Association

Oran Kittithreerapronchai[1]

[1]Department of Industrial Engineering, Chulalongkorn University
Bangkok 10330 THAILAND

last updated: November 2, 2019

Outline

1. Association Theory
2. Titanic Survival
3. Workshop Association

Association Task

Definition

Given a set of transactions, find rules that predict the occurrence of items based on the transaction

# Trans	Items	Example of Association
1	{Bread, Milk}	{Diaper} → {Beer }
2	{Bread, Diaper, Beer, Eggs}	{Milk, Bread} → {Eggs, Coke }
3	{Milk, Diaper, Beer, Coke}	{Beer, Bread} → {Milk }
4	{Bread, Milk, Diaper, Beer}	
5	{Bread, Milk, Diaper, Coke}	

Terminology

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Itemset:** a collection of one or more items, e.g. {Milk, Bread, Diaper}
- *k*-**itemset:** an itemset that contains *k* items
- **Association Rule:** an implication of itemset $X \rightarrow Y$, e.g., {Milk, Diaper} $\rightarrow$ {Beer}
- **Support count** (σ): frequency of occurrence of an itemset, e.g, σ({Milk, Bread,Diaper}) = 2

Mathematics of Association Rule

- **Support** (s): fraction of transactions that contain an itemset.,e.g., $s(\{Milk, Diaper\} \rightarrow \{Beer\}) = \frac{2}{5}$

$$s(A \rightarrow B) = P(A \cup B)$$

- **Confidence** (c): how often items in Y appear in transactions that contain X, e.g., $c(\{Milk, Diaper\} \rightarrow \{Beer\}) = \frac{\sigma(\{\text{Milk, Diaper,Beer}\})}{\sigma(\{\text{Milk, Diaper}\})} = 2/3$

$$c(A \rightarrow B) = P(B|A)$$

- **Lift** (l): performance of a targeting association rule at predicting cases, e.g., $l(\{Milk, Diaper\} \rightarrow \{Beer\}) = \frac{2}{5}/\frac{3}{5}\frac{3}{5} = \frac{10}{9}$

$$l(A \rightarrow B) = \frac{P(A \cup B)}{P(A)P(B)}$$

Support and Confidence

# Trans	Items
1	{Bread, Milk}
2	{Bread, Diaper, Beer, Eggs}
3	{Milk, Diaper, Beer, Coke}
4	{Bread, Milk, Diaper, Beer}
5	{Bread, Milk, Diaper, Coke}

Assoc. Rules	$s(\cdot)$	$c(\cdot)$	$l(\cdot)$
	0.4	0.67	1.11
	0.4	0.50	1.25
	0.4	0.50	0.83
	0.4	0.67	1.11
	0.4	1.00	1.25
	0.4	0.67	0.83

Observation

All rules are based on {Milk, Diaper,Beer} with same support, but difference confidence

Support and Confidence

# Trans	Items
1	{Bread, Milk}
2	{Bread, Diaper, Beer, Eggs}
3	{Milk, Diaper, Beer, Coke}
4	{Bread, Milk, Diaper, Beer}
5	{Bread, Milk, Diaper, Coke}

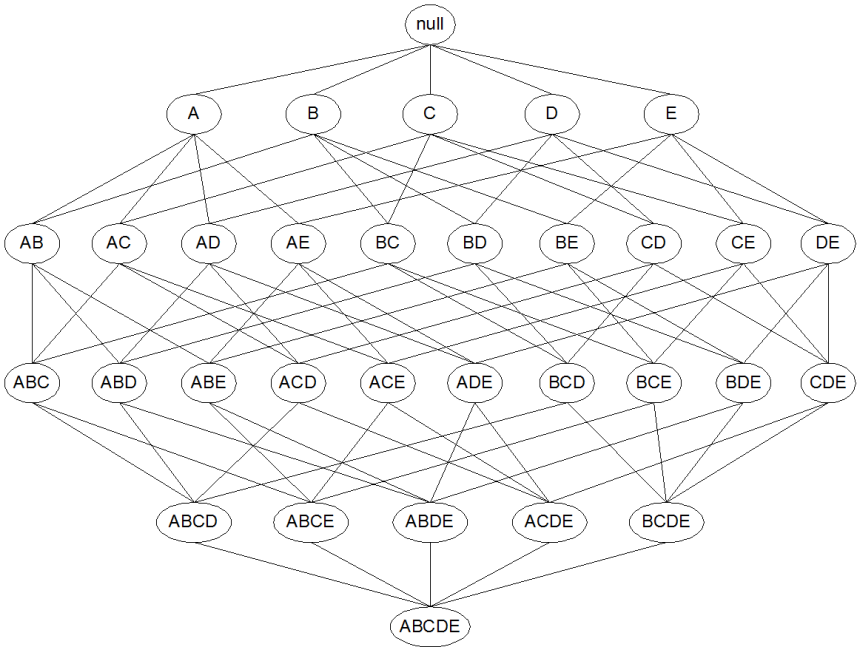
Assoc. Rules	$s(\cdot)$	$c(\cdot)$	$l(\cdot)$
{Beer} $\rightarrow$ {Milk,Diaper}	0.4	0.67	1.11
{Diaper} $\rightarrow$ {Milk,Beer}	0.4	0.50	1.25
{Milk} $\rightarrow$ {Diaper,Beer}	0.4	0.50	0.83
{Milk,Diaper} $\rightarrow$ {Beer}	0.4	0.67	1.11
{Milk,Beer} $\rightarrow$ {Diaper}	0.4	1.00	1.25
{Diaper,Beer} $\rightarrow$ {Milk}	0.4	0.67	0.83

Observation

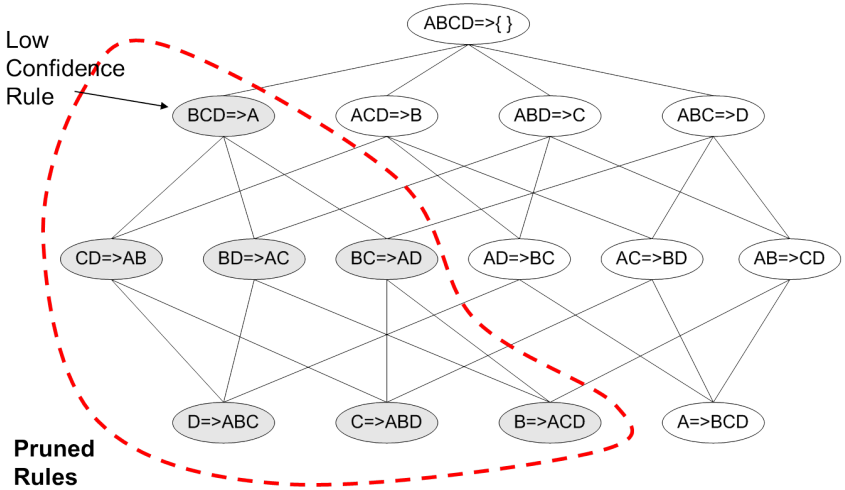
All rules are based on {Milk, Diaper,Beer} with same support, but difference confidence

Processes in Association

- **Frequent Generation:** list all itemsets whose $s(\cdot) \geq \underline{S}$
- **Rule Generation:** generate high confidence rules that exceeds $\underline{C}$ from frequent itemsets



Frequent Generation



Rule Generation

Frequency Generation

Given transaction $\mathbb{T}$, find all non-empty itemset $L \subset \mathbb{T}$ satisfies the minimum support $\underline{S}$

- **Brute Force:** check all combination, 2^M
- **Apriori:** check a lower 'cardinal' before higher combination
- **ECLAT:** separate item and count transaction ID

Apriori Algorithm

given minimum support

if counts of subset exceed minimum support **then**

extend a subset by adding an available element

else

ignore a subset

Example of Apriori at Minimal Support = 3

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

level	item	count	
1	{Bread}	4	
	{Coke}	2	ignore
	{Milk}	4	
	{Beer}	3	
	{Diaper}	4	
	{Eggs}	1	ignore
2	{Bread,Milk}	3	
	{Bread,Beer}	2	ignore
	{Bread,Diaper}	3	
	{Milk,Beer}	2	ignore
	{Milk,Diaper}	3	
	{Beer,Diaper}	3	
3	{Bread,Milk,Diaper}	3	

Example of ECLAT

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Bread	Coke	Milk	Beer	Diaper	Eggs
1	3	1	2	2	2
2	5	3	2	3	
4		4	4	4	
5		5		5	

- k-**itemset:** compute by $k-1$ itemset
- **Advantage:** very fast support counting
- **Disadvantage:** temporary TID-lists may become too large for memory

Rule Generation

Given itemset L, find all non-empty subsets $\{f\} \rightarrow \{l\}$where $\{f, l\} = L$ satisfies the minimum confidence $\underline{C}$

- **Example:** rule generation of $\{A, B, C, D\}$ are $\{A\} \rightarrow \{B, C, D\}$, $\{A, B\} \rightarrow \{C, D\}$, $\{A, B, C\} \rightarrow \{D\}$, $\{B\} \rightarrow \{A, C, D\}$, $\{B, C\} \rightarrow \{A, D\}$ ... $\{D\} \rightarrow \{A, C, D\}$
- **In general:** there are $2^{|L|} - 2$ # why?
- **How to efficiently generate rules?**
  - **Non-Monotonic of group:** $c(ABC \rightarrow D) \not\geq c(AB \rightarrow D)$
  - **Monotonic of series** $c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$

Titanic Survival

- **Data:**

```
head(Titanic)
titanic.df   <- as.data.frame(Titanic)
titanic.tabl <- titanic.df[titanic.df$Freq>0,]
nClass       <- nrow(titanic.tabl)
titanic.raw  <- NULL

for(i in 1:nClass){
  tempDF      <- titanic.tabl[i,1:4]
  tempFreq    <- titanic.tabl[i,5]
  titanic.raw <- rbind(titanic.raw
                       ,do.call("rbind", replicate(tempFreq,tempDF , simplify = FALSE)))
}
dim(titanic.raw)
```

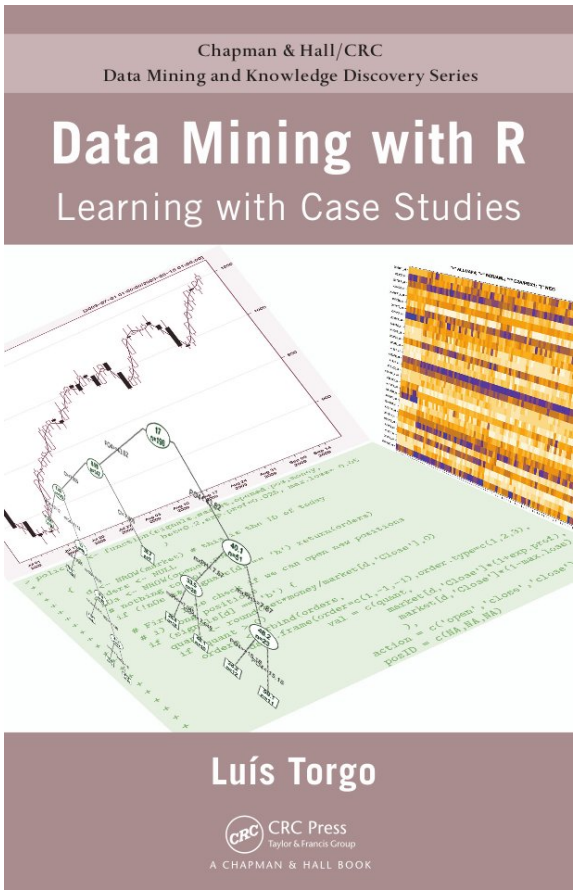
- **Rule Gen.**

```
require(arules)
rules.all  <- apriori(titanic.raw)
inspect(rules.all)
rules.cust <- apriori(titanic.raw,control  =  list(verbose=F)
                      ,parameter= list(minlen=2, supp=0.005, conf=0.8)
                      ,appearance  =  list(rhs=c("Survived=No","Survived=Yes"),default="lhs"))
rules.sorted  <- sort(rules.cust,  by="lift")
inspect(rules.sorted)
```

- **Visualize**

```
require(arulesViz)
plot(rules.cust)
plot(rules.all,  method  =  "grouped")
plot(rules.sorted,  method  =  "graph")
```

TBA



- **Package:** 'DMwR'
- **Code:** rScript
- **Instruction:** step-by-step follow up
- **Goals:**
  -

Transaction Analysis with Groceries

- **Greceries:**

```
require(arules)
data("Groceries")
inspect(head(Groceries))

sort(itemFrequency(Groceries),T)[1:20]
itemFrequencyPlot(Groceries,topN=20,type="absolute")
itemFrequencyPlot(Groceries,topN=20,type="relative")
```

- **Rule Gen.**

```
grocery.rules <- apriori(Groceries,parameter = list(supp=0.001,conf=0.8))
options(digits = 3)
inspect(grocery.rules[1:5])
```

- **Gen Trans**

```
set.seed(101)
orders <- data.frame(
  transactionID = sample(1:500, 1000, replace=T),
  item = paste("item", sample(1:50, 1000, replace=T),sep = "")
)
orders <- unique(orders)


orders.temp <- as.matrix(xtabs(~transactionID+item,data=orders))
head(orders.temp)
orders.mat <- matrix(NA,nrow=dim(orders.temp)[1],ncol=dim(orders.temp)[2],)
orders.mat[orders.temp == 1] <- T
orders.mat[orders.temp == 0] <- F
dimnames(orders.mat) <- dimnames(orders.temp)
head(orders.mat)
```