

# Asociación

## Minería de Datos

Por  
**Elizabeth León Guzmán, Ph.D.**  
Profesora  
Ingeniería de Sistemas y Computación

# Introducción

Una transacción en Base de datos transaccionales puede contener una **lista de “ítems”** (ej: productos comprados por un cliente)

Id	Items
1	{pan, leche}
2	{pan,pañales,cerveza,huevos}
3	{leche, pañales,cerveza, gaseosa
4	{pan,leche,pañales,cerveza}
5	{pan,leche,pañales,gaseosa}

# Introducción

Dado un conjunto de transacciones encontrar patrones, asociaciones, correlaciones o estructuras entre los conjuntos de “*items*” u objetos.

## Análisis de canasta de mercado

¿Pan es comprado con bananos frecuentemente?

¿leche es comprada con bananos frecuentemente?

¿La marca de la leche hace la diferencia?



¿Dónde debe ser ubicado en el almacén el jugo de manzana para maximizar sus ventas?

How Are The Demographics Of The Neighborhood Affecting What Customers Are Buying?

Imagen tomada de <https://bopenguin.com/glossary/market-basket-analysis>

## Reglas de asociación

Encontrar las reglas que predicen la **ocurrencia** de un ítem u objeto basado en las **ocurrencias** de otros ítems en la transacción

# Introducción

Interés en analizar los datos para aprender el comportamiento de las compras de sus clientes

- Promociones de mercadeo
- Manejo de inventario
- Relación con el cliente

## Análisis de canasta de mercado

¿Pan es comprado con bananos frecuentemente?

¿leche es comprada con bananos frecuentemente?

¿La marca de la leche hace la diferencia?

¿Dónde debe ser ubicado en el almacén el jugo de manzana para maximizar sus ventas?

How Are The Demographics Of The Neighborhood Affecting What Customers Are Buying?

Imagen tomada de <https://botpenguin.com/glossary/market-basket-analysis>



# Introducción

Id	Items
1	{pan, leche}
2	{pan, pañales, cerveza, huevos}
3	{leche, pañales, cerveza, gaseosa}
4	{pan, leche, pañales, cerveza}
5	{pan, leche, pañales, gaseosa}

## Ejemplo de Reglas de Asociación

$\{\text{Pañales}\} \rightarrow \{\text{Cerveza}\},$   
 $\{\text{Leche, Pan}\} \rightarrow \{\text{Cerveza, Gaseosa}\},$   
 $\{\text{Cerveza, Pan}\} \rightarrow \{\text{Leche}\},$

La implicación significa

**co-ocurrencia**

**No causalidad!**

# Introducción

Metodología de análisis de asociaciones: usada para descubrir relaciones interesantes ocultas en largos conjuntos de datos

## Aplicaciones

- Análisis de datos de mercado (Basket data análisis)
- Predicción
- Personalización
- Recomendación
- Navegación web

# Reglas de Asociación

Encontrar: todas las reglas que correlacionan la presencia de un conjunto de ítems con otro conjunto de ítems

## Ejemplos

- *98% de las personas que compran **llantas y accesorios de autos** también adquieren **servicios para autos***

**Convenios de mantenimiento de autos**

- *60% de usuarios de la Web que **visitan la Página A y B** compran el **ítem T1***

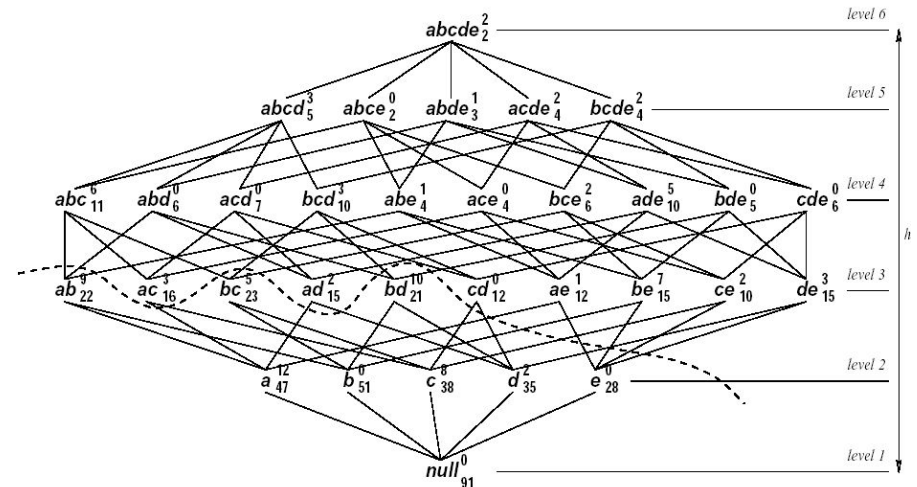
**Recomendaciones en Web (URL1 and URL3 -> URL5)**

# Representación de reglas de asociación

- Regla de asociación

Antecedente  $\rightarrow$  Consecuente [soporte, confianza]

- Conjunto de ítems frecuentes





# Representación de reglas de asociación

Regla: Una expresión de implicación de la forma:

**Antecedente**  $\rightarrow$  **Consecuente** [soporte, confianza]

$X \rightarrow Y$  [soporte, confianza]”, donde X y Y son itemsets

Ejemplo:

pañales  $\rightarrow$  cerveza [50%, 60%]

# Métricas de evaluación de la regla: Soporte y Confianza

- **Soporte (s)**

La fracción de transacciones que contienen a X y Y

- **Confianza (c)**

Medida de que tan frecuente los ítems en Y están en las transacciones que contienen a X

Id	Items
1	{pan,leche}
2	{pan,pañales,cerveza,huevos}
3	{leche,pañales,cerveza,gaseosa}
4	{pan,leche,pañales,cerveza}
5	{pan,leche,pañales,gaseosa}

$\{\text{Leche, Pañales}\} \Rightarrow \text{Cerveza}$

$$s = \frac{\sigma(\text{Leche, Pañales, Cerveza})}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Leche, Pañales, Cerveza})}{\sigma(\text{Leche, Pañales})} = \frac{2}{3} = 0.67$$

# Métricas de evaluación de la regla: Soporte y Confianza

- **Soporte**: Usado para eliminar reglas no interesantes:

*“regla con bajo soporte puede ocurrir por chance”*

- **Confianza**: mide que tan fiable es la inferencia hecha por la regla.

# Ejemplo

id_trans	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Soporte mínimo 50%,  
Confianza mínima 50%,

$A \Rightarrow C$  (?, ?)

$C \Rightarrow A$  (?, ?)

# Ejemplo

id_trans	Items
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%  
Min. confidence 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

- Para regla  $A \Rightarrow C$ :  
soporte = soporte ({A,C}) =  $2/4 = 0.5$       50%  
confianza = soporte ({A,C})/soporte ({A}) =  $2/3 = 0.66$       66.6%
- Para regla  $C \Rightarrow A$   
soporte = soporte ({C,A}) =  $2/4 = 0.5$       50%  
confianza = soporte ({C,A})/soporte ({C}) =  $2/2 = 1$       100%

# Minando reglas de asociación

Dado un conjunto de transacciones  $T$ , el objetivo de minar reglas de asociación es encontrar todas las reglas que tengan:

- Soporte  $\geq$  umbral *minsup*
- Confianza  $\geq$  umbral *minconf*

# Minando reglas de asociación

	Items
1	{pan, leche}
2	{pan, pañal, cerveza, huevo}
3	{leche, pañal, cerveza, gaseosa}
4	{pan, leche, pañal, cerveza}
5	{pan, leche, pañal, gaseosa}

Reglas con k=3:

$\{leche, pañal\} \rightarrow \{cerveza\}$  (s=0.4, c=0.67)  
 $\{leche, cerveza\} \rightarrow \{pañal\}$  (s=0.4, c=1.0)  
 $\{pañal, cerveza\} \rightarrow \{leche\}$  (s=0.4, c=0.67)  
 $\{cerveza\} \rightarrow \{leche, pañal\}$  (s=0.4, c=0.67)  
 $\{pañal\} \rightarrow \{leche, cerveza\}$  (s=0.4, c=0.5)  
 $\{leche\} \rightarrow \{pañal, cerveza\}$  (s=0.4, c=0.5)

Reglas con k=2:

$\{leche\} \rightarrow \{cerveza\}$  (s=0.4, c=0.5) ,  $\{cerveza\} \rightarrow \{leche\}$  (s=0.4, c=0.67)  
 $\{pañal\} \rightarrow \{cerveza\}$  (s=0.6, c=0.75) ,  $\{cerveza\} \rightarrow \{pañal\}$  (s=0.6, c=1)  
 $\{leche\} \rightarrow \{pañal\}$  (s=0.6, c=0.75) ,  $\{pañal\} \rightarrow \{leche\}$  (s=0.6, c=0.75)

## Observaciones:

- Las reglas del itemset K=3 {leche, pañal, cerveza}
  - Todas tienen el mismo soporte pero diferente confianza
  - Si el itemset NO es frecuente, las 6 reglas pueden ser podadas sin computar su confianza

# Minando reglas de asociación

Estrategia de dos pasos:

## 1. Generación de conjuntos de ítemsets frecuentes

Encontrar todos los itemsets que satisfacen el umbral mínimo de soporte

## 2. Generación de reglas

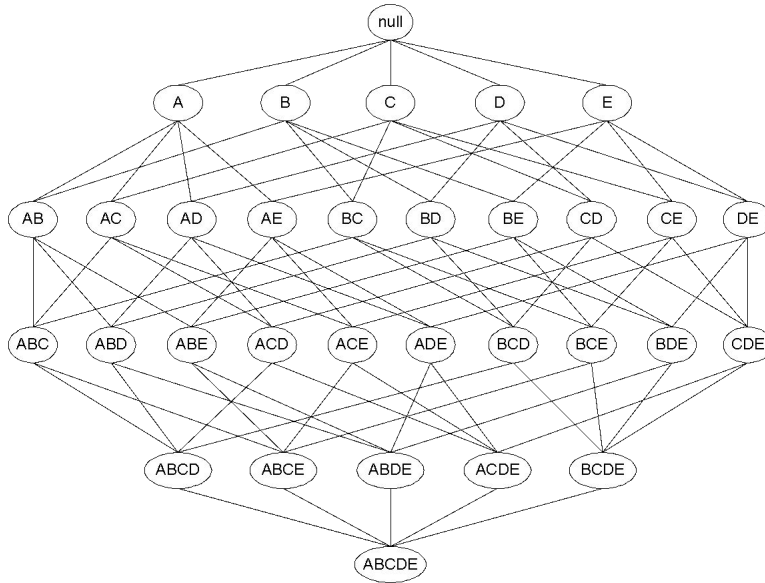
Extraer las reglas con alta confianza de los itemsets frecuentes encontrados en el paso anterior

Generación de conjuntos de Ítems frecuentes demasiado costoso!



# Generación de conjuntos de ítems frecuentes

Una estructura Lattice puede ser usada para enumerar la lista de todos los posibles conjuntos de ítems



Dados  $d$   
items, hay  
 $2^d$  itemsets  
candidatos

# Generación de conjuntos de ítems frecuentes

## Enfoque de **Fuerza Bruta**

- Cada itemset en el lattice es **candidato** a itemset frecuente
- Conteo del soporte de cada candidato scanning la base de datos

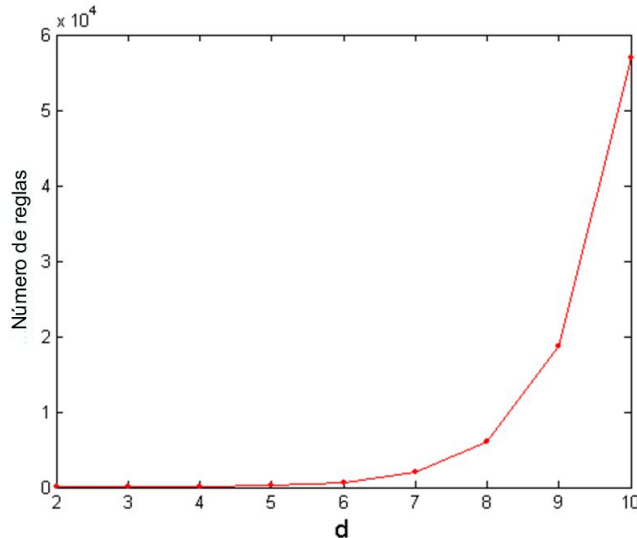


- Comparar cada transacción con cada candidato
- Complejidad  $\sim O(NMw) \Rightarrow$  **Costoso** dado que  $M = 2^d$  !!!

# Generación de conjuntos de ítems frecuentes

Dados  $d$  ítems únicos

- Número total de itemsets =  $2^d$
- Número total de posibles reglas de asociación:



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

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

# Minando reglas de asociación

**Fuerza bruta** : computar soporte y confianza por cada regla posible -> **prohibida** (demasiado costosa)

$$R = 3^d - 2^{d+1} + 1$$

$$R = 3^6 - 2^{7+1} + 1 = 602$$

**Podar reglas antes** de computar sus valores de soporte y confianza (para evitar cálculos innecesarios)

Conjuntos de ítems frecuentes

# Generación de conjuntos de ítems frecuentes

## Reducir el **número de candidatos** (M)

- Búsqueda completa:  $M = 2^d$
- Utilice técnicas de poda para reducir M

## Reducir el **número de transacciones** (N)

- Reducir el tamaño de N mientras el tamaño de los itemset aumenta

## Reducir el **número de comparaciones** (NM)

- Uso eficiente de estructuras de datos para almacenar los candidatos o las transacciones
- No es necesario evaluar todos los candidatos contra cada transacción

# Generación de conjuntos de ítems frecuentes

Reducir el número de itemsets candidatos durante la generación de los itemsets frecuentes con la ayuda del soporte

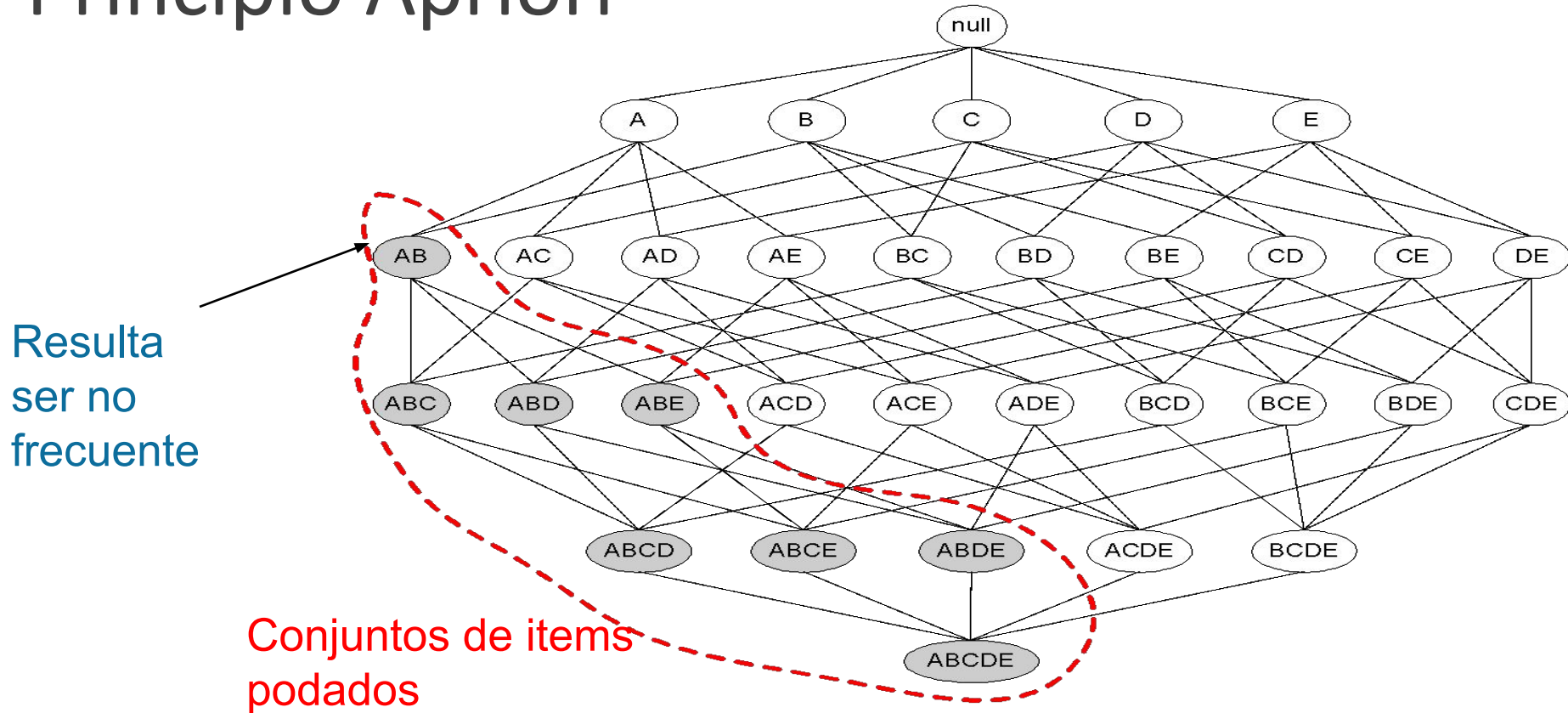
## Principio Apriori

Cualquier subconjunto de un ítemset frecuente debe ser frecuente

Si  $\{AB\}$  es un itemset frecuente, entonces  $\{A\}$  y  $\{B\}$  deben ser itemsets frecuentes:

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


# Principio Apriori



# Principio Apriori

Ítem	Cuenta
Pan	4
Gaseosa	2
Leche	4
Cerveza	3
Pañales	4
Huevos	1

ítems (1-itemsets)



Itemset	Cuenta
{Pan, Leche}	3
{Pan, Cerveza}	2
{Pan, Pañales}	3
{Leche, Cerveza}	2
{Leche, Pañales}	3
{Cerveza, Pañales}	3

ítems (2-itemsets)

Soporte Mínimo= 3

Si se considera cada subconjunto,

$${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$$

Con poda basada en el soporte,

$$6 + 6 + 1 = 13$$

Tríos (3-itemsets)



Itemset	Cuenta
{Pan, Leche, Pañales}	3



# Algoritmo “Apriori”

Primer algoritmo que usa el soporte para controlar el crecimiento de los itemsets candidatos

## Método:

- Generar itemsets de longitud  $k = 1$
- Contar el soporte de cada uno de los itemsets (scan a la BD)
- Podar itemsets que no sean frecuentes
- Repetir hasta que no se identifiquen nuevos itemsets frecuentes
  - Generar itemsets candidatos de longitud  $(k + 1)$  a partir de itemsets frecuentes de longitud  $k$
  - Podar itemsets candidatos que contengan subconjuntos de longitud  $k$  que no sean frecuentes
  - Contar el soporte de cada uno de los candidatos (scan a la BD )
  - Eliminar los candidatos que sean no sean frecuentes, dejando sólo aquellos frecuentes
  - $k=k+1$

# Apriori Algorithm — Example (supp=50%)

Datos D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$C_1$

Scan D

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

$L_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

$L_2$

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

Scan D

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

$C_3$

itemset
{2 3 5}

Scan D

$L_3$

itemset	sup
{2 3 5}	2

# Generación de reglas de asociación

Una vez encontrados los itemsets frecuentes, se generan las reglas de asociación

confianza  $\geq \textit{min\_conf}$

- Por cada itemset frecuente  $I$ , generar los subconjuntos no vacíos de  $I$ ;
- Por cada subconjunto no vacío  $s$  de  $I$ ,
  - Si (  $\text{support\_count}(I) / \text{support\_count}(s) \geq \textit{min\_conf}$  )  
THEN output the rule “ $s \Rightarrow (I-s)$ ”

$$C(\{2,3\} \rightarrow 5) = s(2,3,5)/s(2,3)=2/2 = 100\%$$

$$C\{3 \rightarrow 1\} = ?$$

No todas las reglas encontradas pueden ser interesantes para presentar y usar

# Generación de reglas de asociación

Si  $L = \{A, B, C, D\}$  es un itemset frecuente, reglas candidatas:

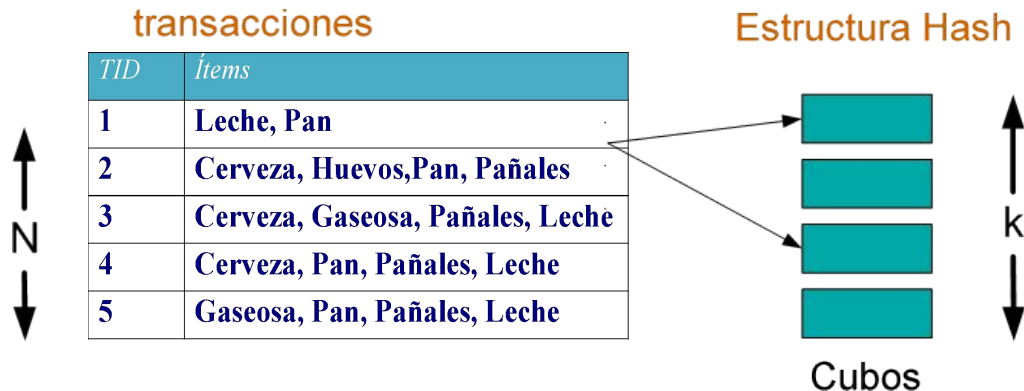
$ABC \rightarrow D,$	$ABD \rightarrow C,$	$ACD \rightarrow B,$	$BCD \rightarrow A,$
$A \rightarrow BCD,$	$B \rightarrow ACD,$	$C \rightarrow ABD,$	$D \rightarrow ABC$
$AB \rightarrow CD,$	$AC \rightarrow BD,$	$AD \rightarrow BC,$	$BC \rightarrow AD,$
$BD \rightarrow AC,$	$CD \rightarrow AB,$		

Si  $|L| = k$ , entonces hay  $2^k - 2$  reglas de asociación candidatas (ignorando  $L \rightarrow \emptyset$  y  $\emptyset \rightarrow L$ )

# Reducir el número de comparaciones

## Conteo de candidatos:

- Explorar la base de datos de transacciones para determinar el soporte de cada itemset candidato
- Para reducir el número de comparaciones, almacenar los candidatos en una estructura de hash
- En lugar de comparar cada transacción contra todos los candidatos, comparar contra los candidatos contenidos en los cubos de la estructura



# Factores que afectan la complejidad

## Elección de un umbral mínimo de soporte

- la reducción del umbral de soporte resulta en más itemsets frecuentes
- esto puede aumentar el número de candidatos y la longitud máxima de los itemsets frecuentes

## Dimensionalidad (número de ítems) del conjunto de datos

- se necesita más espacio para almacenar la cuenta de soporte de cada ítem
- si el número de ítems frecuentes también aumenta, tanto los costos computacionales como de E/S también pueden aumentar

# Factores que afectan la complejidad

## Tamaño de la base de datos

- Como Apriori hace varias pasadas, el tiempo de ejecución del algoritmo puede aumentar con el número de transacciones

## Extensión promedio de transacción

- La extensión de una transacción aumenta en conjuntos de datos más densos
- Esto puede aumentar la longitud máxima de itemsets frecuentes y las travesías del árbol hash (número de subconjuntos en una transacción se incrementa con su extensión)

# Soporte

¿Cómo determinar el valor apropiado para el umbral *minsup*?

- Si *minsup* es muy alto, se podrían perder itemsets que involucren ítems raros e interesantes (e.g., Productos costosos)
- Si *minsup* es muy bajo, es computacionalmente costoso y el número de itemsets es muy alto

Usar sólo un umbral de soporte mínimo puede no ser efectivo



# Medidas de interés

Los algoritmos de reglas de asociación tienden a producir demasiadas reglas

- Muchas de ellas son redundantes o no son interesantes
- Redundante si  $\{A,B,C\} \rightarrow \{D\}$  y  $\{A,B\} \rightarrow \{D\}$  tienen el mismo soporte y confianza

Las medidas de interés pueden ser usadas para podar/ordenar los patrones derivados

En la formulación original de reglas de asociación, el soporte y la confianza son las únicas medidas usadas

# Medidas de interés

- Dada una regla  $X \rightarrow Y$ , La información necesaria para calcular la medida de interés de una regla se puede obtener de una tabla de contingencia

Tabla de Contingencia para  $X \rightarrow Y$

	Y	$\overline{Y}$	
X	$f_{11}$	$f_{10}$	$f_{1+}$
$\overline{X}$	$f_{01}$	$f_{00}$	$f_{0+}$
	$f_{+1}$	$f_{+0}$	$ T $

$f_{11}$ : soporte de X y Y

$f_{10}$ : soporte de X y  $\overline{Y}$

$f_{01}$ : soporte de  $\overline{X}$  y Y

$f_{00}$ : soporte de  $\overline{X}$  y  $\overline{Y}$

El soporte y la confianza pueden ser interpretados como un estimado de la probabilidad.  $s = P(X, Y)$  y  $c = P(Y | X)$

Se usa para definir varias medidas

- soporte, confianza, lift, Gini, etc.

# Desventajas de la confianza

	Café	<u>Café</u>	
Té	150	50	200
<u>Té</u>	650	150	800
	800	200	1000

Interés en analizar la relación entre personas que beben té y café en un grupo de 1000 personas

Té → Café

soporte =  $150/1000 = 15\%$     confianza =  $150/200 = 75\% \rightarrow P(\text{Café} | \text{Té}) = 0.75$

Podría ser aceptable, pero la fracción de personas que:

- beben café independiente de si beben té es del **80%**, y las que
- beben té y también café es de **75%**

El conocer que una persona bebe té decrece la probabilidad de tomar café de 80% a 75%, por lo que la regla **Té → Café** desorienta a pesar de tener alta confianza

# Medidas basadas en estadísticas

Es el ratio entre la confianza de la regla y el soporte del consecuente de la regla.  
Si los valores son binarios es equivalente a la medida “interest factor”

$$Lift = \frac{c(X \rightarrow Y)}{s(Y)} = \frac{P(Y | X)}{P(Y)} = \frac{P(X, Y)}{P(X)P(Y)}$$

Para el ejemplo anterior **Té** → **Café**

$$Lift = I(X, Y) = \frac{s(X, Y)}{s(X) \times s(Y)} = \frac{f_{11}}{f_{1+} f_{+1}}$$

$$Lift = \frac{c(té \rightarrow café)}{s(café)} = \frac{0.75}{0.8} = 0.9375 = \frac{0.15}{0.2 \times 0.8}$$

# Medidas basadas en estadísticas

$$I(X,Y) = \begin{cases} =1 & X \text{ y } Y \text{ independientes} \\ >1 & X \text{ y } Y \text{ correlacionadas Positivamente} \\ <1 & X \text{ y } Y \text{ correlacionadas Negativamente} \end{cases}$$

$$Lift = \frac{c(té \rightarrow café)}{s(café)} = \frac{0.75}{0.8} = 0.9375 = \frac{0.15}{0.2 \times 0.8}$$

# Desventaja de Lift e Interés

Dominio de texto: p,q,r y s son términos en documentos

	p	$\overline{p}$	
q	880	50	930
$\overline{q}$	50	20	70
	930	70	1000

$$s(q \rightarrow p) = \frac{880}{1000} = 0.88$$

$$Lift = \frac{c(q \rightarrow p)}{s(p)} = \frac{0.946}{0.930} = 1.017$$

	r	$\overline{r}$	
s	20	50	70
$\overline{s}$	50	880	930
	70	930	1000

$$s(s \rightarrow r) = \frac{20}{1000} = 0.02$$

$$Lift = \frac{c(s \rightarrow r)}{s(r)} = \frac{0.285}{0.07} = 4.071$$

# Medidas basadas en estadísticas

Medidas que tienen en cuenta la dependencia estadística

$$Lift = \frac{P(Y | X)}{P(Y)} = \frac{P(X, Y)}{P(X)P(Y)} = \frac{f_{11}}{f_{1+}f_{+1}}$$

$$PS = P(X, Y) - P(X)P(Y) = f_{11} - f_{1+}f_{+1}$$

$$coeficiente - \phi = \frac{P(X, Y) - P(X)P(Y)}{\sqrt{P(X)[1 - P(X)]P(Y)[1 - P(Y)]}}$$

$$= \frac{f_{11} - f_{1+}f_{+1}}{\sqrt{f_{1+}(1 - f_{1+})f_{+1}(1 - f_{+1})}}$$

Existen varias medidas propuestas en la literatura

Algunas medidas son buenas en ciertas aplicaciones pero no en otras

¿Qué criterio se debería usar para determinar si una medida es buena o mala?

#	Measure	Formula
1	$\phi$ -coefficient	$\frac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's ( $\lambda$ )	$\frac{\sum_j \max_k P(A_j, B_k) + \sum_k \max_j P(A_j, B_k) - \max_j P(A_j) - \max_k P(B_k)}{2 - \max_j P(A_j) - \max_k P(B_k)}$
3	Odds ratio ( $\alpha$ )	$\frac{P(A,B)P(\bar{A},\bar{B})}{P(A,\bar{B})P(\bar{A},B)}$
4	Yule's $Q$	$\frac{P(A,B)P(\bar{A}\bar{B}) - P(A,\bar{B})P(\bar{A},B)}{P(A,B)P(\bar{A}\bar{B}) + P(A,\bar{B})P(\bar{A},B)} = \frac{\alpha - 1}{\alpha + 1}$
5	Yule's $Y$	$\frac{\sqrt{P(A,B)P(\bar{A}\bar{B})} - \sqrt{P(A,\bar{B})P(\bar{A},B)}}{\sqrt{P(A,B)P(\bar{A}\bar{B})} + \sqrt{P(A,\bar{B})P(\bar{A},B)}} = \frac{\sqrt{\alpha} - 1}{\sqrt{\alpha} + 1}$
6	Kappa ( $\kappa$ )	$\frac{P(A,B) + P(\bar{A},\bar{B}) - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}$
7	Mutual Information ( $M$ )	$\frac{\sum_i \sum_j P(A_i, B_j) \log \frac{P(A_i, B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i) \log P(A_i), -\sum_j P(B_j) \log P(B_j))}$
8	J-Measure ( $J$ )	$\max \left( P(A, B) \log \left( \frac{P(B A)}{P(B)} \right) + P(\bar{A}\bar{B}) \log \left( \frac{P(\bar{B} \bar{A})}{P(\bar{B})} \right), \right. \\ \left. P(A, B) \log \left( \frac{P(A B)}{P(A)} \right) + P(\bar{A}B) \log \left( \frac{P(\bar{A} B)}{P(\bar{A})} \right) \right)$
9	Gini index ( $G$ )	$\max \left( P(A)[P(B A)^2 + P(\bar{B} A)^2] + P(\bar{A})[P(B \bar{A})^2 + P(\bar{B} \bar{A})^2] \right. \\ \left. - P(B)^2 - P(\bar{B})^2, \right. \\ \left. P(B)[P(A B)^2 + P(\bar{A} B)^2] + P(\bar{B})[P(A \bar{B})^2 + P(\bar{A} \bar{B})^2] \right. \\ \left. - P(A)^2 - P(\bar{A})^2 \right)$
10	Support ( $s$ )	$P(A, B)$
11	Confidence ( $c$ )	$\max(P(B A), P(A B))$
12	Laplace ( $L$ )	$\max \left( \frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2} \right)$
13	Conviction ( $V$ )	$\max \left( \frac{P(A)P(\bar{B})}{P(\bar{A}B)}, \frac{P(B)P(\bar{A})}{P(\bar{B}A)} \right)$
14	Interest ( $I$ )	$\frac{P(A,B)}{P(\bar{A})P(\bar{B})}$
15	cosine ( $IS$ )	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's ( $PS$ )	$P(A, B) - P(A)P(B)$
17	Certainty factor ( $F$ )	$\max \left( \frac{P(B A) - P(B)}{1 - P(B)}, \frac{P(A B) - P(A)}{1 - P(A)} \right)$
18	Added Value ( $AV$ )	$\max(P(B A) - P(B), P(A B) - P(A))$
19	Collective strength ( $S$ )	$\frac{P(A,B) + P(\bar{A}\bar{B})}{P(A)P(B) + P(\bar{A})P(\bar{B})} \times \frac{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A,B) - P(\bar{A}\bar{B})}$
20	Jaccard ( $\zeta$ )	$\frac{P(A,B)}{P(A,B) + P(\bar{A}\bar{B})}$
21	Kloggen ( $K$ )	$\sqrt{P(A, B) \max(P(B A) - P(B), P(A B) - P(A))}$



# FP-GROWTH método

Longitud 100 número de candidatos es al menos:

$$\sum_{i=1}^{100} \binom{100}{i} = 2^{100} - 1 \approx 10^{30}$$

Complejidad de computación crece exponencialmente!

**Frequent pattern growth** método: mejora escalabilidad (grandes conjuntos de datos)

# FP-GROWTH

1. Proyección de los conjuntos de ítems frecuentes
2. Construye una estructura en memoria llamada FP-tree
3. Aplicar algoritmo FP-Growth

Id	Items
1	f,a,c,d,g,i,m,p
2	a,b,c,f,l,m,o
3	b,f,h,j,o
4	b,c,k,s,p
5	a,f,c,e,l,p,m,n

1

Scan

Mínimo soporte = 3

Id	Soporte
f	4
c	4
a	3
b	3
m	3
p	3

Frecuencias ordenadas  
descendentemente.

Orden **importante!**

FP-tree sigue ese orden

Id	Items
1	f,a,c,d,g,i,m,p
2	a,b,c,f,l,m,o
3	b,f,h,j,o
4	b,c,k,s,p
5	a,f,c,e,l,p,m,n

1

Scan

Mínimo soporte = 3

Id	Soporte
f	4
c	4
a	3
b	3
m	3
p	3

Frecuencias ordenadas  
descendentemente.

Orden **importante!**

FP-tree sigue ese orden

2

Raíz del árbol es creada

**ROOT**

Id	Items
1	f,a,c,d,g,i,m,p
2	a,b,c,f,l,m,o
3	b,f,h,j,o
4	b,c,k,s,p
5	a,f,c,e,l,p,m,n

1

Scan

Mínimo soporte = 3

Id	Soporte
f	4
c	4
a	3
b	3
m	3
p	3

3 Scan de la primera transacción. Construcción de la primera rama del árbol (solo los ítems que están en la lista de frecuentes)

Frecuencias ordenadas descendentemente.

Orden **importante!**

FP-tree sigue ese orden

2

Raíz del árbol es creada

**ROOT**



Id	Items
1	f,a,c,d,g,i,m,p
2	a,b,c,f,l,m,o
3	b,f,h,j,o
4	b,c,k,s,p
5	a,f,c,e,l,p,m,n

1

Scan

Mínimo soporte = 3

Id	Soporte
f	4
c	4
a	3
b	3
m	3
p	3

Frecuencias ordenadas  
descendentemente.

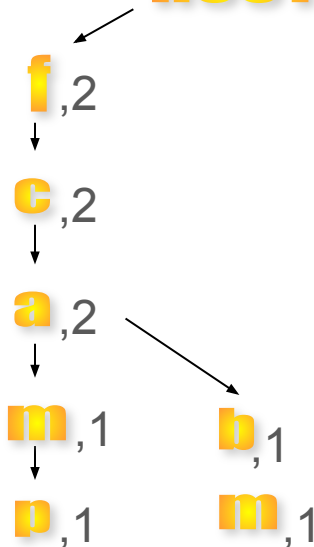
Orden **importante!**

FP-tree sigue ese orden

2

Raíz del árbol es creada

**ROOT**



3 Scan de la primera transacción. Construcción de la primera rama del árbol (solo los ítems que están en la lista de frecuentes)

4 Scan de la segunda transacción

Id	Items
1	f,a,c,d,g,i,m,p
2	a,b,c,f,l,m,o
3	b,f,h,j,o
4	b,c,k,s,p
5	a,f,c,e,l,p,m,n

1

Scan

Mínimo soporte = 3

Id	Soporte
f	4
c	4
a	3
b	3
m	3
p	3

Frecuencias ordenadas  
descendentemente.

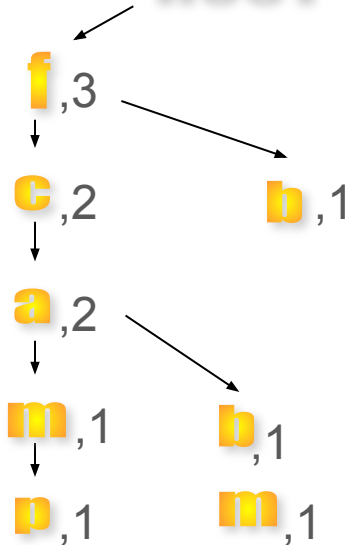
Orden **importante!**

FP-tree sigue ese orden

2

Raíz del árbol es creada

**ROOT**



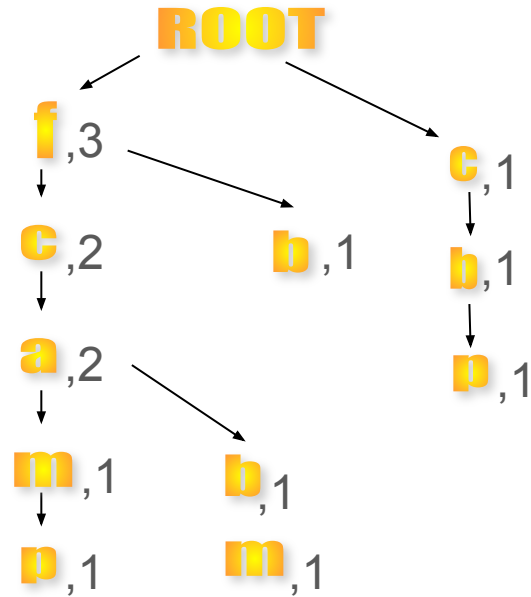
3 Scan de la primera transacción. Construcción de la primera rama del árbol (solo los ítems que están en la lista de frecuentes)

4 Scan de la segunda transacción

5 Scan de la tercera transacción

Terminar el árbol: scan de 4 y 5 transacción!

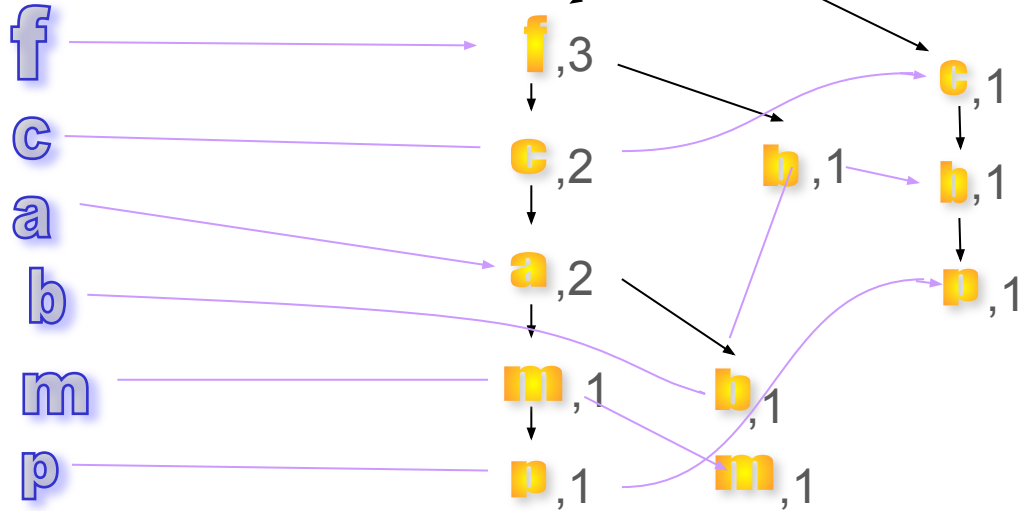
# Final FP-tree





## Final FP-tree

## Lista de cabezas (headers)



# Algoritmo FP-Growth

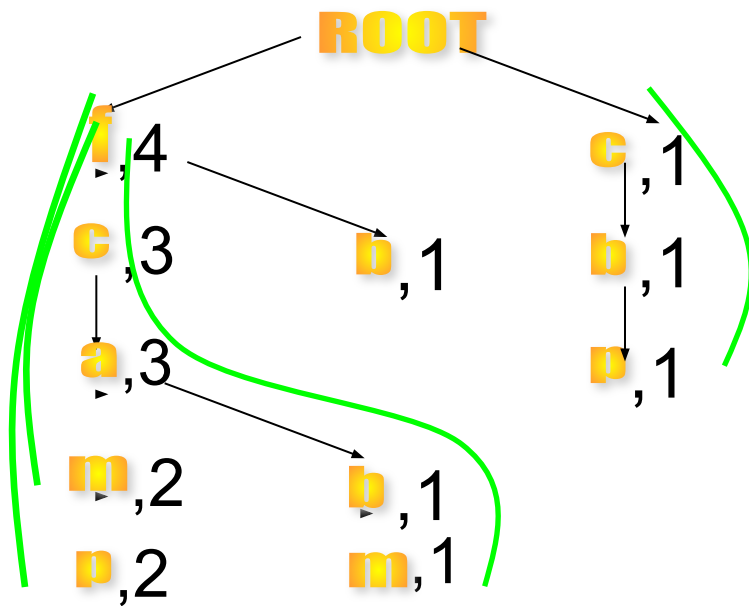
Coleccionar las transacciones donde **p** participa comenzando por el **header** y siguiendo los nodos enlazados,

luego

m sin p

b sin p ni m ....

e ir seleccionando las que al contar dan más del umbral de soporte



**p**

2 caminos

Ejemplos o registros

$\{ (f,2), (c,2), (a,2), (m,2), (p,2) \}$

$\{ (c,1), (b,1), (p,1) \}$

Umbral de 3

$\{ (c,3), (p,3) \}$

$\{ c, p \}$

**m sin p**

Ejemplos o registros acumulados:

$\{ (f,2), (c,2), (a,2), (m,2) \}$

$\{ (f,1), (c,1), (a,1), (b,1), (m,1) \}$

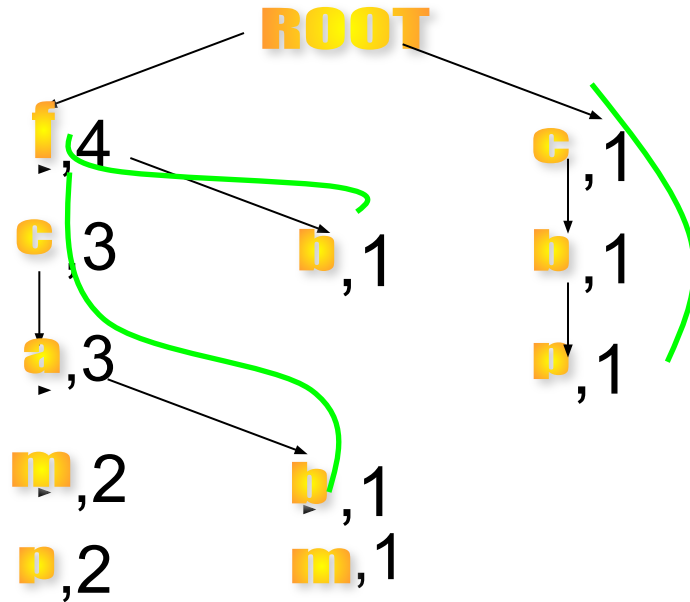
Conjuntos de ítems frecuentes son:

$\{ (f,3), (c,3), (a,3), (m,3) \}$

$\{ f, c, a, m \}$

**b sin m y sin p**

3 caminos



Ejemplos o registros acumulados:

$\{(f,1), (c,1), (a,1), (b,1)\}$

$\{(f,1), (b,1)\}$

$\{(c,1), (b,1)\}$

Conjuntos de ítems frecuentes son:

$\{(b,3)\}$

$\{b\}$

terminar!

# Reglas de asociación multidimensionales

Id	A1	A2	Items
1	a	1	{pan, leche}
2	b	2	{pan,pañales,cerveza,huevos}
3	a	2	{leche, pañales,cerveza, gaseosa
4	a	3	{pan,leche,pañales,cerveza}
5	c	1	{pan,leche,pañales,gaseosa}

# Reglas de asociación multidimensionales

Sentido común

Dividir el proceso de minería en dos pasos:

- Minar información dimensional
- Frequent itemsets

Encontrar combinaciones de valores multidimensionales y luego los correspondientes conjuntos de ítems frecuentes

# Ejemplo

Combinación de los valores de los atributos

Umbral de n (considerados frecuentes)

Patrón Multidimensional

MD-Pattern

Algoritmo BUC

Id	A1	A2	A3	Items
1	a	1	m	x, y, z
2	b	2	n	z, w
3	a	2	m	x, z, w
4	c	3	p	x, w

# Ejemplo

Usando umbral de 2

I.

- Ordenar por atributo A1

Id	A1	A2	A3	Items
1	a	1	m	x, y, z
3	a	2	m	x, z, w
2	b	2	n	z, w
4	c	3	p	x, w



# Ejemplo

Usando umbral de 2

I.

- Ordenar por atributo A1
- MD-pattern es **a**
- Seleccionar las tuplas con el MD-pattern encontrado

Id	A1	A2	A3	Items
1	a	1	m	x, y, z
3	a	2	m	x, z, w
2	b	2	n	z, w
4	c	3	p	x, w

# Ejemplo

Usando umbral de 2

I.

- Ordenar por atributo A1
- MD-pattern es **a**
- Seleccionar las tuplas con el MD-pattern encontrado
- Ordenar el subconjunto de tuplas con respecto a la segunda dimensión A2

Id	A1	A2	A3	Items
1	a	1	m	x, y, z
3	a	2	m	x, z, w

Ningún valor (1y 2) cumple el umbral, la dimensión A2 es ignorada

5. Ordenar con respecto a la tercera dimensión A3
6. Md-Pattern es **m**

Id	A1	A2	A3	Items
1	a	1	m	x, y, z
3	a	2	m	x, z, w

**(a,\*,m)**

II. Repetir el proceso del paso I, comenzando por el atributo A2 (A1 no es analizada en esta iteración)

MD-pattern es 2

continuando con A3 del subconjunto, no hay MD-patterns

Id	A1	A2	A3	Items
1	a	1	m	x, y, z
3	a	2	m	x, z, w
2	b	2	n	z, w
4	c	3	p	x, w

(\* , 2 , \*)

III. Repetir el proceso del paso I, comenzando por el atributo A3 (A1 y A2 no son analizados)

MD-pattern es **m**

continuando con A3 del subconjunto, no hay MD-patterns

Id	A1	A2	A3	Items
1	a	1	m	x, y, z
3	a	2	m	x, z, w
2	b	2	n	z, w
4	c	3	p	x, w

(\*, \*, m)

Minar los conjuntos frecuentes por cada MD-pattern

Otra forma: Minar primero conjuntos de items frecuentes y después encontrar los MD-patterns.

# Bibliografía

- [1] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, 2005, Introduction to Data Mining, Addison-Wesley.
- [2] Kantardzic, 2011. Data Mining: Concepts, Models, Methods an Algorithms