Association rule mining

Luciano Serafini

Fondazione Bruno Kessler

May 6, 2019

Association rules

- Transaction data
- Market basket analysis
- Produce

$$\{Cheese, Milk\} \rightarrow \{Bread\}[sup = 5\%, conf = 80\%]$$

- Association rule: 80% of customers who buy cheese and milk also buy bread. This is supported by 5% of customers who buy all these products together

Association rule mining

- Proposed by Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. "Mining association rules between sets of items in large databases". In: *Acm sigmod record*. Vol. 22. 2. ACM. 1993, pp. 207–216
- It is an important data mining model studied extensively by the database and data mining community.
- Assume all data are categorical.
- Initially used for Market Basket Analysis to find how items purchased by customers are related.

Example (Association rule)

Looking at the fidelity cards, we notice that 95% of the people who buy milk and caffe, also buy biscuits, and this is supported by 5% of the entries of our database. In symbols:

$$milk, caffee \rightarrow biscuits \quad confidence = 98\%, \quad support = 5\%$$

Association Rules

- let Ω be a finite non empty set of items, denoted by $\{a, b, c, \ldots\}$;
- An itemset X over Ω, is a subset of Ω
- A transaction database $\mathcal{D}$ is a sequence $X_1, \ldots, X_{|\mathcal{D}|}$ of itemsets, called transactions of $\mathcal{D}$;
- The cover of and itemset X in $\mathcal{D}$, denoted by $C(X, \mathcal{D})$

$$\{i \in \mathbb{N} \mid X \subseteq X_i \in \mathcal{D}\}$$

- The support of X in $\mathcal{D}$, denoted $S(X, \mathcal{D})$, is the cardinality of $C(X, \mathcal{D})$,

Example

Tid	*Itemsets*						
1	a	b	c	d			
2	a	b			e	f	
3	a	b	c				
4	a		c	d		f	
5							g
6				d			
7				d			g

- $C(\{a, b\}, \mathcal{D}) = \{1, 2, 3\}$;
- $S(\{a, b\}, \mathcal{D}) = 3$;

Association Rule

Definition

An association rule is a pattern of the form

$$X \rightarrow Y$$

where X (called the antecedent) and Y (called the consequent) are two disjoint itemsets.

- The support of an association rule $X \rightarrow Y$ in a transaction database $\mathcal{D}$ is equal to

$$S(X \rightarrow Y, \mathcal{D}) = \frac{S(X \cup Y, \mathcal{D})}{|\mathcal{D}|}$$

- The confidence of $X \rightarrow Y$ in $\mathcal{D}$ is equal to

$$Conf(X \rightarrow Y, \mathcal{D}) = \frac{S(X \cup Y, \mathcal{D})}{S(X, \mathcal{D})}$$

Definition

Given a transaction database $\mathcal{D}$, a minimum support threshold $0 < \alpha \leq 1$ and a minimum confidence threshold $0 < \beta \leq 1$, the problem of mining association rules consists in computing the the $MAR(\mathcal{D}, \alpha, \beta)$ equal to:

$$\{X \to Y \mid S(X \to Y, D) \geq \alpha \text{ and } Conf(X \to Y, \mathcal{D}) \geq \beta\}$$

An example

Example

1	Beef, Chicken, Milk
2	Beef, Cheese
3	Cheese, Boots
4	Beef, Chicken, Cheese
5	Beef, Chicken, Clothes, Cheese, Milk
6	Chicken, Clothes, Milk
7	Chicken, Clothes, Milk

	Beef	Chicken	Clothes	Cheese	Milk	Boots
1	1	1	0	0	1	0
2	1	0	0	1	0	0
3	0	0	0	1	0	1
4	1	1	0	1	0	0
5	1	1	1	1	1	0
6	0	1	1	0	1	0
7	0	1	1	0	1	0

Example

Example

- Assume:
  - minsup $= 30\%$
  - minconf $= 80\%$

$\mathcal{D} =$

1	Beef, Chicken, Milk
2	Beef, Cheese
3	Cheese, Boots
4	Beef, Chicken, Cheese
5	Beef, Chicken,Clothes, Cheese, Milk
6	Chicken, Clothes, Milk
7	Chicken, Clothes, Milk

Example

Example

- Assume:
  - minsup $= 30\%$
  - minconf $= 80\%$

$\mathcal{D} =$

1	Beef, Chicken, Milk
2	Beef, Cheese
3	Cheese, Boots
4	Beef, Chicken, Cheese
5	Beef, Chicken,Clothes, Cheese, Milk
6	Chicken, Clothes, Milk
7	Chicken, Clothes, Milk

- An example frequent itemset:

$$Chicken, Clothes, Milk \qquad [sup = 3/7]$$

Example

Example

- Assume:
  - minsup $= 30\%$
  - minconf $= 80\%$

$\mathcal{D} =$

1	Beef, Chicken, Milk
2	Beef, Cheese
3	Cheese, Boots
4	Beef, Chicken, Cheese
5	Beef, Chicken,Clothes, Cheese, Milk
6	Chicken, Clothes, Milk
7	Chicken, Clothes, Milk

- An example frequent itemset:

$$\begin{array}{rr} Chicken, Clothes, Milk & [sup = 3/7] \\ Chicken, Milk & [sup = 4/7] \end{array}$$

Example

Example

- Assume:
  - minsup $= 30\%$
  - minconf $= 80\%$

$\mathcal{D} =$

1	Beef, Chicken, Milk
2	Beef, Cheese
3	Cheese, Boots
4	Beef, Chicken, Cheese
5	Beef, Chicken,Clothes, Cheese, Milk
6	Chicken, Clothes, Milk
7	Chicken, Clothes, Milk

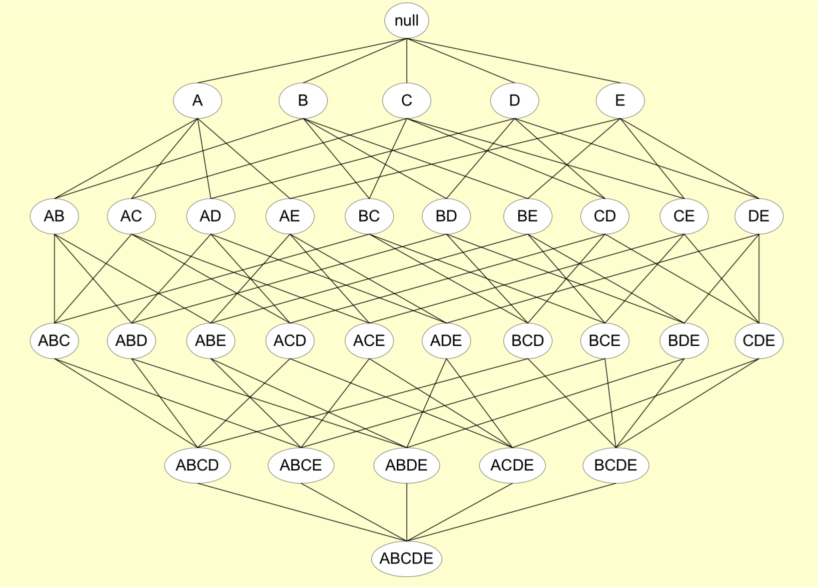
- An example frequent itemset:

$$
\begin{aligned}
Chicken, Clothes, Milk & \qquad [sup = 3/7] \\
Chicken, Milk & \qquad [sup = 4/7]
\end{aligned}
$$

- Association rules from the itemsets:

$$
\begin{aligned}
Clothes \to Milk, Chicken & \qquad [sup = 3/7, conf = 3/3] \\
Clothes, Chicken \to Milk & \qquad [sup = 3/7, conf = 3/3] \\
Chicken \to Milk & \qquad [sup = 4/7, conf = 4/5] \\
Milk \to Chicken & \qquad [sup = 4/7, conf = 5/5]
\end{aligned}
$$

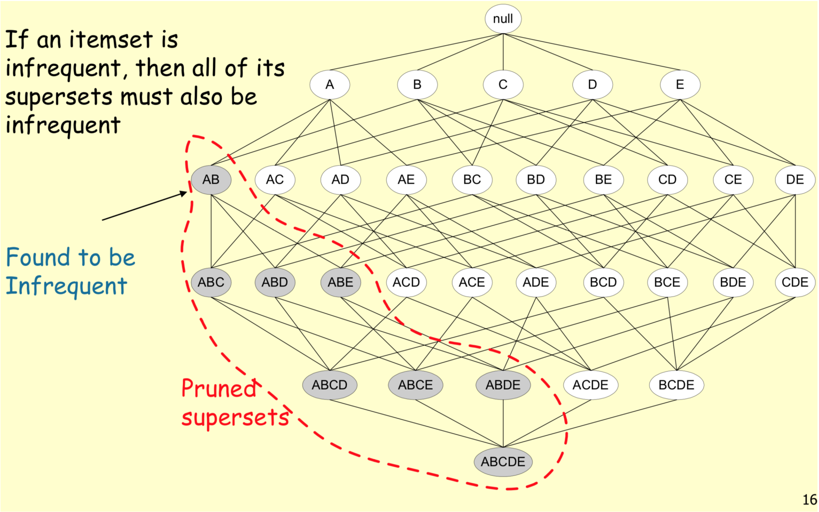
Itemset Lattice



The Downward Closure Property

- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If $\{\mathbf{chicken}, \mathbf{clothes}, \mathbf{milk}\}$ is frequent, so is $\{\mathbf{chicken}, \mathbf{milk}\}$
  - every transaction having $\{chicken, clothes, milk\}$ also contains $\{chicken, milk\}$.

Pruning Itemset Lattice



The Apriori Algorithm

- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested!
- Method:
  - Initially, scan DB once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Test the candidates against DB
  - Terminate when no frequent or candidate set can be generated

The Apriori algorithm

Example (Supp = 0.5)

ACDFG
ABCDF
CDE
ADF
ACDEF
BCDEFG

A	4
B	1
C	5
D	6
E	3
F	5
G	2

The Apriori algorithm

Example (Supp = 0.5)

ACDFG
ABCDF
CDE
ADF
ACDEF
BCDEFG

A	4
B	1
C	5
D	6
E	3
F	5
G	2

$\Rightarrow$

A	4
C	5
D	6
E	3
F	5

The Apriori algorithm

Example (Supp = 0.5)

ACDFG
ABCDF
CDE
ADF
ACDEF
BCDEFG

A	4
B	1
C	5
D	6
E	3
F	5
G	2

$\Rightarrow$

A	4
C	5
D	6
E	3
F	5

$\Rightarrow$

AC	3
AD	4
AE	1
AF	4
CD	5
CE	3
CF	4
DE	3
DF	4
EF	2

The Apriori algorithm

Example (Supp = 0.5)

Transactions
ACDFG
ABCDF
CDE
ADF
ACDEF
BCDEFG

A	4
B	1
C	5
D	6
E	3
F	5
G	2

$\Rightarrow$

A	4
C	5
D	6
E	3
F	5

$\Rightarrow$

AC	3
AD	4
AE	1
AF	4
CD	5
CE	3
CF	4
DE	3
DF	4
EF	2

$\Rightarrow$

AC	3
AD	4
AF	4
CD	5
CE	3
CF	4
DE	3
DF	4

The Apriori algorithm

Example (Supp = 0.5)

Transactions
ACDFG
ABCDF
CDE
ADF
ACDEF
BCDEFG

A	4
B	1
C	5
D	6
E	3
F	5
G	2

$\Rightarrow$

A	4
C	5
D	6
E	3
F	5

$\Rightarrow$

AC	3
AD	4
AE	1
AF	4
CD	5
CE	3
CF	4
DE	3
DF	4
EF	2

$\Rightarrow$

AC	3
AD	4
AF	4
CD	5
CE	3
CF	4
DE	3
DF	4

$\Rightarrow$

ACD	3
ACF	3
ADF	4
CDE	3
CDF	4

The Apriori algorithm

Example (Supp = 0.5)

ACDFG
ABCDF
CDE
ADF
ACDEF
BCDEFG

A	4
B	1
C	5
D	6
E	3
F	5
G	2

$\Rightarrow$

A	4
C	5
D	6
E	3
F	5

$\Rightarrow$

AC	3
AD	4
AE	1
AF	4
CD	5
CE	3
CF	4
DE	3
DF	4
EF	2

$\Rightarrow$

AC	3
AD	4
AF	4
CD	5
CE	3
CF	4
DE	3
DF	4

$\Rightarrow$

ACD	3
ACF	3
ADF	4
CDE	3
CDF	4

$\Rightarrow$

ACDF	3

The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k
L_k: frequent itemset of size k

Algorithm 1 Apriori($\mathcal{D}$,min-support)

```
L_1 ← frequent items;
for (k = 1; L_k ≠ ∅) do
    C_{k+1} ← candidates generated from L_k, Ł_{k−1}, ..., L_1;
    for t ∈ D do
        increment the count of all candidates in C_{k+1} that are contained in t
        L_{k+1} ← candidates in C_{k+1} with min-support
    end for
end for
return ⋃_k L_k
```

Candidate generation methods

- Brute-force: examine all possible itemsets candidates and then delete useless. **weakness:** too many candidates
- Get C_k from $L_{k-1} \times L_1$: extend frequent $(k-1)$-itemsets found from previous step with the frequent 1-itemsets. **weakness:** it is possible to create copies of candidate itemsets, e.g.

$$\{a, b\} \cup \{c\} \rightarrow \{a, b, c\}$$
$$\{b, c\} \cup \{a\} \rightarrow \{a, b, c\}$$

A possible solution: ordering of items

- The items are sorted in lexicographic order (which is a total order).
- $\{w_1, w_2, ..., w_k\}$: a k-itemset w, where $w_1 < w_2 < \cdots < w_k$ according to the total lexicographic order.
- Then, $(k-1)$-itemsets can be extended only to items which are lexicographically higher than its own, e.g.

$$\begin{array}{lr} \{a, b\} \cup c \rightarrow \{a, b, c\} & \text{is allowed} \\ \{a, c\} \cup b \rightarrow \{a, b, c\} & \text{is not allowed} \end{array}$$

- **weakness:** numerous of possible useless candidates itemsets, e.g. merge $\{a, b\}$ and $\{c\}$ is useless if the itemset $\{a, c\}$ is not frequent.

A possible solution

- A-priori candidate generation based on $L_{k-1} \times L_{k-1}$ method:
- Examines all possible pairs of $k-1$-itemsets in L_{k-1} and keeps only those having $k-2$ common items.

$$\{w_1, \ldots, w_{k-1}\} \cup \{w_2, \ldots, w_k\} \rightarrow \{w_1, \ldots, w_k\}$$

- Example (assume lexicographic order):

$$L_2 = \{\{a, c\}, \{b, c\}, \{b, d\}, \{c, d\}\}$$

then candidate

$$C_3 \subseteq \{\{a, b, c\}, \{a, c, d\}, \{b, c, d\}\}$$

however, $\{a, b, c\}$ and $\{a, c, d\}$ can be discarded since $\{a, b\} \notin L_2$ and $\{a, d\} \notin L_2$, respectively. Therefore

$$C_3 \subseteq \{\{b, c, d\}\}$$

A-priori candidate generation

Two steps:

Join step: Generate all possible candidate itemsets C_k of length k from L_{k-1}

prune step: Remove those candidates from C_k that cannot be frequent because some subset is not in L_{k-1}.

Mining association rules

Frequent itemsets $\neq$ association rules

- Every k-itemset Y generates $2 \cdot k - 2$ possible rules.

$X \rightarrow Y \setminus X$	For every non empty proper subset of Y (i.e., $X \neq Y$ and $X \neq \emptyset$)

- Important: every possible rule satisfies the minsup threshold.

$$\frac{|\{Y \in T\}|}{|\{X \in T\}|} \geq \mathit{minconf}$$

A SAT-Based Approach for Mining Association Rules

Abdelhamid Boudane et al. "A SAT-based approach for mining association rules". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press. 2016, pp. 2472–2478

Logic Based Approach

- in the following we describe a SAT encoding for the problem of mining a generic association rule $X \rightarrow Y$
- our objective is to define a theory (= set of formulas) $\mathcal{E}(\mathcal{D}, \alpha, \beta)$ for every $\mathcal{D}$, α and β, such that,
- if $\mathcal{E}(\mathcal{D}, \alpha, \beta)$ is satisfied by an interpretation $\mathcal{I}$ then from $\mathcal{I}$ we can extract an association rule $X \rightarrow Y$ that belongs to $MAR(\mathcal{D}, \alpha, \beta)$.

In other words, the problem of searching for an association rule $X \rightarrow Y$ in a database $\mathcal{D}$ with minimum support α and minimum confidence threshold β is translated in the problem of checking the satisfiability of $\mathcal{E}(\mathcal{D}, \alpha, \beta)$.

The set of propositional variables

the set $\mathcal{P}$ of propositional variables with the respective informal interpretations are the following:

- for all $a \in \Omega$, x_a and y_a
  - x_a means a appears in X (the premise of $X \rightarrow Y$)
  - y_a means a appears in Y (the consequent of $X \rightarrow Y$)
- for $i \in \{1, \ldots, |\mathcal{D}|\}$ p_i and q_i
  - p_i means $i \in C(X, \mathcal{D})$, i.e., the i-th translation is in the cover set of X;
  - q_i means $i \in C(X \cup Y, \mathcal{D})$, i.e., the i-th translation is in the cover set of $X \cup Y$;

Encoding definitions and constraints in propositional formulas

- X and Y are disjoint

$$\bigwedge_{a \in \Omega} x_a \rightarrow \neg y_a \tag{1}$$

Encoding definitions and constraints in propositional formulas

- X and Y are disjoint

$$\bigwedge_{a \in \Omega} x_a \rightarrow \neg y_a \tag{1}$$

- the i-th transaction is in the cover of X;

$$\bigwedge_{i=1}^{|\mathcal{D}|} \left(p_i \equiv \bigwedge_{a \notin X_i} \neg x_a \right) \tag{2}$$

Encoding definitions and constraints in propositional formulas

- X and Y are disjoint

$$\bigwedge_{a \in \Omega} x_a \rightarrow \neg y_a \tag{1}$$

- the i-th transaction is in the cover of X;

$$\bigwedge_{i=1}^{|\mathcal{D}|} \left(p_i \equiv \bigwedge_{a \notin X_i} \neg x_a \right) \tag{2}$$

- the i-th transaction is in the cover of $X \cup Y$;

$$\bigwedge_{i=1}^{|\mathcal{D}|} \left(q_i \equiv p_i \wedge \bigwedge_{a \notin X_i} \neg y_a \right) \tag{3}$$

Example

Tid	*Itemsets*	$p_i \equiv \ldots$	$q_i \equiv p_i \wedge \ldots$
1	$a\ b\ c\ d$		
2	$a\ b\ e\ f$		
3	$a\ b\ c$		
4	$a\ c\ d\ f$		
5	g		
6	d		
7	$d\ g$		

Example

Tid	*Itemsets*	$p_i \equiv \ldots$	$q_i \equiv p_i \wedge \ldots$
1	$a\ b\ c\ d$	$\neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_e \wedge \neg y_f \wedge \neg y_g$
2	$a\ b\ e\ f$		
3	$a\ b\ c$		
4	$a\ c\ d\ f$		
5	g		
6	d		
7	$d\ g$		

Example

Tid	*Itemsets*	$p_i \equiv \dots$	$q_i \equiv p_i \wedge \dots$
1	$a\ b\ c\ d$	$\neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_e \wedge \neg y_f \wedge \neg y_g$
2	$a\ b\ e\ f$	$\neg x_c \wedge \neg x_d \wedge \neg x_g$	$\neg y_c \wedge \neg y_d \wedge \neg y_g$
3	$a\ b\ c$		
4	$a\ c\ d\ f$		
5	g		
6	d		
7	$d\ g$		

Example

Tid	Itemsets	$p_i \equiv \ldots$	$q_i \equiv p_i \wedge \ldots$
1	a b c d	$\neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_e \wedge \neg y_f \wedge \neg y_g$
2	a b e f	$\neg x_c \wedge \neg x_d \wedge \neg x_g$	$\neg y_c \wedge \neg y_d \wedge \neg y_g$
3	a b c	$\neg x_d \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_d \wedge \neg y_e \wedge \neg y_f \wedge \neg y_g$
4	a c d f		
5	g		
6	d		
7	d g		

Example

Tid	Itemsets	$p_i \equiv \ldots$	$q_i \equiv p_i \wedge \ldots$
1	$a\ b\ c\ d$	$\neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_e \wedge \neg y_f \wedge \neg y_g$
2	$a\ b\ e\ f$	$\neg x_c \wedge \neg x_d \wedge \neg x_g$	$\neg y_c \wedge \neg y_d \wedge \neg y_g$
3	$a\ b\ c$	$\neg x_d \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_d \wedge \neg y_e \wedge \neg y_f \wedge \neg y_g$
4	$a\ c\ d\ f$	$\neg x_b \wedge \neg x_e \wedge \neg x_g$	$\neg y_b \wedge \neg y_e \wedge \neg y_g$
5	g		
6	d		
7	$d\ g$		

Example

Tid	*Itemsets*	$p_i \equiv \ldots$	$q_i \equiv p_i \wedge \ldots$
1	*a b c d*	$\neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_e \wedge \neg y_f \wedge \neg y_g$
2	*a b e f*	$\neg x_c \wedge \neg x_d \wedge \neg x_g$	$\neg y_c \wedge \neg y_d \wedge \neg y_g$
3	*a b c*	$\neg x_d \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_d \wedge \neg y_e \wedge \neg y_f \wedge \neg y_g$
4	*a c d f*	$\neg x_b \wedge \neg x_e \wedge \neg x_g$	$\neg y_b \wedge \neg y_e \wedge \neg y_g$
5	*g*	$\neg x_b \wedge \neg x_c \wedge \neg x_d \wedge \neg x_e \wedge x_f$	$\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge \neg x_d \wedge \neg x_e \wedge x_f$
6	*d*		
7	*d g*		

Example

Tid	Itemsets	$p_i \equiv \ldots$	$q_i \equiv p_i \wedge \ldots$
1	$a\ b\ c\ d$	$\neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_e \wedge \neg y_f \wedge \neg y_g$
2	$a\ b\ \ \ e\ f$	$\neg x_c \wedge \neg x_d \wedge \neg x_g$	$\neg y_c \wedge \neg y_d \wedge \neg y_g$
3	$a\ b\ c$	$\neg x_d \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_d \wedge \neg y_e \wedge \neg y_f \wedge \neg y_g$
4	$a\ \ c\ d\ \ f$	$\neg x_b \wedge \neg x_e \wedge \neg x_g$	$\neg y_b \wedge \neg y_e \wedge \neg y_g$
5	g	$\neg x_b \wedge \neg x_c \wedge \neg x_d \wedge \neg x_e \wedge x_f$	$\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge \neg x_d \wedge \neg x_e \wedge x_f$
6	d	$\neg x_b \wedge \neg x_c \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_a \wedge \neg y_b \wedge \neg y_c \wedge \neg y_e \wedge \neg y_f \wedge \neg\neg y_g$
7	$d\ \ g$		

Example

Tid	Itemsets	$p_i \equiv \ldots$	$q_i \equiv p_i \wedge \ldots$
1	$a\ b\ c\ d$	$\neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_e \wedge \neg y_f \wedge \neg y_g$
2	$a\ b\ \ \ e\ f$	$\neg x_c \wedge \neg x_d \wedge \neg x_g$	$\neg y_c \wedge \neg y_d \wedge \neg y_g$
3	$a\ b\ c$	$\neg x_d \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_d \wedge \neg y_e \wedge \neg y_f \wedge \neg y_g$
4	$a\ \ c\ d\ \ f$	$\neg x_b \wedge \neg x_e \wedge \neg x_g$	$\neg y_b \wedge \neg y_e \wedge \neg y_g$
5	g	$\neg x_b \wedge \neg x_c \wedge \neg x_d \wedge \neg x_e \wedge x_f$	$\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge \neg x_d \wedge \neg x_e \wedge x_f$
6	d	$\neg x_b \wedge \neg x_c \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_a \wedge \neg y_b \wedge \neg y_c \wedge \neg y_e \wedge \neg y_f \wedge \neg\neg y_g$
7	$d\ \ g$	$\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge \neg x_e \wedge \neg x_f$	$\neg y_a \wedge \neg y_b \wedge \neg y_c \wedge \neg y_e \wedge \neg y_f$

Example

Tid	Itemsets	$p_i \equiv \ldots$	$q_i \equiv p_i \wedge \ldots$
1	$a\ b\ c\ d$	$\neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_e \wedge \neg y_f \wedge \neg y_g$
2	$a\ b\ \ e\ f$	$\neg x_c \wedge \neg x_d \wedge \neg x_g$	$\neg y_c \wedge \neg y_d \wedge \neg y_g$
3	$a\ b\ c$	$\neg x_d \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_d \wedge \neg y_e \wedge \neg y_f \wedge \neg y_g$
4	$a\ c\ d\ f$	$\neg x_b \wedge \neg x_e \wedge \neg x_g$	$\neg y_b \wedge \neg y_e \wedge \neg y_g$
5	g	$\neg x_b \wedge \neg x_c \wedge \neg x_d \wedge \neg x_e \wedge x_f$	$\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge \neg x_d \wedge \neg x_e \wedge x_f$
6	d	$\neg x_b \wedge \neg x_c \wedge \neg x_e \wedge \neg x_f \wedge \neg x_g$	$\neg y_a \wedge \neg y_b \wedge \neg y_c \wedge \neg y_e \wedge \neg y_f \wedge \neg\neg y_g$
7	$d\ \ g$	$\neg x_a \wedge \neg x_b \wedge \neg x_c \wedge \neg x_e \wedge \neg x_f$	$\neg y_a \wedge \neg y_b \wedge \neg y_c \wedge \neg y_e \wedge \neg y_f$

Notice that every interpretation $\mathcal{I}$ that satisfies (1), (2), and (3) defines:

- a set $X = \{\omega \in \Omega \mid \mathcal{I} \models x_\omega\}$
- a set $Y = \{\omega \in \Omega \mid \mathcal{I} \models y_\omega\}$
- such that $X \cap Y = \emptyset$; (due to (1))
- $\mathcal{I} \models p_i$ iff i is in the support of X; (due to (2))
- $\mathcal{I} \models q_i$ iff i is in the support of $X \cup Y$. (due to (2) and (3))

- we first formalize the general facts that $S(X, \mathcal{D}) \geq n$ and $S(X \cup Y, \mathcal{D}) \geq n$ for any $n \leq |\mathcal{D}|$, by the following two formulas

$$\bigvee_{\substack{N \subseteq \{1,\dots,|D|\} \\ |N|=n}} \bigwedge_{i \in N} p_i \quad \textit{and} \quad \bigvee_{\substack{N \subseteq \{1,\dots,|D|\} \\ |N|=n}} \bigwedge_{i \in N} q_i$$

which can be rewritten in the following two CNF formulas:

$$\bigwedge_{\substack{N \subseteq \{1,\dots,|D|\} \\ |N|=|\mathcal{D}|-n+1}} \bigvee_{i \in N} p_i \quad \textit{and} \quad \bigwedge_{\substack{N \subseteq \{1,\dots,|D|\} \\ |N|=|D|-n+1}} \bigvee_{i \in N} q_i$$

Exercise

Prove that this transformation returns equivalent formulas.

Encoding in propositional logic

- The support of $X \rightarrow Y$ is greater or equal to α; I.e.,

$$\frac{S(X \cup Y, \mathcal{D})}{|D|} \geq \alpha$$

corresponding to the formula[1]

$$S(X \cup Y, \mathcal{D}) \geq \lceil \alpha \cdot |\mathcal{D}| \rceil \tag{4}$$

- The confidence of $X \rightarrow Y$ is greater or equal to β;

$$\frac{S(X \cup Y, \mathcal{D})}{S(X, \mathcal{D})} \geq \beta$$

the last implication can translated in the following conjunction of implications[2]

$$\bigwedge_{n=1}^{|\mathcal{D}|} (S(X, \mathcal{D}) = n \rightarrow S(X \cup Y, \mathcal{D}) \geq \lceil \beta \cdot n \rceil) \tag{5}$$

[1] $\lceil x \rceil$ denotes the smallest integer such that $n \geq x$

[2] $S(X, \mathcal{D}) = n$ denotes $S(X, \mathcal{D}) \geq n \wedge \neg S(X, \mathcal{D}) \geq n + 1$.

Solving $MAR(\mathcal{D}, \alpha, \beta)$ via Satisfiability

If we are able to find an interpetation $\mathcal{I}$ that satisfies (1)–(5), then we extract the association rule $X \rightarrow Y$ in $\mathcal{E}(\mathcal{D}, \alpha, \beta)$, as follows

- $X = \{\omega \in \Omega \mid \mathcal{I} \models x_\omega\}$
- $Y = \{\omega \in \Omega \mid \mathcal{I} \models y_\omega\}$

$X \rightarrow Y \in \mathcal{E}(\mathcal{D}, \alpha, \beta)$ is guaranteed because:

- $X \cap Y = \emptyset$ since $\mathcal{I} \models (1)$
- $S(X \rightarrow Y, \mathcal{D}) \geq \alpha$ since $\mathcal{I} \models (4)$
- $Conf(X \rightarrow Y, \mathcal{D}) \geq \beta$ since $\mathcal{I} \models (5)$.