Mineria de Datos

Reglas de Asociacion

Dr. Edgar Acuna
Departmento de Matematicas

Universidad de Puerto Rico- Mayaguez

math.uprrm.edu/~edgar

Datos transaccionales

Ejemplo de canasta de mercados:

Canasta1: {Pan, queso, Leche}

Canasta 2: {Manzana, huevos, sal, yogur}

…

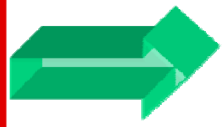
Canasta n: {biscuit, huevos, leches}

Definicions:

- Un *item*: un articulo en una canasta.
- Una *transaccion*: items comprados en una canasta; puede tener unTID (ID de la transaccion).
- Un conjunto de datos transaccional: Un conjunto de transacciones.

Representacionr binaria de datos transaccionales

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 y reglas de asociacion

- Un *itemset* es un conjunto de articulos.
  - E.g., {leche, pan, cereales} es un itemset.
- Un *k-itemset* es un itemset con k articulos.
- Dado un conjunto de datos *D*, un itemset *X* tiene una frecuencia de ocurrencia en D.
- El objetivo es encontrar itemsets que aparecen juntos en muchas transacciones.
- Una *Regla de Associacion* es una relacion entre dos itemsets disjuntos *X* y *Y*. *Asi,*

$$X \Rightarrow Y$$

representa el patron de cuando *X* ocurre, entonces *Y* tambien ocurre.

Uso de Reglas de Asociacion

- Reglas de asociacion no representan ningun tipo de causalidad o correlacion entre los dos itemsets.
  - $X \Rightarrow Y$ no implica que X causa Y.
  - $X \Rightarrow Y$ puede ser diferente a $Y \Rightarrow X$, lo que no ocurre en correlacion.
- Reglas de asociacion ayuda en marketing, publicidad, planificacion del arreglo de pisos en una tienda, control de inventario, seguridad nacional, e-comercio.

Soporte y Confianza

- ***soporte*** de un itemset X en D is *conteo*(X)/|D|.
- Para una regla de asociacion $X \Rightarrow Y$, *podemos calcular*
  - soporte($X \Rightarrow Y$) = soporte ($X \cup Y$)
  - confianza ($X \Rightarrow Y$) = soporte ($X \cup Y$)/soporte (X). Representa la fuerza de la implicacion.
- Soporte (S) y Confianza (C) estan relacionados a probabilidades conjuntas y condicionales.
- Podria haber un numero extremedamente grande de reglas de asociacion.
- Reglas de asociaciones fuertes son aquellas cuyas S y C son mayores o iguales a minSup y minConf (puntos de cortes establecidos de antemano)

Ejemplo

- El conjunto de datos *D*

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

Conteo, Soporte, Confianza:

$Conteo(1,3)=2$

$|D| = 4$

$Soporte(1,3)=0.5$

$Soporte(3 \rightarrow 2)=0.5$

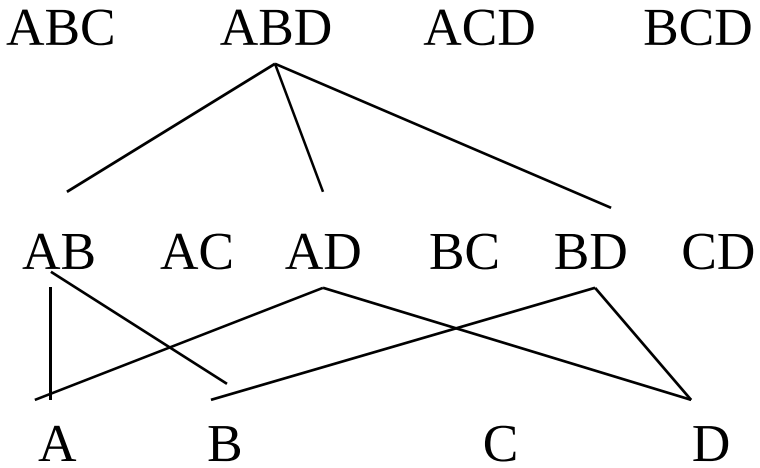
$Confianza(3 \rightarrow 2)=0.67$

- Pasos principales en mineria de reglas de asociacion.
  - Generacion de itemsets frecuentes. Encontrar itemsets que tienen un soporte S mayor o igual que un punto de corte pre-establecido.
  - Derivacion de reglas de asociacion. Usando los itemsets frecuentes obtenidos anteriormente se generan las reglas de asociacion que tienen una confianza C mayor o igual que un punto de corte pre-establecido.

  El rendimiento global de hacer mineria para hallar reglas de asociacion es determinado por el primer paso.

Itemsets frecuentes

- Un itemset frecuente es un itemset cuyo soporte S es ≥ minSup. Si el conjunto de datos tiene m articulos entonces hay 2^m itemsets frecuentes posibles.
- La propiedad Apriori: cualquier subconjunto de un itemset frecuente es tambien un itemset frecuente.



Usando la propiedad Apriori se puede podar ramas innecesarias, pues todos los superconjuntos de itemsets no frecuentes son tambien no frecuentes.

El algoritmo APRIORI (Agrawal et al., 1995)

1. Sean L_1: Los 1-itemsets frecuentes.
2. Para k=2, formar C_k (un k-itemset candidato) de L_{k-1}
3. Hallar el conjunto frecuente L_k de C_k entre todos los itemsets candidatos de tamano k. Hacer k = k + 1.
4. Repetir 2-3 hasta que C_k (y por lo tanto L_{k+1}) se vuelva vacio.
5. Output: La union de todos los L_K.

- En el paso 3, llamado el paso de generacion del itemset frecuente, *D* es escaneado y se cuenta cada itemset en C_k , si es mayor que minSup, es frecuente y se vuelve un miembro de L_k.

Paso 2: La generacion del itemset candidato C_k.

- Para k=1, C_1 = todos los 1-itemsets.
- Para k>1, generar C_k de L_{k-1} como sigue:
  - *El paso de unir*

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

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

(Los articulos se guardan sorteados).

  - *El paso de podar*

Remover $\{a_1, \ldots, a_{k-2}, a_{k-1}, a_k\}$ de C_k si contiene un subconjunto de tamano k-1 que no es frecuente.

Ejemplo – Encontrando itemsets frecuentes

Conjunto de datos 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

   ➔ C_3 podado: a2a3a5 (todos los subonjuntos pertenecen a L_2)

3. scan D ➔ L_3: a2a3a5:2

El orden de los articulos puede afectar el proceso

Conjunto de datos 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**

Suponer que el orden de los articulos es: 5,4,3,2,1

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

➔ C_3: 321, 532

➔ C_3 podado: 532

3. scan D ➔ L_3: 532:2

Programas para el algoritmo Apriori

Christian Borgelt ha escrito programas en C y en Java para el algortimo apriori. Estos programas corren el sistema operativo Windows
http://www.borgelt.net/src/apriori.exe. Una vez descargado el programa se corre asi

```
C:\>apriori -ts -s50 ardata.txt arout
apriori - find association rules with the apriori algorithm
version 4.26 (2005.01.31)        (c) 1996-2005   Christian Borgelt
reading ardata.txt ... [5 item(s), 4 transaction(s)] done [0.00s].
……..
C:\>type arout
1  (50.0%)
2  (75.0%)
5  (75.0%)
3  (75.0%)
1 3  (50.0%)
2 5  (75.0%)
2 3  (50.0%)
5 3  (50.0%)
2 5 3  (50.0%)
```

Resultan 9 itemsets frecuentes

La libreria arules

Contiene funciones `para manipular y analizar datos transaccionales mediante la determinacion de los itemsets frecuentes y reglas de asociacion.

Se basa en los algoritmos apriori y Eclat de M. Zaki, programados pro Borgelt.

Para los datos del ejemplo habria que hacer los siguiente

```
> ardata=list(c(1, 3, 4),c(2, 3, 5),c(1,2,3,5),c(2,5))
> b=as(ardata,"transactions")
> itemfre=apriori(b,parameter = list(supp = 0.5,target="frequent itemsets"))

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen
        0.8    0.1    1 none FALSE            TRUE     0.5      1      5
           target   ext
 frequent itemsets FALSE

algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
```

La libreria arules (cont)

```
apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[5 item(s), 4 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [9 set(s)] done [0.00s].
creating S4 object  ... done [0.00s].
> summary(itemfre)
set of 9 itemsets
most frequent items:
      3       2       5       1       4 (Other)
      5       4       4       2       0       0
element (itemset/transaction) length distribution:sizes
1 2 3
4 4 1
```

La libreria arules (cont)

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.667   2.000   3.000

summary of quality measures:
    support
 Min.   :0.5000
 1st Qu.:0.5000
 Median :0.5000
 Mean   :0.6111
 3rd Qu.:0.7500
 Max.   :0.7500

includes transaction ID lists: FALSE
mining info:
 data ntransactions support confidence
    b            4     0.5          1
```

Derivando reglas de asociacion a partir de itemsets frecuentes

Los itemsets frecuentes son diferentes que reglas de asociacion. Se requiere un paso adicional para hallar las reglas de asociacion.

Para cada itemset frecuente *X*,

Para cada subconjunto propio no vacio A de *X*,

Sea *B* = X - *A*

A $\Rightarrow$B es una regla de asociacion si

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

donde soporte (A $\Rightarrow$ B) = soporte (AB), y

confianza(A $\Rightarrow$ B) = soporte (AB) / soporte (A)

Ejemplo – Derivando reglas de asociacion a partir de itemsets frecuentes

- Suponga que 235 es frecuente, con soporte=50%
  - Subconjuntos propios no vacios son: 23, 25, 35, 2, 3, 5, con soporte=50%, 75%, 50%, 75%, 75%, 75% respectivamente
  - Estos generan estas reglas de asociacion:
    - 23 => 5, confianza=100%
    - 25 => 3, confianza=67%
    - 35 => 2, confianza=100%
    - 2 => 35, confianza=67%
    - 3 => 25, confianza=67%
    - 5=> 23, confianza=67%

    Todas las reglas tienen soporte = 50%

El programa apriori

```
C:\>apriori -tr -s50 ardata.txt arout
apriori - find association rules with the apriori algorithm
version 4.26 (2005.01.31)        (c) 1996-2005   Christian Borgelt
reading ardata.txt ... [5 item(s), 4 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing arout ... [5 rule(s)] done [0.00s].

C:\>type arout
3 <- 1  (50.0, 100.0)
5 <- 2  (75.0, 100.0)
2 <- 5  (75.0, 100.0)
5 <- 2 3  (50.0, 100.0)
2 <- 5 3  (50.0, 100.0)
```

Interpretation of ar1:100 % of transactions that purchase item 1 also purchase item 3. 50% of transactions contain both items.

Usando la libreria arules

```
> ar=apriori(b,parameter = list(supp = 0.5, conf = 0.8,target="rules"))
parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target
      ext
       0.8    0.1    1 none FALSE            TRUE     0.5      1      5  rules FALSE

algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[5 item(s), 4 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [5 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Usando la libreria arules

```
> summary(ar)
set of 5 rules
rule length distribution (lhs + rhs):sizes
2 3
3 2
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    2.0     2.0     2.0     2.4     3.0     3.0
summary of quality measures:
    support       confidence      lift
 Min.   :0.50   Min.   :1    Min.   :1.333
 1st Qu.:0.50   1st Qu.:1    1st Qu.:1.333
 Median :0.50   Median :1    Median :1.333
 Mean   :0.60   Mean   :1    Mean   :1.333
 3rd Qu.:0.75   3rd Qu.:1    3rd Qu.:1.333
 Max.   :0.75   Max.   :1    Max.   :1.333
mining info:
 data ntransactions support confidence
    b            4     0.5        0.8
```

Derivando las reglas de asociacion

- Este paso no consume tanto tiempo como la generacion de los itemsets frecuentes.
- Se puede tambien acelerar usando tecnicas de computacion paralelo.
- Es la generacion de los itemsets frecuentes realmente un paso necesario para derivar reglas de asociacion?
  - Frequent-Pattern Growth (FP-Tree, Han, 2001)

Mejora de la eficiencia

- Como podemos mejorar la eficiencia ?

  Podando sin chequear todos los subconjuntos de tamano k - 1 ?

  Juntando y podando sin dar vueltas sobre todo el L_{k-1}?.

Usar hash trees.

Mejoras adicionales

- Acelerar la busqueda.
- Reducir el numero de transacciones (similar a hacer seleccion de casos)
- Reducir el numero de pasadas sobre los datos en el disco
- Reducir el numero de subconjuntos por transaccion a ser considerados.
- Reducir el numero de candidatos

Trabajando en las transacciones

- Remover las transacciones que no contienen k-itemsets frecuentes en cada escaneado.
- Remover de las transacciones aquellos articulos que no son miembros de algun k-itemset candidato.
  - Por ejemplo, si 12, 24, 14 son los unicos itemsets candidados contenidos en 1234, entonces el articulo 3 puede ser removido.
  - Si12, 24 son los unicos itemsets candidatos contenidos en la transaccion 1234, entonces remover la transaccion de la siguiente etapa de escaneado.
- La reduccion del tamano de los datos produce menos tiempo de procesamiento y lectura pero anade tiempo de escribir.

Reduciendo Scans via Particionamiento

- Dividir el conjunto de datos D en *m* partes, D1, D2,…, Dm, de tal manera que cada porcion pueda entrar en memoria.
- Hallar los itemsets frecuentes F_i in D_i, con soporte ≥ minSup, para cada *i*.
  - *Si un itemset es frecuente en D, entonces debe ser frecuente en algun D_i.*
- La union de todos los F_i forma un conjunto candidato de itemsets frecuentes en D; contar cuantos son.
- A menudo esto requiere solo dos escaneos de D.

Algoritmos para reglas de asociacion

Dependen de la representacion de los datos

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

  FP-Growth (Han et al., 2000)

  H-Mine (Pei et al., 2001)

Aspectos unicos de Reglas de Asociacion

- vs. clasificacion supervisada
  - El lado derecho puede tener cualquier cantidad de articulos.
  - Puede encontrar una clasificacion como una regla X $\Rightarrow$ *c* en una manera diferente: dicha regla no es acerca de diferenciar entre clases, si no acerca de que (X) describe la clase *c*
- vs. clustering
  - No asigna observaciones en grupos.
  - Para X $\Rightarrow$ Y, si Y es considerado como un cluster, entonces puede formar diferentes clusters compartiendo la misma descripcion (X).

Resumen

- Reglas de Asociacion son distintos de otro algoritmos de mineria de datos.
- La propiedad Apriori puede reducir el espacio de busqueda.
- La busqueda de reglas de asociacion es una tarea intimidante.
- Reglas de Associacion tienen muchas aplicaciones.
- Itemsets frequentes son un concepto util en la practica.