Reglas de Asociación

Definición

El objetivo de las reglas de asociación es encontrar asociaciones o correlaciones entre los elementos u objetos de bases de datos transaccionales, relacionales o data warehouse.

Las reglas de asociación tienen diversas aplicaciones como:

- Soporte para la toma de decisiones.
- Diagnóstico y predicción de alarmas en telecomunicaciones.
- Análisis de información de ventas

Procedimiento

Se encuentran usando un procedimiento de **covering**. Sin embargo, en el lado derecho de las reglas, puede aparecer cualquier par o pares atributo-valor.

$$\alpha \Rightarrow \beta$$

Para encontrar ese tipo de reglas se debe de considerar cada posible combinación de pares atributo-valor de ambos lados.

Para posteriormente podarlas usando cobertura (número de instancias predichas correctamente) y precisión (proporción de número de instancias a las cuales aplica la regla).

Ejemplo

Encontrar las reglas de asociación $\alpha \Rightarrow \beta$ de la tabla con la restricción de cumplir con un mínimo de cobertura y de precisión. Las reglas con:

- Cobertura mínima de 50%
- Precisión mínima de 50%

Transacción	Elementos Comprados
1	A, B, C
2	A, C
3	A, D
4	B, E, F

Ejemplo

- A ⇒ C (50%, 66.6%)
- C ⇒ A (50%, 100%)

Una regla de asociación es una expresión de la forma α⇒β donde α y β son conjuntos de elementos.

Significado intuitivo: las transacciones de la base de datos que contienen α tienden a contener β.

Definiciones

$I = \{i_1, i_2, i_3, \ldots, i_m\}$ conjunto de literales, atributos

D conjunto de transacciones T, $T \subseteq I$

TID identificador asociado a cada transacción

α conjunto de elementos $\alpha \in I$

Una **regla de asociación** es una implicación: $\alpha \Rightarrow \beta$, $\alpha \in I$, $\beta \in I$ y $\alpha \cap \beta = \emptyset$

Soporte (o **cobertura**), *s*, es la probabilidad de que una transacción contenga $\{\alpha, \beta\}$

Confianza (o **eficiencia**), *c*, es la probabilidad condicional de que una transacción que contenga $\{\alpha\}$ también contenga $\{\beta\}$.

Evaluación

En reglas de asociación, la cobertura se llama **soporte** (support) y la precisión se llama **confianza** (confidence).

Se pueden leer como:

- soporte($\alpha \Rightarrow \beta$) = P($\alpha \cup \beta$)
- confianza($\alpha \Rightarrow \beta$) = P($\beta \mid \alpha$) = $\dfrac{\text{soporte}(\alpha \cup \beta)}{\text{soporte}(\alpha)}$

Problema

Dado un conjunto de transacciones, encontrar reglas que describen tendencias en los datos:

- Detectar cuándo la ocurrencia de un artículo está asociada a la ocurrencia de otros artículos en la misma transacción.

TID	Artículos
1	Pan, leche, huevos
2	Pan, pañales, cerveza
3	Leche, pañales, cerveza
4	Pan, leche, pañales, cerveza
5	Pan, leche, huevos, cerveza

Problema

Reglas de asociación:

- {cerveza} → {pañales} (60%, 75%)
- {leche, pan} → {huevos} (40%, 66%)
- {cerveza, pan} → {leche, huevos} (20%, 33%)

Objetivo: Identificar artículos que muchos clientes compran conjuntamente.

Solución: Procesar los datos de los terminales de punto de venta proporcionados por los escáneres de códigos de barras.

Aplicaciones

Promociones y ofertas

Si se identifica una regla del tipo: {impresora} → {tóner}

- Tóner en el consecuente => Puede determinarse cómo incrementar sus ventas.
- Impresora en el antecedente => Puede determinarse qué productos se verían afectados si dejamos de vender impresoras.
- Impresora en el antecedente y tóner en el consecuente => Puede utilizarse para ver qué productos deberían venderse con impresoras para promocionar las ventas de tóner.

Extracción de Reglas

Dado un conjunto de transacciones T, encontrar todas las reglas de asociación

- cuyo soporte sea mayor o sea mayor o igual que un umbral mínimo de soporte: supp ($\alpha \Rightarrow \beta$) ≥ MinSupp
- cuya confianza sea mayor o sea mayor o igual que un umbral mínimo de confianza: conf ($\alpha \Rightarrow \beta$) ≥ MinConf

Extracción de Reglas

Solución por fuerza bruta

- Enumerar todas las reglas de asociación posibles.
- Calcular el soporte y la confianza de cada regla.
- Eliminar las reglas que no superen los umbrales de soporte y confianza (soporte y confianza (MinSupp y MinConf).

Demasiado costoso!!

Fuerza bruta…

Ejemplo

- Reglas derivadas de {pan, pañales, cerveza}
- {pan} → {pañales, cerveza}, supp=0.4, conf=2/4=0.5
- {pañales} → {pan, cerveza}, supp=0.4, conf=2/3=0.66
- {cerveza} → {pan, pañales}, supp=0.4, conf=2/4=0.5
- {pan, pañales} → {cerveza}, supp=0.4, conf=2/2=1
- {pan, cerveza} → {pañales}, supp=0.4, conf=2/3=0.66
- {pañales, cerveza} → {pan}, supp=0.4, conf=2/3=0.66

Observaciones

Todas las reglas anteriores son particiones binarias del mismo grupo de productos ({pan, pañales, cerveza}).

Todas las reglas que provienen del mismo grupo tienen el mismo **soporte**, aunque su confianza pueda variar.

Por tanto, podemos separar la parte que depende del soporte de la que depende de la confianza.

Solución

Generación de grupos frecuentes: identificar los grupos con soporte ≥ MinSupp.

Generación de reglas de asociación: obtener reglas de asociación con una confianza elevada a partir de cada grupo frecuente, donde cada regla es una partición binaria del grupo.

Todavía es costosa!!

Estrategias

Reducir el número de candidatos (M)

- Uso de técnicas de poda.
- Ejemplo: Algoritmos Apriori y DHP [Direct Hashing and Pruning]

Reducir el número de transacciones (N)

- Reducir N conforme aumenta el tamaño del grupo.
- Ejemplo: Algoritmo AprioriTID

Reducir el número de comparaciones (NM)

- Uso de estructuras de datos eficientes para almacenar los candidatos o las transacciones, de forma , de forma que no haya que comparar cada candidato con todas las transacciones.

Reducción del número de candidatos

La propiedad Apriori

- **Si un grupo es frecuente, también lo son todos sus subconjuntos**

¿Por qué? Porque el soporte de un grupo nunca puede ser mayor que el de cualquiera de sus subconjuntos:

$$\forall X,Y\ (: X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

Formalmente, esta propiedad se conoce con el nombre de anti-monotonía del soporte.

Algoritmo Apriori

Tablas

- L[k] = Conjunto de k-grupos frecuentes
- C[k] = Conjunto de k-grupos potencialmente frecuentes.

Algoritmo

- Generar L[1] (patrones frecuentes de tamaño 1)
- Repetir mientras se descubran nuevos grupos frecuentes:
  a) Generar los candidatos C[k+1] a partir de los patrones frecuentes L[k].
  b) Contabilizar el soporte de cada candidato de C[k+1] recorriendo la base de datos secuencialmente.
  c) Eliminar candidatos no frecuentes, dejando en L[k+1] sólo aquéllos que son frecuentes.