Análisis de secuencias
Algoritmo AprioriAll

Ramón García Verjaga
José Alberto Gómez García

Índice I

Índice II

01

Introducción

Introducción

Las **reglas de asociación** son un **mecanismo** que permite expresar la **existencia de patrones entre diversos conjuntos de datos**.

Estos **patrones** nos **permiten obtener información** inicialmente oculta entre la gran cantidad de datos que podamos haber recopilado con el paso del tiempo.

Un par de ejemplos de uso de reglas de asociación son:

- el descubrimiento de los **recorridos más frecuentes** que se producen a la hora de navegar entre las diferentes páginas de un sitio **web**;
- o conocer **qué productos suelen comprarse juntos** en un supermercado o tienda online, de manera que se pueda optimizar la asignación de productos a estantes, o establecer técnicas de marketing para vender mejor los productos.

Introducción

Estas **reglas de asociación** nos pueden ayudar a la hora de **tomar decisiones estratégicas** en el marco de un negocio o producto, de manera que intentemos **aumentar** la **rentabilidad** o el **beneficio** que obtenemos del mismo.

A veces, únicamente **tenemos en cuenta las transacciones** sin tener en cuenta el **posicionamiento temporal-espacial** de las mismas.

Cuando dispongamos de **marcas de tiempo** que nos indiquen cuándo se han realizado las transacciones sería interesante **analizarlas e intentar extraer conocimiento de las mismas**. Esto es lo que pretendemos con el análisis de secuencias.

02

Reglas de asociación

Reglas de asociación

co-ocurrencia

Una regla de asociación es una declaración del tipo:

«**SI** *x* **ENTONCES** *y*»

Es decir, es una proposición sobre la **ocurrencia** de cierta situación con una determinada **probabilidad** asociada, en una base de datos de tipo transaccional.

Al conjunto que representa la ***x*** se le denomina **antecedente** y al conjunto que representa la ***y*** se le llama **consecuente**.

Un **ejemplo** de regla de asociación podría ser:

«**SI** *condimentos* **ENTONCES** *carne*»

Aunque se expresan de forma similar, **no debemos confundir estas proposiciones** con aquellas propias de la **lógica proposicional** (utilizadas en reglas de clasificación).

Reglas de asociación
Conceptos básicos

Bases de datos de transacciones

Id	Ítemsets
1	{leche, pan}
2	{pan, mantequilla}
3	{cerveza}
4	{leche, pan, mantequilla}
5	{pan}

- **Transacción**: operaciones o acciones sobre las que obtenemos un identificador asociado a un conjunto de ítems. Filas.
- **Ítem**: es cada uno de los artículos que forman parte de una transacción. Por ejemplo: leche.
- **Ítemset**: es un conjunto de uno o más ítems. Por ejemplo:
  {leche, pan y mantequilla}
  - **K-ítemset**: es un conjunto de ítems que posee k elementos. Por ejemplo: los 2-ítemsets de nuestro conjunto de datos son: {leche, pan} y {pan, mantequilla}.

Reglas de asociación
Conceptos básicos

Bases de datos de transacciones

Id	Ítemsets
1	{leche, pan}
2	{pan, mantequilla}
3	{cerveza}
4	{leche, pan, mantequilla}
5	{pan}

- **Soporte**: para un conjunto de ítems X se define como la proporción de transacciones que contiene dicho conjunto de ítems.

  Soporte({leche}) = **2 / 5** = 0,4 o **40 %**

- **Itemset frecuente**: es aquel cuyo **soporte** es igual o superior a un **umbral** establecido de antemano.
  - Es **maximal** si es frecuente y ningún superconjunto del ítemset es frecuente.

Reglas de asociación
Medidas de efectividad

Bases de datos de transacciones

Id	Ítemsets
1	{leche, pan}
2	{pan, mantequilla}
3	{cerveza}
4	{leche, pan, mantequilla}
5	{pan}

La efectividad de una regla de asociación determinada se mide por **dos parámetros** principales: el soporte y la confianza.

- El **soporte** se refiere a la **frecuencia relativa** con la que una determinada **regla aparece** en la **base de datos** que se está analizando.
- La **confianza** se refiere a la **fiabilidad** o **soporte** de una **regla** de asociación determinada.

También, tenemos otra medida denominada **lift**, que mide la **correlación** entre **antecedente** y **consecuente**. Es decir, mide **hasta qué punto ocurren el antecedente** y el **consecuente conjuntamente** más o menos de lo esperado si fuesen independientes.

Reglas de asociación
Medidas de efectividad

Bases de datos de transacciones

Id	Ítemsets
1	{leche, pan}
2	{pan, mantequilla}
3	{cerveza}
4	{leche, pan, mantequilla}
5	{pan}

Si tuviéramos la siguiente regla:

R = «**SI** *pan* **ENTONCES** *leche*»

Tendríamos los siguientes valores de efectividad:

- **Soporte** (R): como hay 5 transacciones y el ítemset {pan, leche} aparece dos veces, tenemos un soporte para la regla de 2 entre 5 que es 0,4 o el **40%**.
- **Confianza** (R): Como {pan}, ítemset del antecedente, aparece en 4 transacciones y {leche}, ítemset del consecuente, solamente aparece en 2 de esas 4 transacciones, tenemos una confianza para la regla de 2 entre 4 que es 0,5 o el **50 %**.

Generalmente, nos interesan reglas con un **soporte** mayor al **30 %** y una **confianza** mayor al **70 %**.

03

Patrones de secuencias

Patrones de secuencias

¿Qué nos interesa?

Con el momento en que uno o varios artículos fueron comprados por los clientes tenemos una **información temporal** muy valiosa.

¿Qué sucede con esta información?

Esta información puede utilizarse para **construir la secuencia de transacciones** realizadas por un cliente en un determinado periodo de tiempo.

¿Qué significa esto?

Esto significa que existe una **relación ordinal**, normalmente basada en la **precedencia temporal o espacial**, entre los datos basados en sucesos.

¿Qué sucede con las reglas de asociación?

Hasta el momento las reglas de asociación sólo enfatizan las relaciones de co-ocurrencia y no tienen en cuenta la información secuencial de los datos.

La **información secuencial** de los datos puede ser **valiosa** para identificar **características recurrentes** de un sistema dinámico o para predecir futuras ocurrencias de eventos.

04

Análisis de secuencias

Análisis de secuencias

Una secuencia es una lista ordenada de elementos:

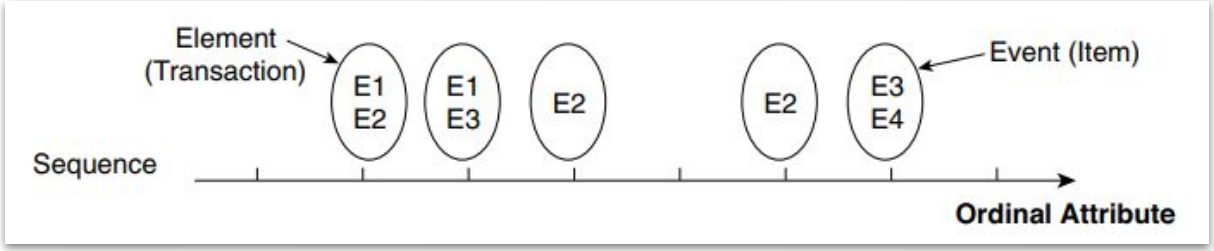
s = ‹ e1, e2, …, en ›

Donde:

- Cada elemento contiene una **colección de eventos**: **ei = {i1, i2, …, ik}**
- Cada elemento tiene una **localización temporal asociada**.

La **longitud de la secuencia** es el **número de elementos** de la secuencia

Una **k-secuencia** es una **secuencia** de **k eventos**.



Análisis de secuencias

BBDD	Secuencia	Elemento (Transacción)	Evento (Ítem)
Clientes	Historial de compras de un cliente determinado	Conjunto de artículos comprados por un cliente en un instante concreto	Libros, productos...
Web	Navegación de un visitante de un sitio web	Colección de ficheros vistos por el visitante tras un único clic de ratón	Página inicial, información de contacto, fotografía
Eventos	Eventos generados por un sensor	Eventos generados por un sensor en un instante t	Tipos de alarmas generadas
Genoma	Secuencia de ADN	Elemento de la secuencia de ADN	Bases A, T, G, C

Análisis de secuencias

Algunos **ejemplos**:

- **Visitas de una página web**:

  <{Página de inicio}, {Electrónica}, {Tablets}, {iPad}, {Shopping cart}, {Order confirmation}, {Volver al carrito}>

- **Préstamos de libros de una biblioteca**:

  <{Harry Potter y la Piedra Filosofal}, {Harry Potter y la Cámara Secreta}, {Harry Potter y el Prisionero de Azkaban}>

Análisis de secuencias

ID-Cliente	Inst-Temporal	Items
1	15/03/2003	{23, 56}
1	17/03/2003	{42, 13}
1	18/03/2003	{45, 33}
2	12/03/2003	{12, 13}
2	18/03/2003	{23, 34, 5, 8}

ID-Cliente	Secuencia
1	< {23, 56}, {42, 13}, {45, 33} >
2	< {12, 13}, {23, 34, 5, 8} >

Análisis de secuencias

Decimos que una **secuencia** ⟨a1, a2, ..., an⟩ **está contenida** en otra **secuencia** ⟨b1, b2, ..., bm⟩ si existe un conjunto de enteros a1 ⊆ bi1, a2 ⊆ bi2, ..., an ⊆ bin.

Secuencia	Subsecuencia	¿Incluida?
<{2, 4}, {3, 5, 6}, {8}>	<{2}, {3, 5}>	Sí
<{1, 2}, {3, 4}>	<{1}, {2}>	No
<{2,4}, {2,4}, {2,5}>	<{2}, {4}>	Sí

El **soporte** de una **subsecuencia** S se define como la **fracción de secuencias** de la base de datos **que incluyen** la **subsecuencia** S.

Un **patrón secuencial** es una **subsecuencia frecuente** (esto es, una subsecuencia con **soporte ≥ MinSupp**)

05

Algoritmo AprioriAll

Algoritmo AprioriAll

Es un algoritmo basado en la versión normal del algoritmo Apriori.
Fue propuesto por Rakesh Agrawal y Ramakrishnan Skirant en 1995

- Es un algoritmo que requiere de bastantes recursos.
- Conviene gestionar muy bien el almacenamiento. Se suelen usar “hash trees”.

```
L1 = {large 1-sequences}
for (k = 2; Lk-1 ≠ {}; k++) do
   begin
   Ck = New candidates generated from Lk-1
   foreach customer-sequence c in the database do
      Increment the count of all candidates in Ck
         that are contained in c.
   Lk = Candidates in Ck with minimum support.
   end
Answer = Maximal Sequences in Lk

Notation:
Lk: Set of all large k-sequences
Ck: Set of candidate k-sequences
```

Algoritmo AprioriAll
Base de datos de transacciones

Tenemos una base de datos como la mostrada en la siguiente tabla.

Como vemos tenemos:

- **ID-Cliente**: el identificador del cliente que realiza la transacción
- **Inst-Temporal**: la fecha en la que se realiza la transacción.
- **Ítems**: Conjunto de ítems a los que aplica la transacción.

ID-Cliente	Inst-Temporal	Items
1	15/03/2003	{30}
1	17/03/2003	{90}
2	18/03/2003	{10, 20}
2	18/03/2003	{30}
2	18/03/2003	{40, 60, 70}
3	18/03/2003	{30, 50, 70}
4	13/03/2003	{30}
4	15/03/2003	{40, 70}
4	17/03/2003	{90}
5	14/03/2003	{90}

Algoritmo AprioriAll

El algoritmo está compuesto por 5 fases:

1. Ordenación
2. Selección de itemsets
3. Transformación y renombramiento
4. Construcción de itemsets frecuentes
5. Selección de secuencias maximales

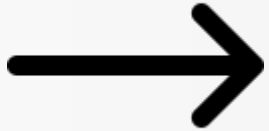
Vamos a ir comentando cada una de las fases con un ejemplo

Algoritmo AprioriAll
Fases | 1. Ordenación

Ordenación. Debemos ordenar la base de datos haciendo uso de los identificadores únicos de clientes. Posteriormente, ordenamos en función del instante temporal de forma ascendente, de manera que transacciones más antiguas sean las primeras. Este paso implica convertir una base de datos de transacciones en una base de datos de secuencias de clientes.

Algoritmo AprioriAll
Fases | 1. Ordenación

ID-Cliente	Inst-Temporal	Items
1	15/03/2003	{30}
1	17/03/2003	{90}
2	18/03/2003	{10, 20}
2	18/03/2003	{30}
2	18/03/2003	{40, 60, 70}
3	18/03/2003	{30, 50, 70}
4	13/03/2003	{30}
4	15/03/2003	{40, 70}
4	17/03/2003	{90}
5	14/03/2003	{90}



ID-Cliente	Secuencia
1	< {30} ; {90} >
2	< {10, 20} ; {30} ; {40, 60, 70} >
3	< {30, 50, 70} >
4	< {30} ; {40, 70} ; {90} >
5	< {90} >

Algoritmo AprioriAll
Fases | 2.Selección de ítemsets

Selección de ítemsets. Para cada cliente, seleccionamos los conjuntos de ítems que tienen una cobertura mínima que hayamos seleccionado de antemano. La cobertura es calculada con respecto a la presencia de un itemset en la secuencia de cada cliente.

ID-Cliente	Secuencia
1	< {30} ; {90} >
2	< {10, 20} ; {30} ; {40, 60, 70} >
3	< {30, 50, 70} >
4	< {30} ; {40, 70} ; {90} >
5	< {90} >

{30} con ID = 1

{40} con ID = 2

{70} con ID = 3

{40, 70} con ID = 4

{90} con ID = 5

Soporte mínimo: 2 / 40 %

Algoritmo AprioriAll
Fases | 3.Transformación y renombramiento

Transformación y renombramiento. Cada secuencia se transforma de manera que sólo tenga ítems frecuentes. Hecho esto, cada conjunto de ítems frecuentes en cada secuencia es renombrado con un identificador único.

{30} con ID = 1

{40} con ID = 2

{70} con ID = 3

{40, 70} con ID = 4

{90} con ID = 5

ID-Cliente	Secuencia	Secuencia transformada	Secuencia final renombrada
1	< {30} ; {90} >	< {30}; {90} >	< {1} ; {5} >
2	< {10, 20} ; {30} ; {40, 60, 70} >	< {30}; {{40} , {70} , {40, 70}} >	< {1} ; {2, 3, 4} >
3	< {30, 50, 70} >	< {{30} , {70}} >	< {1, 3} >
4	< {30} ; {40, 70} ; {90} >	< {30}; {{40}, {70}, {40,70}}, {90} >	< {1} ; {2, 3, 4} ; {5} >
5	< {90} >	< {90} >	< {5} >

Algoritmo AprioriAll

Fases | 4.Construcción de secuencias frecuentes

Construcción de secuencias frecuentes. A partir del conjunto de ítems frecuentes, se construye iterativamente el conjunto de secuencias que cumplan con el criterio de cobertura.

Tamaño 1		Tamaño 2		Tamaño 3		Tamaño 4	
Seq.	Soporte	Seq.	Soporte	Seq.	Soporte	Seq.	Soporte
< 1 >	4	< 1, 2 >	2	< 1, 2, 3 >	2	< 1, 2, 3, 4 >	2
< 2 >	2	< 1, 3 >	3	< 1, 2, 4 >	2		
< 3 >	3	< 1, 4 >	2	< 1, 3, 4 >	2		
< 4 >	2	< 1, 5 >	2	< 2, 3, 4 >	2		
< 5 >	3	< 2, 3 >	2				
		< 2, 4 >	2				
		< 3, 4 >	2				

Algoritmo AprioriAll
Fases | 5.Selección de secuencias maximales

Selección de secuencias maximales. Se filtra el conjunto de secuencias frecuentes de manera que no haya subsecuencias partiendo desde las secuencias de mayor tamaño.

Nos quedan 2 secuencias maximales:

- <1, 2, 3, 4> → <{30}; {40} ; {70}; {40, 70}
- <1, 5> → < {30}, 90} >

Con ellas, y mientras respetemos el orden, podemos generar las reglas.

06

Otras formas de análisis

Otras formas de análisis

Existen otras versiones de Apriori:

- MinApropri
- AprioriSome
- Hybrid Apriori

Muchos otros algoritmos:

- GSP
- FP-growth
- TANIMOTO...

Reglas de asociación multinivel aplicable a lo anterior.

07

Conclusiones

Conclusiones

- Hemos recordado lo que son las reglas de asociación, el uso que se les puede dar, la relación que tienen con el análisis de secuencias y cómo pueden ayudar a tomar determinadas decisiones.
- Hemos aprendido lo que son los patrones de secuencias y la importancia que tienen para representar la espacio-temporalidad en relación a datos transaccionales.
- Hemos expuesto el algoritmo AprioriAll para análisis de secuencias explicando las diferentes fases en las que está estructurado y ejemplificando cada una de ellas.
- Hemos nombrado otras técnicas para análisis de secuencias.
- En definitiva, tener en cuenta la temporalidad de los eventos a la hora de obtener información a través de la generación de reglas de asociación es muy importante. Para multitud de acciones, como hemos podido ver, es importante saber en qué orden suceden los eventos y qué relación tienen los unos con los otros.

08

Bibliografía

Bibliografía

- P-N. Tan, M. Steinbach and V. Kumar. Introduction to Data Mining. Pearson, 2005. Capítulo 7.
- R. Agrawal and R. Srikant. Mining sequential patterns. Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, 1995, pp. 3-14.
- P. Fournier-Viger, J.C.W. Lin, R.U. Kiran, Y.S. Koh. A survey in sequential pattern mining, Data Science and Pattern Recognition. 2017.
- José Hernández Orallo, M.José Ramírez Quintana, Cèsar Ferri Ramírez. Introducción a la Minería de Datos. Pearson, 2004. Capítulo 9.
- Fernando Berzal. Patrones secuenciales. Obtenido de: https://elvex.ugr.es/idbis/dm/slides/22%20Pattern%20Mining%20-%20Sequences.pdf
- Wikipedia. GSP algorithm. Obtenido de: https://en.wikipedia.org/wiki/GSP_algorithm
- Gabriel Navarro. Material académico proporcionado en la asignatura Tratamiento Inteligente de Datos del Máster Profesional Universitario en Ingeniería Informática de la Universidad de Granada.
- Association Rules Discovery. Obtenido de: https://www.cs.upc.edu/~mmartin/D7-%20Association%20rules.pdf

¡Gracias!

¿Alguna pregunta?