rightarrow A, \\ AB \rightarrow CD, & AC \rightarrow BD, & AD \rightarrow BC, & BC \rightarrow AD, \\ BD \rightarrow AC, & CD \rightarrow AB, & A \rightarrow BCD, & B \rightarrow ACD, \\ C \rightarrow ABD, & D \rightarrow ABC & \end{array}$$

Generación de Reglas

¿Podemos mejorar la eficiencia del proceso de generación cuando extraemos reglas con más de un ítem en el consecuente?

- La medida de confianza no tiene la propiedad anti-monótona:
NO: $\text{confianza}(\text{ABC} \rightarrow \text{D})$ puede ser más grande o más pequeña que $\text{confianza}(\text{AB} \rightarrow \text{D})$
- Pero la confianza de las reglas generadas a partir del mismo ítemset sí tiene la propiedad de anti-monótona

□ Ejemplo: $L = \{\text{A}, \text{B}, \text{C}, \text{D}\}$:

$$\text{conf}(\text{BCD} \rightarrow \text{A}) \geq \text{conf}(\text{CD} \rightarrow \text{AB}) \geq \text{conf}(\text{D} \rightarrow \text{ABC})$$

a medida que le quito cosas al antecedente y le doy al sucediente más grande el antecedente y más pequeño el sucediente

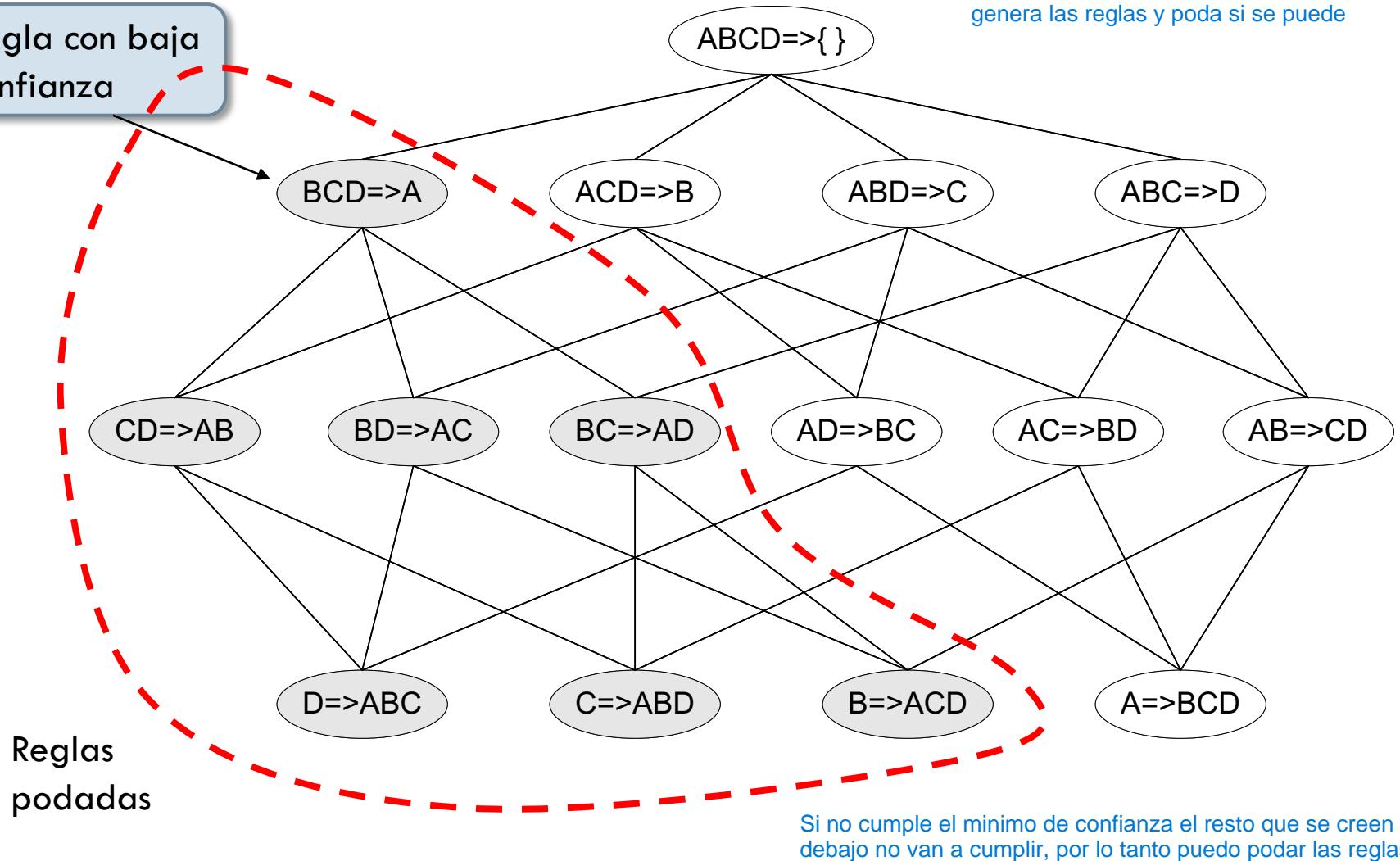
El soporte de AB es mayor igual que el de ABC

si le quito letras al antecedente la confianza baja

Generación de Reglas

Regla con baja
confianza

Possible ejercicio de examen:
genera las reglas y poda si se puede



Contenidos

- Aprendizaje no supervisado - contexto
- Definición de regla de asociación
- Medidas clásicas de las reglas de asociación
- Métodos clásicos de extracción de reglas
- Conjuntos maximales y cerrados
- Generación de reglas
- **Problemas abiertos**
- Aplicaciones

Problemas abiertos

Reglas de Asociación Cuantitativas

- Todas las reglas que hemos visto hasta ahora se conocen como **reglas de asociación binarias**, es decir, reglas extraídas a partir de BD donde todos los datos o todas las variables toman valores categóricos, por ejemplo: (color,rojo), (puesto,administrativo), pan, etc.
- Sin embargo, **en la mayoría de los problemas reales las BD contienen variables que toman valores numéricos**. Si intentamos extraer reglas a partir de estas BDs usando pares (Atributo,valor) tenemos 2 problemas:
 - El soporte de la mayoría de los items es muy bajo por lo general
 - Reglas pobres semánticamente

Problemas abiertos: R.A. Cuantitativas

- **Solución:** dividir el dominio de estos atributos en intervalos. Dos enfoques:
 - Definir unos intervalos a priori (conocimiento experto). Riqueza semántica, pero quizá los intervalos no sean los más adecuados para obtener buenas reglas ya que: Si hay muchos intervalos pequeños, podemos tener bajo soporte de cada uno de ellos. Por el contrario, los intervalos grandes dan lugar a reglas muy generales y a reglas inútiles cuando aparen en el consecuente de las reglas.
 - Búsqueda de reglas cuantitativas: dejar que el algoritmo busque los mejores intervalos. Posibles problemas: intervalos de semántica pobre, complejidad computacional (orden cuadrático en el número de valores del atributo), muchas reglas.

Problemas abiertos

Representación de las reglas

- La representaciones clásicas para las reglas de asociación son muy limitadas.
- Solución: extender el modelo de representación para poder representar más información de una forma más compacta:
 - Reglas negativas: $\neg A \rightarrow B$, $A \rightarrow \neg B$ ó $\neg A \rightarrow \neg B$
 - Reglas que incluyan operadores lógicos AND, OR, etc, u operadores aritméticos $<$, $>$, etc
 - Etc.

poder hacer reglas más sencillas te da una gran ventaja, para representar info mas compleja

Problemas abiertos

Medidas de Calidad

- Las medidas clásicas de soporte y confianza presentan problemas cuando las utilizamos para guiar la búsqueda de las reglas de asociación:
 - Soportes alto: dan lugar a reglas con soportes altos en el consecuente → reglas poco útiles
 - Confianza: al estar basada en frecuencias, no detecta cuando el soporte del consecuente es muy alto. Ej:

Confianza ($A \rightarrow B$) = 1.0

No existe ninguna medida perfecta

Confianza ($C \rightarrow D$) = 0.84

itemsets muy frecuentes desbordará el proceso, porque siempre estará presente

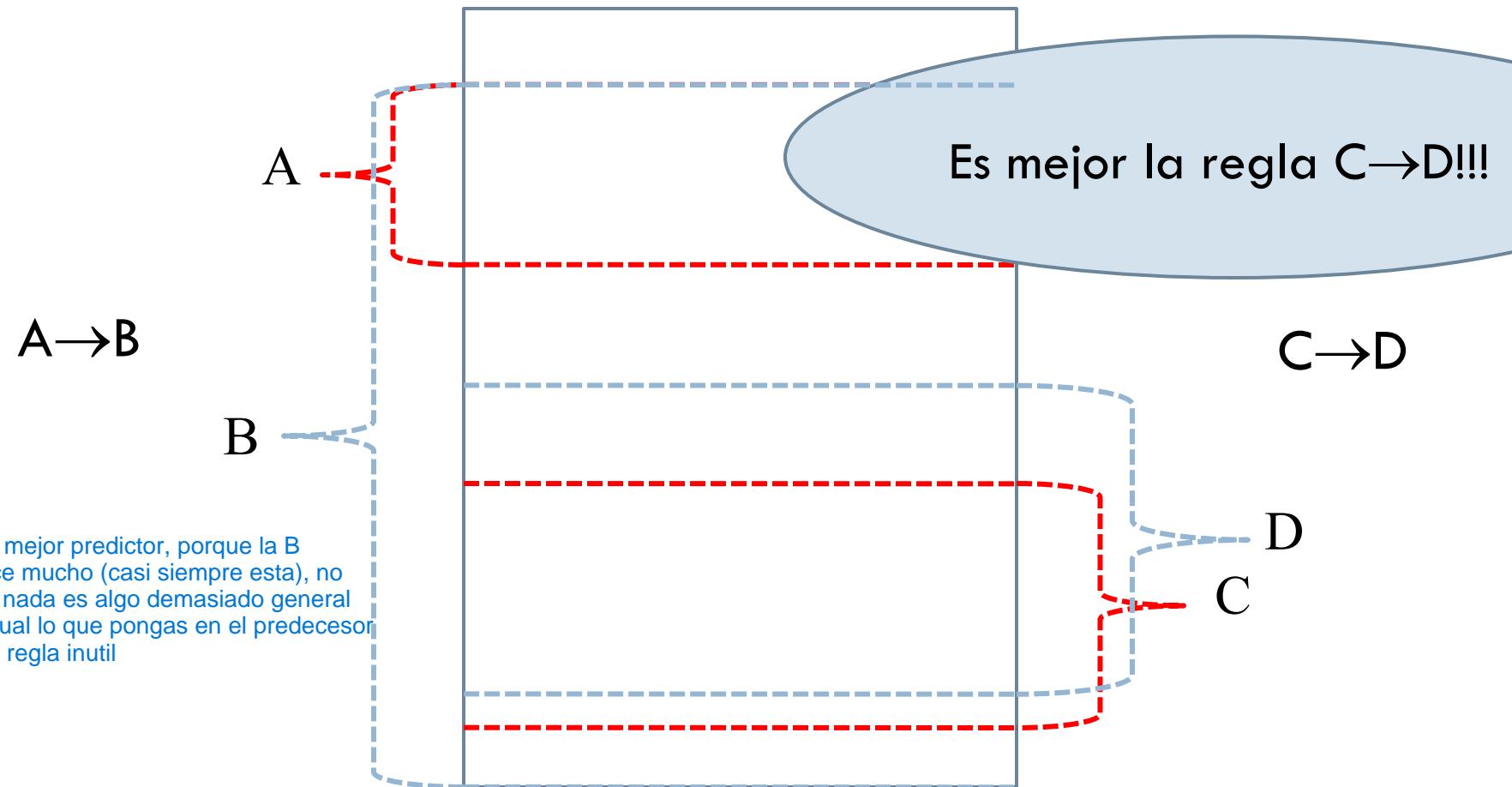
¿Qué regla es mejor?

Problemas abiertos: Medidas

Base de datos

$$\text{Sop}(A) = 0.2; \text{Sop}(B) = 0.9$$

$$\text{Sop}(C) = 0.3; \text{Sop}(D) = 0.4$$



Problemas abiertos: Medidas

- **Solución:** introducir nuevas medidas de calidad/interés que junto a las clásicas permitan evitar estos problemas.

- **Problema:** No existe la medida de calidad/interés que no tenga algún problema, por lo que tenemos que utilizar varias medidas que se complementen: **lift**, **factor de certeza**, etc.

Contenidos

- Aprendizaje no supervisado - contexto
- Definición de regla de asociación
- Medidas clásicas de las reglas de asociación
- Métodos clásicos de extracción de reglas
- Conjuntos maximales y cerrados
- Generación de reglas
- Problemas abiertos
- **Aplicaciones**

Aplicaciones

- El objetivo final de las técnicas de minería de datos es aportar conocimiento que ayude a la toma de decisiones.
- El conocimiento que aportan las reglas de asociación permite comprender mejor los procesos que generaron los datos (por ejemplo, tendencias en compra).
- También pueden usarse en ocasiones para tareas de predicción (no siempre en sentido futuro). Por ejemplo, si sabemos que el salario es alto, podemos deducir que los estudios son superiores. Esto requiere refinamiento en ocasiones.

Aplicaciones: Ejemplos

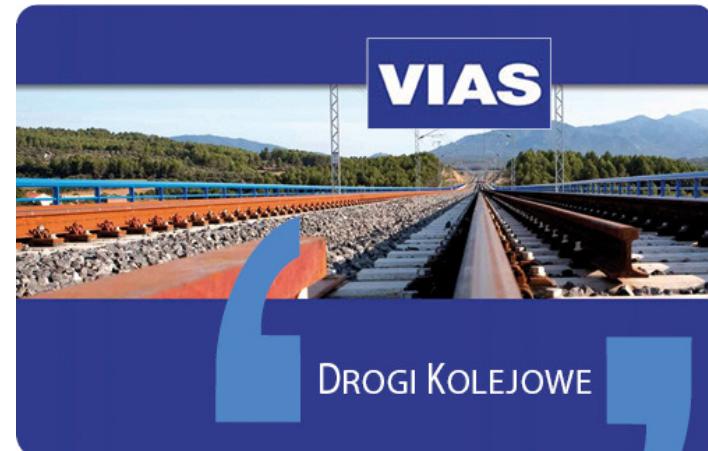
- Extracción de información a partir de datos bancarios



- Extracción de información a partir de los datos recopilados para el sistema de monitorizar la cabeza de una turbina de viento.



- Extracción de información a partir de los datos recopilados para las líneas de ferrocarril



Aplicaciones: Otras

- Minería de textos: se asocia la presencia de términos en documentos.



- Minería social: extraer asociaciones a partir de la información en redes sociales y mecanismos de comunicación:



- Minería de web:
 - Se asocian características de los internautas con el acceso a páginas, tiempos de acceso.
 - Minería de patrones secuenciales. Proporciona secuencias de visitas a páginas web que se repiten con frecuencia.

amazon

Google™