Universidad Autónoma de Nuevo León
Facultad de Ciencias Físico Matemáticas

Semestre: 7

Materia:
Minería de datos

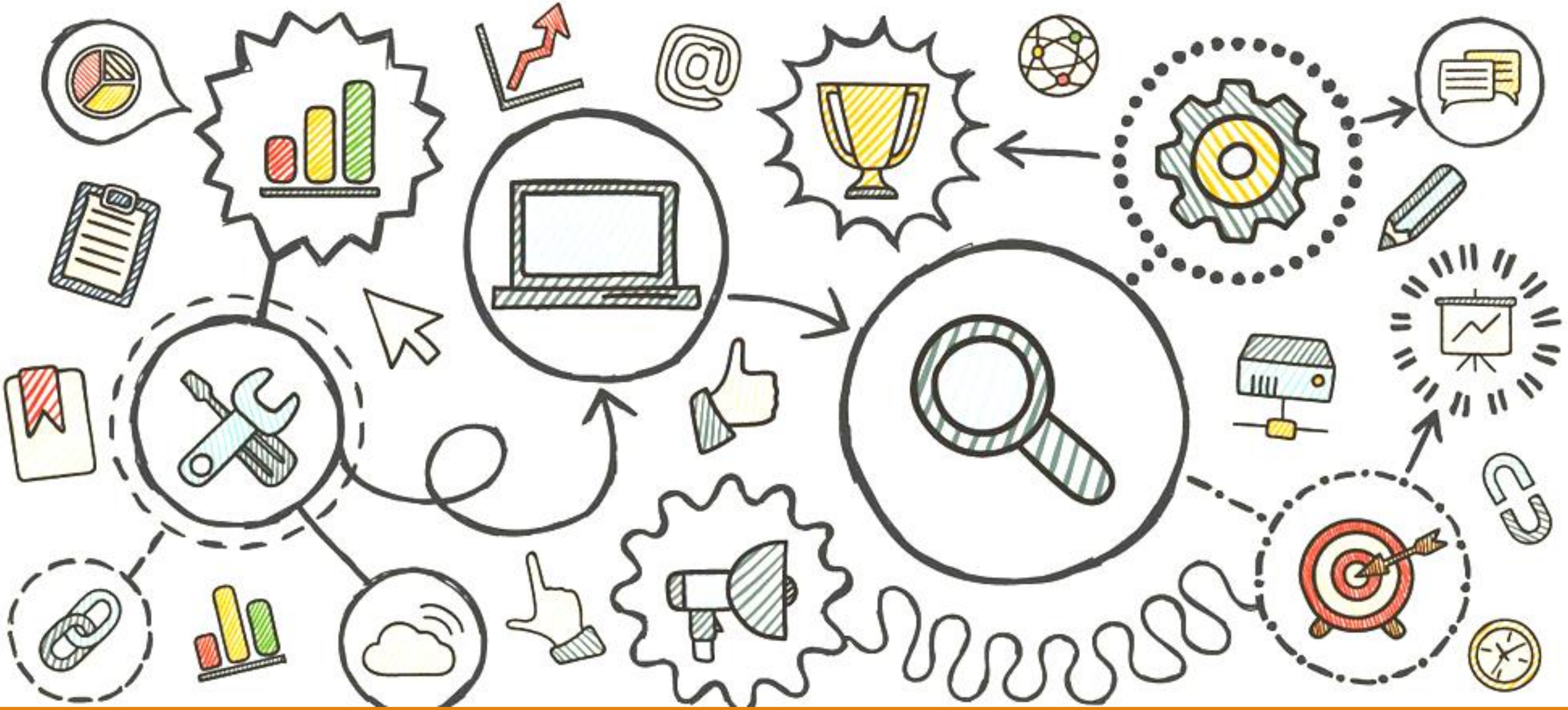
Maestra:
Mayra Cristina Berrones Reyes

Alumnos :

Andrea López Solís #1822031
Daniela Govea Serna #1722714
Francisco García Sánchez Armáss #1816358
Jesús Eduardo Valencia González #1630606
Karyme Mayela Gauna Rodríguez #1819032

Grupo: 003
Aula: AVI2

San Nicolas de los Garza Nuevo León a 18 de septiembre del 2020



Reglas de Asociación

(Association Rules)

Reglas de asociación

Búsqueda de patrones frecuentes, asociaciones, correlaciones o estructuras causales entre conjuntos de elementos u objetos en bases de datos de transacciones, bases de datos relacionales y otros repositorios de información disponibles.

Aplicaciones

-Análisis de datos de la banca.

-Cross-marketing (poner la crema batida junto a las fresas).

-Diseño de catálogos.

Ejemplo

Dado un conjunto de transacciones, encontrar reglas que predigan la ocurrencia de un artículo según las ocurrencias de otros artículos en la transacción.

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

Ejemplos:

- La gente que comprara pan, comprara también leche.
- La gente que comprara soda, comprara también papas de funda.
- La gente que comprara pan, comprara también mermelada.

Soporte:
Fracción de transacciones que contiene un itemset.
s({Leche, Pan}) = 3
s ({Soda, Chips}) = 4

Conjunto de elementos frecuente:
Un conjunto de elementos cuyo soporte es mayor o igual que un umbral de mínimo.

Conjunto de elementos:
Una colección de uno o más artículos, por ejemplo, {leche, pan, mermelada}. k-itemset, un conjunto de elementos que contiene k elementos.

Recuento de soporte:
Frecuencia de ocurrencia de un itemset.

S({Leche, Pan}) = 3

S({Soda, Chips}) = 4

Confianza (c):
Mide que tan frecuente items en Y aparecen en transacciones que contienen X.

$$s = \frac{\sigma(\{Leche, Pan\})}{\#Núm.\ transiciones} = \frac{3}{8} = 0.375 \qquad c = \frac{\sigma(\{Leche, Pan\})}{\sigma(\{Pan\})} = \frac{3}{4} = 0.75$$

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

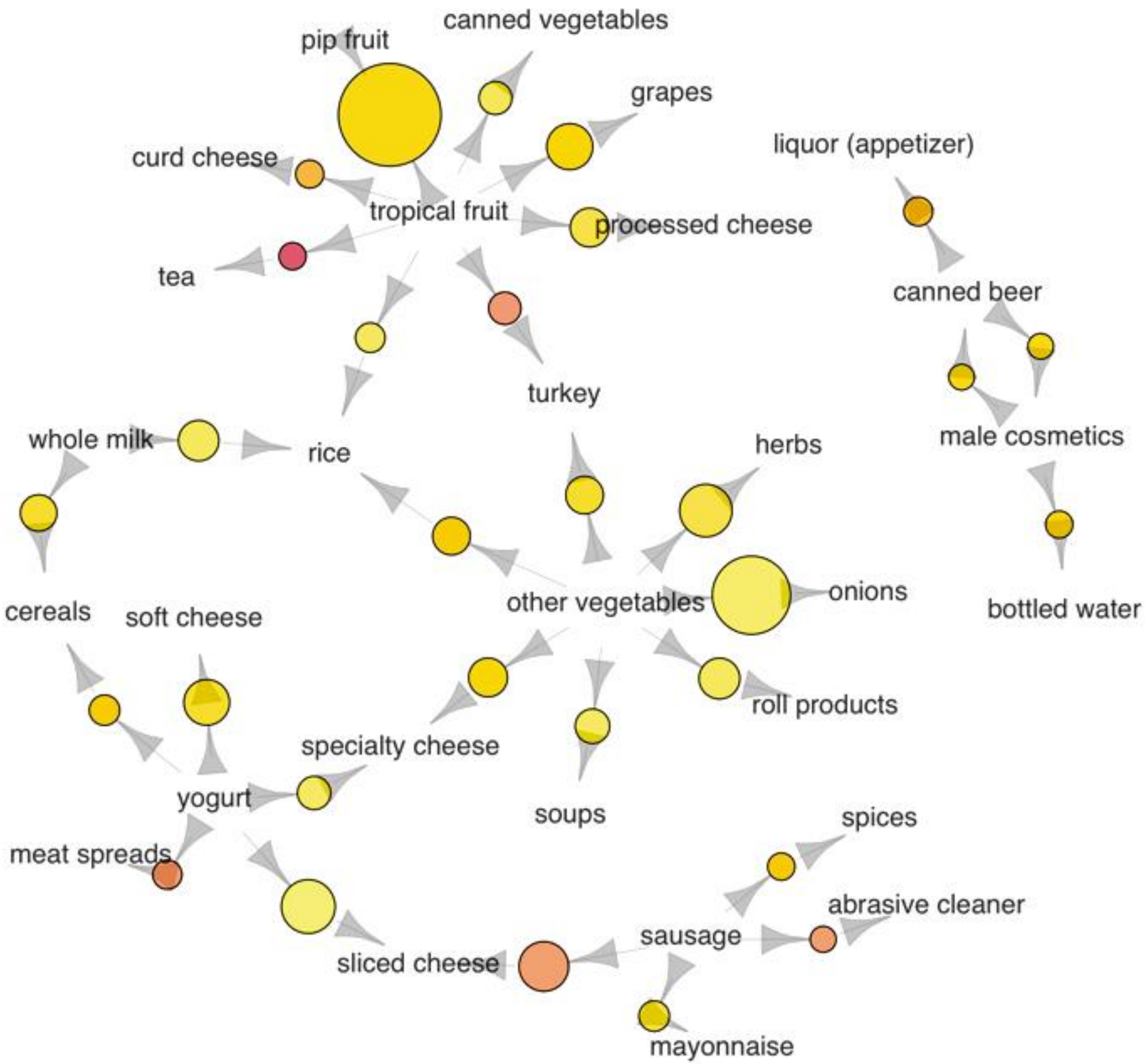
El Objetivo

Dado un conjunto de transacciones T, el objetivo de la minería de reglas de asociación es encontrar todas las reglas teniendo:

- Umbral mínimo de soporte
- Umbral mínimo de confianza

Enfoque de fuerza bruta:

- Lista todas las reglas de asociación posibles.
- Compruebe el soporte y la confianza para cada regla .
- Elimine las reglas que fallan en los umbrales mínimos.

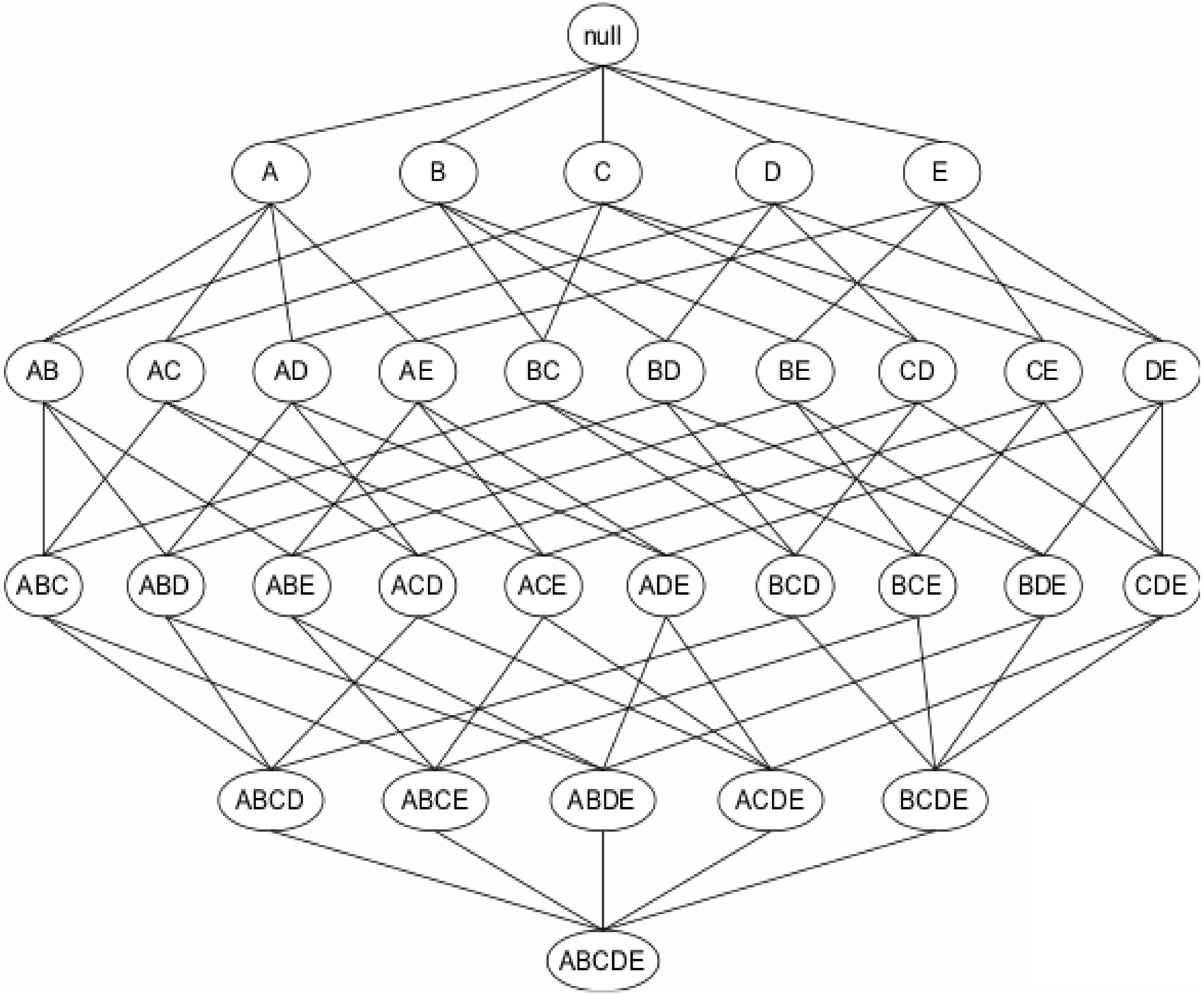


Reglas de la Asociación Minera: Enfoque de dos pasos

RAM: Enfoque 2 pasos

<u>Generación de elementos frecuentes:</u>
Generar todos los conjuntos de elementos cuyo soporte ≥ min sup.

<u>Generación de reglas:</u>
Generar reglas de alta confianza a partir de un conjunto de elementos frecuentes. Cada regla es una partición binaria de un conjunto de elementos frecuente.

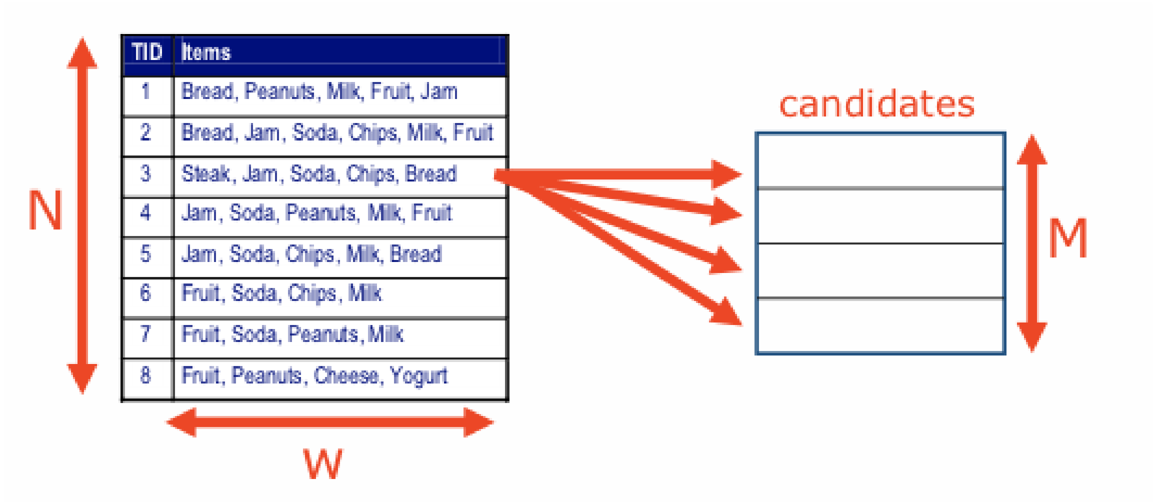


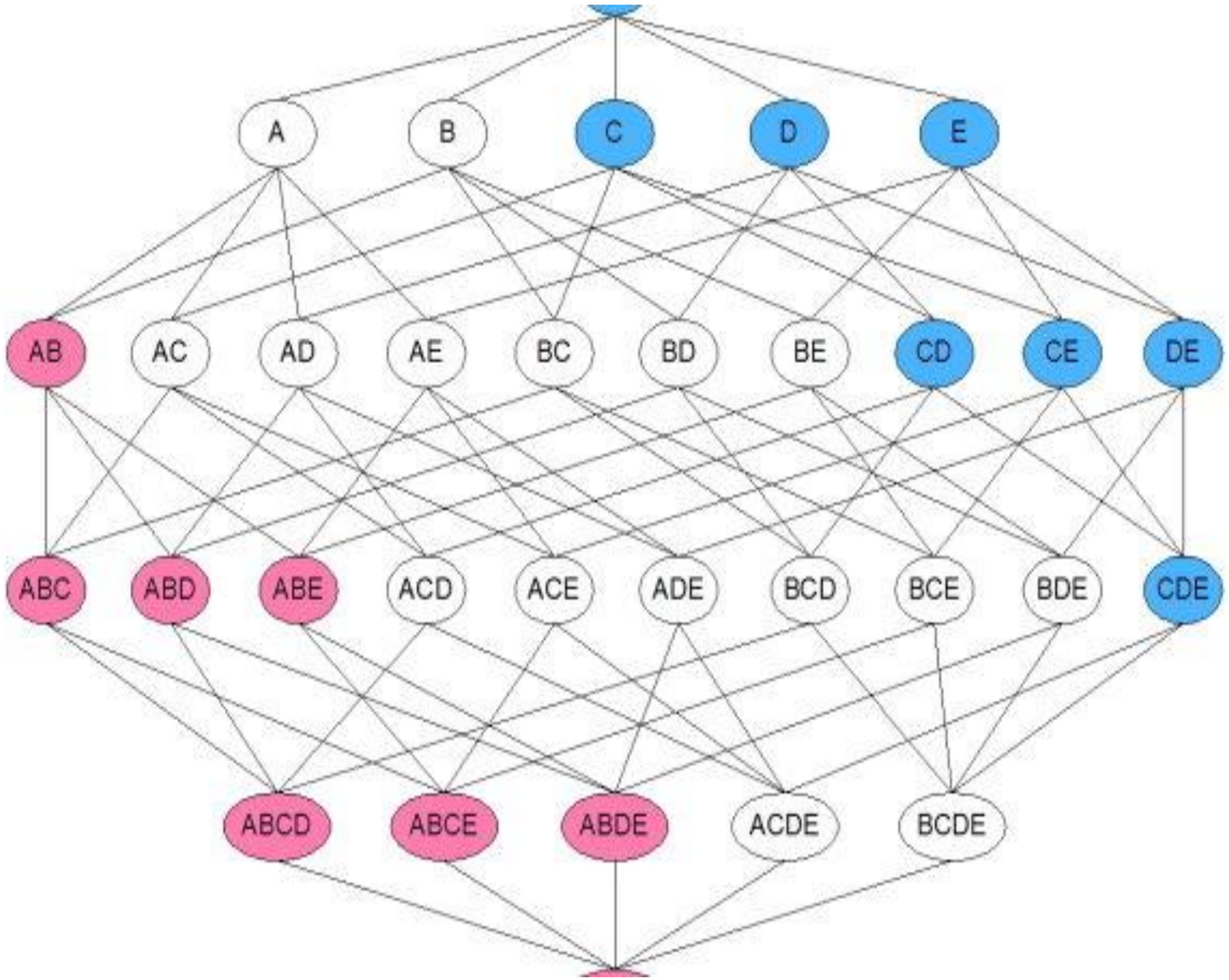
Cada conjunto de elementos en la red es un conjunto de elementos frecuente candidato.

- Calcular el soporte de cada candidato escaneando la base de dato
- Empareja cada transacción con cada candidato

Estrategias de generación de elementos frecuentes

- Reducir el número de candidatos (M)
- Reducir el número de transacciones (N)
- Reducir el número de comparaciones (NM)





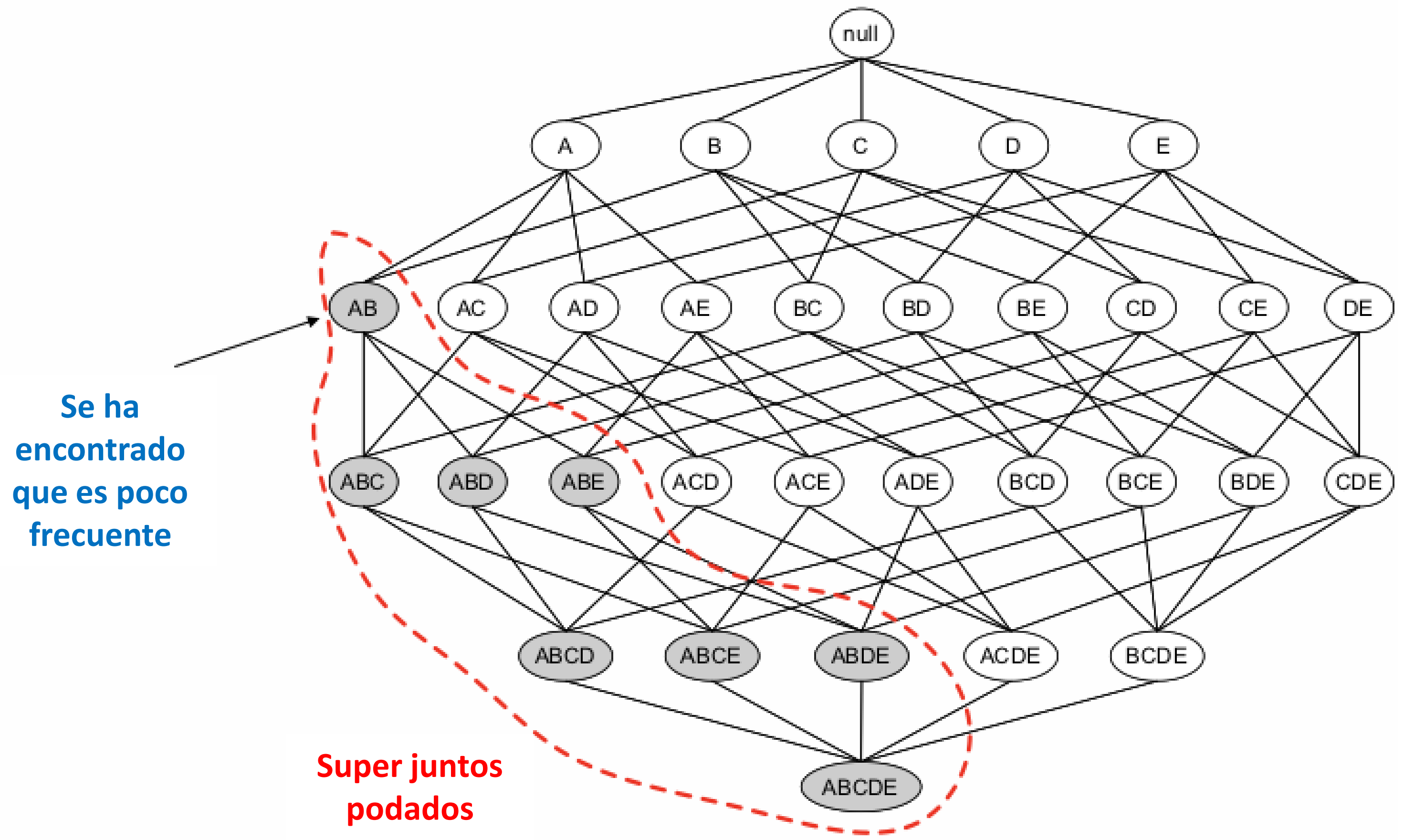
Reglas de la Asociación: Principio “apriori”

Reduciendo el número de candidatos

Principio de Apriori: si un conjunto de elementos es frecuente, entonces todos sus subconjuntos también deben ser frecuentes. El principio de Apriori se mantiene debido a la siguiente propiedad de la medida de soporte:

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

El soporte de un conjunto de elementos nunca excede el soporte de sus subconjuntos. Esto se conoce como la propiedad anti-monótona de soporte



Algoritmo Apriori

El núcleo del algoritmo de Apriori.

- Utilizar los conjuntos frecuentes (k-1) para generar candidatos a k-items frecuentes.
- Utilizar el escaneo de la base de datos y la coincidencia de patrones para recoger los recuentos de los conjuntos de elementos candidatos.

Comprimir una gran base de datos en una estructura compacta de árbol de patrones frecuentes (FP-tree). Muy condensado, pero completo para la minería de patrones frecuentes. Evita costosos análisis de bases de datos.

Utilice un método de minería de patrones frecuentes, basado en el árbol de FP. Una metodología de dividir y conquistar: descomponer las tareas de minería en los más pequeños.

Algoritmo Apriori

APRIORI (**D**, $\mathcal{I}$, *minsup*):

$\mathcal{F} \leftarrow \emptyset$
$\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$ `// Initial prefix tree with single items`
foreach $i \in \mathcal{I}$ **do** Add i as child of $\emptyset$ in $\mathcal{C}^{(1)}$ with $sup(i) \leftarrow 0$
$k \leftarrow 1$ `// k denotes the level`
while $\mathcal{C}^{(k)} \neq \emptyset$ **do**
    COMPUTESUPPORT ($\mathcal{C}^{(k)}$, **D**)
    foreach *leaf* $X \in \mathcal{C}^{(k)}$ **do**
        if $sup(X) \geq minsup$ **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$
        else remove X from $\mathcal{C}^{(k)}$
    $\mathcal{C}^{(k+1)} \leftarrow$ EXTENDPREFIXTREE ($\mathcal{C}^{(k)}$)
    $k \leftarrow k + 1$
return $\mathcal{F}^{(k)}$

COMPUTESUPPORT ($\mathcal{C}^{(k)}$, **D**):

foreach $\langle t, \mathbf{i}(t) \rangle \in$ **D do**
    foreach *k-subset* $X \subseteq \mathbf{i}(t)$ **do**
        if $X \in \mathcal{C}^{(k)}$ **then** $sup(X) \leftarrow sup(X) + 1$

EXTENDPREFIXTREE ($\mathcal{C}^{(k)}$):

foreach *leaf* $X_a \in \mathcal{C}^{(k)}$ **do**
    foreach *leaf* $X_b \in$ SIBLING(X_a), *such that* $b > a$ **do**
        $X_{ab} \leftarrow X_a \cup X_b$
        `// prune candidate if there are any infrequent subsets`
        if $X_j \in \mathcal{C}^{(k)}$, ***for all*** $X_j \subset X_{ab}$, *such that* $|X_j| = |X_{ab}| - 1$ **then**
            Add X_{ab} as child of X_a with $sup(X_{ab}) \leftarrow 0$
    if *no extensions from* X_a **then**
        remove X_a, and all ancestors of X_a with no extensions, from $\mathcal{C}^{(k)}$
return $\mathcal{C}^{(k)}$

Ejemplo

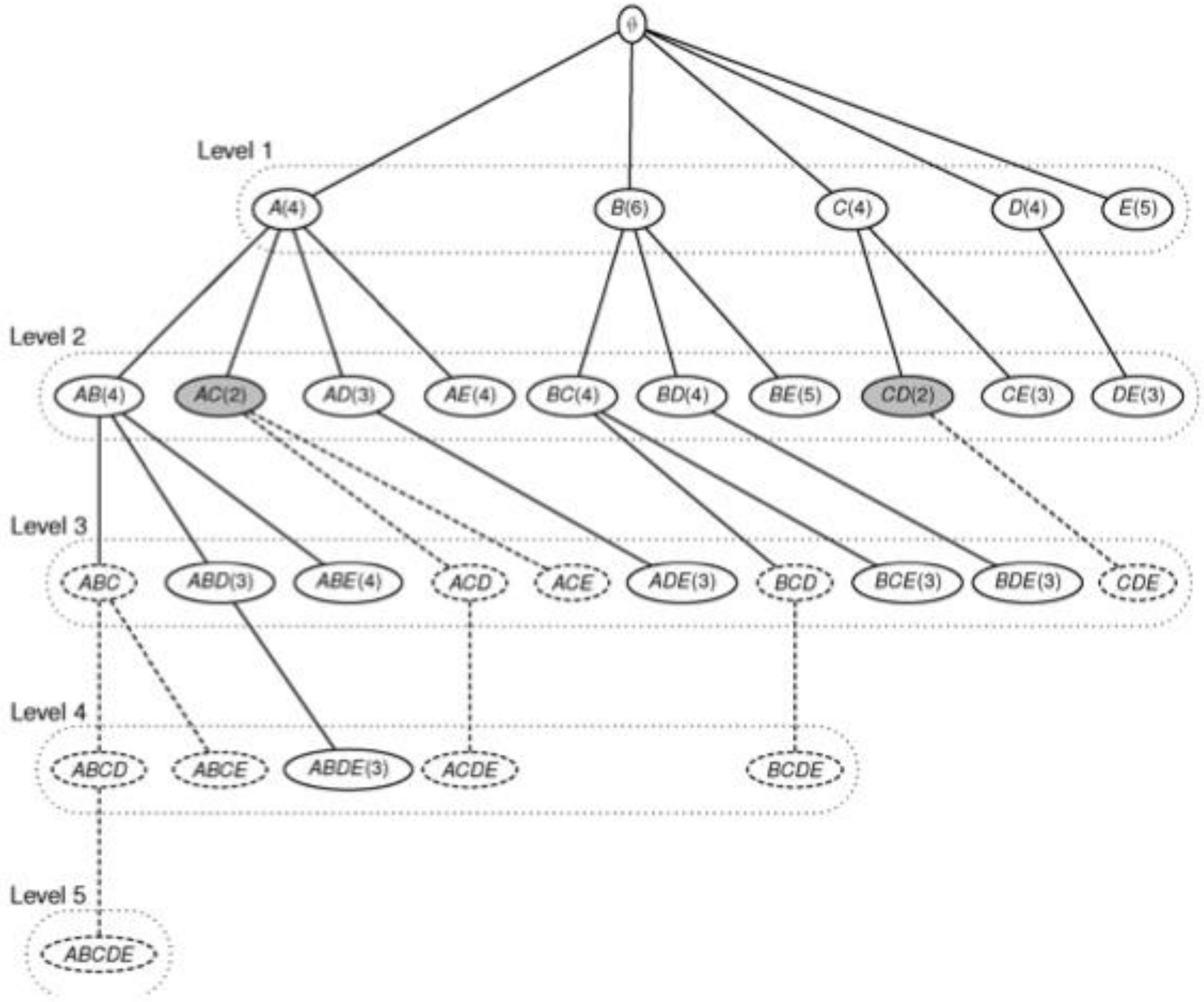
Dada la siguiente base de datos y un soporte mínimo de 3, genere todos los conjuntos de elementos frecuentes.

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

Base de Datos Binaria

t	$\mathbf{i}(t)$
1	*ABDE*
2	*BCE*
3	*ABDE*
4	*ABCE*
5	*ABCDE*
6	*BCD*

Base de Datos Transaccionaría



Ejercicio de practica

Se procede a identificar los *itemsets* frecuentes y, a partir de ellos, crear reglas de asociación.

Transacción
{A, B, C, D}
{A, B, D}
{A, B}
{B, C, D}
{B, C}
{C, D}
{B, D}

Para este problema se considera que un *item* o *itemset* es frecuente si aparece en un mínimo de 3 transacciones, es decir, su soporte debe de ser igual o superior a *3/7 = 0.43*. Se inicia el algoritmo identificando todos los *items* individuales (*itemsets* de un único *item*) y calculando su soporte.

Itemset (k=1)	Ocurrencias	Soporte
{A}	3	0.43
{B}	6	0.86
{C}	4	0.57
{D}	5	0.71

A continuación, se generan todos los posibles *itemsets* de tamaño *k = 2* que se pueden crear con los *itemsets* que han superado el paso anterior y se calcula su soporte.

Item k=2

Los *itemsets* {A, B}, {B, C}, {B, D} y {C, D} superan el límite (>=0.43) de soporte, por lo que son frecuentes. Los *itemsets* {A, C} y {A, D} no superan el soporte mínimo (<=0.43) por lo que se descartan.

Itemset (k=2)	Ocurrencias	Soporte
✔ {A, B}	3	0.43
{A, C}	1	0.14
{A, D}	2	0.29
✔ {B, C}	3	0.43
✔ {B, D}	4	0.57
✔ {C, D}	3	0.43

Item k=2

Se repite el proceso, esta vez creando *itemsets* de tamaño *k = 3*.

Itemset (k=2)	Ocurrencias	Soporte
{A, B}	3	0.43
{B, C}	3	0.43
{B, D}	4	0.57
{C, D}	3	0.43

Item k=3

Los *itemsets* {A, B, C}, {A, B, D} y {C, D, A} contienen subconjuntos infrecuentes, por lo que son descartados. Para los restantes se calcula su soporte.

Itemset (k=3)	Ocurrencias	Soporte
{A, B, C}	0	0
{A, B, D}	1	0.14
{B, C, D}	2	0.29
{C, D, A}	0	0

El *items* {B, C, D} no supera el soporte mínimo por lo que se considera infrecuente. Al no haber ningún nuevo *itemset* frecuente, se detiene el algoritmo.

Como resultado de la búsqueda se han identificado los siguientes *itemsets* frecuentes:

Itemset frecuentes
{A, B}
{B, C}
{B, D}
{C, D}

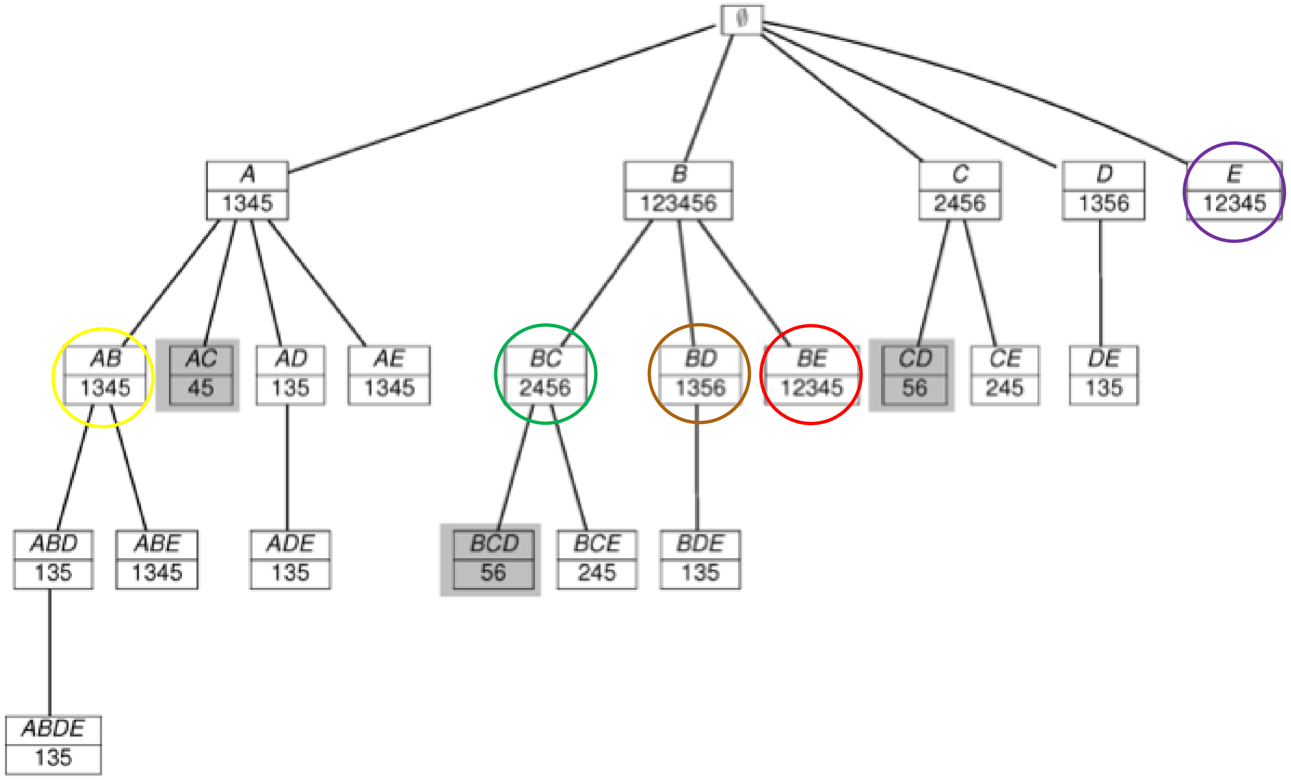
Supóngase que se desean únicamente reglas con una confianza igual o superior a 0.7, es decir, que la regla se cumpla un 70% de las veces.

De todas las posibles reglas, únicamente:

- {C} => {D}
- {C} => {B}
- {A} => {B}

superan el límite de confianza.

Reglas	**Confianza**	**Confianza**
{A} => {B}	soporte{A, B} / soporte {A}	0.43 / 0.43 = 1
{B} => {A}	soporte{A, B} / soporte {B}	0.43 / 0.86 = 0.5
{B} => {C}	soporte{B, C} / soporte {B}	0.43 / 0.86 = 0.5
{C} => {B}	soporte{B, C} / soporte {C}	0.43 / 0.57 = 0.75
{B} => {D}	soporte{B, D} / soporte {B}	0.43 / 0.86 = 0.5
{D} => {B}	soporte{B, D} / soporte {D}	0.43 / 0.71 = 0.6
{C} => {D}	soporte{C, D} / soporte {C}	0.43 / 0.57 = 0.75
{D} => {C}	soporte{C, D} / soporte {D}	0.43 / 0.71 = 0.6



Eclat (Ejemplo)

Dado t(X) y t(Y) para dos conjuntos de elementos frecuentes X e Y,
entonces:

t(XY) = t(X)^t(Y).
sup (XY) = | t(XY) |

D	*A*	*B*	*C*	*D*	*E*
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

Binary Database

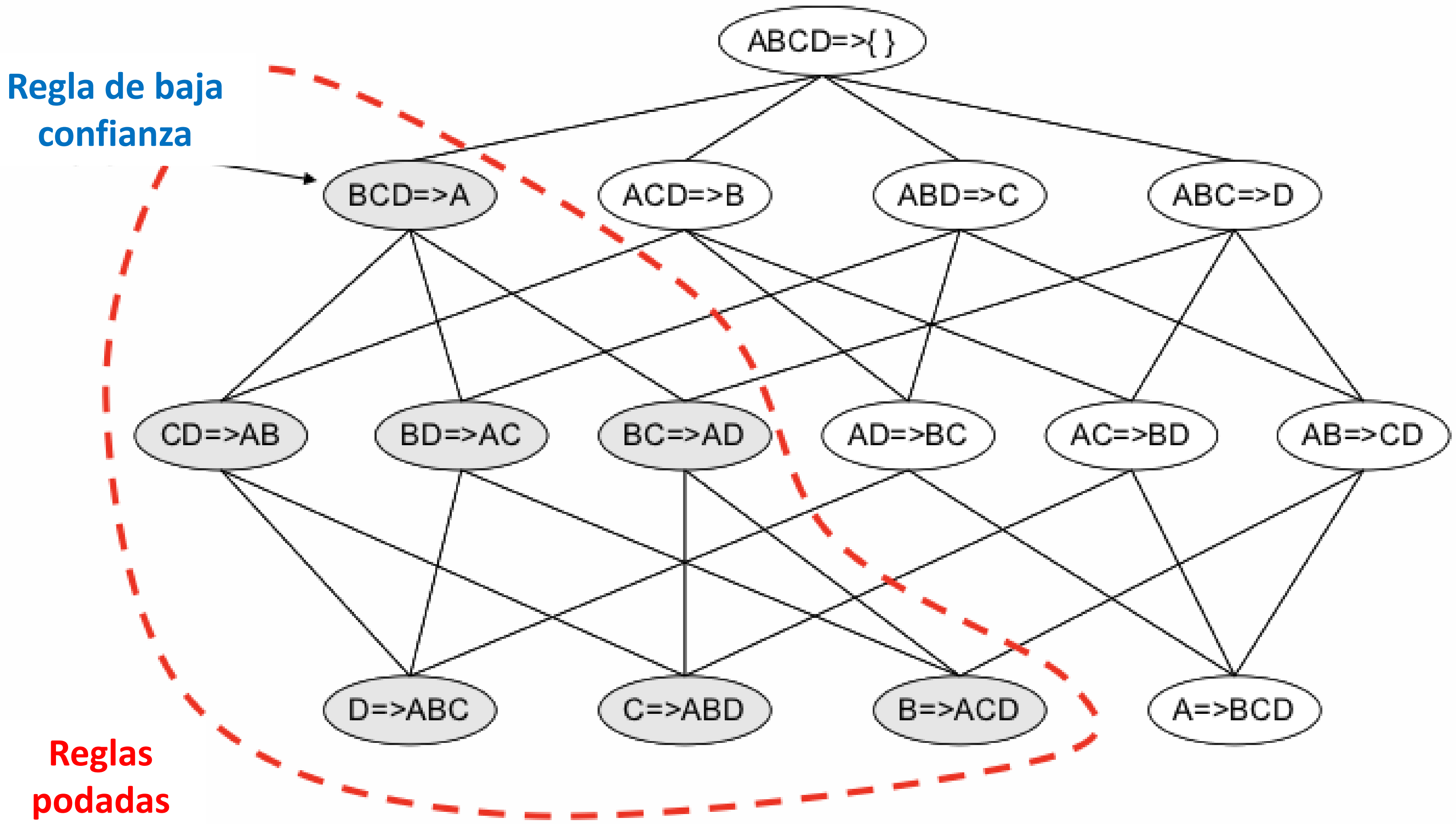
t	**i**(*t*)
1	*ABDE*
2	*BCE*
3	*ABDE*
4	*ABCE*
5	*ABCDE*
6	*BCD*

Transaction Database

t(*x*)				
A	*B*	*C*	*D*	*E*
1	1	2	1	1
3	2	4	3	2
4	3	5	5	3
5	4	6	6	4
	5			5
	6			

Vertial Database

¿Cómo generar reglas de manera eficiente a partir de elementos frecuentes?



Generar Reglas

La confianza no tiene una propiedad anti-monótona.

c (ABC → D) puede ser mayor o menor que c (AB → D).

Pero la confianza en las reglas generadas desde el mismo conjunto de elementos tiene una propiedad anti monotónica.

L= {A, B, C, D}: c (BCD → A)> = c (BC→ AD)> = c (B → ACD)
= c (BD→ AC)> = c (D → ABC)
= c (CD→ AB)> = c (C → ABD)

L = {A, B, C, D}: c (ACD → B)> = c (AC → BD)> = c (A → BCD)
= c (AD → BC)> = c (D → ABC)
= c (CD → AB)> = c (C → ABD)

L= {A, B, C, D}: c (ABD → C)> = c (AD → BC)> = c (A → BCD)
= c (AB → CD)> = c (B → ACD)
= c (BD → AC)> = c (D → ABC)

L = {A, B, C, D}: c (ABC → D)> = c (AB → CD)> = c (A → BCD)
= c (AC → BD)> = c (C → ABD)
= c (BC → AD)> = c (B → ACD)

La confianza es anti monótona con respecto al número de artículos en el lado derecho de la regla.

Bibliografía

- JayWrkr (Mayo 5,2019).*Association Rules.* [en línea]. México. Disponible en: https://medium.com/@jaywrkr/miner%C3%ADa-de-datos-3-f75d15f90c46 [2020, 12 septiembre]
- Ansel Yoan Rodríguez González, José Francisco Martínez Trinidad, Jesús Ariel Carrasco Ochoa, José Ruiz Shulcloper (Marzo 31,2009).*Minería de Reglas de Asociación sobre Datos Mezclados.* [en línea]. Puebla, México. Disponible en: http://ccc.inaoep.mx/portalfiles/file/CCC-09-001.pdf [2020, 12 septiembre]
- Amat Rodrigo Joaquín (Junio, 2018). *Reglas de asociación y algoritmo a priori con R.* [en línea]. México. Disponible en: https://www.cienciadedatos.net/documentos/43_reglas_de_asociacion [2020, 14 septiembre]