Association Rule Mining

Market basket analysis: to find associations between different itemsets that customers place in shopping basket

Applications: Cross marketing, catalog / floor design, design attractive packs, web log analysis for e-commerce,

Rule generation:

Antecedent $\Rightarrow$ Consequent (support, confidence)

Data structure:

$$T = \{t_1, \ldots, t_n\} \text{ set of transcations}$$

Each t_K is an itemset

$$I = \{i_1, \ldots, i_m\}$$

Typical data

Cid	A1	A2	A3	. . . . .	A_K
c_1	0	0	1	0 1 1 - - -	0
c_2	1	0	0	1 1 0 - - -	1
⋮					
c_n	-	-	-	- - -	.

Aim:
- Find frequent patterns, i.e. associations among sets of items in T
- Represent these relationships as association rules of the form

$$X \Rightarrow Y \text{ (support, confidence)}$$

Important definitions:

Support count : # of occurrences of an itemset in the database

$\sigma(\{\text{itemset}\})$

Support : Fraction of transactions containing the itemset

$$S(\text{itemset}) = \frac{\sigma(\text{itemset})}{|T|}$$

Frequent itemset : An itemset whose support $\geq$ a threshold, minsup

Confidence : A measure of how often B appears in transactions containing A.

[Rule: $(A \Rightarrow B)$]

% of transactions containing A which also contains B

$$C(A \Rightarrow B) = \frac{S(A, B)}{S(A)}$$ (est of conditional prob $B|A$)

Example:

Tid	items
T1	bread, egg, peanut-butter
T2	bread, peanut-butter
T3	bread, milk, peanut-butter
T4	beer, bread
T5	beer, milk

Rule	support	confidence
bread ⇒ peanut-butter	$\frac{3}{5} = 0.6$	$\frac{3}{4} = 0.75$
peanut-butter ⇒ bread	$\frac{3}{5} = 0.6$	$\frac{3}{3} = 1.0$
beer ⇒ bread	$\frac{1}{5} = 0.2$	$\frac{1}{2} = 0.5$
peanut-butter ⇒ egg	$\frac{1}{5} = 0.2$	$\frac{1}{3} = 0.33$
egg ⇒ peanut-butter	$\frac{1}{5} = 0.2$	$\frac{1}{1} = 1$
egg ⇒ milk	0	0

ARM task: Given a set of transactions, the goal of ARM is to find all rules ∋
(i) support ≥ min sup
& (ii) confidence ≥ min conf
(min sup, min conf) : fixed apriori

Brute force approach
- List all possible association rules
- compute support & confidence for each rule
- prune as per threshold

Approach is computationally prohibitive

The Apriori Algorithm

Agrawal & Srikant : "Fast algorithms for mining association rules in large databases", Int Conf on VLDB, 1994.

2-step ARM of Apriori algorithm

S1: Generate all frequent itemsets with support ≥ min sup

S2: Generate association rules using these frequent itemsets

Anti-monotonicity property of Apriori algorithm

Downward closure property

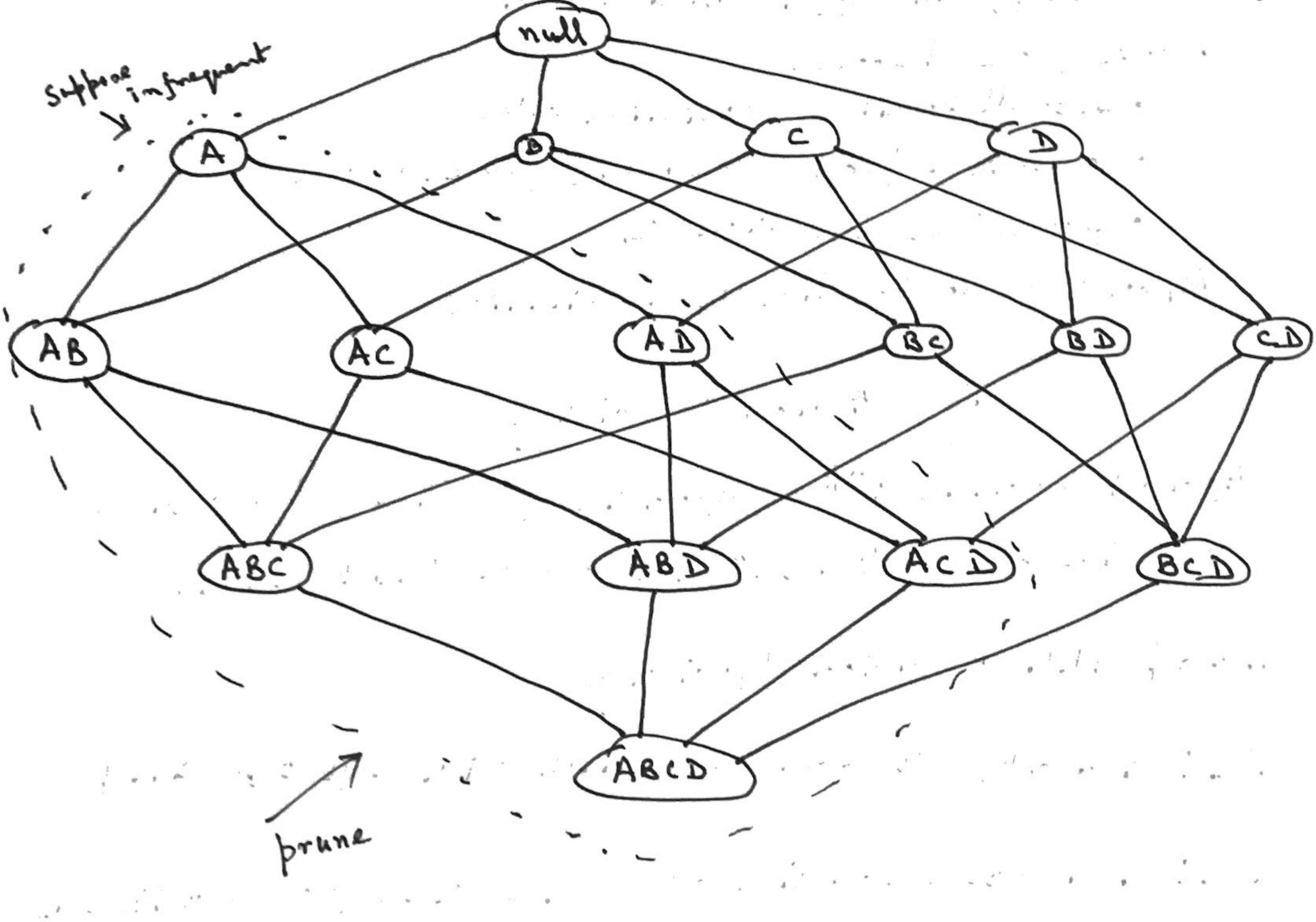
- Any subset of a frequent itemset is frequent

$$\text{i.e. } \forall\ X \subseteq Y$$
$$S(X) \geq S(Y) \quad \forall\ X, Y$$

$\Rightarrow$ If an itemset is not frequent, none of it's supersets can be frequent ($\Rightarrow$ prune all such sets)

$\Rightarrow$ If an itemset is not frequent, there is no need to explore its supersets.

Example: Itemset lattice → apriori principle of prunning



- ~~As~~ Suppose B is found to be infrequent (support < min sup)

  ⇒ all itemsets with B will also be infrequent

  ⇒ prune all such branches. (i.e. all it's supersets)

  i.e. exclude itemset AB, BC, BD, ABC, ABD, BCD, ABCD

- Suppose AB is found to be infrequent

  ⇒ all itemsets with AB will also be infrequent

  ⇒ prune all such branches (i.e. all supersets)

  i.e. exclude itemsets, ABC, ABD, ABCD.

Apriori Step-1 in pseudo codes

- K = 1
- Generate frequent itemsets of length 1
- Prune itemsets of higher orders (i.e. supersets), if necessary
- Generate itemsets of length K+1 from frequent itemsets of length K
- Compute the support of new candidate itemsets w.r.t. minsup. Prune if necessary.
- K = K+1
- Repeat until no frequent itemsets are found.

Generation (step) of Itemsets for next level

Let L_K denote the frequent itemsets at level K and C_K denote the set of all candidates at level K

- Items in L_{K-1} are listed in an order

Step 1: Self joining $L_{K-1} * L_{K-1}$

i.e. joining of 2 items from L_{K-1}

$$\{p.item_1,\ p.item_2 \ldots\ p.item_{K-1}\}$$
$$\quad = \qquad\quad = \qquad\qquad\quad = \quad\ <$$
& $\{q.item_1\ q.item_2 \ldots\ q.item_{K-1}\}$ under the given order

insert into C_K the itemset $p.item_1\ p.item_2 \ldots p.item_{K-2}\ p.item_{K-1}\ q.item_{K-1}$

Step 2: Prunning of C_K set

$\forall$ itemsets c in C_K and

$\forall$ $(K-1)$ order subsets s of c (under the given order)

If (s is not in L_{K-1}) delete c from C_K

Justification is anti-monotone property

Example

$$L_3 = \{ABC, ABD, ACD, ACE, BCD\}$$

Self joining step: $L_3 * L_3$

ABC, ABD → ABCD

ACD, ACE → ACDE

Prunning step:

Consider ACDE → subsets ACD, ACE, CDE, ADE

CDE/ ADE is not in L_3

⇒ Prune ACDE from C_4

Consider ABCD → subsets ABC, ABD, ACD, BCD

all subsets in L_3

⇒ Prunning not req'd for ABCD

⇒ Pass ABCD to C_4

Example

T1 : {A, C, D}

T2 : {B, C, E}

T3 : {A, B, C, E}

T4 : {B, E}

min sup = .5

1st Scan →

C_1

item	support count
A	2
B	3
C	3
~~D~~	~~1~~ prune
E	3

→ Freq itemset L_1

A	2
B	3
C	3
E	3

→ C_2

AB
AC
AE
BC
BE
CE

$\xrightarrow{\text{2nd Scan}}$

prune

Itemset	Support Count
~~AB~~	~~1~~
AC	2
~~AE~~	~~1~~
BC	2
BE	3
CE	2

$\longrightarrow$

L_2	
AC	2
BC	2
BE	3
CE	2

$\longrightarrow$

C_3
BCE

L_3

all other joins prunned.

$S(BCE) = 2$

Under C_3, the only Itemset BCE has support count 2, i.e $\geq$ minsup

Step 2 of apriori algorithm

Generate association rules using frequent itemsets

Given any frequent itemset L;

- find all non-empty subsets F of L
- output each rule $F \Rightarrow \{L-F\}$ that satisfies the threshold on confidence (min conf)

Example: Let $L = \{A, B, C\}$ is a frequent itemset

Candidate rules are

$AB \Rightarrow C$; $AC \Rightarrow B$; $BC \Rightarrow A$

$A \Rightarrow BC$; $B \Rightarrow AC$; $C \Rightarrow AB$

In general, there are $2^{|L|} - 2$ candidate rules.

<u>Remark</u>: Efficiency in rule generation

Confidence of rules generated from the same Itemset have anti-monotone property

$$c(ABC \Rightarrow D) \geq c(AB \Rightarrow CD) \geq c(A \Rightarrow BCD)$$

Sly $c(ABC \Rightarrow D) \geq c(AC \Rightarrow BD) \geq c(C \Rightarrow ABD)$

- - - .

[Consider, e.g., $ABC \Rightarrow D$ & $AB \Rightarrow CD$

$AB \subset ABC$

$s(AB) \geq s(ABC)$

$$\Rightarrow \frac{1}{s(ABC)} \geq \frac{1}{s(AB)}$$

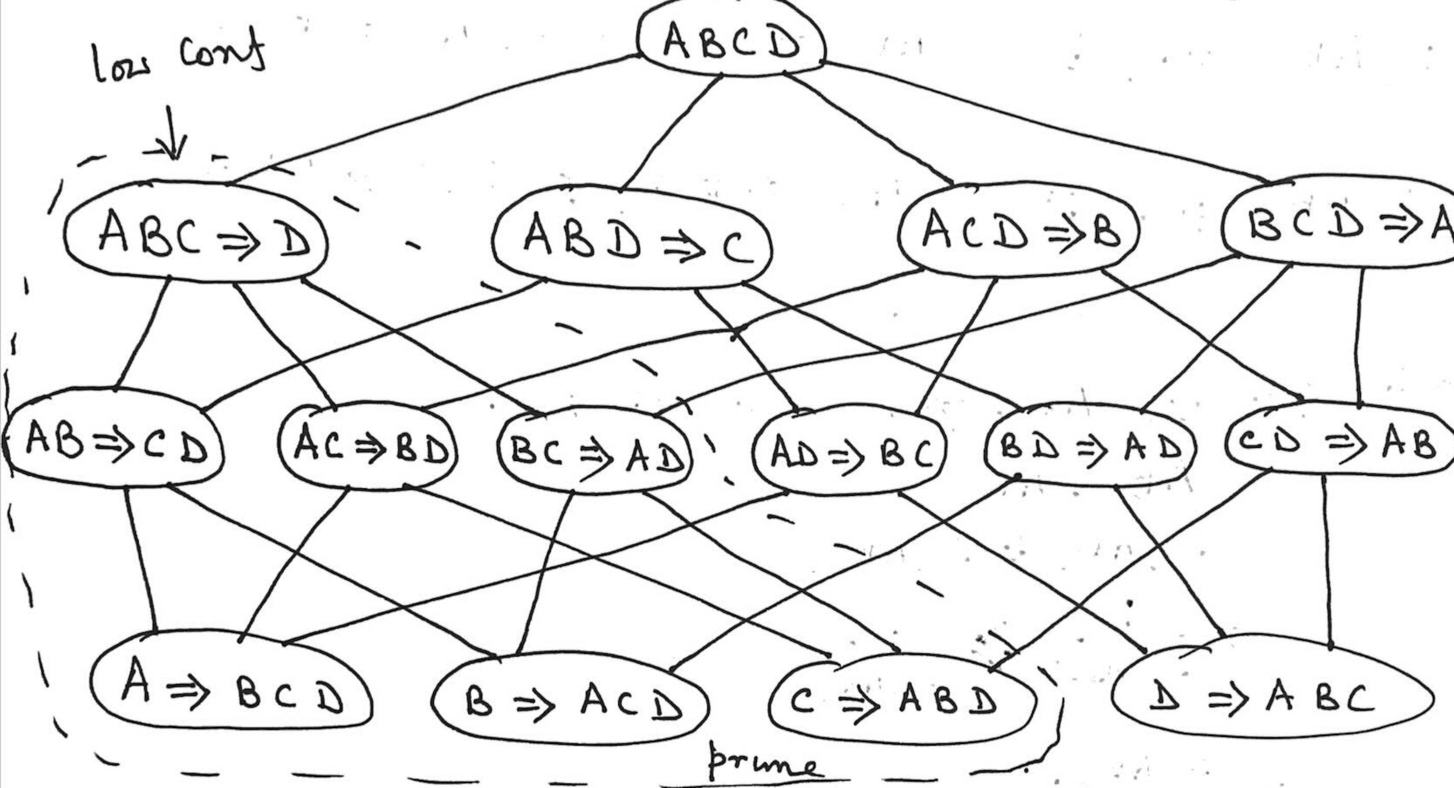
$$\Rightarrow \frac{s(ABCD)}{s(ABC)} \geq \frac{s(ABCD)}{s(AB)}$$

i.e. $c(ABC \Rightarrow D) \geq c(AB \Rightarrow CD)$]

We apply this property to prune rule generation sequence.

Example of prunning using anti-monotone property

Suppose ABCD is a frequent itemset and look at possible rules for ARM $L \to F \Rightarrow \{L-F\}$



~~Supposed~~ Suppose the rule $ABC \Rightarrow D$ is low conf i.e. $C(ABC \Rightarrow D) <$ min conf

then by anti-monotone property, all sub rules below that in the Lattice will be < min conf and can be prunned, i.e.

$AB \Rightarrow CD$, $AC \Rightarrow BD$, $BC \Rightarrow AD$

$A \Rightarrow BCD$, $B \Rightarrow ACD$ & $C \Rightarrow ABD$

are prunned.

Example

T1 : {A, C, D}

T2 : {B, C, E}

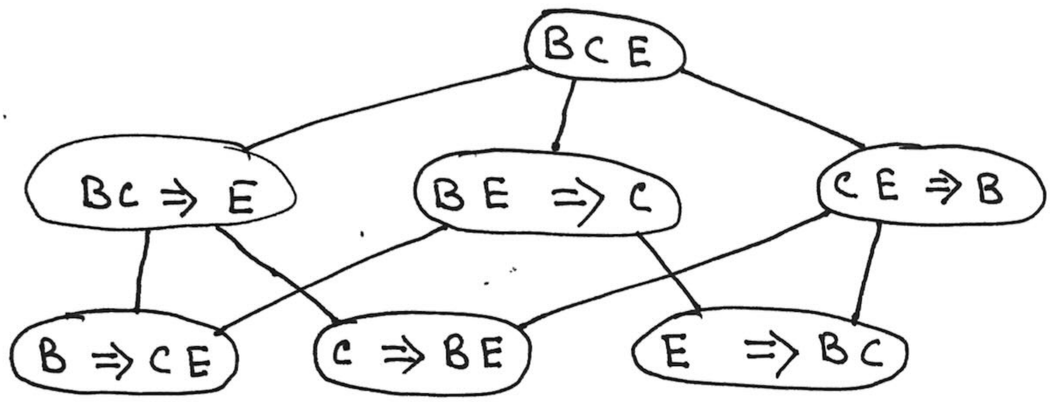
T3 : {A, B, C, E}

T4 : {B, E}

Frequent itemsets with min sup ≥ 0.5 are
$S(\cdot)=2$

BCE , AC, BC , BE , CE
$S(\cdot)=2$ $S(\cdot)=2$ $S(\cdot)=3$ $S(\cdot)=2$

Consider rules for BCE Itemset



$$C(BC \Rightarrow E) = \frac{S(BCE)}{S(BC)} = \frac{2}{2} = 1$$

$\Rightarrow BC \Rightarrow E \; (.5, 1)$

$$C(BE \Rightarrow C) = \frac{S(BCE)}{S(BE)} = \frac{2}{3}$$

if min conf = 0.75, then $BE \Rightarrow C$ along with $B \Rightarrow CE$ & $E \Rightarrow BC$ are prunned.

$$C(CE \Rightarrow B) = \frac{S(BCE)}{S(CE)} = \frac{2}{2} = 1$$

$$C(C \Rightarrow BE) = \frac{S(BCE)}{S(C)} = \frac{2}{3}$$ prune at min conf 0.75