Minería de patrones frecuentes y reglas de asociación

Máster Online en Ciencia de Datos

UCO ONLINE

Dr. José Raúl Romero

Profesor Titular de la Universidad de Córdoba y Doctor en Ingeniería Informática por la Universidad de Málaga. Sus líneas actuales de trabajo se centran en la democratización de la ciencia de datos (*Automated ML* y *Explainable Artificial Intelligence*), aprendizaje automático evolutivo y analítica de software (aplicación de aprendizaje y optimización a la mejora del proceso de desarrollo de software).

Miembro del Consejo de Administración de la *European Association for Data Science*, e investigador senior del Instituto de Investigación Andaluz de *Data Science and Computational Intelligence*.

Director del **Máster Online en Ciencia de Datos** de la Universidad de Córdoba.

Algoritmo FP-Growth

Algoritmo

UCO ONLINE

- Surge como respuesta a propuestas de generación-y-prueba de itemsets candidatos (p.ej. Apriori)

  La generación de candidatos es costosa cuando hay patrones largos o un gran número de ellos

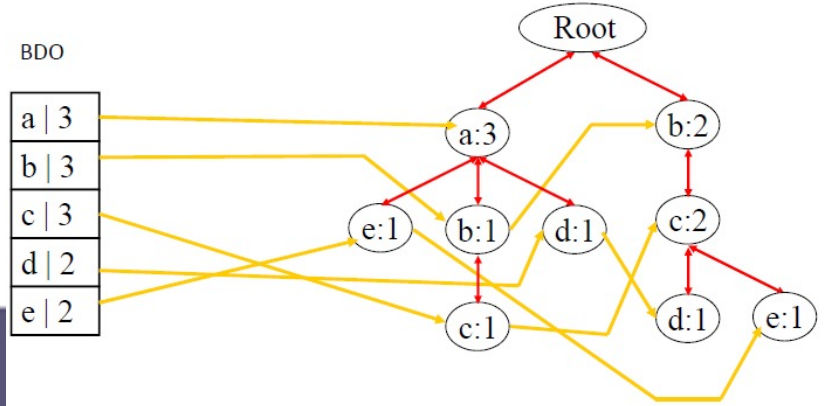
- El modelo FP-Growth consigue la eficiencia de la extracción con:
  1. La **base de datos** se comprime en un **árbol FP** (*frequent pattern tree*) evitando escaneos repetitivos y costosos
  2. Método basado en el crecimiento parcial de fragmentos de los patrones evitando generar un gran número de itemsets candidatos
  3. Método de partición, basado en "divide y vencerás" que reduce drásticamente la **base de patrones** condicionales generados para el siguiente nivel de búsqueda

Algunos de los conceptos que hemos escuchado:

- El conjunto o **base de datos iniciales** (BD): donde se van a recopilar todos los datos de frecuencia, patrones, …
- **Árbol de patrones frecuentes** (*FP-Tree*): es la base de todo el algoritmo, ya que mediante su recorrido se determinan los patrones
- **Lista ordenada de patrones** (BDO): donde se almacenan los patrones obtenidos y se ordenan descendentemente en función de su frecuencia con apuntadores al FP-Tree

FP-Tree

- Estructura básica de FP-Growth, para facilitar el recorrido del árbol, contiene dos tipos de enlaces:
  - **Enlaces padre-hijos**, como cualquier estructura de árbol
  - Aquellos que **enlazan nodos de igual tipo**, cuya cabecera va a venir dada por la lista BDO
- Cada nodo contiene información referente a su frecuencia en la BD de partida, teniendo en cuenta la rama a la que pertenece



Algoritmo – Pasos

1. Construir **igual que en Apriori** el 1-itemset teniendo en cuenta la frecuencia de cada *item*
2. Reordenar la **BD** de forma descendente (**BDO**)
3. Construir el ***FP-Tree***:
   a. Crear un ***nodo root*** (raíz)
   b. Añadir caminos desde el raíz que describan cada entrada del conjunto de datos contenida en la **BD** de partida – se recorren una a una las transacciones
4. Recorrer las listas **BDO** de los *items*
   a. Analizar posibles **patrones base** que llevan a ese nodo
   b. Anotar dichos patrones base
5. Construir **patrones frecuentes** a partir de los patrones base

Algoritmo – Propiedad FP-Growth

Sea α un itemset frecuente en la base de datos, sea **B** patrón base condicional de α, y β un itemset en **B**. Entonces $\alpha \cup \beta$ es un itemset frecuente en la base de datos sii β es también frecuente en B.

"abcdef " es un patrón frecuente si y sólo si

"abcde " es un patrón frecuente y

"f " es frecuente en el conjunto de transacciones que contienen "abcde"

Algoritmo FP-Growth

Ejemplo

UCO ONLINE

TID	Items de compras
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}

Frecuencia mínima = 3

Construcción de 1-itemset y lista BDO

TID	Items frecuentes
1	{F, C, A, M, P}
2	{F, C, A, B, M}
3	{F, B}
4	{C, B, P}
5	{F, C, A, M, P}

[PASO 1] Se construye el 1-itemset ordenado

Item	Frecuencia
F	4
C	4
A	3
B	3
M	3
P	3

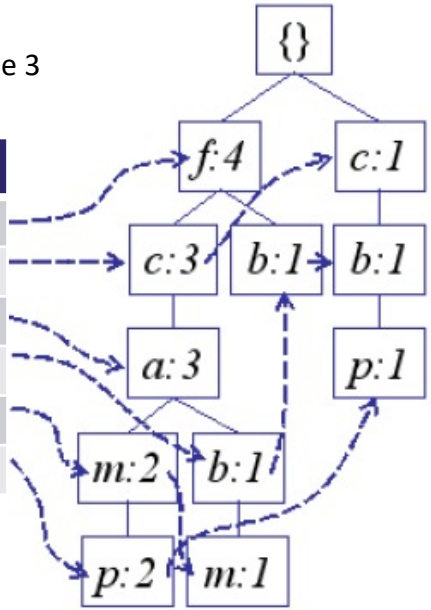
[PASO 2] Se construye la **lista BDO**

Construcción del FP-Tree

[Paso 3] Se construye el FP-Tree

No se consideran items de frecuencia menor de 3

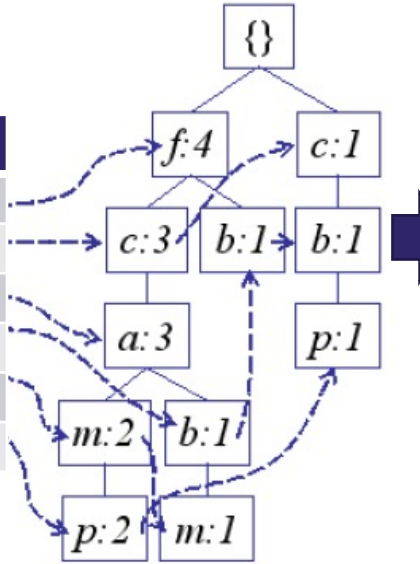
Item	Frecuencia
F	4
C	4
A	3
B	3
M	3
P	3



Generación de patrones base

[Paso 4] Recorrer BDO para obtenerlos patrones base

Item	Frecuencia
F	4
C	4
A	3
B	3
M	3
P	3



Item	Patrón base condicional
C	F:3
A	FC:3
B	FCA:1, F:1, C:1
M	FCA:2, FCAB:1
P	FCAM:2, CB:1

Generación de patrones frecuentes

[Paso 5] Construir patrones frecuentes a partir de patrones base

Para ello, se obtiene el FP-Tree condicional

Va a depender de **min_supp** (=3 en ejemplo)

Item	Patrón base
C	F:3
A	FC:3
B	FCA:1, F:1, C:1
M	FCA:2, FCAB:1
P	FCAM:2, CB:1

Empezamos con el último item de la lista (P)

¿Por qué?

P sucede en dos ramas del árbol

Las ramas formadas son:

F C A M P :2

C B P :1

Considerando P como sufijo, teníamos los patrones base:

F C A M : 2

C B : 1

El FP-Tree condicional para P {(C:3)}|P

Patrones frecuentes que implican P: {CP:3}

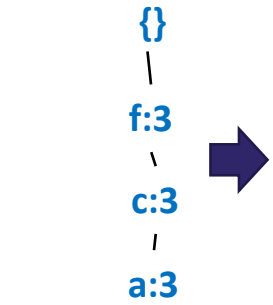
Algoritmo FP-Growth: Ejemplo

[Paso 5] Para cada patrón base, consultamos el FP-Tree para extraer sus items frecuentes

Item	Patrón base
C	F:3
A	FC:3
B	FCA:1, F:1, C:1
M	FCA:2, FCAB:1
P	FCAM:2, CB:1

Seguimos con M

Tenemos 2 patrones base: FCA:2 y FCAB:1



FP-tree condicional de M

Según la propiedad FP-Growth, los ***patrones frec.*** *serían:*

M,

FM, CM, AM,

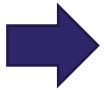
FCM, FAM, CAM,

FCAM

Algoritmo FP-Growth: Ejemplo

[Paso 5] Para cada patrón base, consultamos el FP-Tree para extraer sus items frecuentes

Item	Patrón base
C	F:3
A	FC:3
B	FCA:1, F:1, C:1
M	FCA:2, FCAB:1
P	FCAM:2, CB:1



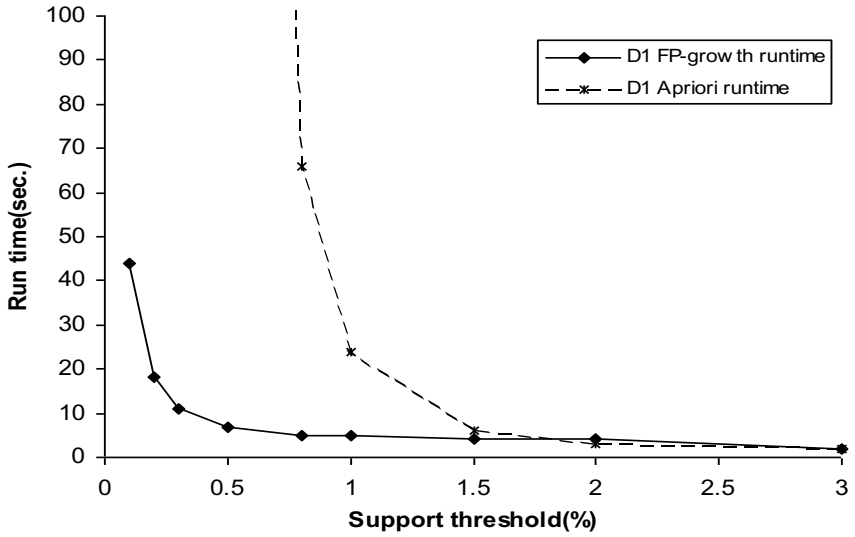
Item	FP-Tree condicional
P	{(c:3)}|p
M	{(f:3, c:3, a:3)}|m
B	*-vacío-*
A	{(f:3, c:3)}|a
C	{(f:3)}|c
F	*-vacío-*

Beneficios de la estructura FP-Tree

Los estudios de rendimiento demuestran que FP-Growth es **un orden de magnitud más rápido que Apriori**

Causas:

- No hay generación de candidatos
- Uso de una estructura de datos compacta
- Elimina escaneos repetitivos en la base de datos
- Las operaciones básicas son el conteo y la construcción del FP-Tree



Algoritmo FP-Growth

Uso de SPMF para la ejecución de FP-Growth

UCO ONLINE

¡Gracias!

UCO ONLINE