ESMA 4016 Mineria de Datos

Association Rules

Dr. Edgar Acuna
Departmento de Matematicas

Universidad de Puerto Rico- Mayaguez
academic.uprm.edu/eacuna

Datos Transaccionales

Ejemplo de Canasta de Mercado:

Basket1: {bread, cheese, milk}

Basket2: {apple, eggs, salt, yogurt}

…

Basketn: {biscuit, eggs, milk}

Definiciones:

- *Un item*: es un articulo en un basket.
- Una *transaction*: items comprados en una canasta; puede tener TID (transaction ID)
- *Un conjunto de datos transaccionales* : Es un conjunto de transacciones

Representacion binaria de datos transactionales

Tid	Items
1	3 4 5 6 7 9
2	1 3 4 5 13
3	1 2 4 5 7 11
4	1 3 4 8
5	1 3 4 10

1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	1	1	1	1	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	0	0	0	0	1
1	1	0	1	1	0	1	0	0	0	1	0	0
1	0	1	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	1	0	0	0

Itemsets and Association Rules

- Un *itemset es un conjuntos de items*.
  - E.g., {milk, bread, cereal} es un itemset.
- Un *k-itemset* es un itemset con k items.
- Dado un conjunto de datos *D*, un itemset *X* tiene una frecuencia de ocurrencia (conteo) en *D.*
- El objetivo es encontrar itemsets que aparecen juntos en muchas transaciones.
- Una regla de asociacion es una relacion entre dos itemsets disjuntos *X* y *Y*

$$X \Rightarrow Y$$

Representa el patron que cuando X ocurre entonces Y tambien ocurre.

Uso de Reglas de Associacion

- Association rules no representan cualquier tipo de causalidad o correlacion entre los dos itemsets.
  - $X \Rightarrow Y$ no significa que X *causa Y,*
  - $X \Rightarrow Y$ puede ser diferente de $Y \Rightarrow X$, distinto a correlacion
- Association rules se puede aplicar en marketing, en publicidad, planificacion de una tienda, control de inventarios, seguridad nacional, comercio electronico,etc.

Support y Confidence

- ***support de*** itemset *X* en *D* es *count*(*X*)/|*D*|
- Para una regla de asociacion $X \Rightarrow Y$, *Podemos calcular*
  - support ($X \Rightarrow Y$) = support ($X \cup Y$)
  - confidence ($X \Rightarrow Y$) = support ($X \cup Y$)/support (X), la cual representa la fuerza de la implicacion.
- Support (S) y Confidence (C) se relacionan con probabilidad conjunta y probabilidad condicional respectivamente.
- Puede haber una cantidad exponencial de reglas de asociacion.
- Reglas de asociacion son aquellas cuyo S y C son mayors o iguales minSup y minConf (umbrales que son puesto por el investigador)

Ejemplo

- Data set *D*

TID	Itemsets
T100	1 3 4
T200	2 3 5
T300	1 2 3 5
T400	2 5

Count, Support, Confidence:

Count(1,3)=2

$|D| = 4$

Support(1,3)=0.5

Support(3 → 2)=0.5

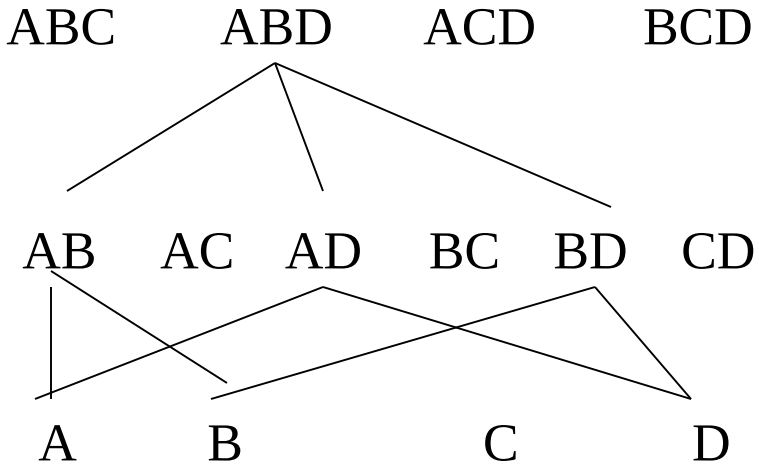
Confidence(3 → 2)=0.67

- Pasos en mineria de reglas de asociacion:
  - Generacion de itemsets Frequentes. Se encuentran los itemsets que tienen support S mayor o igual que un umbral minimo predeterminado.
  - Derivacion de las reglas. Usando los itemsets frecuentes obtenidos en el paso anterior se generan las reglas de asociacion que tienen una confianza C, mayor o igual que un umbral predeterminado.

  El primer paso es el mas importante.

Frequent itemsets

- Un itemset frecuente es un itemset cuyo support (S) es ≥ minSup. Si hay m items en el conjunto de datos entonces Habra 2^m posibles frequent itemsets.
- Propiedad Apriori: Cualquier subconjunto de un itemset frecuente es tambien un itemset frequente any



- Usando esta propiedad se puede podar algunas ramas.

El algoritmo APRIORI (Agrawal et al., 1995). Paso 1.

1. Sea L_1: los frequent 1-itemsets
2. Para k=2, formar C_k a partir de L_{k-1}
3. Hallar frequent set L_k de C_k , conjunto de todos los itemsets candidatos de tamano k. Hacer k = k + 1
4. Repetir los pasos 2-3 hasta que C_k (y por lo tanto L_{k+1}) se vuelvan vacios.
5. Output: Union of todos los L_K.

- En el paso 2, cllamdo el paso de de generacion de los frequent itemset, *D* es escaneado y se cuenta cada itemset en C_k , si es mayor que minSup, es frequente y se vuelve un miembro L_k.

Paso 2: Generacion del itemset candidato

- For k=1, C_1 = all 1-itemsets.
- For k>1, generate C_k from L_{k-1} as follows:
  - *The join step*

Juntar $L_{k-1}=\{a_1, \ldots, a_{k-2}, a_{k-1}\}$ con $L_{k-1}=\{b_1, \ldots, b_{k-2}, b_{k-1}\}$ solo si $a_i=b_i$ y $a_{k-1}<b_{k-1}$. Entonces,

anadir $\{a_1, \ldots, a_{k-2}, a_{k-1}, b_{k-1}\}$ a C_k

(Los items deben mantenerse **en orden**).

  - *The prune step*

Remover $\{a_1, \ldots, a_{k-2}, a_{k-1}, a_k\}$ de C_k si contiene un subset non-frecuente de tamano k-1.

Ejemplo – Hallando los itemsets frecuentes

Dataset D

TID	Items
T100	a1 a3 a4
T200	a2 a3 a5
T300	a1 a2 a3 a5
T400	a2 a5

minSup=0.5

1. scan D ➔ C_1: a1:2, a2:3, a3:3, a4:1, a5:3

   ➔ L_1: a1:2, a2:3, a3:3, a5:3

   ➔ C_2: a1a2, a1a3, a1a5, a2a3, a2a5, a3a5

2. scan D ➔ C_2: a1a2:1, a1a3:2, a1a5:1, a2a3:2, a2a5:3, a3a5:2

   ➔ L_2: a1a3:2, a2a3:2, a2a5:3, a3a5:2

   ➔ C_3: a2a3a5

   ➔ Pruned C_3: a2a3a5 (all subsets belong to L_2)

3. scan D ➔ L_3: a2a3a5:2

El orden de los items puede afectar el proceso

Dataset D

TID	Items
T100	1 3 4
T200	2 3 5
T300	1 2 3 5
T400	2 5

minSup=0.5

1. scan D ➔ C_1: 1:2, 2:3, 3:3, 4:1, 5:3

   ➔ L_1: **1**:2, **2**:3, **3**:3, **5**:3

   ➔ C_2: 12, 13, 15, 23, 25, 35

2. scan D ➔ C_2: 12:1, **13:2**, 15:1, **23:2**, **25:3**, **35:2**

   Suppose the order of items is: 5,4,3,2,1

   ➔ L_2: **31**:2, **32**:2, **52:**3, **53**:2

   ➔ C_3: 321, 532

   ➔ Pruned C_3: 532

3. scan D ➔ L_3: 532:2

arules: R package for association rules

La Libreria arules de R haya las reglas de association de una base de datos transaccionales

El primer paso es convertir la base de datos dada una base de datos transaccional

Esto puede ser a partir de los datos originales usando una lista

```
data1=list(c("a1","a3","a4"),c("a2","a3","a5"),c("a1","a2","a3","a5"),c("a2","a5"))
names(data1)=paste("Tr",c(1:4), sep = "")
trans1=as(data1,"transactions")
```

O a partir de la matriz de datos binários

```
 data
   a1 a2 a3 a4 a5
t1  1  0  1  1  0
t2  0  1  1  0  1
t3  1  1  1  0  1
t4  0  1  0  0  1
trans2=as(data,"transactions")
```

arules: [2]

Luego se aplica el algoritmo apriori para encontra los itemsets frequent, usando

```
a=apriori(trans1,parameter=list(sup=0.5,target="frequent itemsets"))
summary(a) #da informacion acerca de los frequent itemsets
Inspect(a) #muestra los frequent itemsets
> inspect(a)
    items      support count
[1] {a1}       0.50    2
[2] {a2}       0.75    3
[3] {a5}       0.75    3
[4] {a3}       0.75    3
[5] {a1,a3}    0.50    2
[6] {a2,a5}    0.75    3
[7] {a2,a3}    0.50    2
[8] {a3,a5}    0.50    2
[9] {a2,a3,a5} 0.50    2
>
```

Hallando las reglas a partir de los frequent itemsets

Los Frequent itemsets son distintos de las reglas de asociacion. Se require un paso adicional para obtenerlas

For each frequent itemset *X*,

For each proper nonempty subset *A* of *X*,

Let *B* = X - *A*

A $\Rightarrow$B is an association rule if

Confianza (A $\Rightarrow$ B) ≥ minConf,

donde Confianza(A $\Rightarrow$ B) = support (AB) / support (A)

Example – derivando rules a partir de los frequent itemsets

- El itemset 235 es frequente, con supp=50%
  - Los subconjuntos propios no vacios son: 23, 25, 35, 2, 3, 5, con supp=50%, 75%, 50%, 75%, 75%, 75% respectivamente
  - La siguientes relaciones son candidatos a reglas de asociacion:
    - 23 => 5, confidence=100%
    - 25 => 3, confidence=67%
    - 35 => 2, confidence=100%
    - 2 => 35, confidence=67%
    - 3 => 25, confidence=67%
    - 5=> 23, confidence=67%

Example – derivando rules a partir de los frequent itemsets[2]

Usando la function apriori de arules, se tiene que;

```
ar=apriori(trans1,parameter=list(sup=0.5,conf=0.8,target="rules"))
 inspect(ar)
    lhs        rhs  support confidence lift     count
[1] {a1}    => {a3} 0.50    1          1.333333 2
[2] {a2}    => {a5} 0.75    1          1.333333 3
[3] {a5}    => {a2} 0.75    1          1.333333 3
[4] {a2,a3} => {a5} 0.50    1          1.333333 2
[5] {a3,a5} => {a2} 0.50    1          1.333333 2
```

Interpretation of ar1:100 % of transactiones que compran el item 1 tambien compran el item 3. 50% de las transacciones compran los dos articulos.

Derivando las reglas de asociacion

- Este paso no consume tanto tiempo como la generacion de los itemsets frecuentes.
- Se pude reducir la busqueda usando computacion en paralelo (particonando los datos)
- El algoritmo Frequent-Pattern Growth (FP-Tree, Han, 2001) considera que nos es necesario generar los itemset frecuentes para encontrar las reglas de asociacion

Otras mejoras que se puedan hacer

- Reducir el numero de transacciones, esd decir hacer una especie de seleccion de instancias.
- Reducir el numero de veces que se pasa sobre todos los datos.
- Reducir el numero de candidatos

Algoritmos para hallar reglas de asociacion

Depend on the data Representation

- Horizontal (Apriori)
- Vertical (Eclat, Zaki 2000)

  FP-Growth (Han et al., 2000)

  H-Mine (Pei et al., 2001)

R tiene una libreria **arules** que implementa los algoritmos Apriori y .

Reglas de asociacion versus clasificacion y clustering

- vs. clasificacion
  - El lado derecho puede tener cualquier numero de items.
  - Puede encontrar una clasificacion como regla $X \Rightarrow c$ de una manera distinta: la regla no es acerca de diferenciar clases sino de acerca de que X describe la clase c.
- vs. clustering
  - Reglas de asociacion no require las etiquetas de las clases.
  - Para $X \Rightarrow Y$, si Y es considerado como cluster, ertonces se pueden formar diferentes clase que tienen la misma descripcion (X).

Discusion de Support y Confidence

- Support y confidence no son suficientes para medir la importancia de reglas de asociacion.
- Si los thresholds de support y confidence aumentan ➔ solo se consiguen unas pocas reglas de asociacion y ellas problablemente no son importantes.
- Por el contrario se los hresholds de support y confidence son pequenos ➔ entonces se consiguen demasiadas reglas de asociacion

Resumen

- Reglas de asociacion son distintos a otros algoritmos de mineria de datos.
- La propiedad Apriori puede reducer el espacio de busqueda.
- Es complicado encontrar las reglas de asociacion que son largas.
- Reglas de asociaion tiene muchas aplicaciones.