Reglas de asociación

Luis Carlos Molina Félix
Ramon Sangüesa i Solé

PID_00165732

Universitat Oberta de Catalunya
www.uoc.edu

Índice

Introducción

A las grandes superficies les interesa saber qué productos forman la cesta de la compra de sus clientes. El hecho de saber que dos productos tan dispares como la cerveza y los pañales aparecen con frecuencia en los grupos de productos registrados en las transacciones de los puntos de venta puede permitir redistribuir los objetos dentro de las estanterías de manera que su probabilidad de compra aumente. Con la información recogida en las cajas registradoras se construye una base de datos en la que los atributos son los diferentes productos. Cada atributo representa una transacción que corresponde a un único cliente, por ejemplo, patatas, leche, café, pan tostado, cerveza, pañales, azúcar, vino, pan integral, salmón, casetes vírgenes, pilas de 1,5 V, Coca-Cola, Fanta, queso, jamón, embutidos, cacahuetes, paté, galletas de aperitivo, vasos de plástico, platos de plástico o servilletas de papel.

Los **episodios frecuentes** son coocurrencias del mismo acontecimiento.

Ejemplo de episodio frecuente

En el caso de la cesta de la compra, el acontecimiento es la compra de un producto.

Hay otro tipo de información que sería de utilidad, pero que este tipo de modelos no considera, como las cantidades y los precios de cada producto. No es relevante. El objetivo no es otro que conocer la composición de la cesta de la compra por tipo de producto, de manera que la representación de las transacciones no es otra que una tupla con valores binarios (0 ó 1, verdadero o falso, presente o ausente) que indica si un cliente ha comprado o no un tipo de producto determinado. No es necesario añadir que el número de atributos de esta base de datos puede estar en el orden de los millares. Asimismo, podemos ver que muchas de las transacciones pueden tener gran cantidad de valores a cero que indican que el cliente no ha comprado el producto correspondiente.

Una última observación: la identidad del cliente es completamente irrelevante para este tipo de estudio. Vosotros mismos podéis imaginaros otras aplicaciones en ámbitos como las transacciones bancarias, los seguros, el apoyo a clientes, accesos a un sitio web, el diagnóstico de averías frecuentes, etc.

Uso del análisis de episodios frecuentes

El análisis de episodios frecuentes no se limita al ámbito del análisis de la cesta de la compra.

Objetivos

Con los materiales didácticos asociados a este módulo, el estudiante alcanzará los objetivos siguientes:

1. Conocer las características principales de las reglas de asociación.

2. Saber hacer uso de las mismas para varias técnicas de *data mining*.

1. ¿Qué son las reglas de asociación?

Una **regla de asociación** tal como nosotros la entendemos es una expresión de la forma

$$X \Rightarrow Y$$

donde tanto X como Y son conjuntos de elementos. Se entiende por **elementos** los atributos que pueden adoptar valores binarios. Estos elementos permiten formar una expresión lógica compuesta de conjunciones, disyunciones y negaciones.

Reglas de asociación

Atención, no os confundáis. Una regla de asociación no es una regla de clasificación.

La interpretación que hay que hacer de una expresión como $X \Rightarrow Y$ es básicamente frecuentativa: se observa que cuando aparece un objeto en un determinado conjunto de datos que contiene el elemento X, también suele aparecer el elemento Y. Un ejemplo típico es que los clientes que compran casetes (X) también compran pilas (Y).

Fijaremos la nomenclatura y la notación con el fin de poder explicar de una manera más formal cómo actúan los diferentes métodos de construcción de reglas de asociación.

Fijamos en primer lugar cómo están formados los antecedentes de este tipo de reglas.

Sea $L = \{e_1, e_2, ..., e_n\}$ un conjunto de literales (en el sentido que recibe este término en lógica). Cada e_i es un *elemento*. Sea B un conjunto de datos, en el que cada componente C de este conjunto está definido sobre los mismos literales: $C \subseteq L$.

Para caracterizar estos elementos según la nomenclatura utilizada hasta ahora, deberíamos decir que C es un conjunto de atributos binarios, $C = \{e_1, e_2, ..., e_n\}$, o que cada e_i solamente puede adoptar los valores {0, 1} que indican que aquel atributo está ausente (0) o presente (1).

Supondremos que cada componente C (un registro de la base de datos, una transacción de la base de datos de ventas, etc.) del conjunto de datos B tiene un identificador único C_{id}.

Un conjunto de elementos X, $X \subset L$ se denomina **grupo de elementos**. Diremos que un componente C del conjunto de datos (un registro o una transacción) contiene un grupo X si $X \subseteq C$.

Una regla de asociación es una implicación de la forma $X \Rightarrow Y$ que cumple las propiedades siguientes:

Propiedades de las reglas de asociación

Las propiedades de las reglas de asociación se pueden resumir en la frase siguiente: intersección vacía de antecedente y consiguiente.

a) $X \subset L$, todos los elementos del grupo X pertenecen al conjunto de elementos sobre los cuales está definido el conjunto de datos (registros, tuplas, transacciones).

b) $Y \subset L$, todos los elementos del grupo Y pertenecen al conjunto de elementos sobre los cuales está definido el conjunto de datos (registros, tuplas, transacciones).

c) $X \cap Y = \varnothing$, no hay ningún elemento repetido a ambos lados de la implicación.

La implicación de la regla de asociación $X \Rightarrow Y$ se cumple en el conjunto de datos B con una **confianza** c si el c por ciento (c%) de los componentes (registros, tuplas, transacciones) de B que contienen el grupo X también contienen el grupo Y.

La regla de asociación $X \Rightarrow Y$ tiene un **soporte** s dentro del conjunto de datos B si el s por ciento (s%) de los componentes de B contiene la unión de su antecedente y su consiguiente, $X \cup Y$.

Ejemplo de regla de asociación

Aquí tenemos parte de la base de datos B de nuestro gimnasio Hyper-Gym:

Horario	Act1	Act2	Entrenador	Uso piscina
Tarde	Aeróbic	*Stretch*	No	Sí
Tarde	Aeróbic	*Stretch*	No	Sí
Mañana	Aeróbic	Yoga	No	Sí
Tarde	TBC	*Steps*	No	No
Tarde	TBC	*Stretch*	No	Sí
Mañana	Yoga	TBC	Sí	Sí
Mañana	*Stretch*	TBC	No	Sí
Tarde	TBC	TBC	No	Sí
Tarde	TBC	TBC	No	Sí
Tarde	TBC	*Steps*	No	Sí

De hecho, para poder aplicar correctamente el método de construcción de reglas de asociación, habría que distinguir entre los atributos que corresponden a la primera y a la segunda actividad. Por lo tanto, deberíamos desdoblar estos dos atributos en Act1-TBC-Sí, Act1-Yoga, Act1-Stretch-Sí, Act1-Aeróbic, Act1-TBC, Act2-TBC, Act2-TBC-Yoga, Act2-Stretch, Act2-Aeróbic. Podemos ver que la regla:

{'Tarde', 'Act1-TBC'} ⇒ {'No entrenador personal', 'Piscina'}

que nos señala que los usuarios del horario de tarde que practican como primera actividad el TBC no tienen entrenador personal pero hacen uso de la piscina, tiene una confianza del 80% y un soporte del 40%.

En cambio, la regla siguiente:

{'Tarde', 'Act1-TBC'} ⇒ {'No entrenador personal'}

que nos indica que quienes van al gimnasio por la tarde y hacen como primera actividad TBC no tienen entrenador personal tiene una confianza del 100% y un soporte del 50%.

La regla:

{'Tarde'} ⇒ {'No entrenador personal'}

que nos indica que quienes van en el horario de tarde no disponen de entrenador personal, tiene una confianza del 100% y un soporte del 70%.

El **problema de la obtención de reglas de asociación** consiste en conseguir, a partir de un conjunto de datos *B*, todas las reglas que tienen un soporte mínimo determinado por el usuario (que denotaremos como *min_sop*) y una confianza mínima también determinada por el usuario (que denotaremos como *min_conf*).

Actividad

1.1. Con los datos del ejemplo que acabamos de presentar, si hubiéramos pedido que se obtuviera un conjunto de reglas con un soporte del 85% y una confianza del 95%, no se habrían descubierto las reglas que hemos descrito, pero ¿qué habríamos obtenido?

Es importante que nos demos cuenta de que al tratar de descubrir patrones o reglas con determinadas características de calidad (soporte y confianza) nos situamos en un terreno diferente del de la comprobación que efectivamente verifica una hipótesis determinada. Nos planteamos un objetivo ligeramente diferente al de las pruebas de hipótesis en estadística, que intentan verificar si se cumple una determinada hipótesis –por ejemplo, que las personas que van por la tarde no tienen entrenador personal. Aquí no se trata de determinar si una hipótesis se cumple, sino de saber qué hipótesis se cumple; no se trata de comprobar, sino de descubrir.

Las reglas de asociación descubren, no comprueban.

1.1. Construcción de reglas de asociación simples

Vamos a tratar de construir un método astuto para obtener reglas que cumplan los requerimientos de soporte y confianza que les pedimos. No se nos escapa que nos hallamos ante un problema computacional nada trivial. En efecto, deberíamos ir comprobando si un grupo de un único atributo cumple los requerimientos y, si los cumple, intentar ir añadiendo otros atributos hasta encontrar el conjunto máximo que permita construir la regla. El número de combinaciones posibles que hay que probar puede ser bastante elevado.

1.1.1. Terminología

En primer lugar, fijaremos ciertos términos de nomenclatura.

Dado un conjunto de atributos R, una **base de datos binaria *r* sobre *R*** es una colección (o conjunto múltiple) de subconjuntos de ***R***. Denominamos ***ítems*** a los elementos de ***R***, y **líneas** a los de ***r***. El **número de líneas de *r*** se denota $|r|$ y la **medida de *r*** se denota:

$$\|r\| = \Sigma_{ter}|t|$$

Utilizamos las letras mayúsculas del principio del alfabeto *A, B,* ... para denotar los ítems (o elementos). Denotamos el conjunto de todos los ítems con R. Denotamos otros conjuntos con las letras finales del alfabeto, como X e Y. Los símbolos en negrita denotan la colección de subconjuntos, por ejemplo, $\boldsymbol{S}$. Las bases de datos se denotan por letras minúsculas como r, y las líneas, por letras como t y u.

La primera propiedad que nos interesa para todo conjunto de elementos de R, $X \subseteq R$, nos permite saber en cuántas líneas aparece lo que hemos llamado *soporte*. A continuación, definimos el soporte de una manera más formal.

Conceptos básicos

- Concordancia
- Soporte
- Confianza

Sea R un conjunto y r una base binaria definida sobre R, y sea X, $X \subseteq R$ un conjunto de ítems (elementos). El conjunto de ítems X concuerda con una línea $t \in r$, si $X \subseteq t$. El conjunto de líneas de r con que coincide X se denota como $M(X,r)$, por ejemplo, $M(X,r) = \{t \in r \mid X \subseteq t\}$. El soporte de X en r, denotado por $sop(X,r)$, es:

$$\frac{|M(X,R)|}{|r|}$$

Escribiremos simplemente $M(X)$ y $sop(X)$ si no introducimos ninguna ambigüedad al hablar de esta manera de la base de datos en un contexto determinado. Dado un umbral de soporte $min\_sop \in [0, 1]$, el conjunto X está *soportado* si $sop(X,r) \geq min\_sop$.

En la literatura también se utiliza el término *frecuencia* para definir el soporte.

Dada una colección de conjuntos de ítems, las reglas de asociación realizan una descripción de cómo aparecen diferentes combinaciones de ítems en los mismos conjuntos.

Ejemplo de binarización de las bases de datos

Descubrir reglas de asociación en ocasiones obliga a "binarizar" la base de datos original derivando nuevos atributos de los ya existentes. Por ejemplo, si tenemos una base de datos binaria r sobre el conjunto $R = \{A, ..., K\}$:

Base de datos	
ID de la línea	**Línea**
t_1	{*A, B, C, D, G*}
t_2	{*A, B, E, F*}
t_3	{*B, I, K*}
t_4	{*A, B, H*}
t_5	{*E, G, J*}

podemos tomar el subconjunto {*A, B*}. Entonces podemos definir los subconjuntos o reglas de asociación que se pueden organizar sobre esta base de datos como $M(\{A, B\},r) = \{t_1, t_2, t_4\}$. Tomemos $sop(\{A, B\},r) = 3/5 = 0{,}6$. Como ya hemos mencionado con anterioridad en el ejemplo de Hyper-Gym, podemos expandir la base de datos en forma de relación en la que todos los atributos {*A, ..., K*} son binarios. Entonces la tabla anterior quedaría así:

Expansión a valores binarios											
ID de la línea	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***	***J***	***K***
t_1	1	1	1	1	0	0	1	0	0	0	0
t_2	1	1	0	0	1	1	0	0	0	0	0
t_3	0	1	0	0	0	0	0	0	1	0	1
t_4	1	1	0	0	0	0	0	1	0	0	0
t_5	0	0	0	0	1	0	1	0	0	1	0

Como ya hemos dicho, un conjunto de atributos *X* será lo suficientemente frecuente (tendrá bastante soporte) si coincide, como mínimo, con una proporción *min_sop* de las líneas en la base de datos *r*. El **umbral de soporte** *min_sop* es un parámetro que viene dado por el usuario y depende de cada aplicación. Lo definimos de una manera más formal a continuación.

Sea *R* un conjunto, *r* una base de datos binaria sobre *R* y *min_sop* el **umbral de soporte**. La colección de conjuntos que tienen bastante soporte a *r* segundos en *min_sop* se denota por $F(r,min\_sop)$, y se define como:

$$F(r,min\_sop) = \{X \subseteq R \mid sop(X,r) \geq min\_sop\}$$

Simplificaremos la notación a $F(r)$ cuando no exista ambigüedad. La colección de conjuntos con soporte de medida suficiente *l* se denota por:

$$F_1(r) = \{X \in F(r) \mid |X| = l\}$$

Ejemplo de conjunto con soporte mínimo

Supongamos que el umbral de soporte es 0,3. La colección $F(r,0,3)$ de conjuntos con un soporte m en la base de datos r de la tabla de expansión a valores binarios del ejemplo anterior es $\{\{A\}, \{B\}, \{E\}, \{G\}, \{A, B\}\}$, dado que ningún otro conjunto diferente del vacío aparece en más de una línea. El conjunto vacío $\varnothing$ tiene trivialmente el soporte mínimo en cualquier base de datos binaria; por lo tanto, nunca lo consideraremos como un caso interesante.

Podéis ver la base de datos del ejemplo de binarización de bases de datos en una parte anterior de este mismo subapartado.

Para cada regla podemos tener su soporte y su confianza.

Sea R un conjunto, r una base de datos binaria sobre R y $X,Y \subseteq R$ conjuntos de ítems. Entonces la expresión $X \Rightarrow Y$ es una **regla de asociación sobre *r***. La confianza de $X \Rightarrow Y$ en r, denotada por $conf(X \Rightarrow Y,r)$, se define como:

$$\frac{|M(X \cup Y,r)|}{|M(X,r)|}$$

El **soporte** $sop(X \Rightarrow Y,r)$ **de** $X \Rightarrow Y$ **en** r es $sop(X \cup Y,r)$.

Escribiremos $sop(X \Rightarrow Y)$ para simplificar cuando no haya ambigüedad.

Validez de una regla

Una regla es válida sólo si tiene el soporte y la confianza mínimos.

Dado un umbral de soporte *min_sop* y un umbral de confianza *min_conf*, la regla de asociación $X \Rightarrow Y$ es válida en r si, y sólo si, $sop(X \Rightarrow Y,r) \geq min\_sop$ y $conf(X \Rightarrow Y,r) \geq min\_conf$.

La tarea de descubrimiento de reglas de asociación...

... implica descubrir conjuntos que cumplan ciertos requerimientos.

En este caso, la calidad del modelo se evalúa en función del soporte y la confianza.

En otras palabras, la confianza $conf(X \Rightarrow Y,r)$ es la probabilidad condicional de que una línea elegida aleatoriamente dentro de r que coincida con X, también coincida con Y. El soporte de la regla es la cantidad de evidencia que se puede obtener a partir de la base de datos a favor de la regla. Para considerar una regla interesante, debe ser lo suficientemente frecuente y fuerte (esto es, tener bastante soporte y confianza).

Ya podemos establecer con más propiedad cuál es la tarea de descubrimiento de reglas de asociación. En efecto, dados *R, r, min_sop* y *min_conf*, es preciso encontrar todas las reglas de asociación $X \Rightarrow Y$ que sean válidas a r con respecto a *min_sop* y *min_conf* a fin de que X e Y sean conjuntos disjuntos y no vacíos.

Ejemplo de descubrimiento de reglas de asociación

Volvemos a la base de datos de la tabla de expansión a valores binarios del ejemplo anterior. Supongamos que tenemos un umbral de soporte *min_sop* = 0,3 y un umbral de confianza, *min_conf* = 0,9.

La única regla de asociación con partes izquierda y derecha no vacías y válida en la base de datos es $\{A\} \Rightarrow \{B\}$. El soporte de la regla es $0,6 \geq min\_sop$ y su confianza, $1 \geq min\_conf$. En cambio, la regla $\{B\} \Rightarrow \{A\}$ no es válida dentro de esta base de datos porque su confianza es 0,75, menor que *min_conf*.

Podéis ver la base de datos del ejemplo de binarización de bases de datos más arriba en este mismo subapartado.

Observad que las reglas de asociación no tienen propiedades de monotonía con respecto a la expansión o contracción de la parte izquierda. Si $X \Rightarrow Y$ es válida, entonces $X \cup \{A\} \Rightarrow Y$ no tiene por qué ser necesariamente válida, dado que $X \cup \{A\} \Rightarrow Y$ no tendrá necesariamente suficiente soporte o confianza. Por otra parte, las reglas de asociación tampoco mantienen propiedades de monotonía con respecto a la expansión de su parte derecha. En efecto, si $X \Rightarrow Y$ es válida, entonces $X \Rightarrow Y \cup \{A\}$ no tiene que ser necesariamente válida con el suficiente soporte o confianza.

Monotonía

La parte izquierda de las reglas no conservan la monotonía con respecto al soporte.

La parte derecha sí que lo hace.

Las reglas de asociación sólo mantienen la propiedad de monotonía con respecto a la contracción de la parte derecha. En efecto, si $X \Rightarrow Y \cup \{A\}$ es válida, seguramente $X \Rightarrow Y$ también es válida.

Actividad

1.2. Comprobad con los ejemplos que habeis visto en este subapartado las propiedades de monotonía en las reglas de asociación y las reglas obtenidas.

1.2. Generación de reglas

Agrawal propone la división de fases siguiente dentro de todo método de construcción de reglas de asociación:

Lectura recomendada

Encontraréis la división de fases que propone Agrawal en la obra siguiente:

R. Agrawal; T. Imielinski; A. Swami (1993). "Mining Association Rules between Sets of Items in Large Databases". En: P. Buneman; S. Jajodia (ed.). *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data: SIGMOD'93* (pág. 207-216). Washington: ACM.

a) Encontrar todas las combinaciones de elementos que tienen un valor de soporte que se ha fijado como mínimo, *min_sop*. Estas combinaciones de elementos se conocen como *grandes grupos* (*large itemsets*).

b) Utilizar los grandes grupos para construir finalmente las reglas. La idea es que si tenemos dos grandes grupos, por ejemplo, ABCD y AB, hay que determinar si se cumple la regla AB $\Rightarrow$ CD. Para ello, conviene que veamos si posee la confianza necesaria o superior a la confianza exigida por el usuario, *min_conf*.

Por norma general, los algoritmos de construcción de reglas de asociación hacen, como mínimo, dos pasadas sobre el conjunto de datos, una para extraer los grandes grupos y otra para construir las reglas. Los grandes grupos de elementos se construyen considerando un nuevo elemento cada vez y encontrando gradualmente las reglas que tienen ese nuevo elemento en el consiguiente y que tienen la confianza máxima (100%).

El proceso habitual consiste en seguir los pasos siguientes:

1) Elegir una *semilla* de grandes grupos de elementos.

2) Utilizar esa semilla para generar nuevos posibles grandes grupos, los grupos candidatos.

3) Se evalúan estos grupos comparando su soporte con respecto al soporte mínimo; si superan el soporte mínimo, *min_sop*, se consideran grandes grupos.

4) Estos grupos se convierten en las semillas de la fase siguiente.

5) Repetir el proceso hasta que no se encuentren más conjuntos grandes.

A continuación, presentaremos el algoritmo de construcción de reglas de asociación propuesto por Agrawal y otros, y mejorado por el mismo Agrawal en 1995.

Como ya hemos comentado, la idea es obtener en primer lugar todos los conjuntos de ítems $X \subseteq R$ y calcular sus respectivos soportes. Entonces, hay que comprobar por separado para todos $Y \subset X$, $Y \neq \emptyset$ si la regla $X \setminus Y \Rightarrow Y$ es válida con suficiente confianza. El algoritmo siguiente utiliza esta aproximación para generar todas las reglas de asociación válidas con la base de datos de entrada. En el subapartado siguiente abordamos la parte clave de todo algoritmo: cómo encontrar los conjuntos que tienen suficiente soporte en la base de datos.

Lecturas recomendadas

Encontraréis el algoritmo de construcción de reglas de asociación propuesto por Agrawal y el algoritmo mejorado, respectivamente, en las obras siguientes:

R. Agrawal; T. Imielinski; A. Swami (1993). "Mining Association Rules between Sets of Items in Large Databases". En: P. Buneman; S. Jajodia (ed.). *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data: SIGMOD'93* (pág. 207-216). Washington: ACM.

R. Agrawal; H. Mannila; R. Srikant; H. Toivonen; I. Verkamo (1995). "Fast discovery of Association Rules". En: U.M. Fayyad; G. Piatetsky-Sahpiro; P. Smyth; R. Uthurusamy (ed.). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press.

El algoritmo de Agrawal presenta la entrada y la salida siguientes:

- Entrada. Un conjunto *R*, una base de datos binaria *r* sobre *R*, un umbral de soporte *min_sop* y un umbral de confianza *min_conf*.
- Salida. Las reglas de asociación que son válidas para *r* con respecto a *min_sop* y *min_conf* y sus respectivos soporte y confianzas.

Los pasos que sigue el algoritmo de Agrawal son los que enumeramos a continuación:

1) Encontrar los conjuntos frecuentes.

2) Calcular $F(r,min\_sop) := \{X \subseteq R \mid fr(X,r) \geq min\_sop$.

3) Ejecutar el bucle siguiente para generar reglas:

```
para todo X ∈ F(r,min_sop) hacer
    para todo Y ⊂ X como Y ≠ ∅ hacer
        si sop(X) / sop(X \ Y) ≥ min_conf entonces
            crear la regla X \ Y ⇒ Y , sop(X) y sop(X) / sop(X \ Y);
        fsi
    fpara
fpara
```

Vamos a ver cómo funciona este algoritmo. En primer lugar, se verifica la igualdad siguiente:

$$conf(X \Rightarrow Y,r) = \frac{|M(X \cup Y,r)|}{|M(X,r)|} = \frac{sop(X \cup Y,r)}{sop(X, r)}$$

Está claro que todas las reglas de asociación $X \Rightarrow Y$ que genera el algoritmo son válidas con respecto a la base de datos *r*:

$sop(X \Rightarrow Y) \geq min\_sop$, dado que (tercera línea del bucle):

$sop(X \cup Y) \geq min\_sop$ (línea 2) y $conf(X \Rightarrow Y) \geq min\_conf$.

El algoritmo genera todas las reglas de asociación $X \Rightarrow Y$ que son válidas con la base de datos de entrada. Como $sop(X \Rightarrow Y) \geq min\_sop$, entonces también $sop(X \cup Y) \geq min\_sop$ y $X \cup Y$ debe estar dentro de $F(r,min\_sop)$ (punto 2). Si esto ocurre, la posible regla $X \Rightarrow Y$ quedará verificada (punto 3). Como $conf(X \Rightarrow Y) \geq min\_conf$, la regla correspondiente se generará efectivamente (tercera línea del bucle).

1.3. La clave de todo el método: cómo encontrar conjuntos frecuentes

Buscar exhaustivamente los conjuntos frecuentes, es decir, conjuntos con soporte suficiente, lo que Agrawal denomina *large itemsets*, es una tarea claramente pesada desde el punto de vista computacional, por no decir imposible. Sólo resulta factible para los conjuntos *R*: "pequeños". En efecto, el espacio de búsqueda de conjuntos frecuentes está formado por $2^{|R|}$ subconjuntos de *R*.

Espacio de búsqueda de conjuntos frecuentes

Con $R = 100$, que representa un número minúsculo de productos para una cadena de supermercados, el número de configuraciones por explorar es esta terrorífica cifra:

$1{,}26765060022823 \cdot 10^{30}$.

Por tanto, se impone otra aproximación. El concepto clave ahora es el de **conjunto candidato**.

La idea es construir progresivamente conjuntos cada vez mayores. Hay que ir probando varias cardinalidades o medidas de los conjuntos. Para cada cardinalidad, cada $l = 1, 2, \ldots$, se determina primero una colección de conjuntos C_l de elementos de medida *l* tales que $F_l(r) \subseteq C_l$. De esa manera, se obtiene la colección de **conjuntos candidatos frecuentes** (*large itemsets*) $F_l(r)$ que calculan sus frecuencias.

El problema de encontrar conjuntos candidatos

Encontrar conjuntos candidatos representa un problema que requiere una aproximación claramente diferente de la exploración exhaustiva de combinaciones posibles.

Para una base de datos grande con un elevado número de atributos y con conjuntos candidatos grandes, el cálculo de la frecuencia es bastante costoso. Por

ello, resulta útil minimizar el número de candidatos igualándolo al de la fase de generación.

Con el fin de construir una colección de conjuntos candidatos pequeña pero suficiente, debemos prestar atención a las propiedades de los conjuntos de datos que a describimos continuación.

a) Un subconjunto de líneas es, como mínimo, tan frecuente (tiene el mismo soporte) como el superconjunto dentro del que está incluido. En otras palabras, el soporte es monótono con respecto a la contracción del conjunto para el que se calcula, lo cual quiere decir que para cualquier subconjunto X, Y de ítems tales que $Y \subseteq X$, se cumple $M(Y) \supseteq M(X)$ y $sop(Y) \geq sop(X)$ y, además, si X tiene suficiente soporte (es bastante frecuente), entonces Y también.

b) A partir de la propiedad anterior podemos conseguir información útil para la generación de candidatos. En efecto, dado un conjunto X, si cualquiera de los subconjuntos de X no tiene suficiente soporte, entonces podemos descartar X como conjunto frecuente posible, es decir, como candidato posible, y lo eliminamos del conjunto de candidatos $C_{|X|}$.

La proposición siguiente, establece además, que lo que acabamos de ver es suficiente para saber si todos los subconjuntos de X tienen suficiente soporte.

Proposición: sea $X \subseteq R$ un conjunto. Si cualquiera de los subconjuntos propios Y de X, $Y \subset X$ no tiene suficiente soporte, se verifican las propiedades siguientes:

1) X no tiene suficiente soporte.

2) Existe un subconjunto Z, $Z \subset X$, sin suficiente soporte, de medida: $|X| - 1$.

Demostración: el primer punto se deriva directamente de la observación de que si X tiene suficiente soporte, entonces todos sus subconjuntos de X, $Y \subset X$ tienen suficiente soporte. Podemos aplicar el mismo argumento para el segundo punto: para cualquiera Y, $Y \subset X$ existe Z tal que $Y \subseteq Z \subset X$ y $|Z| = |X| - 1$. Si Y no tiene suficiente soporte, Z tampoco lo tiene.

Ejemplo de conjunto candidato

Si sabemos que un conjunto:

$$F_2(r) = \{\{A, B\}, \{A, C\}, \{A, E\}, \{A, F\}, \{B, C\}, \{B, E\}, \{C, G\}\}$$

podemos concluir que $\{A, B, C\}$ y $\{A, B, E\}$ son los únicos componentes de $F_3(r)$ posibles, dado que son los únicos conjuntos de medida 3 para los cuales todos los subconjuntos de medida 2 aparecen en $F_2(r)$. Ahora ya sabemos que $F_4(r)$ debe ser vacío.

Con posterioridad, este algoritmo ha recibido una serie de mejoras.

Lecturas recomendadas

Con relación al algoritmo de Agrawal mejorado podéis consultar la primera de las obras mencionadas a continuación. Asimismo, podéis ver otros algoritmos alternativos en las obras siguientes, cuyas referencias completas encontraréis en el apartado de bibliografía del módulo:

R. Agrawal; H. Mannila; R. Srikant; H. Toivonen; I. Verkamo (1995).

R. Agrawal; R. Srikant (1994).

M. Houtsma; A. Swami (1993).

H. Mannila; H. Toivonen; I. Verkamo (1994).

2. Ponderación de las reglas de asociación

Las reglas de asociación resultan especialmente tentadoras en muchas aplicaciones de *marketing,* pero tienen unos requerimientos tanto de almacenamiento como computacionales bastante elevados. A continuación mencionamos algunos:

1) Por una parte, trabajan sobre **grandes conjuntos de atributos**; por ejemplo, los productos de una cadena de supermercados.

En cuanto a transformación de datos, requieren la eliminación de parte de los datos que normalmente se recogen en las transacciones (cantidad, precio, etc.). Como paso siguiente, obligan a la binarización de los datos; es decir, a transformar los valores de todos los atributos en los equivalentes a {Presente, Ausente}. Este hecho comporta la ampliación de la dimensión de una tabla de la base de datos que, de entrada, ya tiene una dimensión bastante elevada.

Asimismo, debemos tener en cuenta que en aplicaciones reales, el número de atributos de la base de datos puede ser muy elevado, del orden de millares en el caso de aplicaciones de análisis de la cesta de la compra. Pero el número de éstos presente en cada transacción será pequeño. Ello hace que, al construir los conjuntos candidatos de dimensión *i*, nos veamos obligados a guardar los resultados intermedios entre el barrido *i* sobre la base de datos y el barrido *i* + 1, no en memoria principal, sino en disco.

Conjuntos candidatos de medida

Para generar conjuntos de medida 5, por ejemplo, debemos partir de los de medida 4 y volver a efectuar recuentos de frecuencia. El coste de rehacer los recuentos no es trivial.

La forma en que se van construyendo los conjuntos candidatos es ideal para intentar aportar soluciones desde el proceso paralelo.

Lectura recomendada

Podéis ver una discusión general sobre el paralelismo en *data mining* con una sección sobre reglas de asociación en la obra siguiente:

M. Holsheimer; M.L. Kersten (1994). "Architectural Support for Data Mining. Knowledge". *Discovery in Databases. Papers from the 1994 AAAI Workshop* (pág. 217-228). Menlo Park.

2) Por otra parte, podemos ver que el **coste** también depende del soporte exigido por el usuario. Cuanto más soporte, más pasadas hay que hacer sobre la base de datos.

Al igual que sucede en otros sistemas de construcción de reglas (por ejemplo, de clasificación), las colecciones de reglas de asociación resultantes pueden ser muy grandes y complicar la interpretación.

3) La **capacidad predictiva** de las reglas obtenidas es otro aspecto que no podemos dejar de considerar. El **soporte** y la **confianza** tienen evidentemente, relación con la capacidad de predicción, pero no son determinantes. El hecho de que una gran proporción de transacciones muestre una asociación determinada entre el objeto *X* y el objeto *Y* con una confianza bastante alta no significa que se pueda generalizar con mucha facilidad.

A pesar de los aparentes inconvenientes que acabamos de mencionar, las reglas de asociación resultan bastante comprensibles y se emplean en muchas aplicaciones prácticas.

La extensión de las reglas de asociación a datos no binarios es un problema bastante interesante que cuenta con muchas aplicaciones en potencia. Para una discusión de métodos de descubrimiento sobre las reglas de asociación sobre atributos que pueden tomar varios valores, podéis consultar Miller, 1997.

Lectura recomendada

Hallaréis una discusión interesante con respecto a la capacidad predictiva de las reglas de asociación en la obra siguiente:

A. Siebes (1994). "Homogenenous Discoveries Contain no Surprises: Inferring Risk-Profiles from Large Databases". *Knowledge Discovery in Databases. Papers from the 1994 AAAI Workshop* (pág. 97-108). Menlo Park.

En general, las reglas de asociación exploran dependencias, lo que es un problema que ya había sido atacado desde las bases de datos. En efecto, es importante detectar en los datos relaciones de dependencia funcional que permitan después modificar el diseño inicial de las bases de datos. Podéis consultar al respecto Mannila, 1994.

Finalmente, se está explorando la relación entre las reglas de asociación y las de clasificación y se han encontrado métodos que permiten mejorar la precisión de las clasificaciones obtenidas mediante una combinación de ambas perspectivas. Si deseáis ampliar información sobre esta cuestión, podéis leer la obra siguiente: Liu, 1998. También, es interesante ver cómo se pueden obtener reglas de clasificación con poder predictivo a partir de grandes conjuntos de reglas de asociación; para ello, podéis consultar Klemettinen, 1994.

Resumen

Las reglas de asociación son un modelo que describe un dominio en función de las dependencias entre conjuntos de valores.

Los atributos que aparecen en las partes izquierda y derecha de una regla corresponden a atributos cuyos valores coocurren en el conjunto de datos original con un soporte y una confianza determinados por el usuario.

Los datos deben estar en formato binario: los valores de los atributos sólo pueden ser 'presente' o 'ausente', lo cual supone, por norma general, un primer proceso de transformación del conjunto original de datos.

La complejidad del descubrimiento de conjuntos de reglas con las características requeridas es un proceso costoso desde un punto de vista computacional. Cada grupo de atributos de cierta cardinalidad *C* requiere al menos *C* barridos previos para encontrar subconjuntos de cardinalidad menor.

La propiedad de no-monotonía del soporte para subconjuntos de un conjunto dado permite mejorar los algoritmos de construcción de reglas de asociación. Se trata de un modelo bastante comprensible, si bien sus propiedades estadísticas (en concreto, las características de predicción) no son demasiado claras. No obstante, tienen una gran cantidad de aplicaciones.

Actividades

1. Acceded a la dirección de Internet que encontraréis al margen y comparad las especificaciones de los diferentes *software* orientados a construir reglas de asociación.

Para realizar la actividad 1, acceded a la dirección http://www.kdnuggets.com.

2. Para el problema que se os había propuesto en la actividad 1 del módulo "Extracción de conocimiento a partir de datos" de esta asignatura, ¿os sirven las reglas de asociación? ¿Qué método creéis que os resultaría más conveniente?

Ejercicios de autoevaluación

1. Dada la base de datos siguiente que corresponde al ejemplo de las lentes de contacto:

Podéis ver el ejemplo de las lentes de contacto en el subapartado 2.1. del módulo "Clasificación: árboles de decisión", de esta asignatura.

Edad	Diagnóstico	Astigmatismo	Lágrima	Recomendación
Joven	Miope	No	Reducida	Ninguna
Joven	Miope	No	Normal	Blandas
Joven	Miope	Sí	Reducida	Ninguna
Joven	Miope	Sí	Normal	Duras
Joven	Hipermétrope	No	Reducida	Ninguna
Joven	Hipermétrope	No	Normal	Blandas
Joven	Hipermétrope	Sí	Reducida	Ninguna
Joven	Hipermétrope	Sí	Normal	Duras
Prepresbicia	Miope	No	Reducida	Ninguna
Prepresbicia	Miope	No	Normal	Blandas
Prepresbicia	Miope	Sí	Reducida	Ninguna
Prepresbicia	Miope	Sí	Normal	Duras
Prepresbicia	Hipermétrope	No	Reducida	Ninguna
Prepresbicia	Hipermétrope	No	Normal	Blandas
Prepresbicia	Hipermétrope	Sí	Reducida	Ninguna
Prepresbicia	Hipermétrope	Sí	Normal	Ninguna
Presbicia	Miope	No	Reducida	Ninguna
Presbicia	Miope	No	Normal	Ninguna
Presbicia	Miope	Sí	Reducida	Ninguna
Presbicia	Miope	Sí	Normal	Duras
Presbicia	Hipermétrope	No	Reducida	Ninguna
Presbicia	Hipermétrope	No	Normal	Blandas
Presbicia	Hipermétrope	Sí	Reducida	Ninguna
Presbicia	Hipermétrope	Sí	Normal	Ninguna

a) Transformad los datos a fin de poder deducir reglas de asociación.
b) Encontrad un conjunto de reglas con soporte mínimo 0,2 y confianza 0,4.
c) ¿Podéis encontrar algún conjunto con soporte y confianza superiores?

2. Dada la base de datos siguiente:

Horario	Act1	Act2	Entrenador personal	Uso de la piscina
Tarde	Aeróbic	*Stretch*	No	Sí
Tarde	Aeróbic	*Stretch*	No	Sí
Mañana	Aeróbic	Yoga	No	Sí
Tarde	TBC	*Steps*	No	No
Tarde	TBC	*Stretch*	No	Sí
Mañana	Yoga	TBC	Sí	Sí
Mañana	*Stretch*	TBC	No	Sí
Tarde	TBC	TBC	No	Sí
Tarde	TBC	TBC	No	Sí
Tarde	TBC	*Steps*	No	Sí

a) Binarizadla.
b) ¿Cuál es el soporte de la regla {'Tarde', 'TBC', '*Stretch*'}⇒ {'Entrenador personal', 'Piscina'}?
c) ¿Y su confianza?
d) Encontrad, si existe, la regla mínima de soporte 0,9 y confianza 0,9.
e) ¿Cuál es el conjunto de soporte y confianza mínimos superiores a 0,1?
f) Y ¿cuál es el conjunto máximo?
g) Repetid los apartados **e)** y **f)** con el valor 0,4.

Bibliografía

Agrawal, R.; Imielinski, T.; Swami, A. (1993). "Mining Association Rules between Sets of Items in Large Databases". En: P. Buneman; S. Jajodia (eds.). *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data: SIGMOD'93* (págs. 207-216). Washington: ACM.

Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; Verkamo, I. (1995). "Fast discovery of Association Rules". En: U.M. Fayyad; G. Piatetsky-Sahpiro; P. Smyth; R. Uthurusamy (eds.). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press.

Agrawal, R.; Srikant, R. (1994). "Fast Algoritmics for Mining Tools". *Proceedings of the International Conference on Very Large DataBases,VLDB-94*.

Holsheimer, M.; Kersten, M.L. (1994). "Architectural Support for Data Mining". *Knowledge Discovery in Databases. Papers from the 1994 AAAI Workshop* (págs. 217-228). Menlo Park.

Houtsma, M.; Swami, A. (1993). En: *Set-Oriented Mining of Association Rules*. Research Report RJ 9567 (octubre). San José: IBM Almaden Research Center.

Klemettinen, M.; Mannila, H.; Ronkainen, H.; Toivonen, H.; Verkamo, A. (1994). "Finding Interesting Rules from Large Sets of Discovered Association Rules". *Proceedings of CIKM'94, Conference on Information and Knowledge Management* (págs. 40-407).

Liu, B.; Hsu, W.; Ma, Y. (1998). "Integrating Classification and Association Rule Mining". *4th International Conference on Knowledge Discovery and Data Mining: KDD 98* (agosto, págs. 23-27). Nueva York.

Liu, B.; Hsu, W.; Ma, Y. (1998). *Building an Accurate Classifier using Association Rules*. Technical Report.

Mannila, H.; Räihä, K-J. (1994). "Algorithms for Inferring Functional Dependencies from Relations". *Data & Knowledge Engineering* (vol. 12, núm. 1, págs. 83-99).

Mannila, H.; Toivonen, H.; Verkamo, I. (1994). "Efficient Algorithms for Discovering Association Rules". *A Knowledge Discovery in Databases, Technical Report WS-94-03*. AAAI.

Miller, R.J.; Yang, Y. (1997). "Association Rules over Interval Data". *ACM SIGMOD. International Conference on the Management of Data* (vol. 27, núm. 2, págs. 452-461).

Siebes, A. (1994). "Homogenenous Discoveries Contain no Surprises: Inferring Risk-Profiles from Large Databases". *Knowledge Discovery in Databases. Papers from the 1994 AAAI Workshop* (págs. 97-108). Menlo Park.