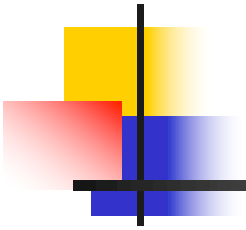
Aprendizaje de Máquina

ITAM



Menú

- Aprendizaje no supervisado
  - Reglas de asociación
    - Búsqueda de asociaciones
    - Búsqueda de Reglas
      - Medidas de interés
    - Algoritmo Apriori
    - Un ejemplo

Reglas de Asociación

- Tipo de Método
  - No supervisado
- Qué suponemos de los datos
  - Datos discretos
    - <atributo1, atributo2,…,atributon>
- Aplicaciones
  - Descubrimiento de patrones
  - Canastas, análisis de datos, predicción

Reglas de Asociación Objetivo

- Objetivo:
  - Encontrar conjuntos de valores para los atributos que aparecen de manera simultánea en una fracción significativa de la base de datos
  - Por ejemplo:
    - Una base de datos de transacciones de un supermercado
    - <Bebidas alcohlicas, Botanas, Productos de Bebe>

    donde:

    Bebidas alcohlicas={cerveza, vino}

    Botanas={jamón,papas, pastelillos}

    Productos de Bebe={talco, pañales}
  - Cada entrada en la base de datos representa una transacción sobre estos productos para un cliente

Reglas de Asociación Objetivo

- El objetivo es descubrir relaciones insospechadas
  - Después de analizar la base podemos asociar la compra de jamón con vino y la venta de cerveza con papas
- ¿Cómo?

Reglas de Asociación Dificultades

- En general va a ser difícil que una relación destaque en una base de datos grande con muchos atributos
- El número de posibles combinaciones de subconjuntos de valores de atributos es intratable
  - ¿Cuántos subconjuntos posibles hay?

Reglas de Asociación Simplificación

- Enfocamos nuestra atención en un solo valor para cada atributo a la vez
- Por ejemplo, en lugar de encontrar relaciones como

  "Si compra cerveza o vino, entonces compra pañales"

  Buscamos independientemente la asociación de cerveza y pañales y la de vino y pañales
  - Las asociaciones que buscamos son conjunciones de valores para un subconjunto de los atributos
- Esta simplificación nos permite convertir cada entrada en la bases de datos (cada transacción) en un vector indicador binaro
  - Para cada posible valor de interés de cada atributo creamos una variable binaria Z_k
  - Z_k es 1 si aparece dicho valor en una transacción y 0 de otro modo

Reglas de Asociación Simplificación

- Por ejemplo:
  - <Bebidas alcohólicas, Botanas, Productos de Bebe>

  donde:

  Bebidas alcohólicas={cerveza, vino}

  Botanas={jamón, papas, pastelillos}

  Productos de Bebe={talco, pañales}

  Supongamos que no nos interesa ninguna relación que tenga que ver con papas y con talco así que:

  Botanas={jamón,pastelillos}

  Productos de Bebe={pañales}
- Creamos para cada transacción un vector binario
  - $<Z_1, Z_2, Z_3, Z_4, Z_5>$ con $Z_k \in \{0,1\}$
  - Donde Z_1 es 1 si en la transacción se compro cerveza y 0 si no. Z_2 es 1 si se compro vino y 0 si no. Z_3 es 1 si se compro jamón, Z_4 si se compro pastelito y Z_5 si se compro pañal….

Reglas de Asociación

- Buscamos el subconjunto K de las variables Z_k para los que la proporción de valores $Z_k=1$ es grande. Es decir:

$$\frac{1}{N}\sum_{j=1}^{N}\prod_{k\in K} z_{j,k}$$

- es grande. Donde $z_{j,k}$ es el valor de la variable Z_k correspondiente al renglón j en la base de datos
- Este número se conoce como el **soporte** del conjunto K y se escribe como T(K)

Reglas de Asociación

- Vamos a buscar relaciones que sean significativas en relación a su tamaño (número de atributos) y a su soporte
- En el minado de asociaciones normalmente se establece el soporte mínimo *t* a buscar y se buscan todos los subconjuntos K_l que cumplan con esto:
  - $\{K_l: T(K_l)>t\}$
- ¿Cómo encontramos todos los subconjuntos? Seguimos con el mismo problema

El Algoritmo Apriori

-Calcular el soporte de todas las variables Z_k. Almacenar en *P* y en *F* sólo aquellas con soporte mayor a t

Repeat{

-Escribir *P* a una base de datos

-Calcular *F* x *P*: todas las combinaciones de las variables sobrevivientes del primer paso con todos los subconjuntos de variables del paso n-1. (Esto crea subconjuntos con n atributos).

-Almacenar en *P* sólo aquellos subconjuntos de tamaño n con soporte mayor a t

}Until P={};

El Algoritmo A priori Características

- Barre la bases de datos |K| veces. Si *t* es suficientemente grande el algoritmo termina en tiempo “razonable”

El Algoritmo Apriori (Agrawal et al.)

- Características
  - Rápido
  - Pocos barridos de los datos
  - Útil para bases de datos muy grandes
  - Encuentra sólo conjuntos de variables K_l (asociaciones) para las que:
    - Su presencia sea mayor a t en la BD
    - La presencia de cualquier subconjunto de K_l es mayor a t
- Por ejemplo si el 90% de las veces que se compra caviar también se compra vodka, pero la venta de caviar es menor a t, el algoritmo Apriori será incapaz de encontrar dicha asociación

Ejemplo

- <cerveza, vino, jamón, pastelillos, pañales>

0	0	1	0	0
1	0	1	0	1
0	1	1	0	1
0	0	0	1	0
1	0	0	0	1
1	0	1	0	0
0	0	0	1	0
0	1	1	0	1
0	1	1	0	1
1	0	0	0	1

- < Z_1, Z_2, Z_3, Z_4, Z_5>

Ejemplo
1er Paso

- t= 0.25

Z1	Z2	Z3	Z4	Z5
0	1	1	0	0
1	0	1	0	1
0	1	1	0	1
0	0	0	1	0
1	0	0	0	1
1	0	1	0	0
0	0	0	1	0
0	1	1	0	1
0	1	1	0	1
1	0	0	0	1

- $T(Z_1)=0.4$, $T(Z_2)=0.4$, $T(Z_3)=0.6$, $T(Z_4)=0.2$, $T(Z_5)=0.6$

Ejemplo
2do Paso

- Combinaciones {Z1, Z2}, {Z1,Z3}, {Z1,Z5}, {Z2,Z3}, {Z2,Z5}, {Z3,Z5}

Z1	Z2	Z3	Z4	Z5
0	1	1	0	0
1	0	1	0	1
0	1	1	0	1
0	0	0	1	0
1	0	0	0	1
1	0	1	0	0
0	0	0	1	0
0	1	1	0	1
0	1	1	0	1
1	0	0	0	1

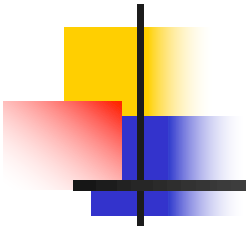
- T({Z1, Z2})=0,T({Z1,Z3})=0.2, T({Z1,Z5})=0.3, T({Z2,Z3})=0.4, T({Z2,Z5})=0.3, T({Z3,Z5})=0.4

Ejemplo t=0.25

- Combinaciones faltantes…. {Z2, Z3,Z5}

Z1	Z2	Z3	Z4	Z5
0	1	1	0	0
1	0	1	0	1
0	1	1	0	1
0	0	0	1	0
1	0	0	0	1
1	0	1	0	0
0	0	0	1	0
0	1	1	0	1
0	1	1	0	1
1	0	0	0	1

- T({Z2,Z3,Z5})=0.3
- Ya no hay mas combinaciones a investigar



Ejemplo

- Los conjuntos sobrevivientes son:
- {Z1}, {Z2}, {Z3},{Z5}
- {Z1,Z5}, {Z2,Z3}, {Z2,Z5}, {Z3,Z5}
- {Z2, Z3, Z5} (vino, jamón y pañales)

Encontrar Reglas de Asociaciones

- Hasta ahora sólo hemos encontrado conjuntos de valores de atributos relacionados.
- Sería útil tener reglas del tipo
  - La compra de vino implica la compra de pañales
- Para cada conjunto Ki encontrado por Apriori lo dividimos en dos conjuntos Ki= A U B donde $A \cap B=\{\}$ de manera que podamos escribir
  - $A \rightarrow B$
- El primer término es el antecedente y el segundo es el consecuente

Reglas de Asociación
Medidas de Interés

- El **soporte** de la regla T(A→B) es igual al soporte de K
- Definimos criterios que establecen el interés de una regla:
  - La **confianza** de la regla C(A→B) es:
    - C(A→B)=T(A→B)/T(A)
    - Es un estimado de la probabilidad de observar B dado que se observo A
  - El "**lift**" de la regla es:
    - L(A→B)=C(A→B)/T(B)
    - Es un estimado de la dependencia entre A y B:
      - P(A y B)/(P(A)P(B))
- Existen otros criterios

Reglas de Asociación
Medidas de Interés

- Leverage
  - Leverage(A→B)= T(A→B)- T(A)T(B)
  - Es un estimado de: P(A y B)- P(A)P(B)
- Conviction
  - Conviction(A→B) =(1-T(B))/(1-C(A→B))
  - Es un estimado de: P(A)P(not B)/P(A y not B)

Reglas de Asociación
Ejemplo {vino, jamón, pañales}

- Reglas
  - {vino, jamón} →{pañales}
  - {vino, pañales} → {jamón}
  - {jamón, pañales} →{vino}
  - {vino} → {jamón, pañales}
  - {jamón} → {vino, pañales}
  - {pañales} →{vino, jamón}
- La regla {vino, jamón} →{pañales} tiene:
  - Soporte =0.3
  - Confianza =0.3/ 0.3=1
  - Lift=1/0.6=1.66

Reglas de Asociación
Medidas de Interés

- Para que la confianza C(A→ B) sea útil debe estar cerca de 1
- El soporte es importante pues nos dice que asociaciones tienen suficiente presencia en la base de datos para ser de interés
- Un “lift” mayor a 1 nos indica que existe una correlación entre los valores encontrados

Reglas de Asociación

- El objetivo es producir reglas con alto soporte y alta confianza (o alto el criterio de interés que se defina)
- El algoritmo Apriori reporta todas las reglas para las que:
  - $T(A \rightarrow B) > t$ y $C(A \rightarrow B) > c$
  - Dado un valor soporte y confianza mínimos t y c.
- El resultado se almacena, en ocasiones, un una base de datos que puede ser consultada:
  - “Dame las reglas que implican la venta de pañales con confianza mayor a 0.2”

Reglas de Asociación Caracterízticas

- Reglas inteligibles
- Aplicable a bases de datos muy grandes
- Es crítico establecer bien el valor mínimo del soporte, de otra manera la búsqueda es exponencial
- Reglas con alta confianza pero bajo soporte no son encontradas por este método
  - Tequila → jumiles
  - No se encontrará por las escasas ventas de jumiles
  - El supuesto es que estas reglas no son importantes. Pero puede ser que sean transacciones poco frecuentes que devengan mayores utilidades a una compañía