Practica 5: Algoritmo A priori y FP Growth

Profesor: Lauro Reyes Cocoletzi

Edkir Nava y Diego Castro
{enavam2001, dcastroe2100} @alumno.ipnx.mx

UPIIT: Unidad Profesional Interdisciplinaria en Ingeniería Campus Tlaxcala Instituto Politécnico Nacional, Tlaxcala, Tlaxcala, México 9000

Ingeniera en Inteligencia Artificial

6 de octubre 2023

***Resumen*— El algoritmo A priori y el algoritmo FP Growth son dos técnicas populares en la minería de datos utilizadas para descubrir patrones de asociación en conjuntos de datos grandes. El algoritmo A priori se basa en el principio de que los elementos que aparecen juntos con frecuencia en un conjunto de datos tienden a estar asociados. Utiliza un enfoque de "frecuencia de soporte" para identificar conjuntos de elementos que cumplen con un umbral de apoyo especificado. Por otro lado, el algoritmo FP Growth (Frequent Pattern Growth) se centra en la construcción de una estructura de datos llamada "árbol FP" para reducir la complejidad del proceso. Este enfoque mejora la eficiencia en la búsqueda de patrones frecuentes.**

Palabras clave* — *Reglas de Asociación, A priori, FP Growth, soporte, confianza,

I. Marco Teorico

A. Reglas de Asociación

Las reglas de asociación son declaraciones *if-then* que ayudan a descubrir hechos que ocurren en común en un determinado conjunto de datos. Una de sus aplicaciones mas comunes es determinar relaciones entre los productos vendidos por una tienda. Las reglas no extraen la preferencia de un individuo, sino que encuentran relaciones entre un conjunto de elementos de cada transacción distinta.

Una regla de asociación es una implicación de la forma $A \rightarrow B$, usando el ejemplo de la tienda, esta regla de asociación significaría que cuando se compra el conjunto de productos *A* entonces es probable que también se compre el conjunto *B*.

Formalmente podemos expresar lo anterior como:

- Sea $I = \{i_1, i_2, i_3, \dots, i_n\}$ un conjunto de *n* ítems
- Sea $D = \{t_1, t_2, t_3, \dots, t_m\}$ un conjunto de *m* transacciones

Cada transacción en *D* tiene un identificador único que contiene un subconjunto de ítems pertenecientes a *I*.
Una regla se puede definir como:

$$X \rightarrow Y$$

Donde *X* y *Y* son un subconjunto de *I* pero no tienen elementos en común:

$$X, Y \subseteq I$$
$$X \cap Y = 0$$

Ejemplificando lo anterior:

$$I = \{leche, pan, mantequilla, cerveza\}$$
$$X = \{leche, pan\}$$
$$Y = \{mantequilla\}$$

$$\{leche, pan\} \rightarrow \{mantequilla\}$$

Las transacciones suelen tener la siguiente estructura:

Transacción 1	Leche	Pan	
Transacción 2	Pan	Mantequilla	
Transacción 3	cerveza		
Transacción 4	Leche	Pan	Mantequilla
Transacción 5	Pan		

B. Algoritmo A Priori

El algoritmo Apriori es un algoritmo desarrollado para la búsqueda de reglas de asociación. Obtiene los llamados conjuntos de ítems frecuentes, los cuales son aquellos conjuntos formados por los ítems cuyo soporte obtenido de la base de datos es superior al soporte mínimo solicitado por el usuario.

Podemos resumir el algoritmo a priori en 4 pasos:

1. Establecer un valor mínimo para el soporte y la confianza. Esto significa que solo nos interesa encontrar reglas para los elementos que tienen cierta existencia por defecto, por ejemplo, el apoyo, y tienen

un valor mínimo de concurrencia con otros elementos, por ejemplo, la confianza.
2. Extraer todos los subconjuntos que tengan un valor de soporte superior a un umbral mínimo.
3. Seleccionar todas las reglas de los subconjuntos con un valor de confianza superior al umbral mínimo.
4. Ordenar las reglas por orden descendente de lift.

VENTAJAS	DESVENTAJAS
Fácil de implementar y entender.	Es necesario encontrar muchas reglas candidatas.
Útil en conjuntos de elementos grandes.	Tiene un costo computacional alto.

C. FP GROWTH

Este algoritmo es una mejora del método A priori; se genera un patrón frecuente sin necesidad de generación de candidatos. El algoritmo de crecimiento FP representa la base de datos en forma de un árbol llamado árbol de patrones frecuentes o árbol FP, esta estructura de árbol mantendrá la asociación entre los conjuntos de elementos y su propósito es extraer el patrón más frecuente.

Pasos para la construcción del algoritmo FP-Growth:

1. Construcción de la estructura de datos FP-Tree:
   - Con el conjunto de datos transaccionales obtenido se recorre para contar la frecuencia de cada elemento individual y construir una tabla de frecuencia de elementos.
   - Posteriormente se hace un filtrado de elementos infrecuentes, los elementos que no cumplen con un umbral de soporte mínimo (umbral de frecuencia) se eliminan del conjunto de datos, ya que no se considerarán para la extracción de patrones frecuentes.
   - Por ultimo los elementos se ordenan en orden descendente de acuerdo con su frecuencia en el conjunto de datos.
2. Construcción del FP-Tree:
   - El segundo paso es construir el árbol FP. Para ello, cree la raíz del árbol. La raíz está representada por nulo.
   - Para cada transacción en el conjunto de datos, se recorren los elementos de la transacción y se insertan en el árbol de manera que se mantenga el orden de frecuencia.
   - Se actualiza la tabla de frecuencia de elementos para reflejar la estructura del árbol.
3. Generación de patrones frecuentes:
   - Se inicia con el elemento menos frecuente del conjunto de datos y se crea un camino condicional desde la raíz del árbol hasta cada hoja que contenga ese elemento.
   - Para cada camino condicional, se extraen los patrones frecuentes al combinar el elemento actual con sus antecesores en el camino.
   - Los patrones frecuentes se acumulan en una lista de patrones frecuentes.
4. Recursión en el árbol FP:
   - Se repite el proceso de generación de patrones frecuentes para cada elemento en orden descendente de frecuencia.
   - Cada iteración implica la construcción de un árbol condicional basado en el elemento actual y la generación de patrones frecuentes en ese subconjunto de datos.
5. Combinación de patrones frecuentes:
   - Los patrones frecuentes generados en cada iteración se combinan para formar un conjunto completo de patrones frecuentes en el conjunto de datos original.

D. CONJUNTO DE DATOS 1

Transacción 1	A	B	C	D
Transacción 2	A	B	D	
Transacción 3	A	B		
Transacción 4	B	C	D	
Transacción 5	B	C		
Transacción 6	C	D		
Transacción 7	B	D		

E. CONJUNTO DE DATOS 2

Transacción 1	I1	I2	I3	
Transacción 2	I2	I3	I4	
Transacción 3	I4	I5		
Transacción 4	I1	I2	I4	
Transacción 5	I1	I2	I3	I5
Transacción 6	I1	I2	I3	I4

F. CONJUNTO DE DATOS "GROCERY"

MILK	BREAD	BISCUIT	
BREAD	MILK	BISCUIT	CORNFLAKES
BREAD	TEA	BOURNVITA	
JAM	JUICE	BREAD	MILK
JUICE	TEA	BISCUIT	

BREAD	TEA	BOURNVITA	COOKIES
JUICE	TEA	CORNFLAKES	
JUICE	BREAD	TEA	BISCUIT
JAM	JUICE	BREAD	TEA
BREAD	MILK		
COFFEE	COOKIES	BISCUIT	CORNFLAKES
COFFEE	COOKIES	BISCUIT	CORNFLAKES
COFFEE	SPLENDA	BOURNVITA	
BREAD	COFFEE	COOKIES	
BREAD	SUGAR	BISCUIT	
COFFEE	STEVIA	CORNFLAKES	
BREAD	SUGAR	BOURNVITA	
BREAD	COFFEE	SPLENDA	
BREAD	COFFEE	STEVIA	
TEA	MILK	COFFEE	CORNFLAKES

II. DESARROLLO

A. A PRIORI

En primer lugar, saca los conjuntos de ítems frecuentes de tamaño 1 y, luego, los de tamaño 2 y así sucesivamente hasta que no se encuentren más conjuntos cuyos ítems no tengan el soporte mayor al soporte mínimo. Después convierte los *ítems* frecuentes en reglas de asociación.

Suponiendo que tenemos un conjunto de transacciones que pueden estar formados por 5 diferentes *items*.

Los conjuntos de un solo ítem son los obtenidos de una pasada en la base de datos y son los ítems cuyo soporte calculado en esa pasada es superior al soporte mínimo propuesto por el usuario.

Los conjuntos de dos o más *ítems* se generan haciendo las demás posibles combinaciones entre todos los *ítems.*

CONJUNTOS	POSIBLES TRANSACCIONES
1 *item*	{a}, {b}, {c}, {d}, {e}
2 *item*	{a,b}, {a,c}, {a,d}, {a,e}, {b,c}, {b,d}, {b,e}, {c,d}, {c,e}, {d,e}
3 *item*	{a,b,c}, {a,b,d}, {a,b,e}, {a,c,d}, {a,c,e}, {a,d,e}, {b,c,d}, {b,c,e}, {b,d,e}, {c,d,e}
4 *item*	{a,b,c,d}, {a,c,b,e}, {a,b,d,e}, {a,c,d,e}, {b,c,d,e}
5 *item*	{a,b,c,d,e}

Considerando por ejemplo los conjuntos de 3 items de la Tabla 2, se podrían generar las siguientes reglas de asociación:

$$\{a, b\} \rightarrow \{c\}$$
$$\{a, c\} \rightarrow \{b\}$$
$$\{b, c\} \rightarrow \{a\}$$

Además de las reglas de Asociación, el algoritmo se sirve de otras herramientas matemáticas para hacer un análisis correcto.

SOPORTE

El soporte mide la proporción de transacciones en las que se cumple la regla. Un alto soporte indica que la regla se cumple en una gran cantidad de transacciones.

El soporte de un item es la frecuencia con la cual este ítem se encuentra en las transacciones dividido entre el número de transacciones.

$$Soporte(X) = \frac{No.transacciones\ que\ contienen\ el\ item\ X}{No.transacciones\ de\ la\ BD}$$

Para obtener el soporte de una regla de decisión, por ejemplo $X \rightarrow Y$ se obtiene con la siguiente ecuación:

$$Soporte(X \rightarrow Y) = \frac{No.transacciones\ que\ contienen\ \ X\ y\ Y}{No.transacciones\ de\ la\ BD}$$

CONFIANZA

Mide la probabilidad de que el conjunto de elementos en el lado derecho de la regla se compre dado que el conjunto de elementos en el lado izquierdo de la regla ya se ha comprado. Una confianza alta indica una relación fuerte entre los elementos.
Se calcula de la siguiente manera:

$$confianza(X \rightarrow Y) = \frac{soporte(X \cup Y)}{soporte(X)}$$

LIFT

Indica como de probable el item *Y* sea comprando cuando el item *X* es comprado, controlando como de popular es el item *Y*.
El lift mide la importancia de la relación entre los elementos en la regla. Un lift mayor que 1 indica una relación positiva, donde la ocurrencia de un elemento aumenta la probabilidad de que ocurra el otro. Un lift igual a 1 indica que los elementos son independientes.

$$sustento(X \rightarrow Y) = \frac{soporte(X \cup Y)}{soporte(X) * soporte(Y)}$$

B. FP GROWTH

Conjunto de datos 1

Obtenido el conjunto de transacciones, el primer paso es calcular la frecuencia

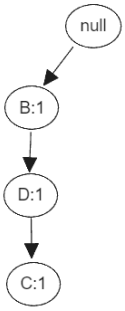
A	3
B	6
C	4
D	5

Ahora definimos el mínimo conteo de soporte, donde funciona como un umbral, en el ejemplo lo seleccionaremos con 4, por lo que debemos ordenar nuestra

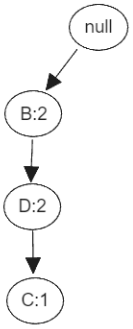
B	6
D	5
C	4

Construimos el árbol FP

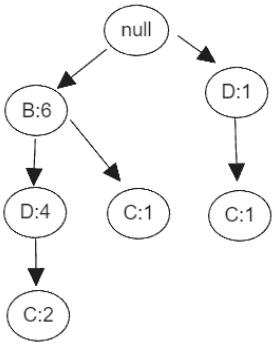
1. Considerando nulo el nodo raíz.
2. En el primer escaneo de la transacción, tenemos T1: B, C, D, los cuales tiene tres elementos {B:6}, {D: 5}, {C: 4}, donde por el orden tendremos que B sera la raíz por ser el de mayor frecuencia, D estará ligado a B y C estará ligado a D, obteniendo el siguiente árbol:



3. Se realiza en la transacción 2 tenemos T2: B, D donde anteriormente ya teníamos a B y D ligados por lo que solamente sumaremos a ese nodo:



4. El proceso anterior lo realizaremos por todas las transacciones obteniendo:



Con ese resultado, juntamos todos los conjuntos que superen la frecuencia, es decir: {B:6}, {D:5}, {C:4}, {B:6,D:4}

Conjunto de datos 2

Se realiza el mismo procedimiento, pero con el conjunto de datos 2, donde se sigue el mismo procedimiento, en los resultados se reportarán los diferentes resultados obtenidos usando distintos soportes.

C. CONJUNTO DE DATOS "GROCERY"

Utilizamos un conjunto de datos de transacciones de una tienda, con el fin de simular una de las principales utilidades que es recomendación de productos, en este caso no fue necesario hacer una limpieza de los datos por el tamaño de la data set donde al ser limitado, no tenemos datos redundantes.

En la implementación agregamos:

- Frecuencia mínima:
  Determina el umbral de frecuencia mínimo que un conjunto de elementos debe cumplir para ser considerado "frecuente". Cuando colocamos un valor como por ejemplo 0.5, solo se considerarán frecuentes los conjuntos de elementos que aparecen en al menos el 50% de las transacciones. Caso contrario, seleccionar uno muy bajo llevará a identificar más conjuntos de elementos como frecuentes, lo que resultará en más reglas de asociación, pero algunas de estas reglas pueden ser débiles y menos significativas

- Mínimo de confianza:
  Controla el umbral mínimo de confianza que una regla de asociación debe cumplir para ser considerada "fuerte". Cuando colocamos un valor alto resultará en un conjunto más pequeño de reglas, pero serán reglas muy sólidas y confiables, a diferencia de colocar un valor bajo llevará a identificar un conjunto más amplio de reglas, pero algunas de ellas pueden ser menos confiables

III. RESULTADOS

A. A PRIORI

Conjunto de datos 1

Al aplicar el algoritmo Apriori con un conteo minimo de 6 para el conjunto de datos 1, obtenemos los siguientes resultados:

```
Mínimo conteo de soporte: 6

Frecuencia 1-itemset ['B']
```

Esto quiere decir que el elemento *B* es el uncio con una frecuencia de 6.

En cambio, al utilizar conteo minimo de 4, podemos saber que los conjuntos {B}, {C}, {D} y {B, D} se repiten al menos 5 veces.

```
Mínimo conteo de soporte: 4

Frecuencia 1-itemset  ['B', 'C', 'D']
Frecuencia 2 -itemset  ['BD']
```

Continuamos aplciando el mismo algoritmo al mismo conjunto de datos pero para conteos se soporte minimos de 3, 2 y 1.

```
Mínimo conteo de soporte: 3

Frecuencia 1-itemset  ['A', 'B', 'C', 'D']
Frecuencia 2 -itemset  ['AB', 'BC', 'BD', 'CD']
Frecuencia 3 -itemset  []
```

```
Mínimo conteo de soporte: 2

Frecuencia 1-itemset  ['A', 'B', 'C', 'D']
Frecuencia 2 -itemset  ['AB', 'AD', 'BC', 'BD', 'CD']
Frecuencia 3 -itemset  ['ABD', 'BCD']
```

```
Mínimo conteo de soporte: 1

Frecuencia 1-itemset  ['A', 'B', 'C', 'D']
Frecuencia 2 -itemset  ['AB', 'AC', 'AD', 'BC', 'BD', 'CD']
Frecuencia 3 -itemset  ['ABC', 'ABD', 'ACD', 'BCD']
Frecuencia 4 -itemset  ['ABCD']
```

Conjunto de datos 2

DfgsdfgSi usamo el el conjunto de datos 2, al indicar conteos de soporte 5, 4 y 3.

```
Mínimo conteo de soporte: 5

Frecuencia 1-itemset  ['I2']
```

```
Mínimo conteo de soporte: 4

Frecuencia 1-itemset  ['I1', 'I2', 'I3', 'I4']
Frecuencia 2 -itemset  ['I12', 'I23']
```

```
Mínimo conteo de soporte: 3

Frecuencia 1-itemset  ['I1', 'I2', 'I3', 'I4']
Frecuencia 2 -itemset  ['I12', 'I13', 'I23', 'I24']
Frecuencia 3 -itemset  ['I123']
```

```
Mínimo conteo de soporte: 2

Frecuencia 1-itemset  ['I1', 'I2', 'I3', 'I4', 'I5']
Frecuencia 2 -itemset  ['I12', 'I13', 'I14', 'I23', 'I24', 'I34']
Frecuencia 3 -itemset  ['I123', 'I124', 'I234']
Frecuencia 4 -itemset  []
```

B. FP GROWTH

Conjunto de datos 1

A comparación del algoritmo a priori, podemos comparar los resultados, donde:

```
Mínimo conteo de soporte:
6
Todas las frecuencias de los itemsets:
[{'B'}]
```

Nos indica que con un mínimo de soporte de 6 solamente tenemos como resultado “B”, pero si número de soporte:

```
Mínimo conteo de soporte:
4
Todas las frecuencias de los itemsets:
[{'C'}, {'D'}, {'D', 'B'}, {'B'}]
```

Nos indica que al menos {'C'}, {'D'}, {'D', 'B'}, {'B'} tiene una al menos una frecuencia de aparición de 5. Con ello hacer todas las posibles combinaciones

```
Mínimo conteo de soporte:
3
Todas las frecuencias de los itemsets:
[{'A'}, {'B', 'A'}, {'C'}, {'D', 'C'}, {'C', 'B'}, {'D'}, {'D',
'B'}, {'B'}]
```

```
Mínimo conteo de soporte:
2
Todas las frecuencias de los itemsets:
[{'A'}, {'D', 'A'}, {'D', 'B', 'A'}, {'B', 'A'}, {'C'}, {'D',
'C'}, {'C', 'B'}, {'D', 'C', 'B'}, {'D'}, {'D', 'B'}, {'B'}]
```

```
Mínimo conteo de soporte:
1
Todas las frecuencias de los itemsets:
[{'A'}, {'C', 'A'}, {'D', 'C', 'A'}, {'C', 'B', 'A'}, {'D', 'C',
'B', 'A'}, {'D', 'A'}, {'D', 'B', 'A'}, {'B', 'A'}, {'C'}, {'D',
'C'}, {'C', 'B'}, {'D', 'C', 'B'}, {'D'}, {'D', 'B'}, {'B'}]
```

Conjunto de datos 2

Se realiza el mismo procedimiento, pero usando el conjunto de datos 2, donde como el máximo soporte obtenemos:

```
Mínimo conteo de soporte:
5
Todas las frecuencias de los itemsets: [{'I2'}]
```

```
Mínimo conteo de soporte:
4
Todas las frecuencias de los itemsets:
[{'I3'}, {'I2', 'I3'}, {'I1'}, {'I2', 'I1'}, {'I4'}, {'I2'}]
```

```
Mínimo conteo de soporte:
3
Todas las frecuencias de los itemsets:
[{'I3'}, {'I3', 'I2'}, {'I1'}, {'I3', 'I1'}, {'I3', 'I1', 'I2'},
{'I1', 'I2'}, {'I4'}, {'I2', 'I4'}, {'I2'}]
```

```
Mínimo conteo de soporte:
2
Todas las frecuencias de los itemsets:
[{'I5'}, {'I3'}, {'I2', 'I3'}, {'I1'}, {'I3', 'I1'}, {'I3', 'I2',
'I1'}, {'I2', 'I1'}, {'I4'}, {'I3', 'I4'}, {'I2', 'I3', 'I4'}, {'I1',
'I4'}, {'I2', 'I1', 'I4'}, {'I2', 'I4'}, {'I2'}]
```

C. CONJUNTO DE DATOS "GROCERY"

Además de sacar la frecuencia de los conjuntos, agregamos los "umbrales" de frecuencia mínimo y mínimo de confianza que nos dan los resultados más relevantes:

1. BISCUIT, COFFEE → COOKIES, CORNFLAKES:
   - Confianza: 1.000
   - Soporte: 0.100
   - Lift: 10.000
   - Convicción: 900,000,000.000

   Esta regla indica que cuando un cliente compra "BISCUIT" y "COFFEE", es seguro que también comprará "COOKIES" y "CORNFLAKES". El valor de convicción extremadamente alto sugiere que esta asociación es casi una certeza.

2. COFFEE, COOKIES, CORNFLAKES → BISCUIT:
   - Confianza: 1.000
   - Soporte: 0.100
   - Lift: 2.857
   - Convicción: 650,000,000.000

   Esta regla indica que cuando un cliente compra "COFFEE," "COOKIES," y "CORNFLAKES," es seguro que también comprará "BISCUIT." El valor de convicción es alto, lo que sugiere una fuerte relación entre estos productos.

3. BISCUIT, COFFEE, CORNFLAKES → COOKIES:
   - Confianza: 1.000
   - Soporte: 0.100
   - Lift: 5.000
   - Convicción: 800,000,000.000

   Esta regla indica que cuando un cliente compra "BISCUIT," "COFFEE," y "CORNFLAKES," es seguro que también comprará "COOKIES." El valor de convicción es alto, lo que sugiere una fuerte relación.

4. COOKIES, CORNFLAKES → BISCUIT, COFFEE:
   - Confianza: 1.000
   - Soporte: 0.100
   - Lift: 10.000
   - Convicción: 900,000,000.000

   Esta regla indica que cuando un cliente compra "COOKIES" y "CORNFLAKES," es seguro que también comprará "BISCUIT" y "COFFEE." El valor de convicción es extremadamente alto, lo que sugiere una relación extremadamente fuerte.

5. JAM → BREAD, JUICE:
   - Confianza: 1.000
   - Soporte: 0.100
   - Lift: 6.667
   - Convicción: 850,000,000.000

   Esta regla indica que cuando un cliente compra "JAM," es seguro que también comprará "BREAD" y "JUICE." El valor de convicción es alto, lo que sugiere una relación fuerte.

IV. CONCLUSIONES

En esta práctica, implementamos los algoritmos A priori y FP Growth para el análisis de un conjunto de datos y obtuvimos resultados positivos en términos de eficiencia y utilidad en la extracción de patrones de asociación. Ambos algoritmos demostraron su capacidad para descubrir relaciones interesantes entre los elementos del conjunto de datos, identificando patrones de compra o uso que son valiosos para la toma de decisiones empresariales.

El algoritmo A priori, aunque simple y fácil de entender, mostró su utilidad con conjuntos de datos al reducir el espacio de búsqueda y realizar un filtrado inicial de elementos infrecuentes. Por otro lado, el algoritmo FP

Growth destacó por su capacidad para construir una estructura de árbol condicional que optimiza la búsqueda de patrones, lo que lo hace especialmente útil en conjuntos de datos más densos. En general, esta práctica ilustra cómo estas técnicas de minería de datos pueden proporcionar información valiosa y ayudar a tomar decisiones basadas en datos sólidos.

V. BIBLIOGRAFIA

[1] González L. (S. f.) reglas de asociación. https://aprendeia.com/reglas-de-asociacion/

[2] Reglas de asociación. (2023, 3 de agosto). Wikipedia, La enciclopedia libre. https://es.wikipedia.org/w/index.php?title=Reglas_de_asociaci%C3%B3n&oldid=152844375.

[3] Anónimo (s. f.) Efficient-A priori https://efficient-apriori.readthedocs.io/en/latest/

[4] Anónimo (S. f.) Algoritmo de crecimiento de patrón frecuente (FP) en minería de datos https://spa.myservername.com/glaseado-gym-completion-guide-pokemon-scarlet-violet

[5] Amat J. (Septiembre, 2023) Reglas de asociación con Python https://cienciadedatos.net/documentos/py50-reglas-de-asociacion-python

[6] Data Mining Algorithms In R/Frequent Pattern Mining/The FP-Growth Algorithm. (2021, December 15). Wikibooks https://en.wikibooks.org/w/index.php?title=Data_Mining_Algorithms_In_R/Frequent_Pattern_Mining/The_FP-Growth_Algorithm&oldid=4016791.