IV Reglas de Asociación

4.1. Introducción

Objetivos:

- Comprender el entorno de problemas y la génesis de las reglas de asociación.
- Conocer las definiciones formales de las reglas de asociación.
- Comprender y manipular los parámetros necesarios para evaluar en forma cualitativa una regla.
- Comprender los problemas combinatorios subyacentes a la búsqueda de relaciones frecuentes.
- Evaluar las diferentes medidas de calidad de una regla de asociación.
- Dominar los algoritmos para la búsqueda de reglas frecuentes.

4.2. Presentación del problema

El problema de minería en reglas de asociación fue introducido por Agrawal y col en 1993.

La motivación principal surge de los problemas que tienen los gerentes de supermercados, donde existe una gran cantidad de productos, quienes deben tomar diferentes decisiones como:

- Que productos colocar en venta.
- Como diseñar los cupones de ventas.
- Como colocar la mercadería en los estantes para maximizar las ventas, etc.

La idea básica consiste en analizar las compras que se han realizado en el pasado para mejorar la calidad de estas decisiones.

Analizar las compras en detalle, sólo fue posible en los últimos años con la introducción de los lectores ópticos, lo que ha

permitido almacenar la canasta de compras ("basket-market"), que consiste en el conjunto de productos reunidos en una sola compra (transacción).
Este análisis no es realizado necesariamente en una sola compra, pueden ser compras realizadas por un cliente durante un periodo de tiempo.

Un ejemplo típico es:

"El 30% de las compras que contienen cerveza y papas fritas, también contienen maní salado" y el 2% de todas las compras del supermercado contienen los tres productos.

Esta aseveración se puede expresar como una regla:

Si *A* ***Entonces*** *B* (c,s)

Donde:

- *A* es el conjunto de productos (atributos) de la condición de la regla, denominado ***Antecedente***.
- *B* es el conjunto de productos (atributos) de la conclusión de la regla, denominado ***Consecuente***.
- *c (30%)* se denomina ***confianza*** de la regla.
- *s (2%)* se denomina ***soporte*** de la regla.

El problema general de minería de datos en reglas de asociación consiste en encontrar, en una base de datos, la totalidad de las reglas que cumplen con un conjunto de restricciones como pueden ser: soporte mínimo y confianza mínima.

Antes de considerar este problema general (más restrictivo) es conveniente considerar problemas más relajados que pueden ser de gran interés para la toma de decisiones.

Ejemplos:

- Encontrar todas las reglas que tienen "coca-cola" como consecuente.

- Ayuda a la planificación de la tienda para aumentar la venta de coca-cola.

- Encontrar todas las reglas que tienen "salsa verde" en el antecedente.

- Ayuda a determinar que productos pueden ser impactados si se discontinua la venta de "salsa verde".

- Encontrar todas la reglas que tienen "ketchup" en el antecedente y "mostaza" en el consecuente.

Ayuda a realizar el pedido de un producto adicional (mostaza) que puede ser vendido junto con ketchup.

- Encontrar todas las reglas que tienen productos localizados en los estantes *x* e *y* en la tienda.

Ayuda a planificar los estantes, determinando si las ventas de los productos en los estantes *x* están relacionadas con las ventas de los productos de los estantes *y*.

Los problemas generales y específicos enunciados anteriormente tienen una gran variedad de aplicaciones que no se relacionan necesariamente con la organización de ventas.
Algunos ejemplos son:

- Descubrir la canasta de compras mínima para un tipo de cliente.
- Análisis de marketing cruzado (dado un grupo de productos cual es la preferencia por otro tipo de productos).
- Diseño de catálogos de ventas.
- Detección de fraudes
- Análisis de pérdidas.

4.3. Definiciones formales

Considere una base de datos o Sistema de Información operacional $SI=<U, Q, V, f>$ como la definida en el Capítulo I.

- $S \subseteq U$ universo cerrado: un conjunto finito, no vacío, de n *objetos* $\{x_1, x_2, \ldots, x_n\}$. Denominados ***transacciones*** (compras).

- Q: un conjunto finito, no vacío, de p *atributos* $\{q_1, q_2, \ldots, q_p\}$, llamados también ***productos*** *o* ***ítems***.

- $V = \bigcup_{q \in Q} V_i^q$, donde V_i^q es el dominio (i indica los posibles valores de cada atributo o instancias) de cada uno de los productos (atributos) q.

Las instancias de estos atributos son consideradas siempre como binarias, pueden ser nominales pero se debe realizar una etapa de codificación para transformarlas en binario.

Transacciones	**Productos** (Atributos)						
S	q_1	q_2	...	q_j	...	q_{p-1}	q_p
x_1	V_1^1	V_2^2		V_3^j		V_1^{p-1}	V_1^p
x_2	V_3^1	V_1^2		V_2^j		V_2^{p-1}	V_2^p
. .	V_4^1	V_4^2		V_3^j		V_2^{p-1}	V_k^p

Sea S un conjunto de transacciones donde cada transacción x_i contiene un conjunto de productos tal que $x_i \subseteq V$.

Se dice que cada transacción x_i contiene un conjunto A de algunos productos en V, si $A \subseteq x_i$.

Una regla de asociación es una implicación de la forma:

$$A \Rightarrow B$$

Donde $A \subset V$, $B \subset V$ y $(A \cap B)_V = \varnothing$

Definición:

Soporte: El soporte de un conjunto A de transacciones $Sop(A)$, se define como el número de transacciones de los atributos de A que toman el valor verdadero.

Soporte de una regla: El soporte de una regla $A \Rightarrow B$, $Sop(A \Rightarrow B)$, es el numero de transacciones en el conjunto S tal que A y B son verdaderos simultáneamente.

Para mantener esta cantidad normalizada se usa, en general, como una proporción de las transacciones conjuntas entre A y B y el número total de transacciones del conjunto S (se indica con n).

Al valor normalizado se denominará soporte normalizado

$$Sopn(A \Rightarrow B) = Sop(A \Rightarrow B)/n.$$

Nótese que este soporte normalizado es la estimación de la probabilidad de la intersección entre A y B (probabilidad de juntura).

$$s = Sopn(A \Rightarrow B) = \hat{p}(A \cap B) = \frac{Sop(A \Rightarrow B)}{n}.$$

confianza: La confianza de la regla $A \Rightarrow B$,en el conjunto S, es la proporción entre el número de casos de A y B que aparecen conjuntamente en S contenidos en el número de casos de A.

Esto es: $c = Conf(A \Rightarrow B) = \frac{Sop(A \cap B)}{Sop(A)}$

Al dividir el numerador y denominador por n.

$$c = \frac{Sop(A \cap B)/n}{Sop(A)/n}$$

Sabiendo que:

- el soporte de A dividido por n es la estimación de la probabilidad de A, $\hat{P}(A) = Sop(A)/n = Sopn(A)$.
- y con la definición de probabilidad condicional.

se puede obtener

$$\frac{Sop(A \Rightarrow B)/n}{Sop(A)/n} = \frac{\hat{p}(A \cap B)}{\hat{P}(A)} = \hat{p}(B/A)$$

Por lo tanto, la *confianza* de la regla $A \Rightarrow B$ representa la probabilidad que se encuentren los productos B en la transacción dado que ésta también contiene los productos del conjunto A.

4.4. El problema de la minería de reglas.

El problema general se puede plantear como:

Dado el conjunto $B \subseteq V$, encontrar todos los posibles sub-conjuntos $A \subseteq V$ que cumplan con un conjunto dado de restricciones, las cuales pueden ser: *mínimo Soporte*, *mínima Confianza o alguna métrica individual o común que las involucre a ambas*.

Así cuando se requiere encontrar:

- todas las reglas ***confiables*** se entenderá que es el conjunto de todas las reglas que cumplen con una confianza mínima ***minconf***.
- todas las reglas ***frecuentes*** se entenderá que es el conjunto de todas las reglas que cumplen con un soporte mínimo ***minsop***.

Considerando el conjunto B como dado y siendo verdadero. El conjunto A puede tener dos alternativas.

- Una ***conjunción***, en cuyo caso $A \subseteq V$ será verdadero ssi todas las condiciones de A son verdaderas: $V^1 \wedge V^2 \wedge .... \wedge V^m$.

- Una ***disyunción***, en cuyo caso $A \subseteq V$ será verdadero ssi una o mas condiciones de A son verdaderas: $V^1 \vee V^2 \vee .... \vee V^m$.

Este problema en su forma general, para ambos tipos de reglas, resulta ser un problema *NP-duro*.

Sin embargo instancias específicas de este problema general pueden ser analizadas para obtener tratabilidad.

Los problemas considerados en este capítulo considerarán el tratamiento de reglas conjuntivas.

La búsqueda de los posibles conjuntos A se realiza agregando condiciones al conjunto, esto se denomina ***especialización*** de la regla.

Por el contrario, cuando el antecedente de la regla contiene menos condiciones se dice que la regla está más ***generalizada***.

Ej: Sea: $A_1 = V^1 \wedge V^2$ y $A_2 = V^1 \wedge V^2 \wedge V^3$, A_1 es mas general que A_2, puesto que A_1 contiene a A_2, y $|A_1| < |A_2|$

Para encontrar todas las posibles reglas se requiere un algoritmo que genere todas las posibles combinaciones y realizar una pre-poda de forma tal que al usar las restricciones se detenga la búsqueda para evitar la explosión combinatoria.

El problema del cumplimiento de las restricciones está asociado con la monotonicidad de la restricción, en función de la especialización.

Si se tienen dos especializaciones del antecedente, se generan dos reglas tales que $A_1 < A_2$ y dos restricciones o medidas ***med***(A_i) $i=1,2,$ asociadas a cada una de las reglas.
Se dice que la medida es monótona si: $med(A_1) \leq med(A_2)$.
La medida es anti-monótona si: $med(A_1) \geq med(A_2)$.

Para realizar una pre-poda eficiente se requiere usar restricciones monótonas o anti-monótonas. Con lo cual se descartan ramas completas en el proceso de especialización.

Al analizar el ***Soporte*** se puede observar que esta medida es anti-monótona. Puesto que la especialización de la regla lleva a mantener o disminuir el soporte.

Ej: Si $A_1 = V^1 \wedge V^2$ y $A_2 = V^1 \wedge V^2 \wedge V^3 \Rightarrow A_1 < A_2$, para un B dado.

Ocurrirá que $Sop(A_1 \Rightarrow B) \geq Sop(A_2 \Rightarrow B)$
Puesto que : $Sop(A_1 \Rightarrow B) = P(V^1 \wedge V^2 \cap B)$
y $Sop(A_2 \Rightarrow B) = P(V^1 \wedge V^2 \wedge V^3 \cap B)$

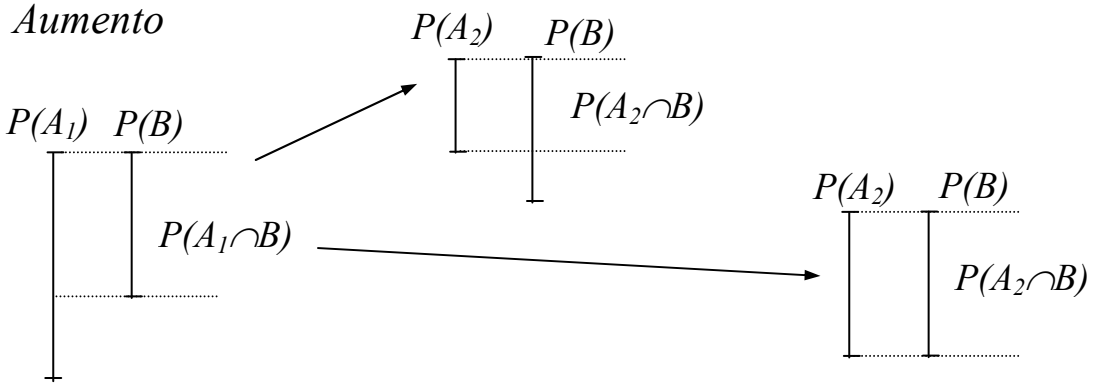
El caso de la ***Confianza*** es diferente puesto que al especializar la regla, su soporte puede mantenerse o disminuir. El soporte del antecedente también puede disminuir, pero en una proporción mayor que el soporte de la regla, en cuyo caso la razón entre los dos, que corresponde a la confianza, puede aumentar.

Dado: $Sop(A_1 \Rightarrow B) = P(V^1 \wedge V^2 \cap B)$ y $Sop(A_1) = P(V^1 \wedge V^2)$

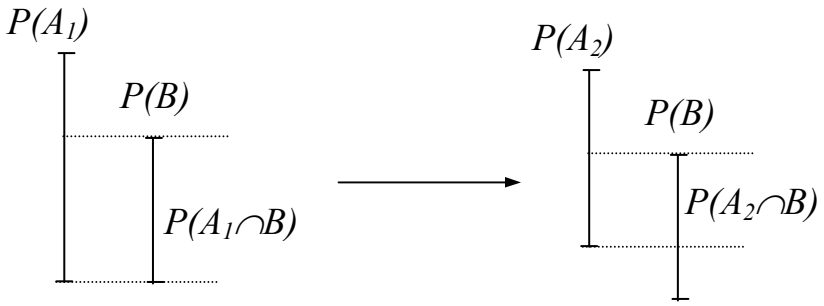
$Sop(A_2 \Rightarrow B) = P(V^1 \wedge V^2 \wedge V^3 \cap B) \downarrow$ y $Sop(A_2) = P(V^1 \wedge V^2 \wedge V^3) \downarrow \downarrow$

o $Sop(A_2 \Rightarrow B) = P(V^1 \wedge V^2 \wedge V^3 \cap B) =$ y $Sop(A_2) = P(V^1 \wedge V^2 \wedge V^3) \downarrow$

Por lo tanto: $Conf(A_2) = \dfrac{Sop(V^1 \wedge V^2 \wedge V^3 \cap B)}{Sop(V^1 \wedge V^2 \wedge V^3)} \uparrow$



Disminución

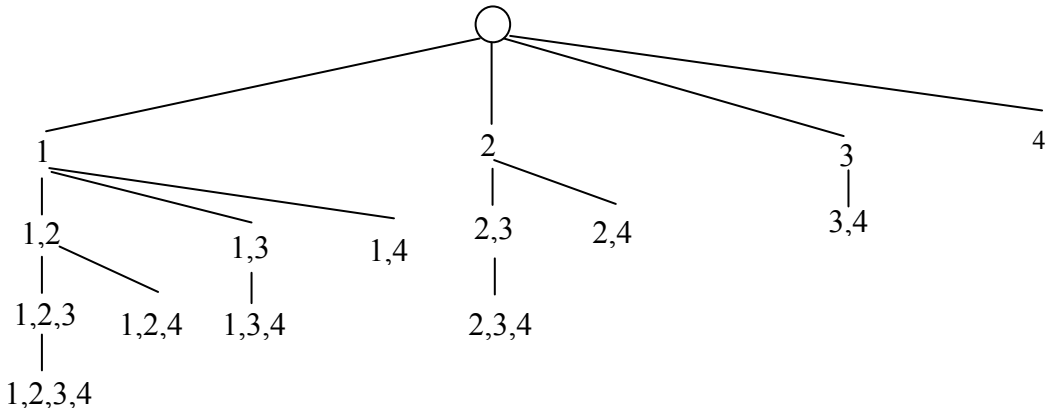


No es posible asegurar monotonicidad o anti-monotonicidad.

4.5. Combinatoria de la búsqueda

El problema consiste en encontrar un algoritmo que permita generar todas las posibles combinaciones para un antecedente y, en el proceso de generación de las reglas, interrumpir la especialización usando medidas monótonas para una pre-poda.

Sea *V={1,2,3,4}* un conjunto de antecedentes del cual se requiere generar todas las posibles combinaciones.



Cada nodo *g* en el árbol es representado por dos grupos:

- uno llamado ***cabeza*** *h(g)* que representa la regla para el nodo.

- Uno llamado ***cola*** *t(g)* que representa todas las posibles combinaciones (ordenadas) que pueden ser agregadas a la cabeza para formar una regla.

El algoritmo inicial presentado por Agrawal utiliza esta combinatoria para generar el árbol paso a paso y va realizando una pre-poda por soporte.

Cuando encuentra un nodo *h(g)* cuyo soporte es inferior al *minsop* entonces no genera las combinaciones *t(g)*, puesto que todas ellas no cumplirán con *minsop*.

Con respecto a la confianza propone simplemente ordenar la totalidad de las reglas generadas y entregar sólo las que cumplen con *minconf*.

La propuesta actual es mantener las "**mejores reglas"** (ej: las más confiables) en una lista y recorrer el árbol por anchura. Si no se consigue agregar nuevas reglas a la lista se detiene el proceso.

Las variaciones de este algoritmo consisten en ordenar la lista de las **mejores reglas** según una medida de calidad de la regla que permita dejar el número mínimo de reglas confiables fuera de la lista.

4.6. Medidas de calidad

Como una forma de evaluar de mejor manera la confianza, se han generado diferentes medidas o métricas de calidad que ayuden a seleccionar el conjunto de las *mejores reglas*.

En general a estas medidas se les exige que sean monótonas en confianza, o soporte y confianza, pero manteniendo uno de ellas constante.

Esto no supera el problema de la no-monotonicidad de la confianza pero, para casos particulares permite realizar una búsqueda más efectiva.

Análisis de monotonicidad de las medidas o métricas de calidad

- ***Lift***: Una medida usada en las herramientas de minería de datos producidas por IBM.

$$lift = \frac{n\, conf(A \Rightarrow B)}{Sop(B)}$$

En términos de probabilidades estimadas se sabe que:

$Conf(A \Rightarrow B) = p(B \cap A)/P(A)$ y que $P(B) = Sop(B)/n$.

Por lo tanto $lift(A \Rightarrow B) = p(B \cap A)/(P(A)P(B))$.

Lo cual representa una medida de independencia entre A y B. Esto es, *lift* tendrá su valor más bajo (1) cuando A y B sean completamente independientes.

Es fácil ver que esta medida es monótona en confianza, puesto que al especializar la regla, *lift* disminuye proporcional a c, puesto que $P(B)$ se mantiene constante para el proceso de especialización.

- ***Convicción***: Una medida similar a la anterior, que mantiene la monotonicidad en confianza.

$$convicción = \frac{n - Sop(B)}{n[1 - conf(A \Rightarrow B)]}$$

Usando estimación de probabilidades y factorizando el numerador por n, se tiene:

$$convicción = \frac{n[1 - P(B)]}{n\left[\frac{P(A) - p(A \cap B)}{P(A)}\right]} = \frac{P(A)[1 - P(B)]}{P(A) - P(A \cap B)}$$

sabiendo que: $P(\overline{B}) = 1 - P(B)$ y que
$P(A \cap \overline{B}) = P(A) - p(A \cap B)$

$$convicción = \frac{P(A)P(\overline{B})}{P(A \cap \overline{B})}$$

Esto tambien representa la independencia de A y B y es monotona en confianza.

Medidas monótonas en soporte y confianza.

- ***Laplace***: $Laplace(A \Rightarrow B) = \frac{Sop(A \Rightarrow B) + 1}{Sop(A) + k}$

Donde la constante k es un entero mayor que 1

Puesto que la confianza $c=p(B/A)=p(A \cap B)/p(A)$ y $Sop(A \Rightarrow B)=p(A \cap B)$, se puede reemplazar en el denominador $P(A)=Sop(A \Rightarrow B)/c$.

Quedando: $Laplace = \frac{Sop(A \Rightarrow B) + 1}{Sop(A \Rightarrow B)/c + k}$

Si la confianza se fija en un valor c, al disminuir el $Sop(A \Rightarrow B)$ por efecto de la especialización, el denominador disminuirá mas rápido que el numerador puesto que $k>1$, haciendo que la medida disminuya.

En el peor caso se mantendrá la relación si k no es lo suficientemente grande en relación a c.

Por lo tanto, la medida es monótona en soporte, manteniendo la confianza constante.

Si el soporte se mantene constante, al especializar la regla, *c* disminuirá haciendo aumentar el denominador, lo que hará que la medida disminuya.

Por lo tanto, la medida es monótona en confianza, para un soporte constante.

- ***Ganancia***: $Gan(A \Rightarrow B) = Sop(A \Rightarrow B) + \theta\, Sop(A)$

con $0<\theta<1$.

Aplicando el mismo concepto anterior $P(A)=Sop(A \Rightarrow B)/c$ y factorizando por $Sop(A \Rightarrow B)$ se tiene:

$$Gan(A \Rightarrow B)=Sop(A \Rightarrow B)(1 - \theta/c)$$

Si la confianza se mantiene constante y $c>\theta$ se observa claramente que la ganancia disminuirá al disminuir el soporte de la regla, puesto que son proporcionales.

Si el soporte de la regla se mantiene constante, la ganancia disminuirá al disminuir la confianza, puesto que aumenta la relación θ/c, aumentando el sustrayendo, y la diferencia $(1-\theta/c)$ disminuirá.
Lo cual implica que la medida será monótona en confianza para soporte constante y viceversa.

- **Métrica de Piatetsky-Shapiro (P-S)**

$$P-S(A \Rightarrow B) = Sop(A \Rightarrow B) - \frac{Sop(A)Sop(B)}{n}$$

Esta métrica presenta la misma estructura que la anterior al considerar $\theta=Sop(B)/n$, como la probabilidad del consecuente. Las condiciones de monotonicidad se pueden estudiar en forma similar a la ganancia.