

# Data Mining

## 資料探勘

### Association Analysis

*Hung-Yu Kao, Fall 2017*

# Top-10 Algorithms in Data Mining

2

## Top 10 algorithms in data mining

Xindong Wu · Vipin Kumar · J. Ross Quinlan · Joydeep Ghosh · Qiang Yang ·  
Hiroshi Motoda · Geoffrey J. McLachlan · Angus Ng · Bing Liu · Philip S. Yu ·  
Zhi-Hua Zhou · Michael Steinbach · David J. Hand · Dan Steinberg

Received: 9 July 2007 / Revised: 28 September 2007 / Accepted: 8 October 2007

Published online: 4 December 2007

© Springer-Verlag London Limited 2007

**Abstract** This paper presents the top 10 data mining algorithms identified by the IEEE International Conference on Data Mining (ICDM) in December 2006: C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, kNN, Naive Bayes, and CART. These top 10 algorithms are among the most influential data mining algorithms in the research community. With each algorithm, we provide a description of the algorithm, discuss the impact of the algorithm, and review current and further research on the algorithm. These 10 algorithms cover classification, clustering, statistical learning, association analysis, and link mining, which are all among the most important topics in data mining research and development.

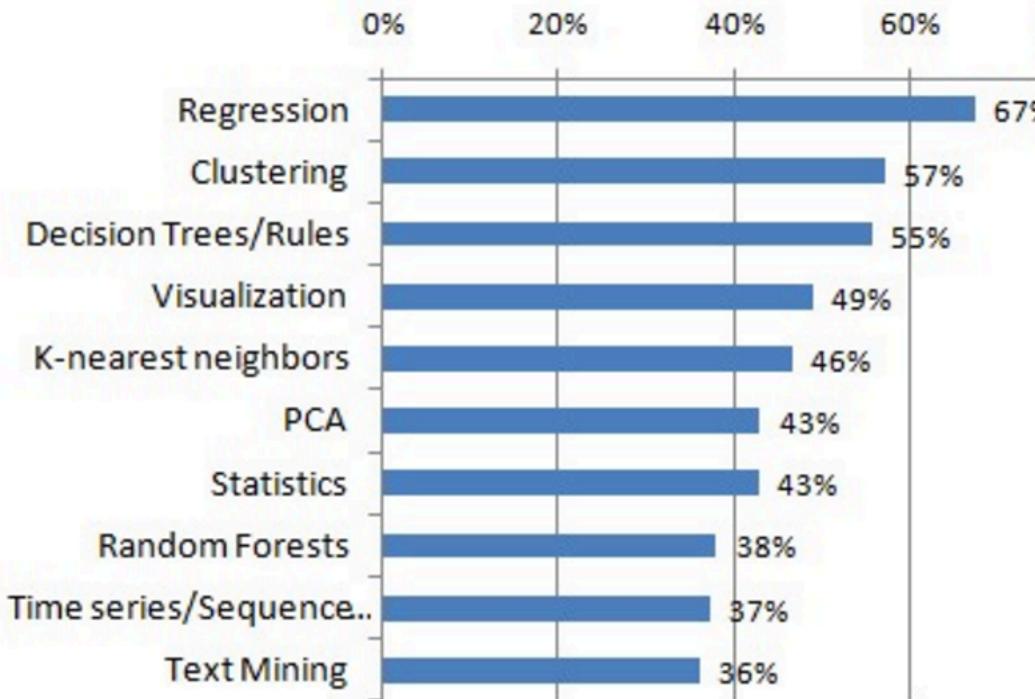


# Top-10 Algorithms and Methods Used by Data Scientists

<http://www.kdnuggets.com/2016/09/poll-algorithms-used-data-scientists.html>

3

## Top 10 Algorithms & Methods used by Data Scientists



The biggest declines are for

- Association rules, down 47% to 15.3% from 28.6%

Which methods/algorithms did you use for data analysis in 2011? [311 voters]	
Decision Trees/Rules (186)	59.8 %
Regression (180)	57.9 %
Clustering (163)	52.4 %
Statistics (descriptive) (149)	47.9 %
Visualization (119)	38.3 %
Time series/Sequence analysis (92)	29.6 %
Support Vector (SVM) (89)	28.6 %
Association rules (89)	28.6 %
Ensemble methods (88)	28.3 %
Text Mining (86)	27.7 %
Neural Nets (84)	27.0 %
Boosting (73)	23.5 %
Bayesian (68)	21.9 %
Bagging (63)	20.3 %
Factor Analysis (58)	18.7 %
Anomaly/Deviation detection (51)	16.4 %
Social Network Analysis (44)	14.2 %
Survival Analysis (29)	9.32 %
Genetic algorithms (29)	9.32 %
Uplift modeling (15)	4.82 %

# Association Rule Mining

4

- Given a set of transactions, find rules that will predict **the occurrence of an item** based on the occurrences of other items in the transaction

## Market-Basket transactions

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

## Example of Association Rules

{Diaper} → {Beer},  
{Milk, Bread} → {Eggs, Coke},  
{Beer, Bread} → {Milk},

Implication means **co-occurrence**,  
not **causality**!

How about {Beer} → {Diaper}?



# Definition: Frequent Itemset

5

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
- **Support count ( $\sigma$ )**
  - An itemset that contains k items
  - Frequency of occurrence of an itemset
  - E.g.  $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g.  $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

# Definition: Association Rule

6

## ● Association Rule

- An implication expression of the form  $X \rightarrow Y$ , where X and Y are itemsets
- Example:  
 $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

## ● Rule Evaluation Metrics

- Support (s)
  - ◆ Fraction of transactions that contain both X and Y
- Confidence (c)
  - ◆ Measures how often items in Y appear in transactions that contain X

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example:

$$\{\text{Milk, Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|T|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

# An Example

7

- if minimum support is 2, minimum confidence is 2/3
- frequent itemset
  - {A}, {B}, {C}, {E}, {A,C}, {B,C}, {B,E}, {C,E}, {B,C,E}
- strong rule
  - {B, E} → C (2/3)
  - C → A (2/3), A → C (2/2)

TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E

# Association Rule Mining Task

8

- Given a set of transactions  $T$ ,  $\text{minsup}$  and  $\text{minconf}$ , the goal of association rule mining is to find all rules having
  - $\text{support} \geq \text{minsup}$  threshold
  - $\text{confidence} \geq \text{minconf}$  threshold
- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the  $\text{minsup}$  and  $\text{minconf}$  thresholds

⇒ Computationally prohibitive!



# Mining Association Rules

9

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

## Example of Rules:

- $\{\text{Milk}, \text{Diaper}\} \rightarrow \{\text{Beer}\}$  ( $s=0.4, c=0.67$ )
- $\{\text{Milk}, \text{Beer}\} \rightarrow \{\text{Diaper}\}$  ( $s=0.4, c=1.0$ )
- $\{\text{Diaper}, \text{Beer}\} \rightarrow \{\text{Milk}\}$  ( $s=0.4, c=0.67$ )
- $\{\text{Beer}\} \rightarrow \{\text{Milk}, \text{Diaper}\}$  ( $s=0.4, c=0.67$ )
- $\{\text{Diaper}\} \rightarrow \{\text{Milk}, \text{Beer}\}$  ( $s=0.4, c=0.5$ )
- $\{\text{Milk}\} \rightarrow \{\text{Diaper}, \text{Beer}\}$  ( $s=0.4, c=0.5$ )

## Observations:

- All the above rules are binary partitions of the same itemset:  
 $\{\text{Milk}, \text{Diaper}, \text{Beer}\}$
- Rules originating from the same itemset have **identical support** but  
*can have different confidence*



# Basic Approach of Association Rule

10

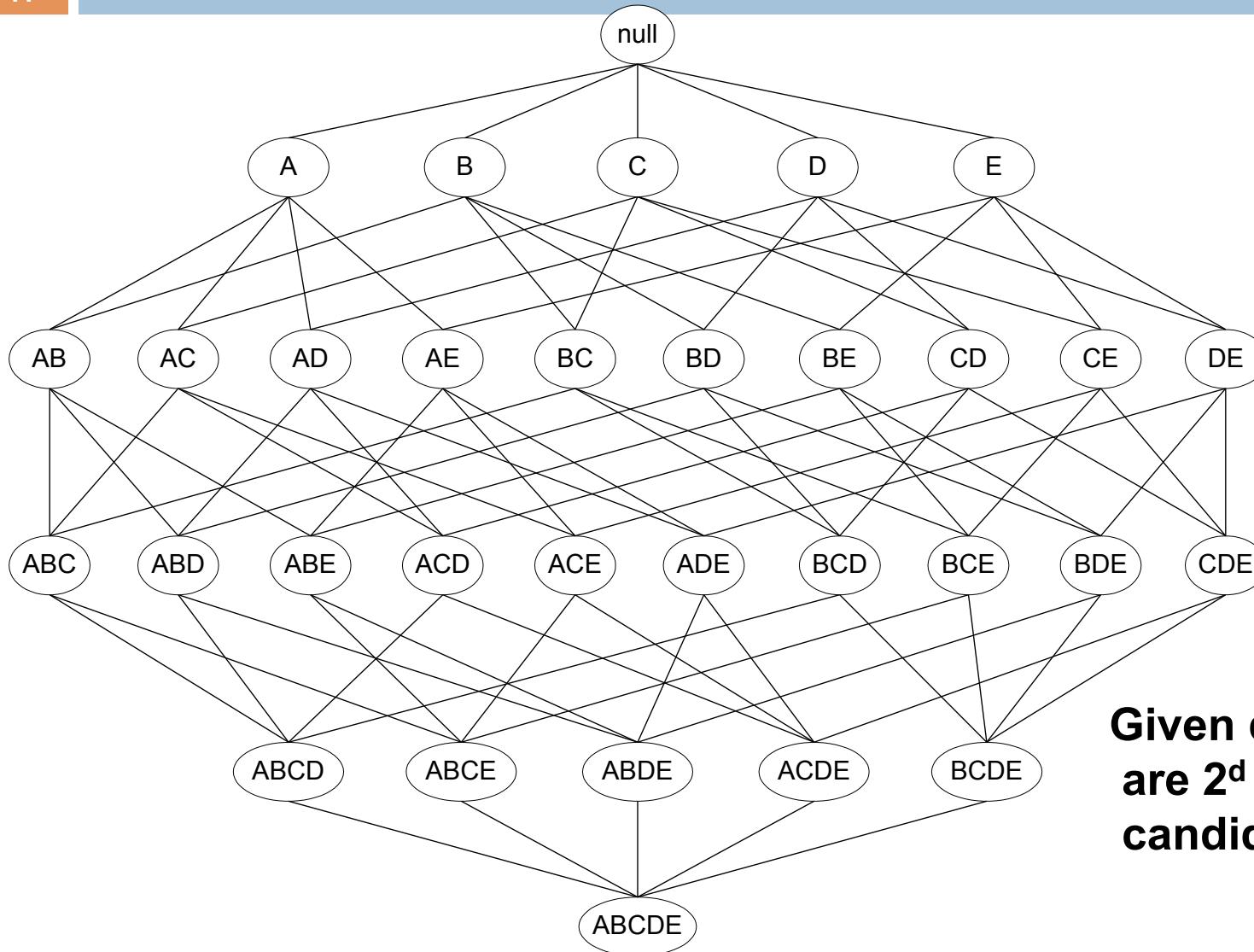
- Process of association rule mining
  - step 1: find all frequent itemsets
  - step 2: generate strong association rules from frequent itemsets

Frequent itemset generation is still computationally expensive.



# Frequent Itemset Generation

11



**Given  $d$  items, there  
are  $2^d$  possible  
candidate itemsets**

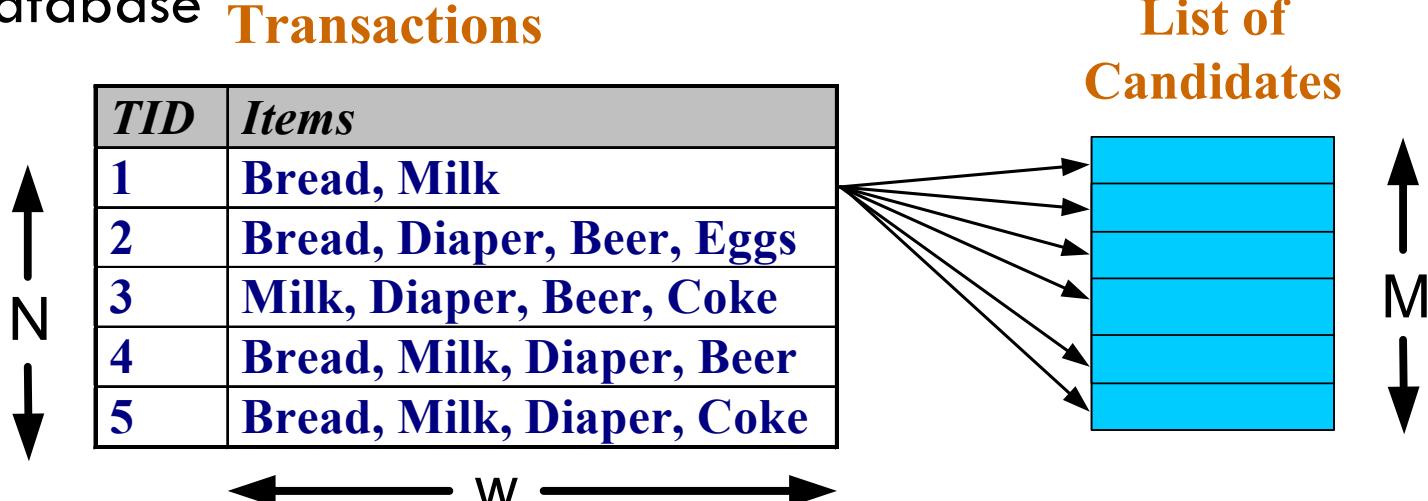


# Frequent Itemset Generation

12

## □ Brute-force approach:

- Each itemset in the lattice is a **candidate** frequent itemset
- Count the support of each candidate by scanning the database **Transactions**



- Match each transaction against every candidate
- Complexity  $\sim O(NMw) \Rightarrow$  **Expensive since *M* =  $2^d$  !!!**

# Frequent Itemset Generation Strategies

13

- Reduce the **number of candidates** ( $M$ )
  - Complete search:  $M=2^d$
  - Use pruning techniques to reduce  $M$
- Reduce the **number of transactions** ( $N$ )
  - Reduce size of  $N$  as the size of itemset increases
  - Used by DHP and vertical-based mining algorithms
- Reduce the **number of comparisons** ( $NM$ )
  - Use efficient data structures to store the candidates or transactions
  - No need to match every candidate against every transaction





# APRIORI ALGORITHM

# Apriori Algorithm

15

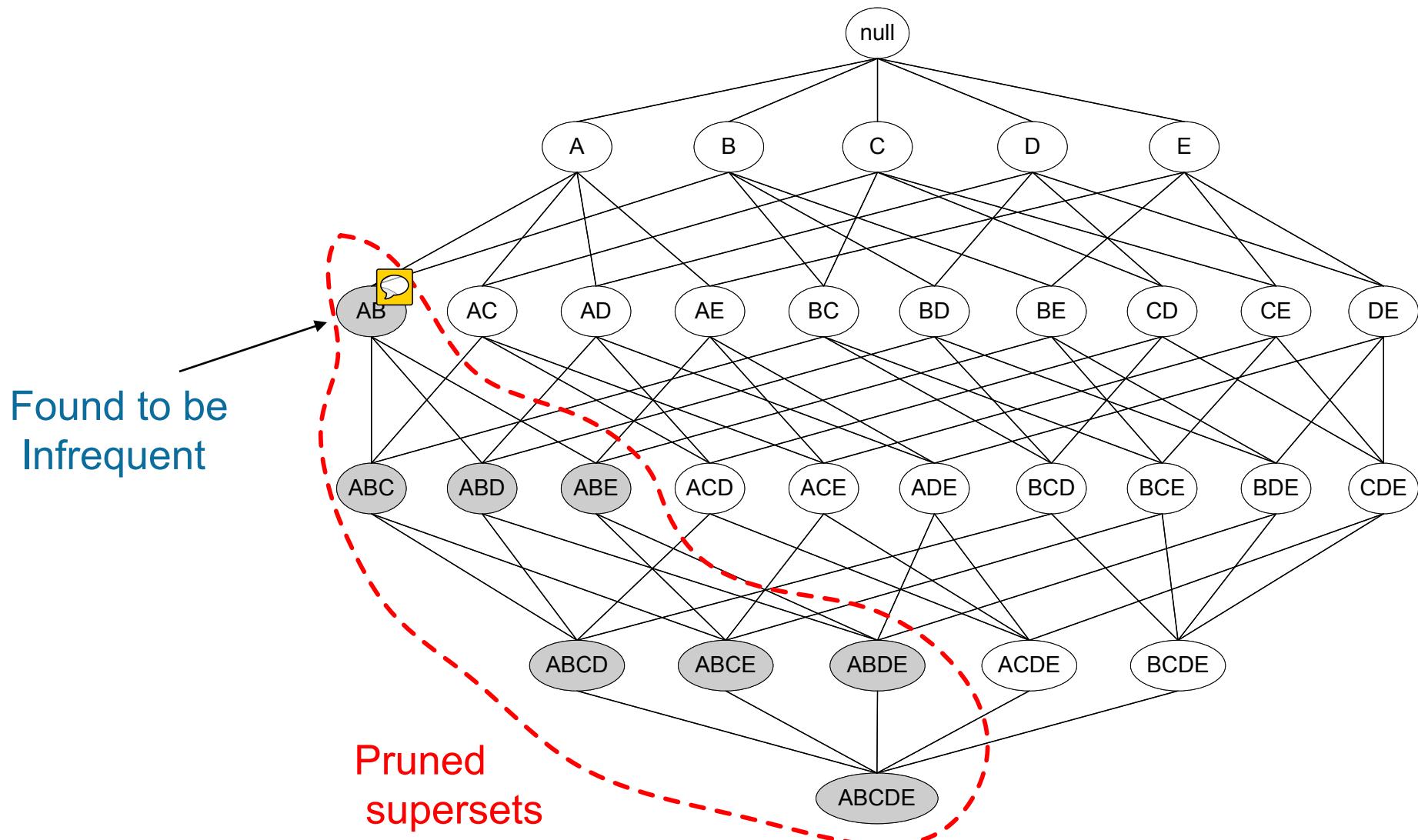
- Observation: Apriori property
  - ***all non-empty subsets of a frequent itemset must also be frequent***
- Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- ***Support of an itemset never exceeds the support of its subsets***
- This is known as the **anti-monotone** property of support



# Illustrating Apriori Principle



# Illustrating Apriori Principle

17

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Items (1-itemsets)



Itemset	Count
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$$

With support-based pruning,

$$6 + 6 + 1 = 13$$



Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3



How about {Beer, Milk, Diaper}?



# Apriori Algorithm

18

## □ Notations

- frequent k-itemset (denoted as  $L_k$ ): satisfy minimum support
- candidate k-itemset (denoted as  $C_k$ ): possible frequent k-itemsets

## □ Level-wise approach

- $(k-1)$ -itemsets are used to explore  $k$ -itemsets
- join  $C_k = L_{k-1} \otimes L_{k-1} = \{A \otimes B \mid A, B \in L_{k-1}, |A \cap B| = k-2\}$
- prune  $C_k$  by subset test
- generate  $L_k$  by scanning transaction DB

Min. support 50% (i.e., 2tx's)

BE=>C conf.:66%

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

$C_1$

1<sup>st</sup> scan

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

$L_1$

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

$C_2$

2<sup>nd</sup> scan

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

$C_2$

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

$L_2$

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

Itemset
{B, C, E}

3<sup>rd</sup> scan

$L_3$

Itemset	sup
{B, C, E}	2

# Algorithm Apriori Candidate Generation

20

## join step

insert into  $C_k$

select  $p.item_1, p.item_2, \dots, p.item_{k-1}, q.item_{k-1}$   
from  $L_{k-1} p, L_{k-1} q$

where  $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2},$   
 $p.item_{k-1} < q.item_{k-1};$

## prune step

for all itemsets  $c \in C_k$  do

    for all  $(k-1)$ -subsets  $s$  of  $c$  do

        if  $(s \notin L_{k-1})$  then

            delete  $c$  from  $C_k$



# Important Details of Apriori

21

## □ How to generate candidates?

- **Step 1: self-joining  $L_k$**
- **Step 2: pruning**

Example of Candidate-generation

$$L_3 = \{abc, abd, acd, ace, bcd\}$$

Self-joining:  $L_3 * L_3$

abcd from abc and abd

acde from acd and ace

Pruning:

acde is removed because ade is not in  $L_3$

$$C_4 = \{abcd\}$$



# Algorithm Apriori

22

```
L1={frequent 1-itemsets};  
for (k=2; Lk-1 ≠ 0; k++) do begin  
    Ck=apriori-gen(Lk-1);  
    for each transactions t ∈ D do begin //scan DB  
        Ct=subset(Ck, t) //get the subsets of t that are candidates  
        for each candidate c ∈ Ct do  
            c.count++;  
    end  
    Lk={c ∈ Ck | c.count ≥ minsup}  
end  
Answer=UkLk;
```

# Challenges of Frequent Pattern Mining

23

- Challenges
  - **Multiple scans** of transaction database
  - Huge number of candidates ( $C_2$ )
  - Tedious workload of support counting for candidates

# How to Count Supports of Candidates?

24

- Why counting supports of candidates a problem?
  - The total number of candidates can be very huge
  - One transaction may contain many candidates
- Method:
  - Candidate itemsets are stored in a *hash-tree*
  - *Leaf node* of hash-tree contains a list of itemsets and counts
  - *Interior node* contains a hash table

# Generate Hash Tree

25

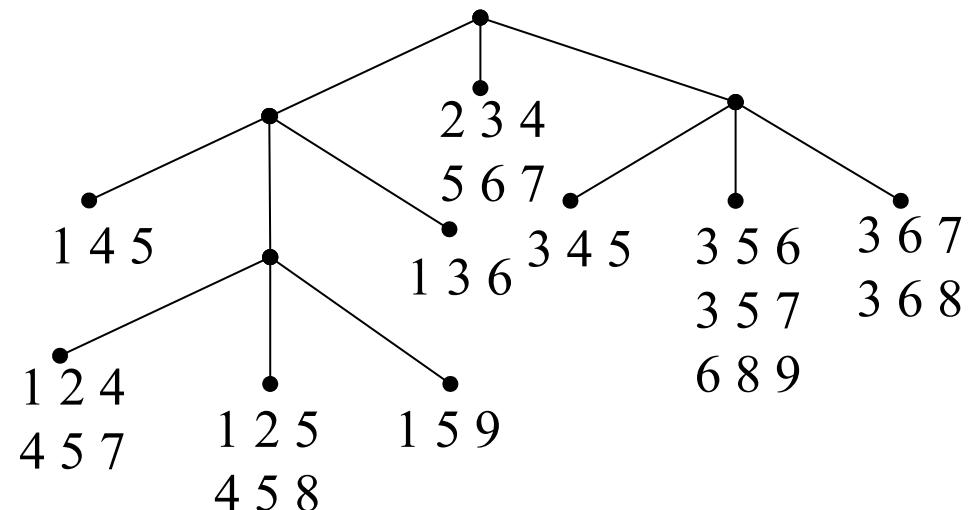
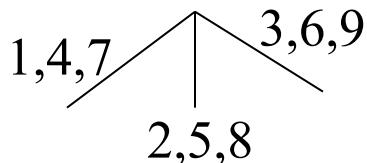
Suppose you have 15 candidate itemsets of length 3:

{1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8}

You need:

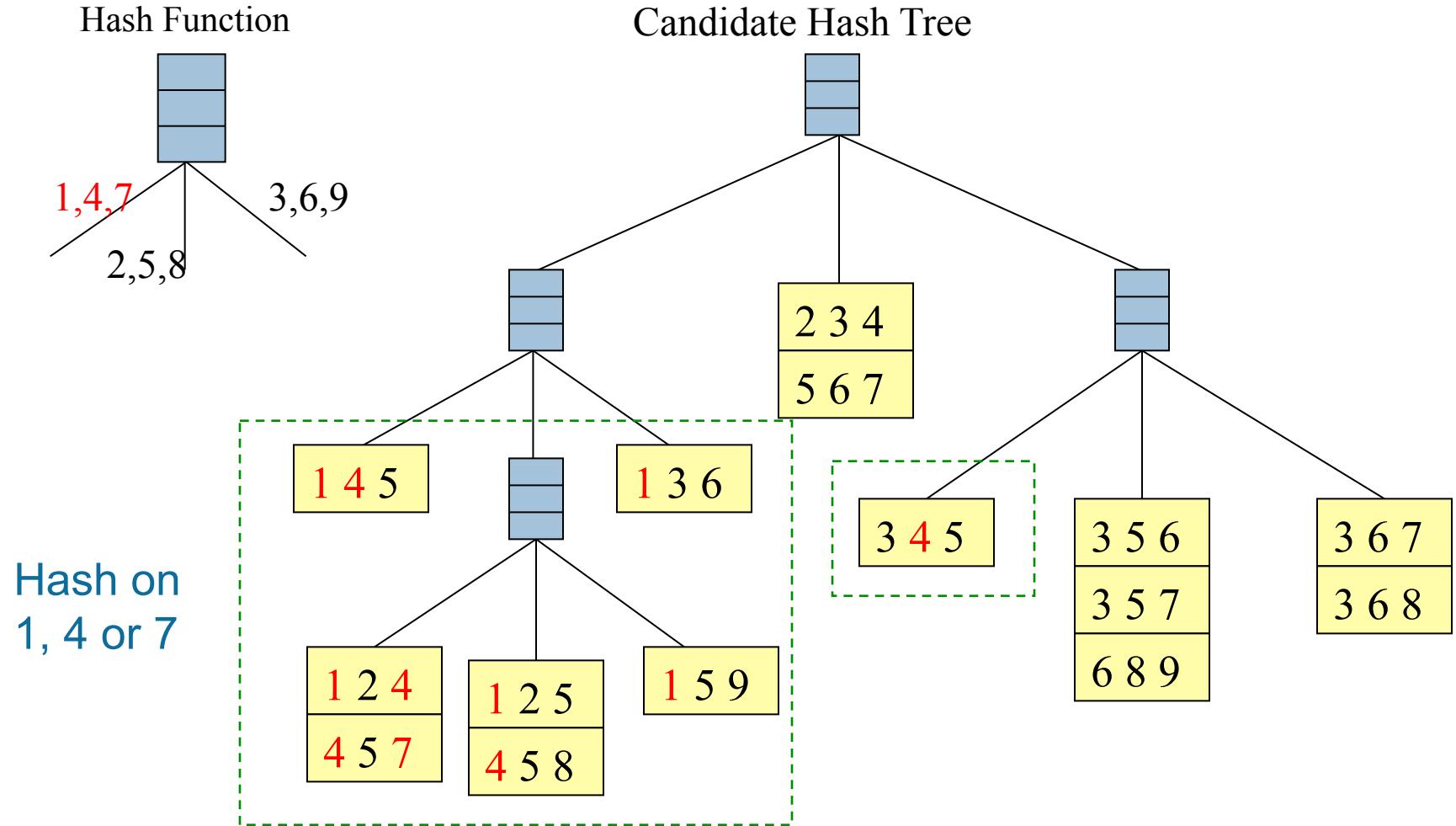
- Hash function
- Max leaf size: max number of itemsets stored in a leaf node (if number of candidate itemsets exceeds max leaf size, split the node)

Hash function



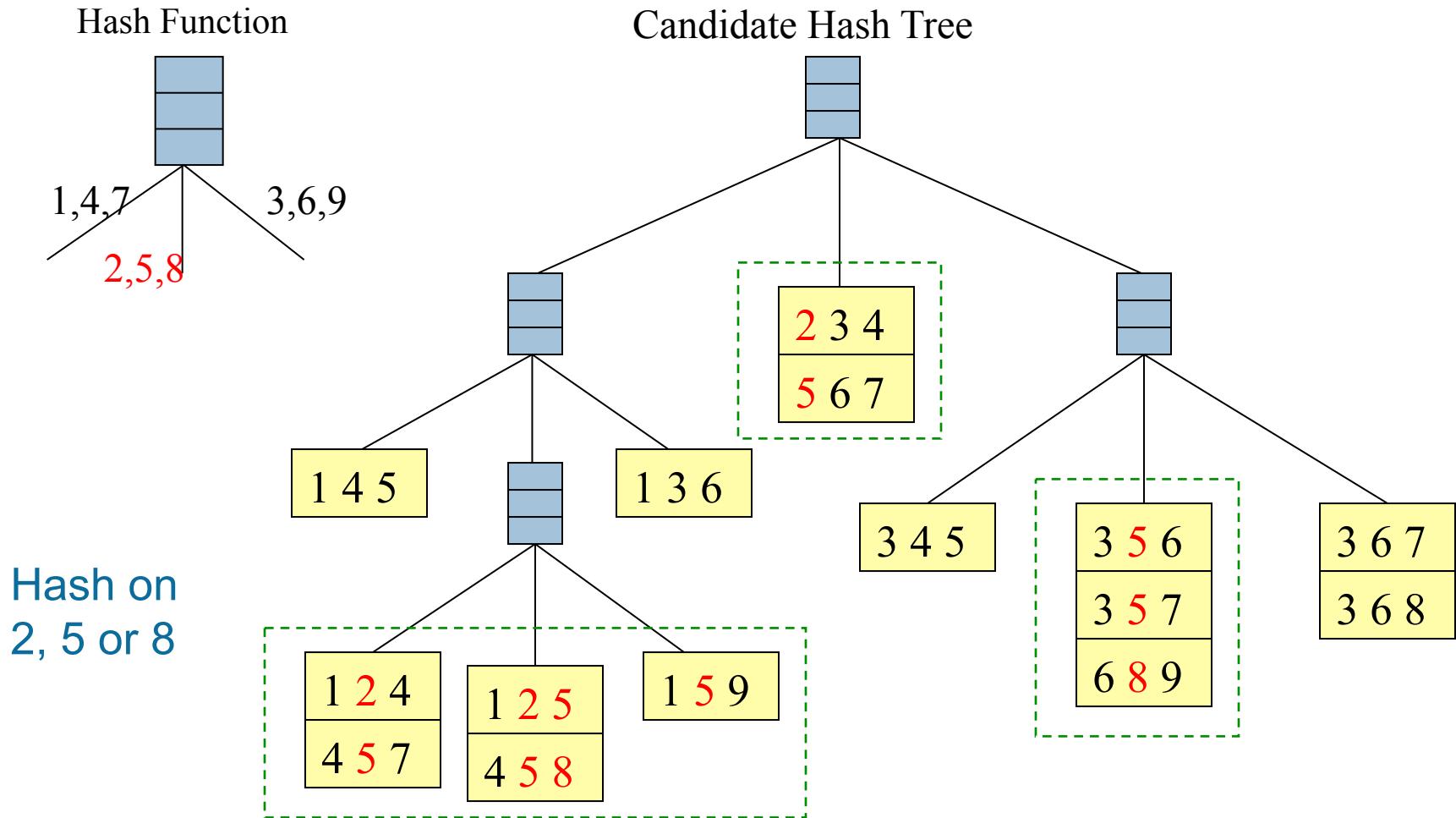
# Association Rule Discovery: Hash tree

26



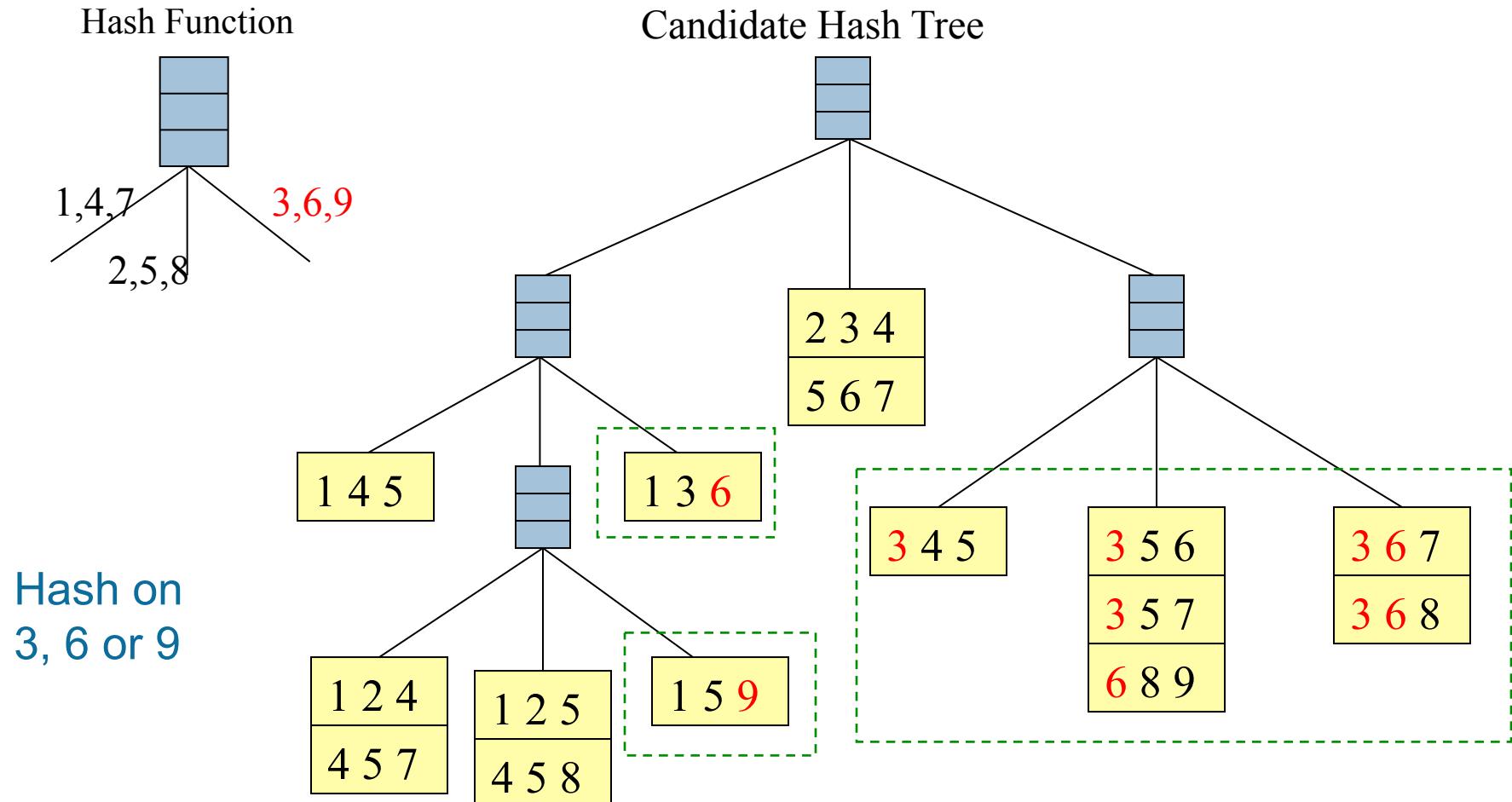
# Association Rule Discovery: Hash tree

27



# Association Rule Discovery: Hash tree

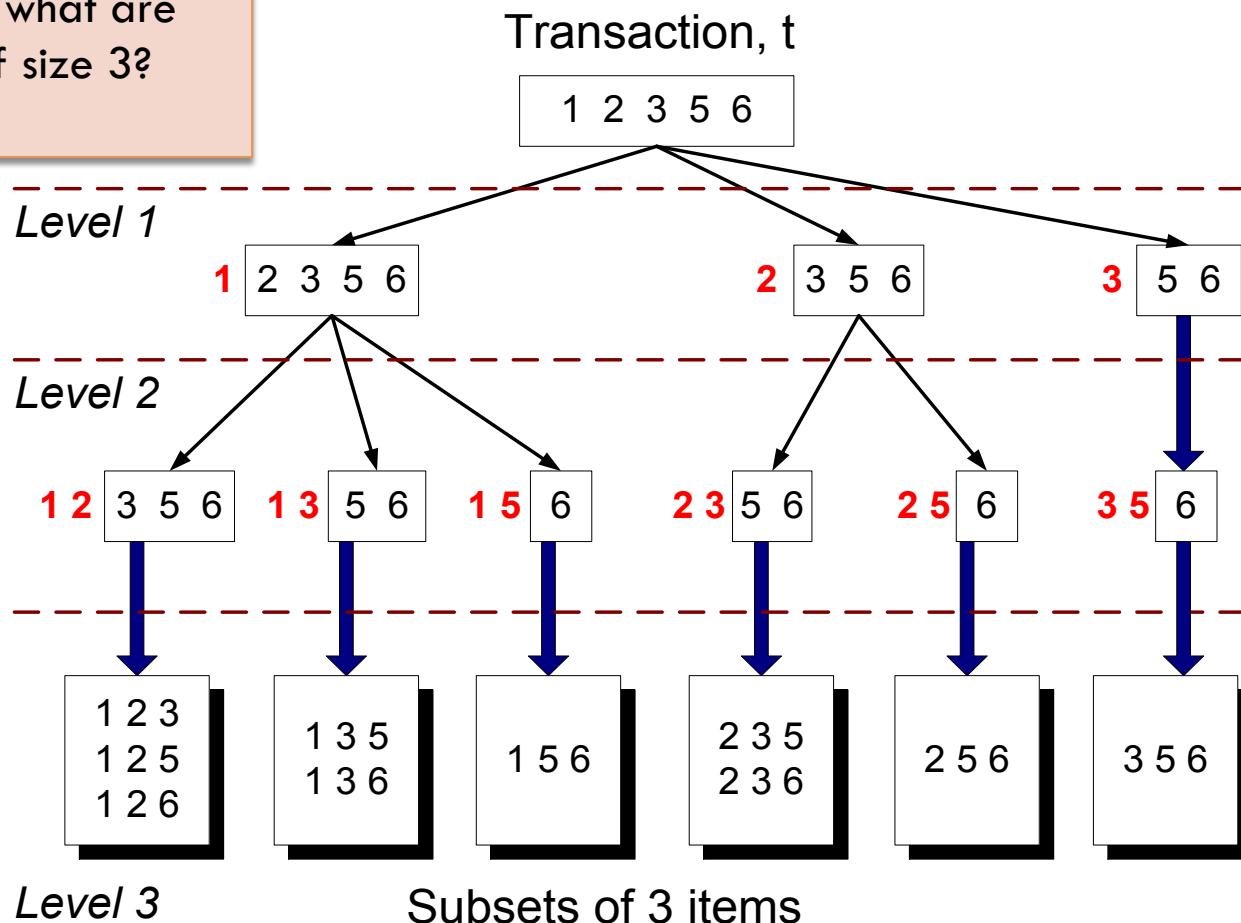
28



# Subset Operation

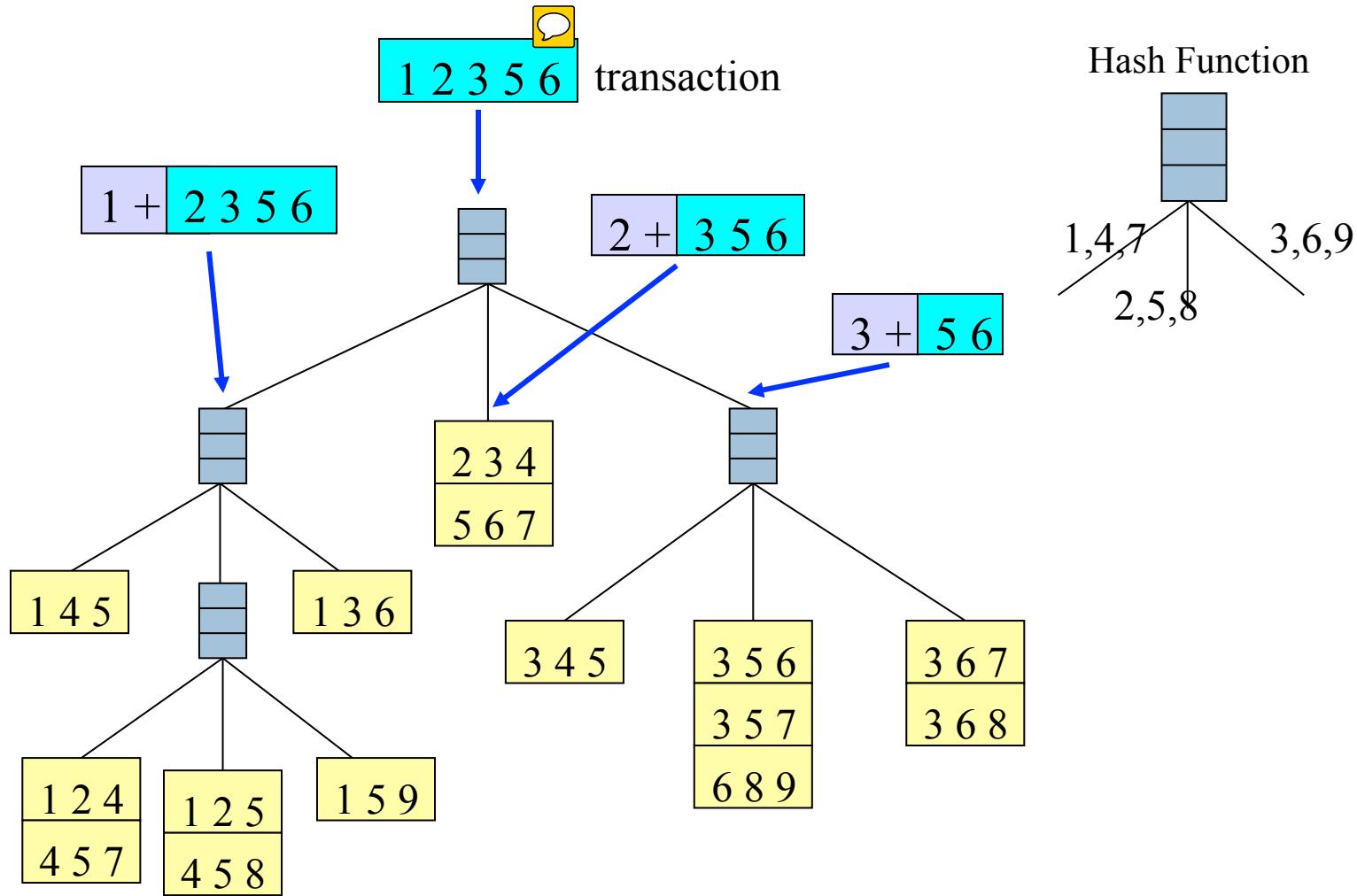
29

Given a transaction  $t$ , what are the possible subsets of size 3?



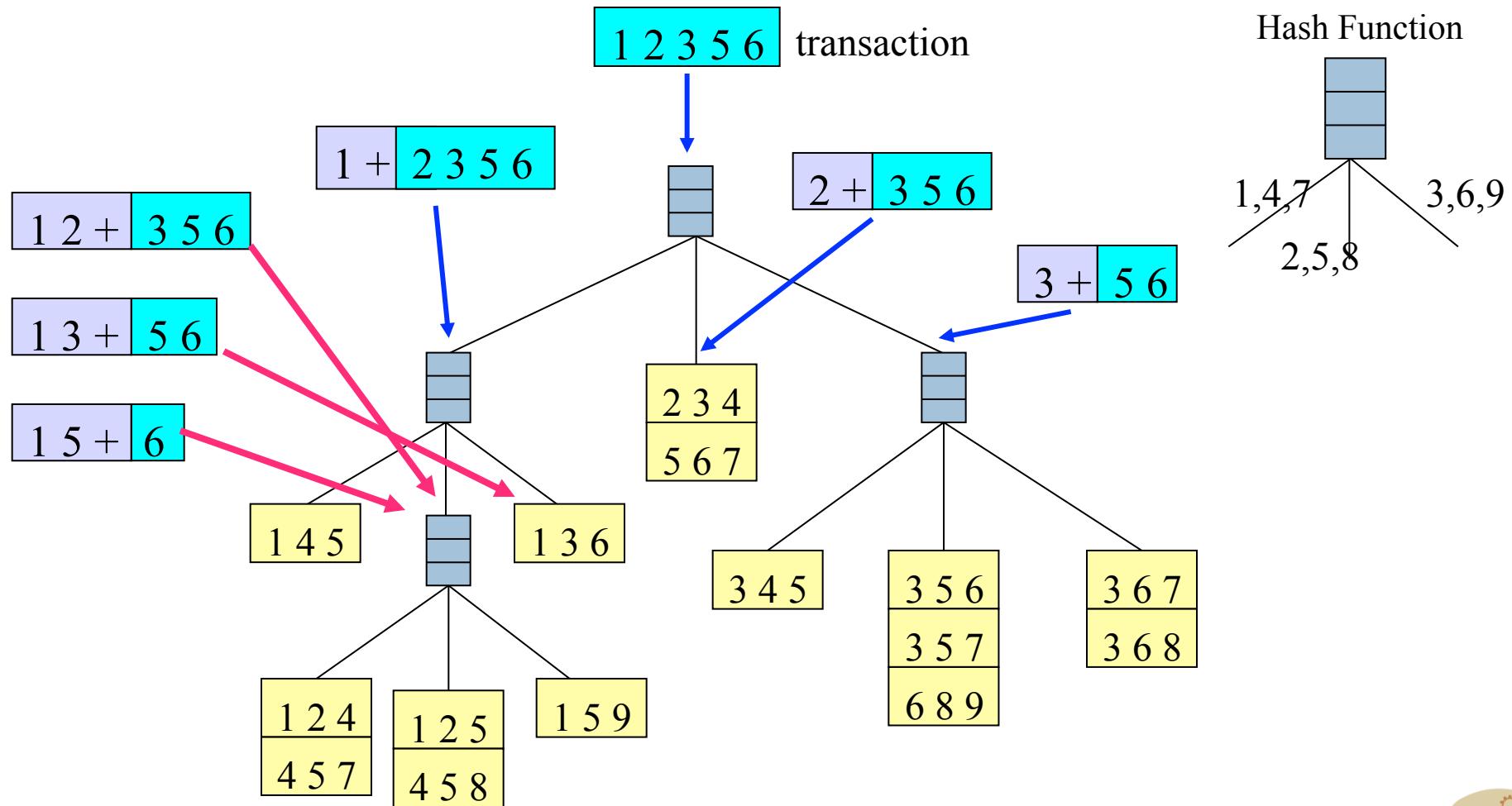
# Subset Operation Using Hash Tree

30



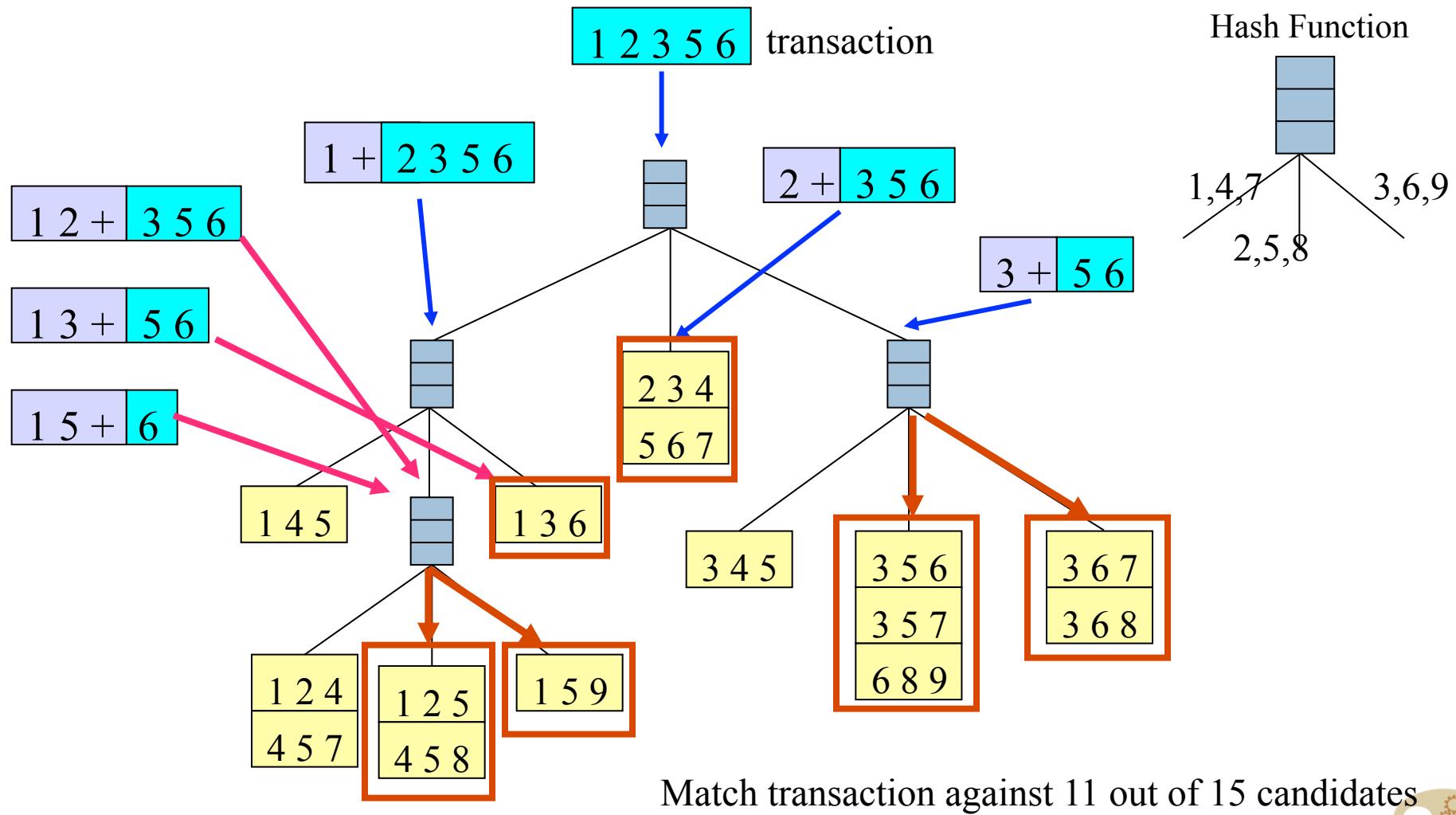
# Subset Operation Using Hash Tree

31



# Subset Operation Using Hash Tree

32



# Implementation Subset Function

33

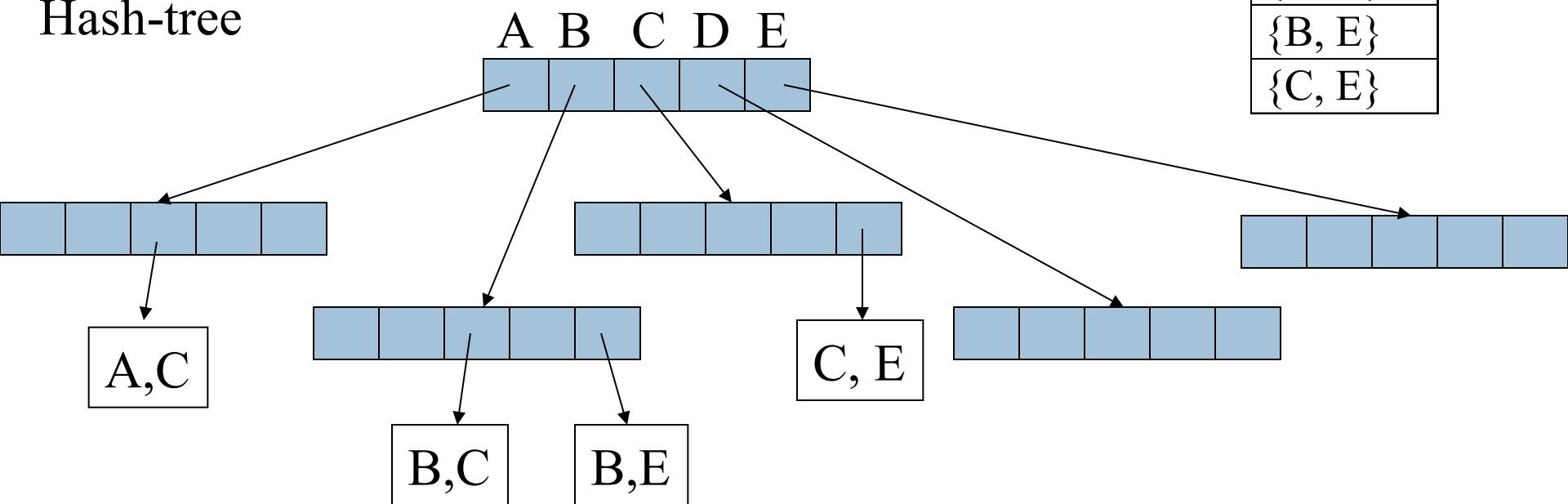
$C_2$

transaction:  $\{B, C, E\}$  has three 2-itemset candidates, i.e.,

$\{\{B, C\}, \{B, E\}, \{C, E\}\}$

Itemset
$\{A, C\}$
$\{B, C\}$
$\{B, E\}$
$\{C, E\}$

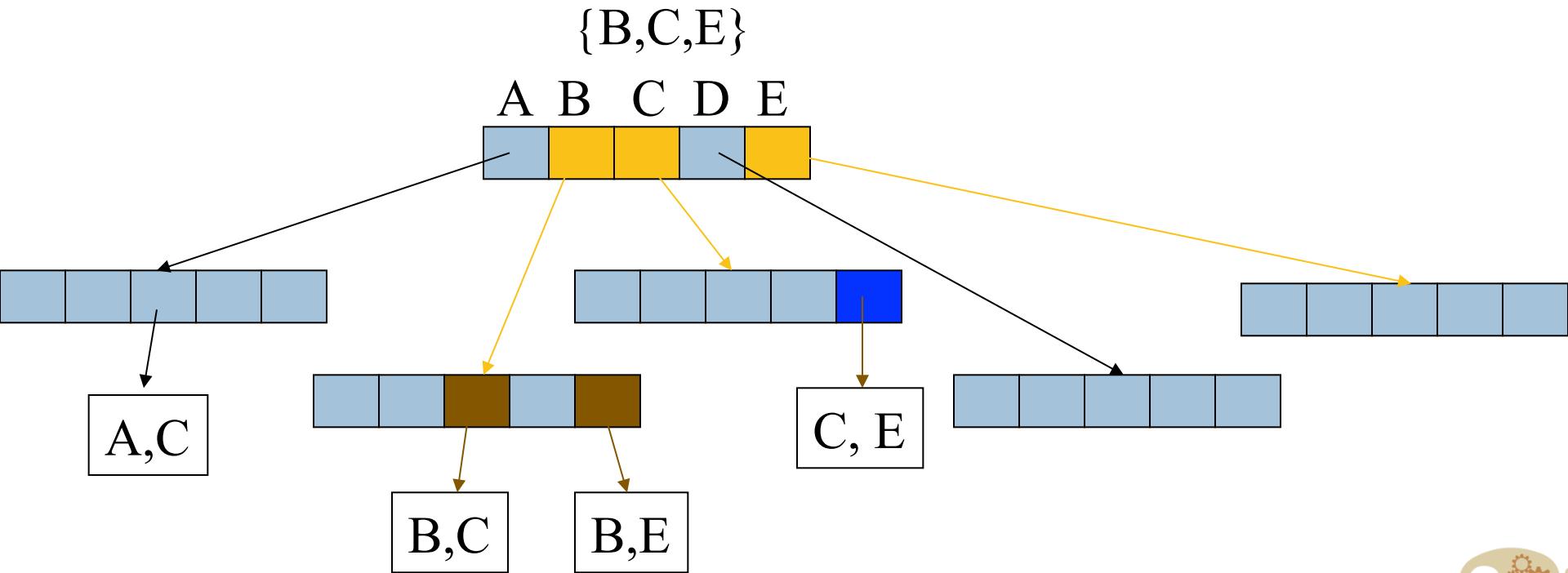
Hash-tree



# Implementation Subset Function (cont.)

34

- For the root, hash on every item in  $t$
- If in the interior node and reach this node by hashing item  $l$ , hash on each item that comes after  $l$  in  $t$  and recursively do.
- If in leaf, add the corresponding itemset into answer set.



# Rules Generation

35

```
For each frequent (large) itemset m do
    for each subset p of m do
        if (support(m)/support(p))>= min_confidence then
            output the rule p→(m-p)
            with confidence=support(m)/support(p)
            support=support(m)
```

$m = \{a, c, d, e, f, g\}$  2000 tx's

$p = \{a, d\}$  5000 tx's

$\{a, d\} \Rightarrow \{c, e, f, g\}$  confidence=40%, support=2000 tx's



# Rule Generation

- Given a frequent itemset  $L$ , find all non-empty subsets  $f \subset L$  such that  $f \rightarrow L - f$  satisfies the minimum confidence requirement

- If  $\{A, B, C, D\}$  is a frequent itemset, candidate rules:

$ABC \rightarrow D,$

$ABD \rightarrow C,$

$ACD \rightarrow B,$

$BCD \rightarrow A,$

$A \rightarrow BCD,$

$B \rightarrow ACD,$

$C \rightarrow ABD,$

$D \rightarrow ABC$

$AB \rightarrow CD,$

$AC \rightarrow BD,$

$AD \rightarrow BC,$

$BC \rightarrow AD,$

$BD \rightarrow AC,$

$CD \rightarrow AB,$

- If  $|L| = k$ , then there are  $2^k - 2$  candidate association rules (ignoring  $L \rightarrow \emptyset$  and  $\emptyset \rightarrow L$ )



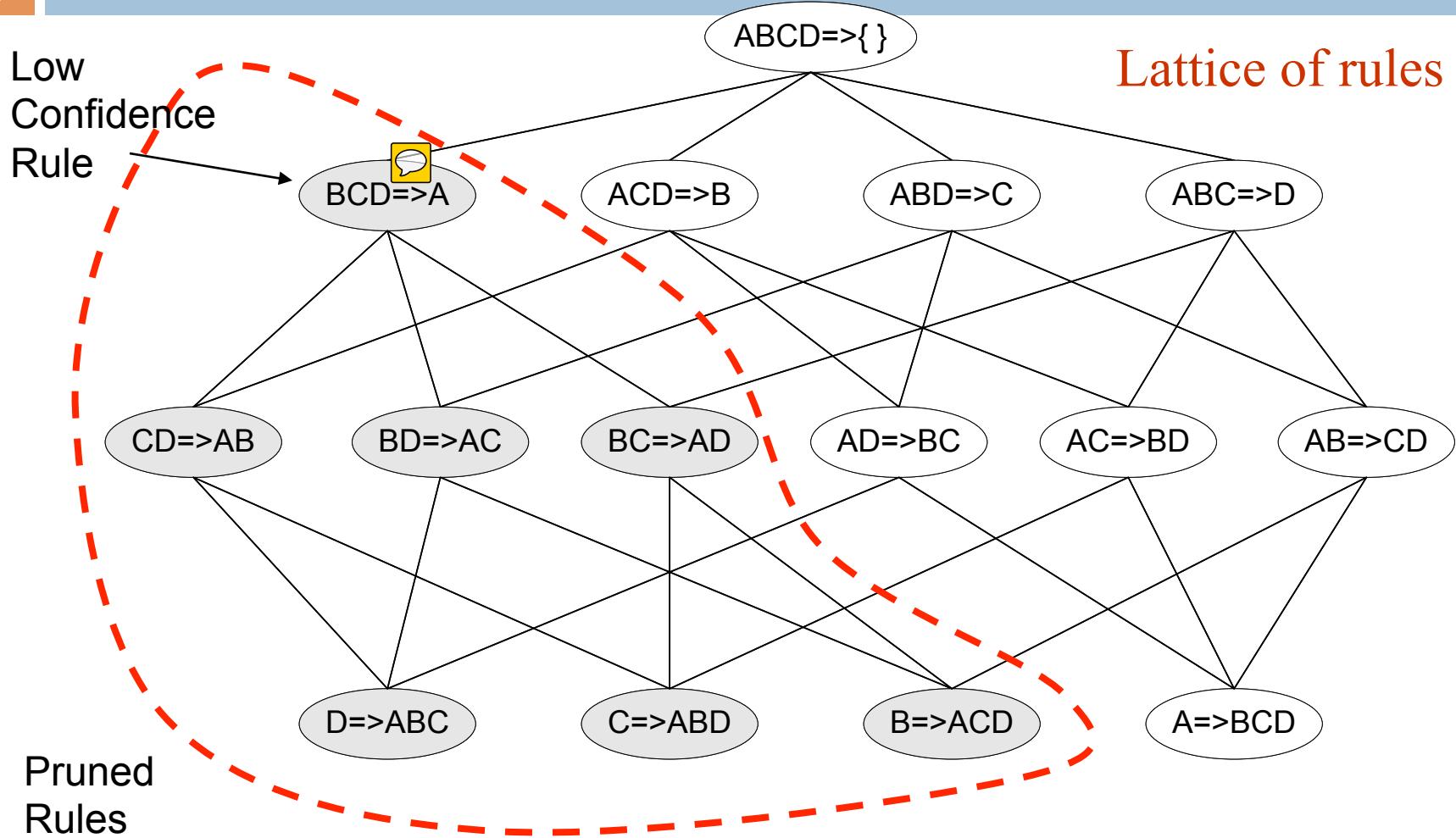
# Rule Generation

37

- How to efficiently generate rules from frequent itemsets?
  - In general, confidence does not have an anti-monotone property  
 $c(ABC \rightarrow D)$  can be larger or smaller than  $c(AB \rightarrow D)$
  - But confidence of rules generated from the same itemset has an anti-monotone property
  - e.g.,  $L = \{A, B, C, D\}$ :  
 $c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$
  - Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

# Rule Generation for Apriori Algorithm

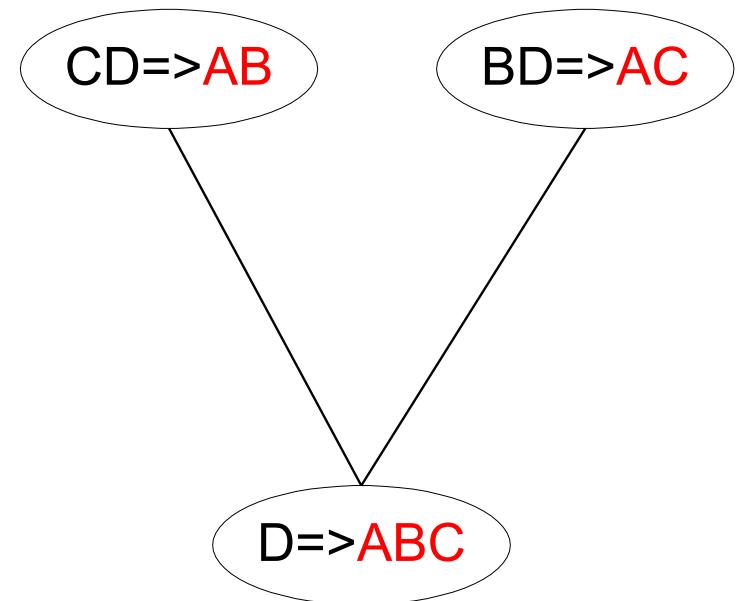
38



# Rule Generation for Apriori Algorithm

39

- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent
- $\text{join}(CD \Rightarrow AB, BD \Rightarrow AC)$  would produce the candidate rule  $D \Rightarrow ABC$
- Prune rule  $D \Rightarrow ABC$  if its subset  $AD \Rightarrow BC$  does not have high confidence



# Redundant Rules

40

- For the same support and confidence, if we have a rule  $\{a,d\} \rightarrow \{c,e,f,g\}$ , do we have
  - $\{a,d\} \rightarrow \{c,e,f\}$
  - $\{a\} \rightarrow \{c,e,f,g\}$
  - $\{a,d,c\} \rightarrow \{e,f,g\}$
  - $\{a\} \rightarrow \{d,c,e,f,g\}$

# Improvement of Apriori Algorithm

41

- Improving Apriori: general ideas
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

# DHP(Direct Hashing & Pruning)

42

- Observation of performance in association rule mining
  - initial candidate set generation is key issue to improve
  - amount of transaction data that must be scanned
- Major features of DHP
  - efficient generation for frequent itemsets
  - effective reduction on transaction database size
  - option of reducing #(database scan) required.

J. Park, M. Chen, and P. Yu. An effective hash-based algorithm for mining association rules.  
In SIGMOD'95



Data Mining

# Efficient Generation of Frequent Itemsets

43

## □ Using hashing

TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E

{A,C}, {A,D}, {C,D}

{B,C}, {B,E}, {C,E}

{A,B}, {A,C}, {A,E}, {B,C}, {B,E}, {C,E}

{B,E}

$$C_2 = L_1 * L_1$$

Support=2

$$h(\{x,y\})=((\text{ord}(x)*10+\text{ord}(y)) \bmod 7$$

{C,E}

{C,E}

{A,D}



{B,C}

{A,E}

{B,C}

{B,E}

{B,E}

{B,E}

{A,C}

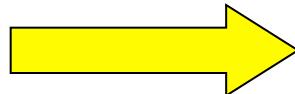
{C,D}

{A,C}

L1={A,B,C,E}

Pruning {A,D}

and {C,D}



C<sub>2</sub>

Itemset

{A, B}

{A, C}

{B, C}

{A, E}

{B, E}

{C, E}

Itemset

{A, B}

{A, C}

{B, C}

{A, E}

{B, E}

{C, E}



# Effective Reduction on Transaction Database Size

44

*A transaction is used to determine  $L_{k+1}$  only if the transaction consists of  $K+1$   $L_k$*

$D_2$

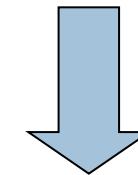
TID	Items
100	A, C, D
200	B, C, E
300	A, B, C, E
400	B, E

{A,C}  
{B,C}, {B,E}, {C,E}  
{A,C}, {B,C}, {B,E}, {C,E}  
{B,E}



$D_3$

TID	Items
200	B, C, E
300	A, B, C, E



TID	Items
200	B, C, E
300	B, C, E

*A qualified item appears in at least  $k$  of  $C_k$  in a transaction.*



# Partitioning

45

- Observation: any potential frequent itemset appears as a frequent itemset in at least one of the partitions.
- Transaction DB is divided into non-overlapping partitions
- Partition size is chosen to be resident in main memory

A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association in large databases. In *VLDB'95*

# Partitioning Algorithm

46

1. Divide D into partitions  $D^1, D^2, \dots, D^p$ ;
2. For  $I = 1$  to  $p$  do
3.      $L^i = \text{Apriori}(D^i)$ ;
4.      $C = L^1 \cup \dots \cup L^p$ ;
5. Count C on D to generate L;

# Partitioning (cont'd)

47

- Two phases scanning
  - **First scan:** generates a set of all **potentially frequent itemsets**
    - Each partition generates the local frequent itemsets
  - **Second scan:** actual support is measured
    - Collection of local frequent itemset = global candidate itemset
    - Global frequent itemsets are found by scan DB



# Partitioning Adv/Disadv

48

## □ ***Advantages:***

- Adapts to available main memory
- Easily parallelized
- Maximum number of database scans is two

## □ ***Disadvantages:***

- May have many candidates during second scan



# Sampling

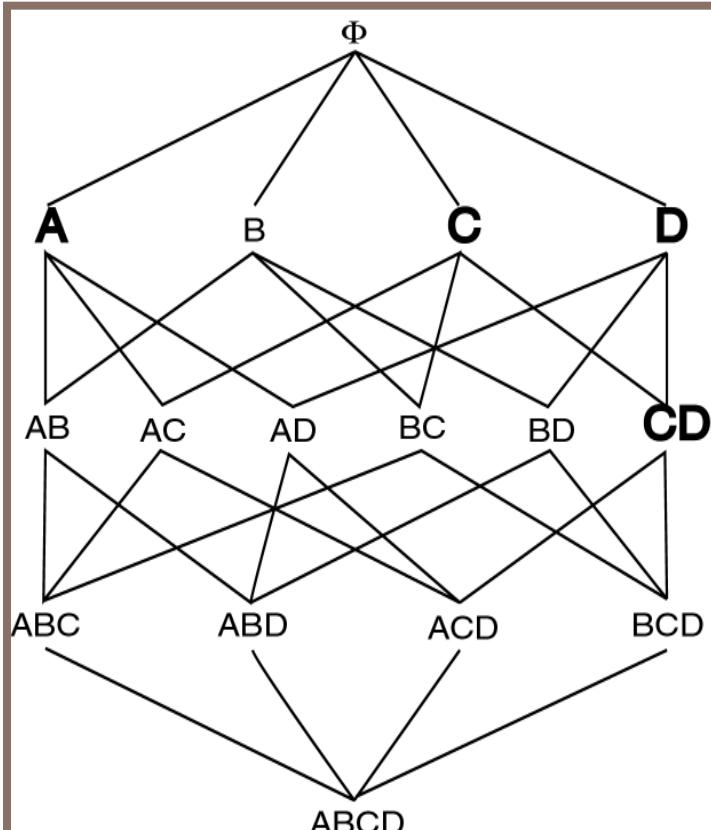
49

- Sample the database and apply Apriori to the sample.
- **Potentially Large Itemsets (PL):** Large itemsets from sample
- **Negative Border ( $BD^-$ ):**
  - Generalization of Apriori-Gen applied to itemsets of varying sizes.
  - Minimal set of itemsets which are not in PL, but whose subsets are all in PL.

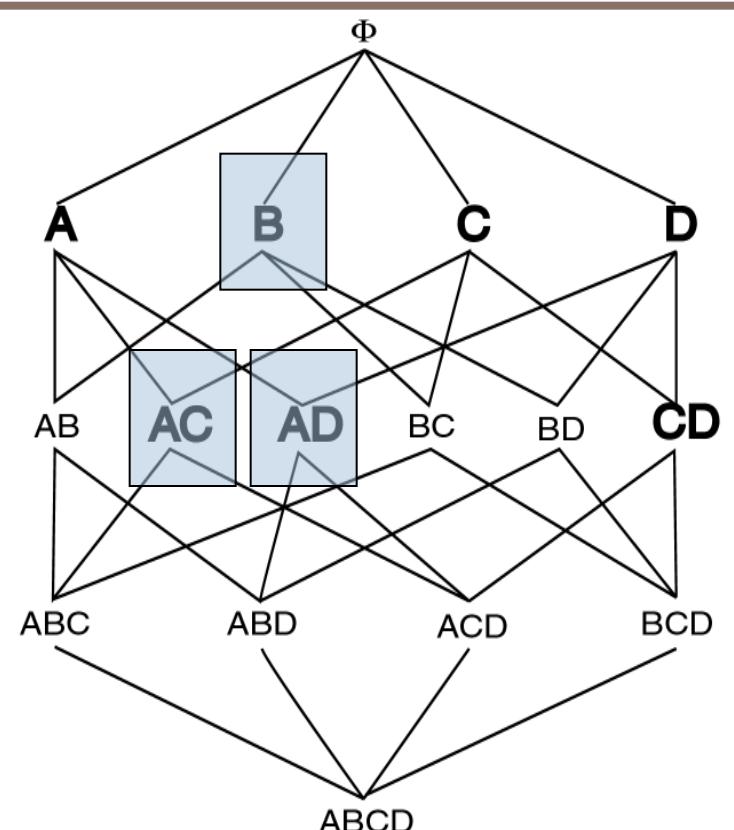
H. Toivonen. Sampling large databases for association rules. In *VLDB'96*

# Negative Border Example

50



$PL$



$PL \cup BD^-(PL)$



Data Mining

# Sampling Algorithm

51

1.  $D_s$  = sample of Database D;
2.  $PL$  = Large itemsets in  $D_s$  using smaller support;
3.  $C = PL \cup BD^-(PL)$ ;
4. Count C in Database using support;
5.  $ML$  = large itemsets in  $BD^-(PL)$ ;
6. If  $ML = \emptyset$  then done
7. else
8.  $C = \text{repeated application of } BD^-;$
9. Count C in Database;

# Sampling Adv/Disadv

52

- **Advantages:**
  - Reduces number of database scans to one in the best case and two in worst
  - Scales better
- **Disadvantages:**
  - Potentially large number of candidates in second pass

# Apriori Example

53

Transaction	Items
$t_1$	Bread, Jelly, PeanutButter
$t_2$	Bread, PeanutButter
$t_3$	Bread, Milk, PeanutButter
$t_4$	Beer, Bread
$t_5$	Beer, Milk



# Sampling Example

54

- Find AR assuming  $s = 20\%$
- $D_s = \{ t_1, t_2 \}$
- Smalls = 10%
- $PL = \{\{\text{Bread}\}, \{\text{Jelly}\}, \{\text{PeanutButter}\}, \{\text{Bread}, \text{Jelly}\}, \{\text{Bread}, \text{PeanutButter}\}, \{\text{Jelly}, \text{PeanutButter}\}, \{\text{Bread}, \text{Jelly}, \text{PeanutButter}\} \}$
- $BD^-(PL) = \{\{\text{Beer}\}, \{\text{Milk}\}\}$
- $ML = \{\{\text{Beer}\}, \{\text{Milk}\}\}$
- Repeated application of  $BD^-$  generates all remaining itemsets

# Bottleneck of Frequent-pattern Mining

55

- Multiple database scans are **costly**
- Mining long patterns needs many passes of scanning and generates lots of candidates
  - To find frequent itemset  $i_1 i_2 \dots i_{100}$ 
    - # of scans: **100**
    - # of Candidates:  $C^{100}_1 + C^{100}_2 + \dots + C^{100}_{100} = 2^{100}-1 = 1.27*10^{30}!$
- Bottleneck: candidate-generation-and-test
- Can we avoid candidate generation?

# FP-Growth

56

## □ Motivation

- Mining in main memory to reduce #(DB scans)
- Without candidate generation
- More frequently occurring items will have better chances of sharing item than less frequently occurring items

J. Han, J. Pei, and Y. Yin: "Mining frequent patterns without candidate generation". In Proc. ACM-SIGMOD'2000, pp. 1-12, Dallas, TX, May 2000.

# FP-Growth (cont'd)

57

- Frequent pattern Growth
- Divide-and-conquer strategy
- Algorithm
  - Phase 1: Construct FP-Tree (frequent-pattern tree)
  - Phase 2: FP-Growth (frequent pattern growth)
    - Divide FP-tree into conditional FP-tree (conditional DB), each associated with one frequent item
    - Mine each such DB separately

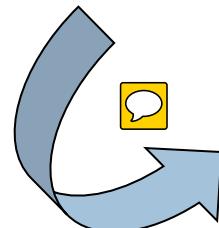
# FP-Trees Construction

58

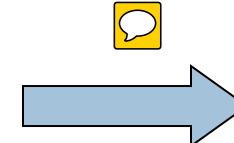
- Step 1: Find frequent 1-item, sorted items in frequency descending order by scanning DB

TID	Items bought
100	{a, c, d, f, g, i, m, p}
200	{a, b, c, f, i, m, o}
300	{b, f, h, j, o}
400	{b, c, k, s, p}
500	{a, c, e, f, l, m, n, p}

$min\_support = 3$



a	3
b	3
c	4
f	4
m	3
p	3



f	4
c	4
a	3
b	3
m	3
p	3

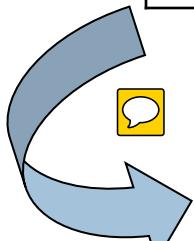


# FP-Trees Construction (cont.)

59

Step 2: Scan DB and construct the FP-tree

f	4
c	4
a	3
b	3
m	3
p	3

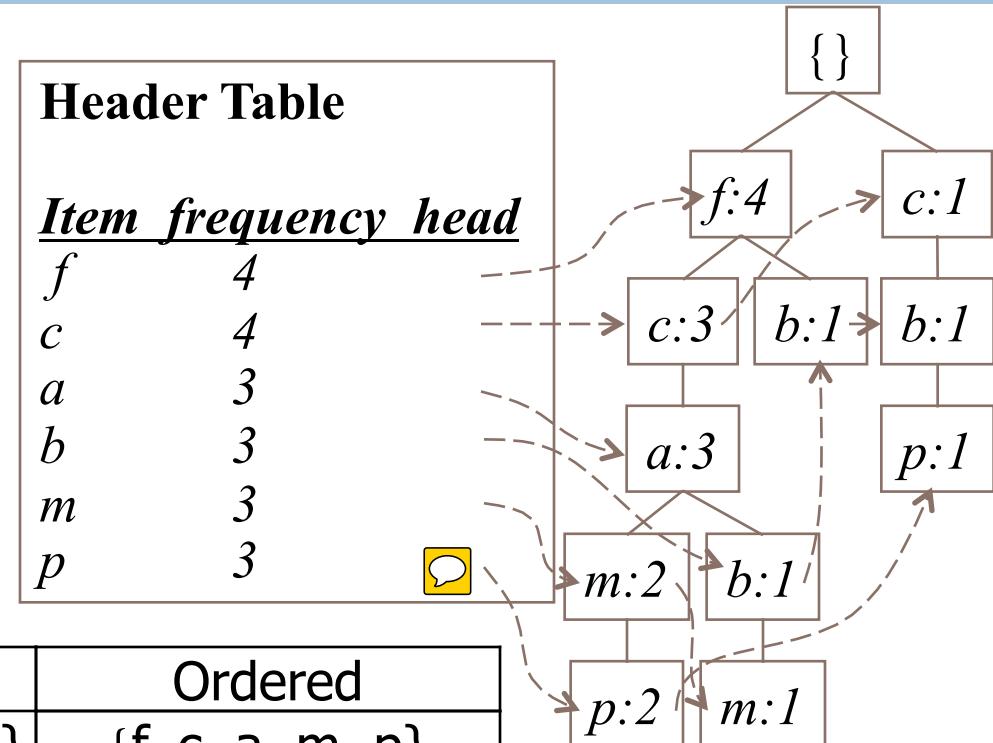


**Header Table**

*Item frequency head*

f	4
c	4
a	3
b	3
m	3
p	3

TID	Items bought	Ordered
100	{a, c, d, f, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, i, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, c, e, f, l, m, n, p}	{f, c, a, m, p}



**TID (ordered) frequent items**

100  $\{f, c, a, m, p\}$

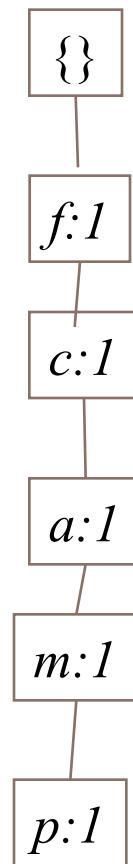
200  $\{f, c, a, b, m\}$

300  $\{f, b\}$

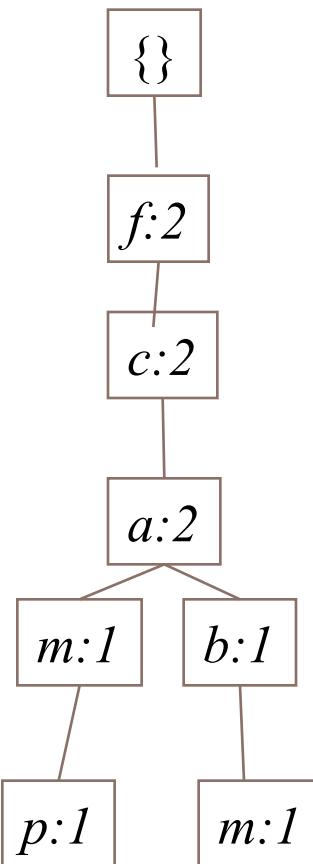
400  $\{c, b, p\}$

500  $\{f, c, a, m, p\}$

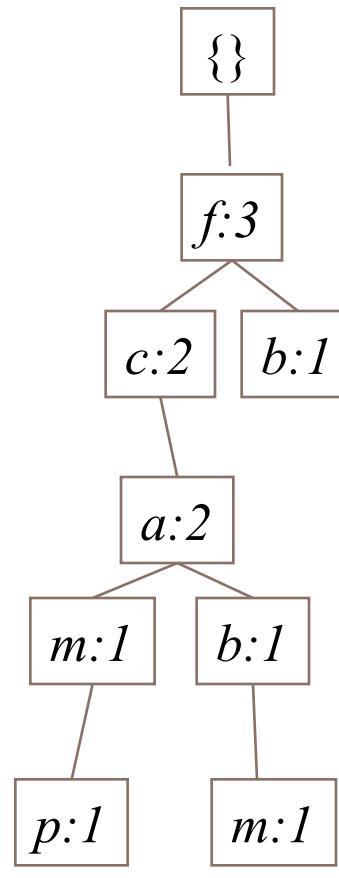
TID:100



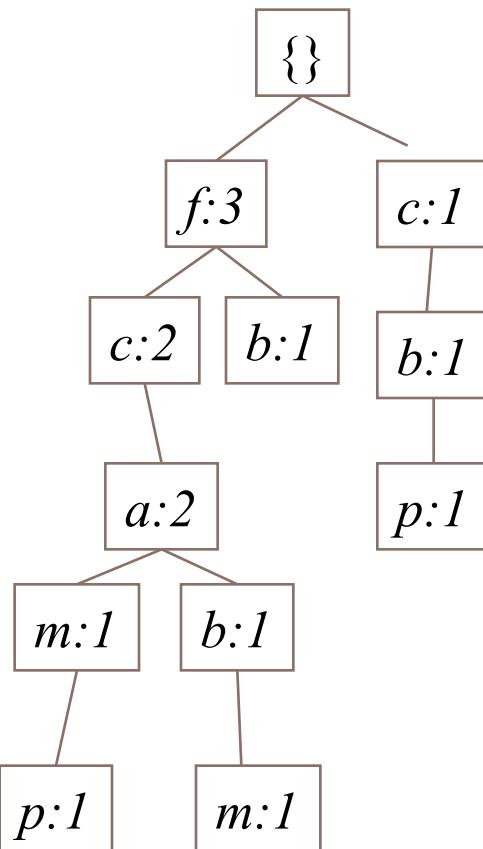
TID:200



TID:300



TID:400



# FP-Growth Overview

61

- Start from each **frequent 1-pattern** (initial suffix pattern), construct **conditional pattern base**
  - **Conditional base**: the set of prefix paths in FP-tree co-occurring with the suffix pattern
- Constructs corresponding **conditional FP-tree**
- Mining recursively on such tree.
- Pattern growth is achieved by the **concatenation of suffix pattern** with frequent patterns generated from conditional FP-tree

# FP-Growth Overview (cont'd)

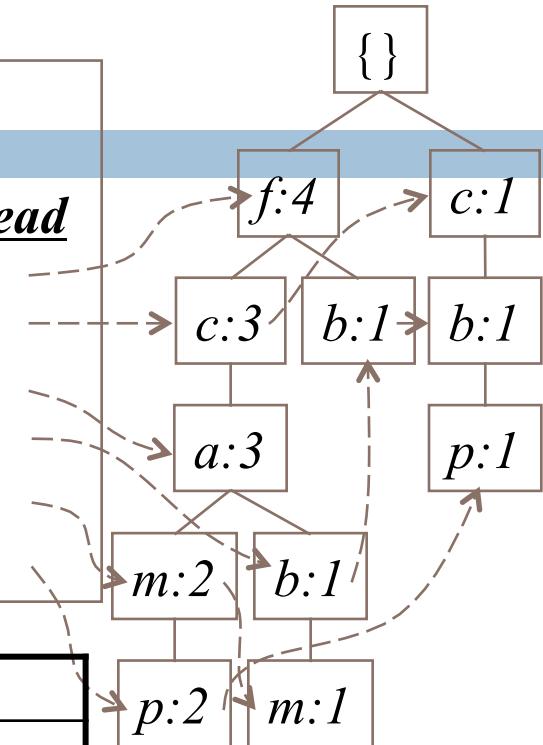
62

*min\_support = 3*

Header Table

Item frequency head

f	4
c	4
a	3
b	3
m	3
p	3



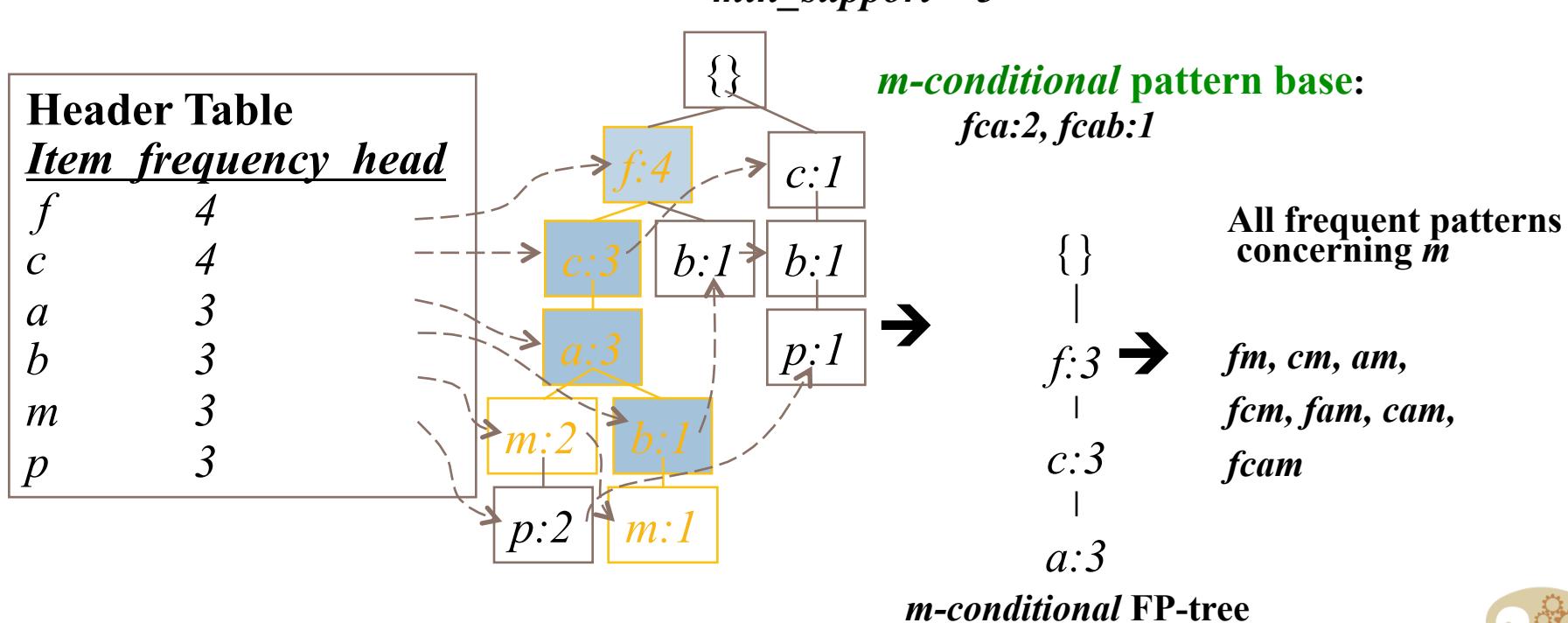
Item	Cond. Pattern base	Cond. FP-tree	Frequent patterns
c	f:3	f:3	fc:3
a	fc:3	fc:3	fca:3,fa:3,ca:3
b	fca:1,f:1,c:1		
m	fca:2,fcab:1	fca:3	fm:3,cm:3,am:3,fcm:3,fam:3,cam:3,fcam:3
p	fcam:2,cb:1	c:3	cp:3



# Construct Conditional FP-tree

63

- For each pattern-base
  - Accumulate the count for each item in the base
  - Construct the conditional FP-tree for the frequent items of the pattern base



# Construct Conditional FP-tree (cont'd)

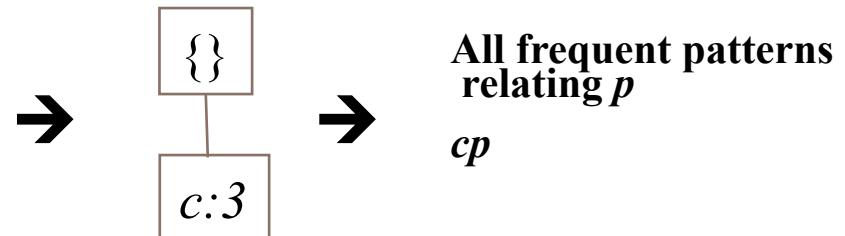
64

Before	Sorted
fcam:2	c:2
cb:1	c:1

□ p's cond. pattern base: fcam:2,cb:1

- Accumulate the count for each item
  - f:2 c:3,a:2,m:2,b:1
- Sort frequent items in count descending order
- Construct cond. FP tree

Min\_support=3



# Example

65

- $\{a, b\}$ ,  $\{b, c, d\}$ ,  $\{a, c, d, e\}$ ,  $\{a, d, e\}$ ,  $\{a, b, c\}$ ,  $\{a, b, c, d\}$ ,  $\{a\}$ ,  $\{a, b, c\}$ ,  $\{a, b, d\}$ ,  $\{b, c, e\}$ , and  
 $\text{min\_sup}=0.2$

# Principles of Frequent Pattern Growth

66

- Pattern growth property
  - Let  $\alpha$  be a frequent itemset in DB, B be  $\alpha$ 's conditional pattern base, and  $\beta$  be an itemset in B. Then  $\alpha \cup \beta$  is a frequent itemset in DB iff  $\beta$  is frequent in B.
- “abcdef” is a frequent pattern, if and only if
  - “abcde” is a frequent pattern, and
  - “f” is frequent in the set of transactions containing “abcde”

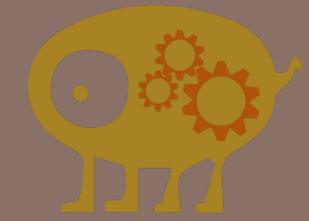


# Why Is FP-Growth the Winner?

67

- Divide-and-conquer:
  - Decompose both the mining task and DB according to the frequent patterns obtained so far
  - Leads to focused search of smaller databases
- Other factors
  - No candidate generation, no candidate test
  - Compressed database: FP-tree structure
  - No repeated scan of entire database
  - Basic ops—counting local freq items and building sub FP-tree, no pattern search and matching





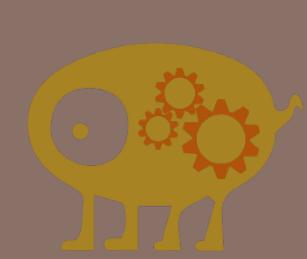
# PRESENTATION OF ASSOCIATION RULES

# Presentation of Association Rules

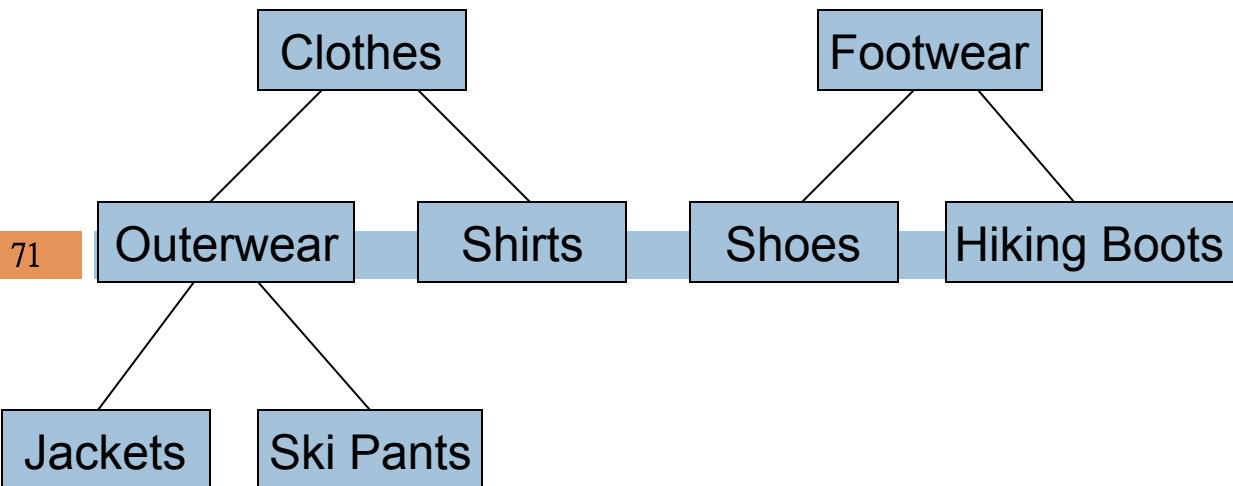
69

	<b>Body</b>	<b>Implies</b>	<b>Head</b>	<b>Supp (%)</b>	<b>Conf (%)</b>	F	G	H	I	J
1	cost(x) = '0.00~1000.00'	$\implies$	revenue(x) = '0.00~500.00'	28.45	40.4					
2	cost(x) = '0.00~1000.00'	$\implies$	revenue(x) = '500.00~1000.00'	20.46	29.05					
3	cost(x) = '0.00~1000.00'	$\implies$	order_qty(x) = '0.00~100.00'	59.17	84.04					
4	cost(x) = '0.00~1000.00'	$\implies$	revenue(x) = '1000.00~1500.00'	10.45	14.84					
5	cost(x) = '0.00~1000.00'	$\implies$	region(x) = 'United States'	22.56	32.04					
6	cost(x) = '1000.00~2000.00'	$\implies$	order_qty(x) = '0.00~100.00'	12.91	69.34					
7	order_qty(x) = '0.00~100.00'	$\implies$	revenue(x) = '0.00~500.00'	28.45	34.54					
8	order_qty(x) = '0.00~100.00'	$\implies$	cost(x) = '1000.00~2000.00'	12.91	15.67					
9	order_qty(x) = '0.00~100.00'	$\implies$	region(x) = 'United States'	25.9	31.45					
10	order_qty(x) = '0.00~100.00'	$\implies$	cost(x) = '0.00~1000.00'	59.17	71.86					
11	order_qty(x) = '0.00~100.00'	$\implies$	product_line(x) = 'Tents'	13.52	16.42					
12	order_qty(x) = '0.00~100.00'	$\implies$	revenue(x) = '500.00~1000.00'	19.67	23.88					
13	product_line(x) = 'Tents'	$\implies$	order_qty(x) = '0.00~100.00'	13.52	98.72					
14	region(x) = 'United States'	$\implies$	order_qty(x) = '0.00~100.00'	25.9	81.94					
15	region(x) = 'United States'	$\implies$	cost(x) = '0.00~1000.00'	22.56	71.39					
16	revenue(x) = '0.00~500.00'	$\implies$	cost(x) = '0.00~1000.00'	28.45	100					
17	revenue(x) = '0.00~500.00'	$\implies$	order_qty(x) = '0.00~100.00'	28.45	100					
18	revenue(x) = '1000.00~1500.00'	$\implies$	cost(x) = '0.00~1000.00'	10.45	96.75					
19	revenue(x) = '500.00~1000.00'	$\implies$	cost(x) = '0.00~1000.00'	20.46	100					
20	revenue(x) = '500.00~1000.00'	$\implies$	order_qty(x) = '0.00~100.00'	19.67	96.14					
21										
22										
23	cost(x) = '0.00~1000.00'	$\implies$	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.45	40.4					
24	cost(x) = '0.00~1000.00'	$\implies$	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.45	40.4					
25	cost(x) = '0.00~1000.00'	$\implies$	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	19.67	27.93					
26	cost(x) = '0.00~1000.00'	$\implies$	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	19.67	27.93					
27	cost(x) = '0.00~1000.00' AND order_qty(x) = '0.00~100.00'	$\implies$	revenue(x) = '500.00~1000.00'	19.67	33.23					





# MULTILEVEL ASSOCIATION RULES



Tx	Items bought
100	Shirt
200	Jacket, Hiking Boots
300	Ski Pants, Hiking Boots
400	Shoes
500	Shoes
600	Jacket

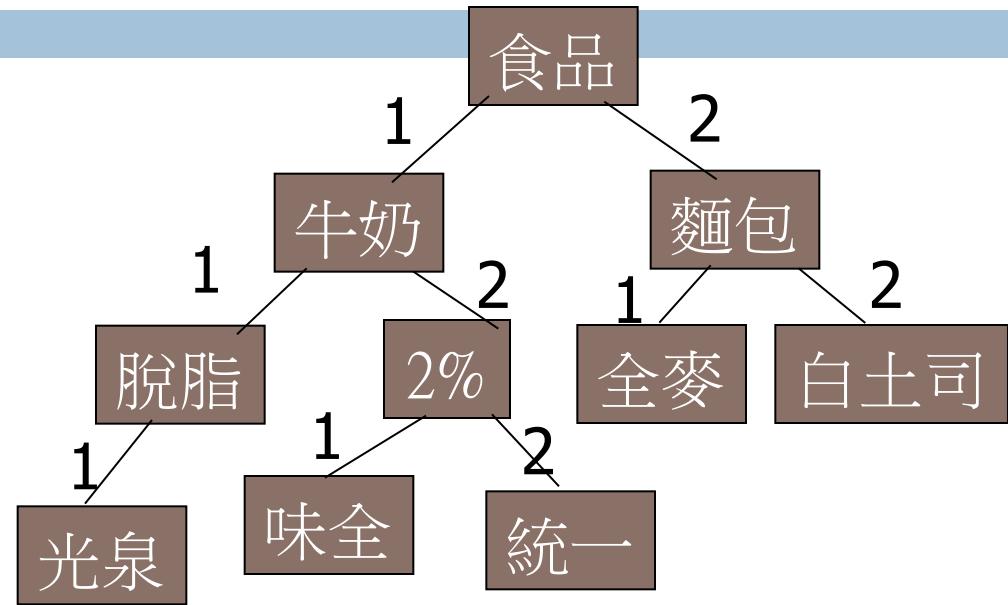
Freq. pattern	Support
Jacket	2
Outerwear	3
Clothes	4
Shoes	2
Hiking Boots	2
Footwear	4
OW, HB	2
Clothes, HB	2
OW, FW	2
Clothes, FW	2

	sup(30%)	conf(60%)
OW -> HB	33%	66%
OW -> FW	33%	66%
HB -> OW	33%	100%
HB -> Clothes	33%	100%
Jacket -> HB	16%	50%
Ski Pants -> HB	16%	100% Data

# Multiple-Level Association Rules

72

- Items often form hierarchy
- Items at the lower level are expected to have lower support
- Rules regarding itemsets at appropriate levels could be quite useful
- Transaction database can be encoded based on dimensions and levels



TID	Items
T1	{111, 121, 211, 221}
T2	{111, 211, 222, 323}
T3	{112, 122, 221, 411}
T4	{111, 121}
T5	{111, 122, 211, 221, 413}



# Mining Multi-Level Associations

73

- A top down, progressive deepening approach:
  - First find high-level strong rules:
$$\text{milk} \rightarrow \text{bread } [20\%, 60\%].$$
  - Then find their lower-level “weaker” rules:
$$2\% \text{ milk} \rightarrow \text{wheat bread } [6\%, 50\%].$$
- Variations at mining multiple-level association rules
  - Level-crossed association rules:
$$2\% \text{ milk} \rightarrow \text{wheat bread}$$
  - Association rules with multiple, alternative hierarchies:
$$2\% \text{ milk} \rightarrow \text{bread}$$



# Uniform Support

74

- Multi-level mining with uniform support

**Level 1**  
**min\_sup = 5%**

**Milk**  
**[support = 10%]**

**Level 2**  
**min\_sup = 5%**

**2% Milk**  
**[support = 6%]**

**Skim Milk**  
**[support = 4%]**

**Level 2**  
**min\_sup = 3%**

# Uniform Support (cont'd)

75

- Uniform Support: the same minimum support for all levels
  - ▣ + No need to examine itemsets containing any item whose ancestors do not have minimum support.
  - ▣ – Lower level items do not occur as frequently. If support threshold
    - too high  $\Rightarrow$  *miss low level associations.*
    - too low  $\Rightarrow$  *generate too many high level associations.*

# Search strategies in Reduced Support

76

- Reduced Support: reduced minimum support at lower levels
  - 4 search strategies:
    - Level-by-level independent
      - Each node is examined, regardless of whether or not its parent node is found to be frequent.
    - Level-cross filtering by single item
      - An item at the i-th level is examined iff its parent node at the (i-1)-th level is frequent
    - Level-cross filtering by k-itemset
      - Check parent k-itemset
    - Controlled level-cross filtering by single item



# Controlled Level-cross Filtering by Single Item

77

**Level 1**  
 $\text{min\_sup} = 12\%$   
 $\text{Level\_passage\_sup} = 8\%$

**Milk**  
[**support = 10%**]

**Level 2**  
 $\text{min\_sup} = 3\%$

**2% Milk**  
[**support = 6%**]

**Skim Milk**  
[**support = 4%**]

# ML Associations with Flexible Support Constraints

78

- Why flexible support constraints?
  - Real life occurrence frequencies vary greatly
    - Diamond, watch, pens in a shopping basket
  - Uniform support may not be an interesting model
- A flexible model
  - The lower-level, the more dimension combination, and the long pattern length, usually the smaller support
  - Special items and special group of items may be specified individually and have higher priority



# Max-patterns

79

- Max-pattern: frequent patterns without proper frequent super pattern
  - BCDE, ACD are max-patterns
  - BCD is not a max-pattern

Min\_sup=2

Tid	Items
10	A,B,C,D,E
20	B,C,D,E,
30	A,C,D,F

R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98, 85-93, Seattle, Washington.



Data Mining

# Closed Association Rules

80

- Large number of frequent itemsets (especially when the support threshold is **low**) and a huge number of association rules
- **Frequent closed itemset:** An itemset  $X$  is a **closed itemset** if there exists no itemset  $Y$  such that
  - $Y$  is a proper superset of  $X$
  - every transaction containing  $X$  also contains  $Y$

N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT'99, 398-416, Jerusalem, Israel, Jan. 1999.

# Closed Association Rules

81

- **Association rule on frequent closed itemsets:** Rule  $X \Rightarrow Y$  is an association rule on frequent closed itemsets if
  - (1) both  $X$  and  $X \cup Y$  are frequent closed itemsets.
  - (2) there does not exist frequent closed itemset  $Z$  such that  $X \subset Z \subset (X \cup Y)$ .
  - (3) the confidence of the rule passes the given min. conf

# Closed Association Rules (cont'd)

82

TID	Items
100	a,c,d,e,f
200	a,b,e
300	c,e,f
400	a,c,d,f
500	c,e,f

Given minimum support 2

Total frequent itemsets: 20: {a}, {c}, {d}, {e}, {f},  
, {a,c}, {a,d}, {a,e}, {a,f}, {c,d}, {c,e}, {c,f}, {d,f},  
, {e,f}, {a,c,d}, {a,c,f}, {a,d,f}, {c,d,f}, {c.e.f}  
, {a,c,d,f}

Closed frequent itemsets:

{a, c, d, f}, {c, e, f}, {a, e}, {c, f}, {a}, {e}

Given minimum confidence 50%,

Closest association rule

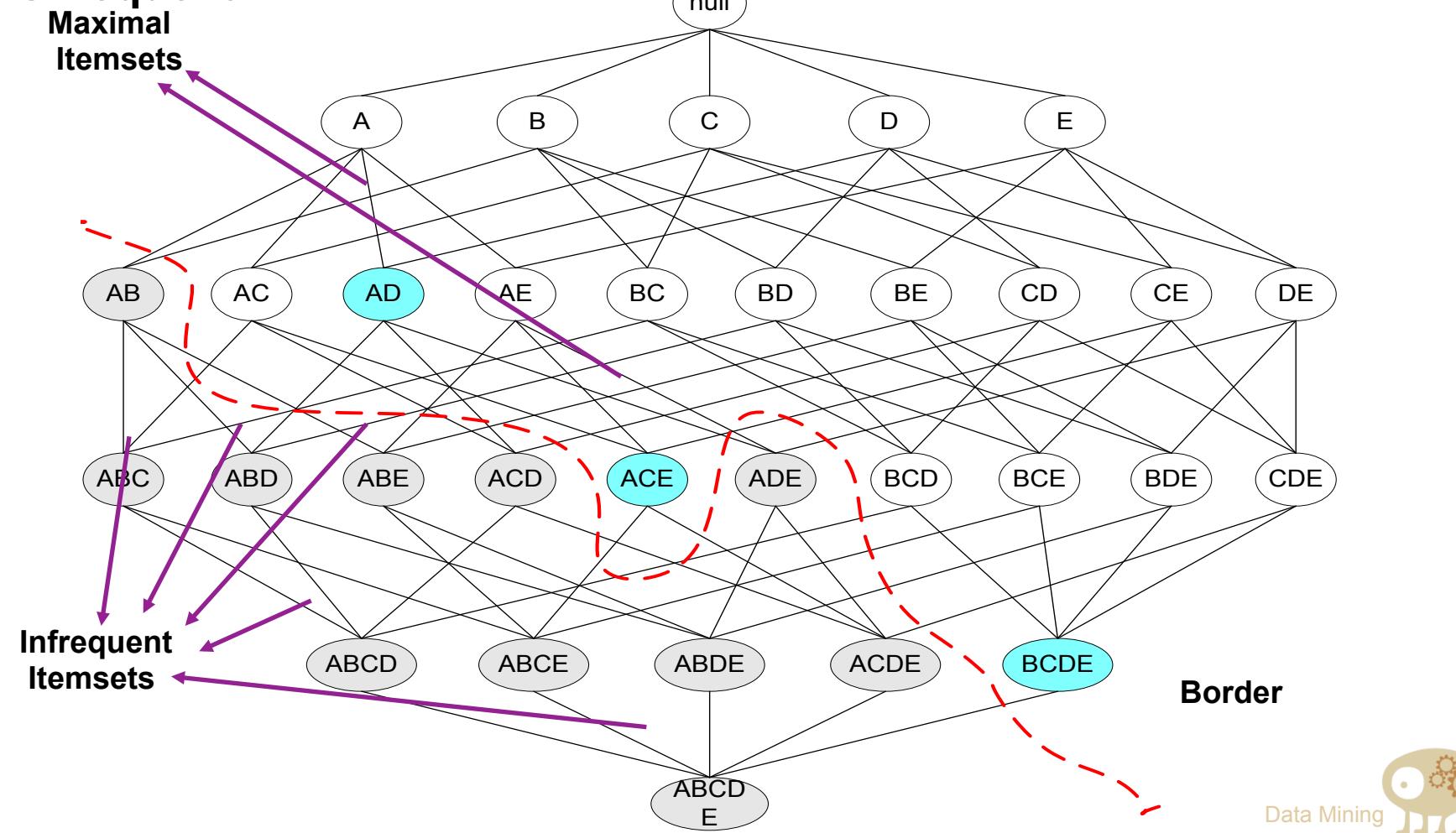
$\{c, f\} \Rightarrow \{a, d\}$  (2,50%),  $\{a\} \Rightarrow \{c, d, f\}$  (2,67%),  
 $\{e\} \Rightarrow \{c, f\}$  (3,75%),  $\{c, f\} \Rightarrow \{e\}$  (3,75%),  
 $\{e\} \Rightarrow \{a\}$  (2,50%),  $\{a\} \Rightarrow \{e\}$  (2,67%)



# Maximal Frequent Itemset

83

An itemset is maximal frequent if none of its immediate supersets is frequent



# Closed Itemset

- An itemset is closed if none of its immediate supersets has the same support as the itemset

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,B,C,D}
4	{A,B,D}
5	{A,B,C,D}

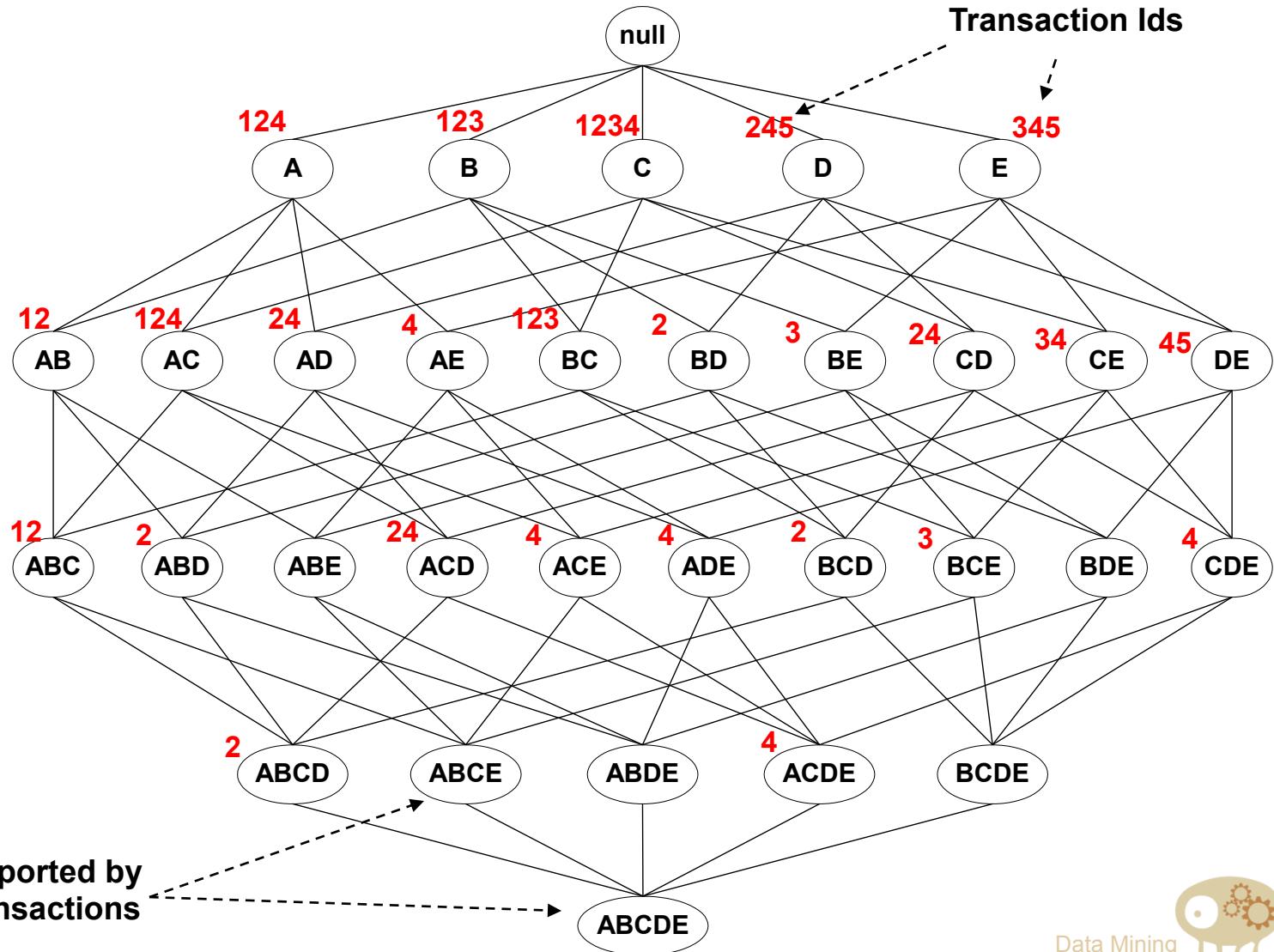
Itemset	Support
{A}	4
{B}	5
{C}	3
{D}	4
{A,B}	4
{A,C}	2
{A,D}	3
{B,C}	3
{B,D}	4
{C,D}	3

Itemset	Support
{A,B,C}	2
{A,B,D}	3
{A,C,D}	2
{B,C,D}	3
{A,B,C,D}	2

# Maximal vs Closed Itemsets

85

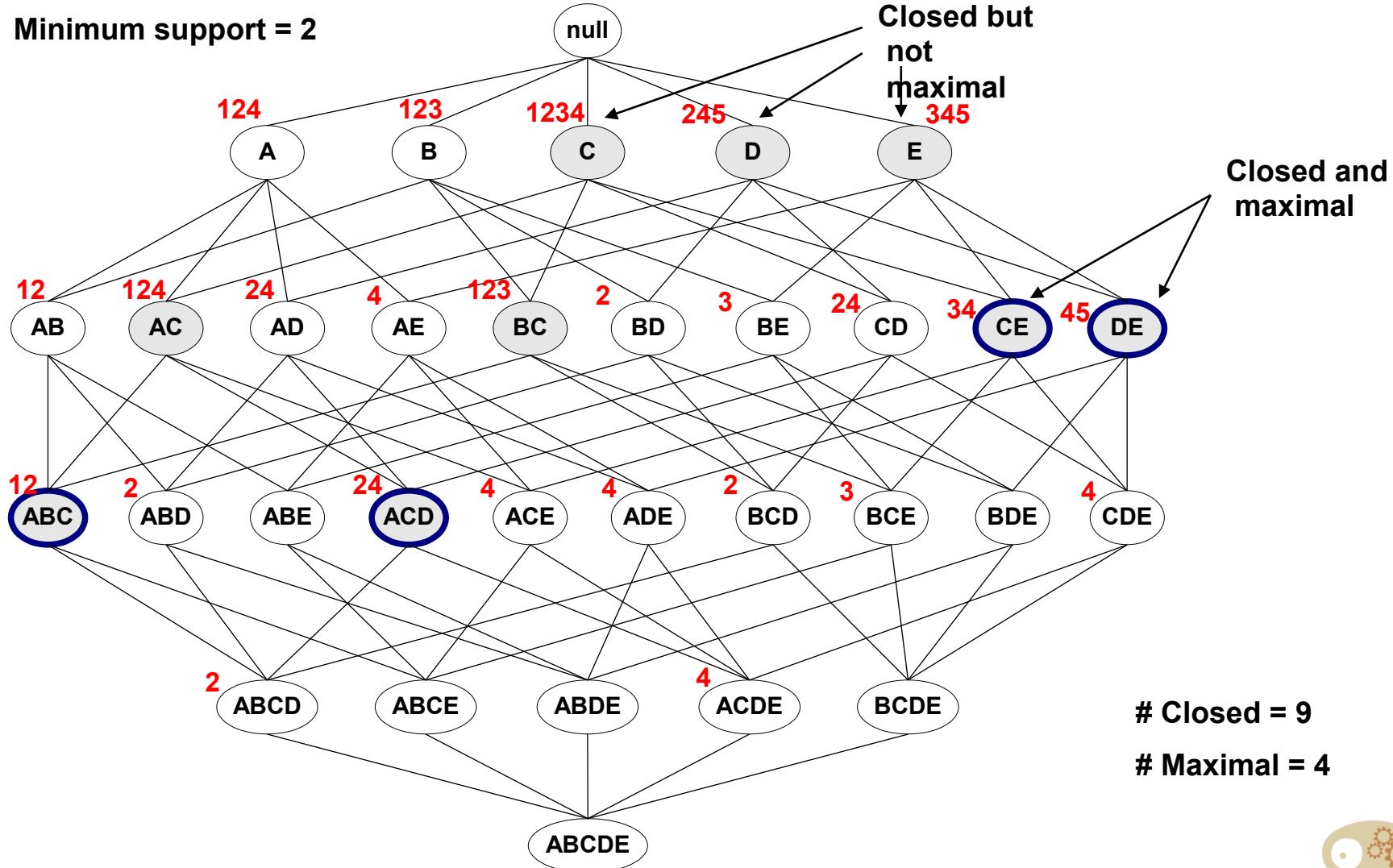
TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE



# Maximal vs Closed Frequent Itemsets

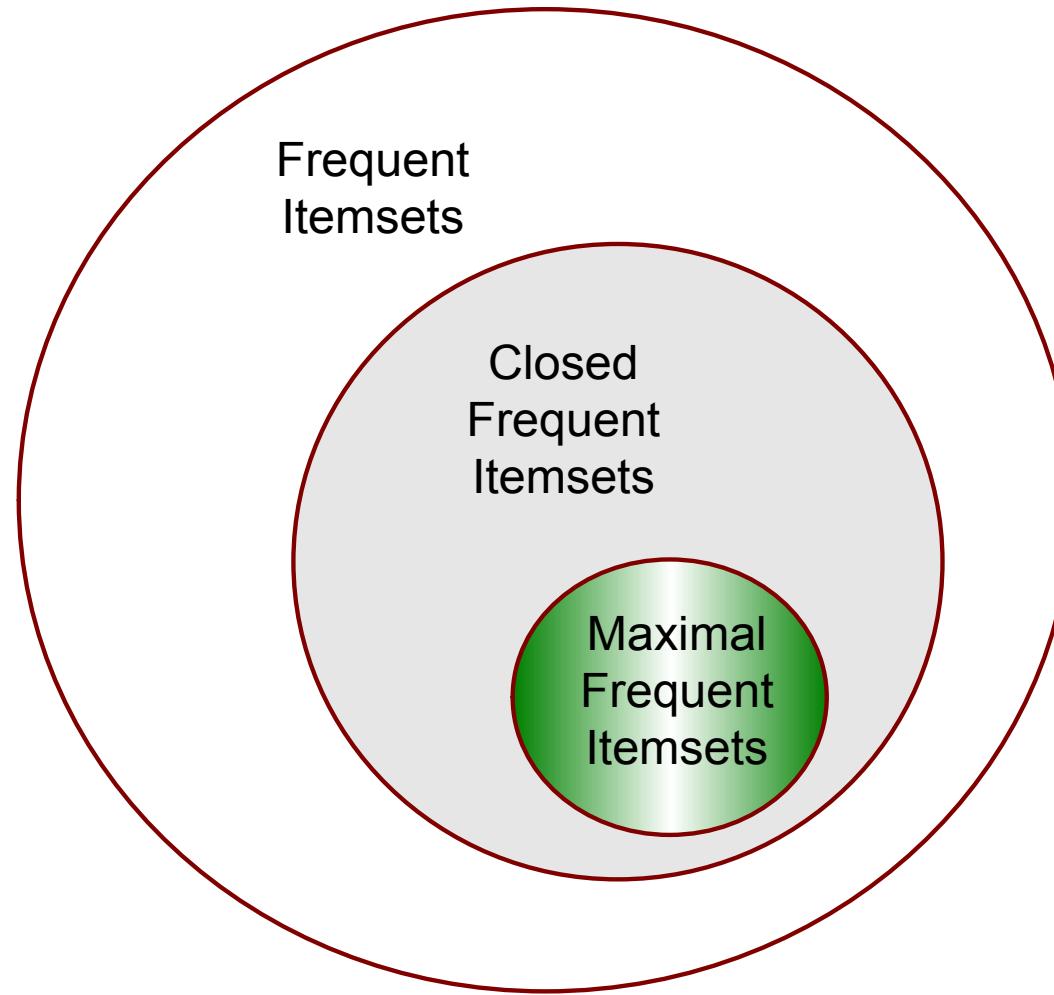
86

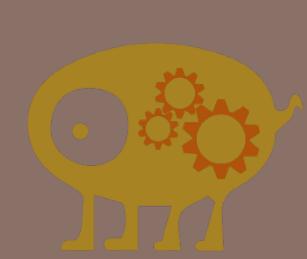
Minimum support = 2



# Maximal vs Closed Itemsets

87





# QUANTITATIVE ASSOCIATION RULES

# Multidimensional Association Rules

89

- Single dimensional association rule
  - E.g.:  $\text{buys}(\text{bread}) \wedge \text{buys}(\text{milk}) \Rightarrow \text{buys}(\text{butter})$
- Multidimensional association rule
  - E.g.:  $\text{age}(34\text{-}35) \wedge \text{income}(30\text{K}\text{-}50\text{K}) \Rightarrow \text{buys}(\text{HDTV})$
- Attributes types
  - Categorical
    - finite number of possible values, no ordering among values
  - Numerical
    - numeric, implicit ordering among values



# Example of Quantitative Association Rules

90

Record ID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

Rule	Support	Confidence
<Age:30..39>and<Married:Yes>=><NumCars:2>	40%	100%
<Age:20..29>=><NumCars:0..1>	60%	<b>100%</b>



# Mapping to Boolean Association Rules Problem

91

TID	Age:20-29 <b>(A)</b>	Age:30-40 <b>(B)</b>	Married: Yes <b>(C)</b>	Married: No <b>(D)</b>	NumCars :0 <b>(E)</b>	Numcars :1 <b>(F)</b>	NumCars :2 <b>(G)</b>
100	1	0	0	1	0	1	0
200	1	0	1	0	0	1	0
300	1	0	0	1	1	0	0
400	0	1	1	0	0	0	1
500	0	1	1	0	0	0	1

TID	Items
100	A,D,F
200	A,C,F
300	A,D,E
400	B,C,G
500	B,C,G



# Dilemma of Discretization

92

- Min\_Support
  - Increasing the number of intervals results in lower support for any single interval.
- Min\_Conf
  - Some rules may have minimum confidence only when an itemset in the antecedent consists of a small interval.

Rule A=>C

confidence =  $\text{support}(\{A\} \cup \{C\}) / \text{support}(\{A\})$



# Mining Quantitative Association Rules

93

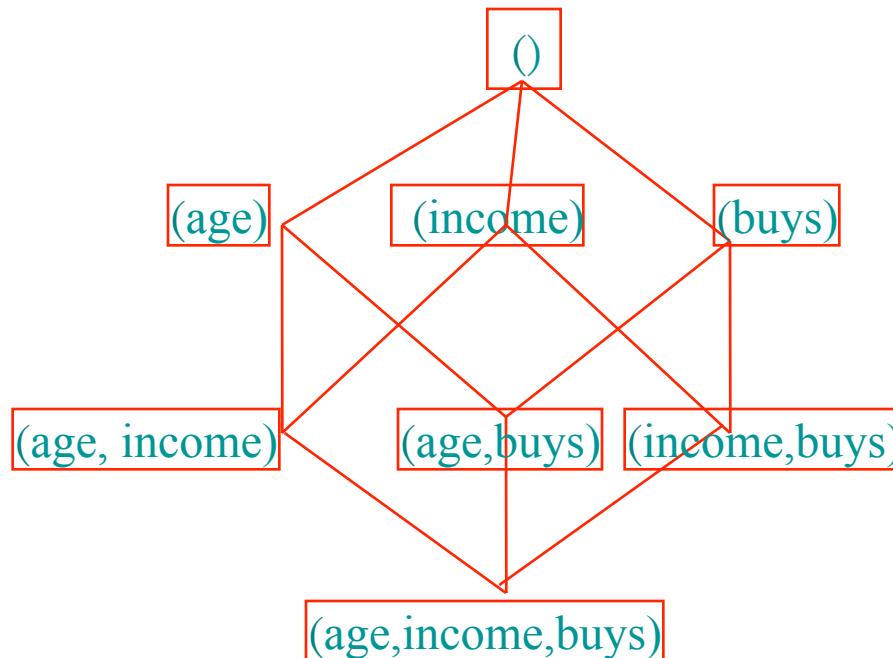
- Approaches
  - Static discretization of quantitative attributes
  - Quantitative association rule (discretized based on distribution of data)
  - Distance-based association rule (discretized based on semantic meaning of interval)



# Static Discretization of Quantitative Attributes

94

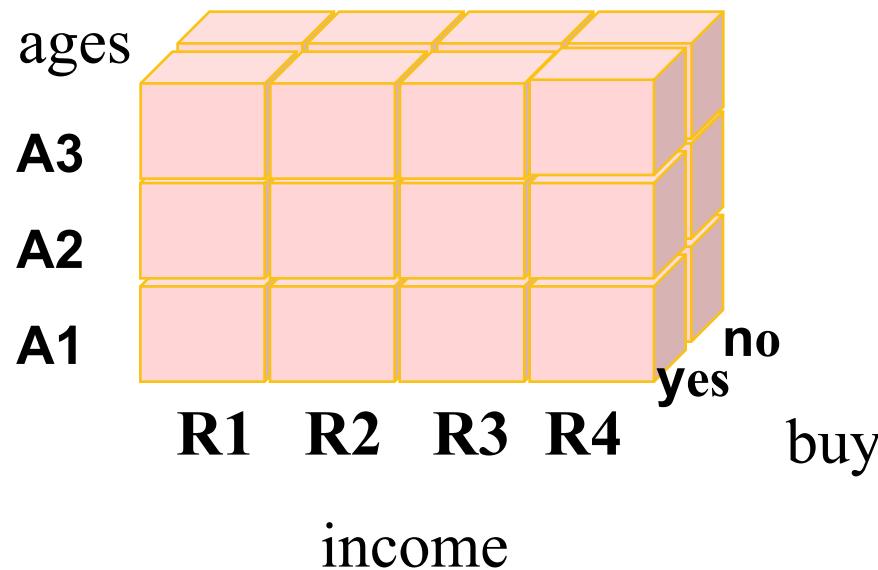
- Discretized prior to mining using concept hierarchy.
- Numeric values are replaced by ranges.



# Static Discretization of Quantitative Attributes (cont'd)

95

- Data cube is well suited for mining.
- The cells of an n-dimensional cuboid correspond to the n-predicate sets.



# Quantitative Association Rules

96

Numeric attributes are *dynamically* discretized

Such that the confidence or compactness of the rules mined is maximized.

2-D quantitative association rules:  $A_{\text{quan1}} \wedge A_{\text{quan2}} \Rightarrow A_{\text{cat}}$

Cluster “adjacent”

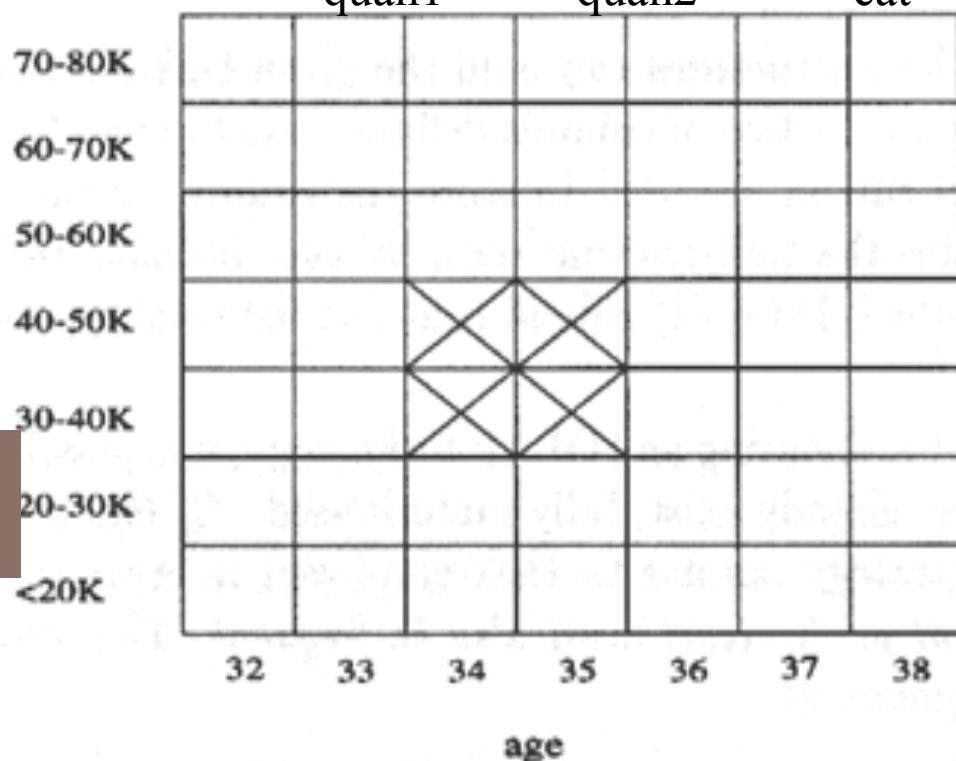
association rules

to form general

rules using a 2-D

grid.

age(34-35)  $\wedge$  income(30K-50K)  
 $\Rightarrow$  buys(high resolution TV)



# Example

**Min-Support=40% = 2 records**  
**Min-confidence=50%**

97

**People**

Record ID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

**Binning**

Age	Married
20..24:1	
25..29:2	Yes:1
30..34:3	No:2
35..39:4	

**After  
Mapping  
attributes**

Record ID	Age	Married	NumCars	Frequent Itemset (Sample)	Support
100	1	2	1	{Age:25..29}	2
200	2	1	1	{Age:30..39}	2
300	2	2	0	{Married:Yes}	3
400	3	1	2	{Married>No}	2
500	4	1	2	{NumCars:1}	2
				{NumCars:2}	2
				{<Age: 30..39>,<Married:Yes>}	2

**Rules:  
Sample**

Rule	Support	Confidence
<Age:30..39>and<Married:Yes>=><NumCars:2>	40%	100%
<Age:20..29>=><NumCars:0..1>	60%	100%



# Mining Distance-based Association Rules

98

## □ Different binning methods

Price(\$)	Equi-width (width \$10)	Equi-depth (depth 2)	Distance-based
7	[0,10]	[7,20]	[7,7]
20	[11,20]	[22,50]	[20,22]
22	[21,30]	[51,53]	[50,53]
50	[31,40]		
51	[41,50]		
53	[51,60]		

- Distance-based partitioning, more meaningful discretization considering:
  - density/number of points in an interval
  - “closeness” of points in an interval

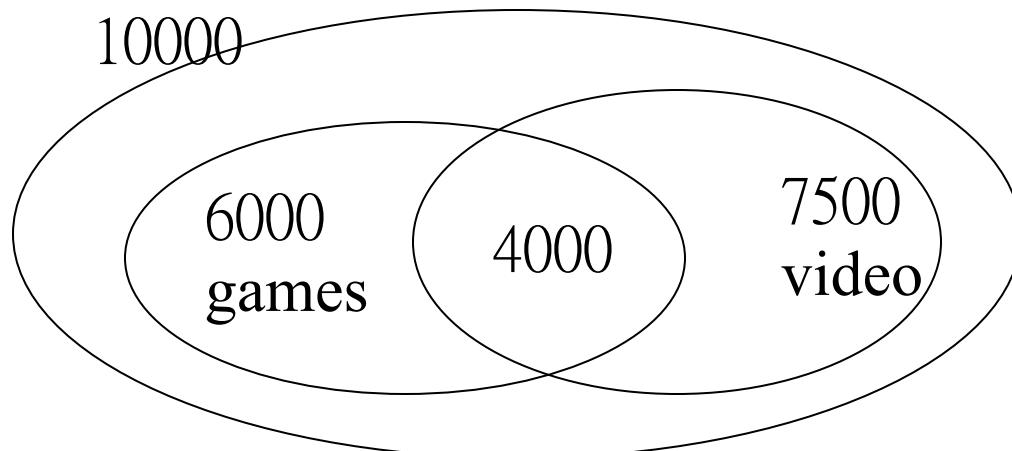




# FROM ASSOCIATION MINING TO CORRELATION ANALYSIS

# Strong Rules & Interesting

100

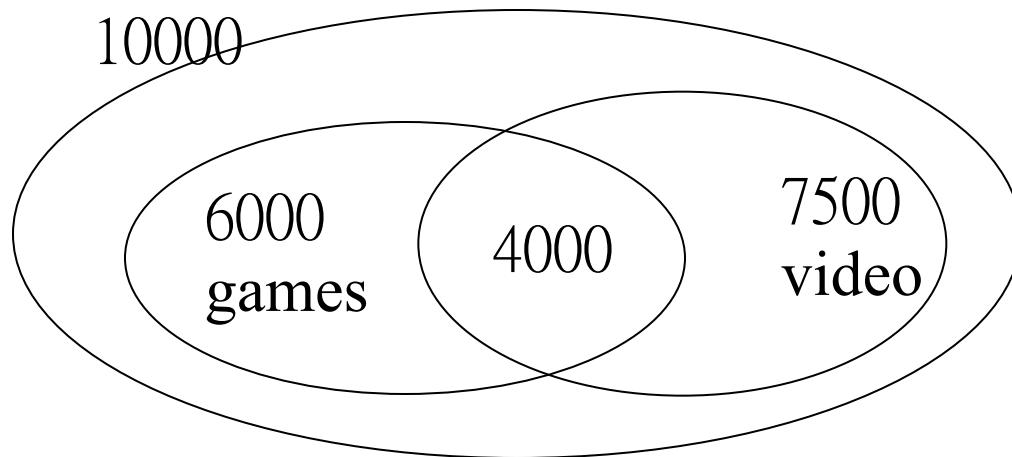


- Games -> Videos,  
 $\text{support} = 4000/10000 = 40\%$ ,  $\text{confidence} = 4000/6000 = 66\%$
- $\text{Prob(videos)} = 7500/10000 = 75\%$
- In fact, games & videos are negatively associated
- Purchase of one actually decrease the likelihood of purchasing the other



# Correlation Analysis

101



- $\text{Corr}(A,B) = P(A \cup B) / (P(A)P(B))$ 
  - E.g.  $\text{Corr}(\text{games}, \text{videos}) = 0.4 / (0.6 * 0.75) = 0.89$
  - $\text{Corr}(A, B) = 1$ , A & B are independent
  - $\text{Corr}(A, B) < 1$ , occurrence of A is negatively correlated with B
  - $\text{Corr}(A, B) > 1$ , occurrence of A is positively correlated with B

How about  $\text{jaccard}(A,B) = |A \cap B| / |A \cup B|$ ? and Dice coefficient ?  $2|A \cap B| / |A| + |B|$ ?

# Mining Association Rules with Weighted Items

102

- Weighted items
- Weighted support
- Association rule with minimum weighted support
- Given minimum weighted support 0.4  
=> {B,E}  $((0.3+0.9)*5/7=0.86)$

code	Item	Profit	Weight
A	Apple	100	0.1
B	Orange	300	0.3
C	Banana	400	0.4
D	Milk	800	0.8
E	Coca	900	0.9

TID	Items
100	A, B, D, E
200	A, D, E
300	B, D, E
400	A, B, D, E
500	A, C, E
600	B, D, E
700	B, C, D, E



# Mining Inter-Transaction Association Rules

103

- **Intra-transaction association rules:**

e.g. When the prices of IBM and SUN go up, at 80% of probability the price of Microsoft goes up **on the same day**

- **Inter-transaction association rules:**

e.g. If the price of IBM and SUN go up, Microsoft's will most likely (80% probability) go up **the next day and then drop four days later**

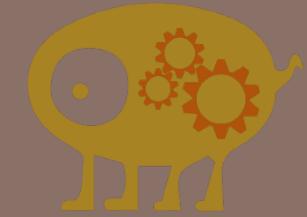


# Summary

104

- Frequent pattern mining: mine regularities in databases
- Basic methods
  - Apriori: candidate generation and test
  - Pattern growth: without candidate generation
- Extensions
  - Various patterns: max-patterns, closed patterns
  - ML/MD patterns, quantitative patterns





# ASSOCIATION RULES WITH CONSTRAINTS

# Constrained Association Rules

106

- Shortcomings of traditional association rules
  - lack of user exploration
  - lack of focus: find associations between itemsets
    - whose types do not overlap
    - total price under \$100 to itemsets whose average price is at least \$1000

# Constraint-Based Mining

107

- Interactive, exploratory mining giga-bytes of data?
  - Could it be real? --- Making good use of constraints!
- Kinds of constraints used in mining
  - knowledge constraint: classification, association, etc.
  - data constraint: SQL-like queries
    - Find product pairs sold together in Vancouver in Dec.'98.
  - dimension/level constraints:
    - in relevance to region, price, brand, customer category.
  - rule constraints:
    - small sales (price < \$10) triggers big sales (sum > \$200).
  - Interestingness constraints:
    - strong rules ( $\text{min\_support} \geq 3\%$ ,  $\text{min\_confidence} \geq 60\%$ ).



# Rule Constraints in Association Mining

108

- Two kind of rule constraints:
  - Rule form constraints
    - meta-rule guided mining.  
e.g.  $P(x, y) \wedge Q(x, w) \rightarrow \text{buys}(x, \text{"Education software"})$ .
  - Rule (content) constraint
    - constraint-based association query optimization  
e.g. **Sum{S.price < 5}**

# Constrained Association Query Optimization Problem

109

- Given a CAQ = {  $(S1, S2) \mid C$  }, the algorithm should be :
  - sound: It only finds frequent sets that satisfy the given constraints C
  - complete: All frequent sets satisfy the given constraints C are found
- A naïve solution:
  - Apply Apriori for finding all frequent sets, and then to test them for constraint satisfaction one by one.
- Proposed approach:
  - Comprehensive analysis of the properties of constraints and try to push them as deeply as possible inside the frequent set computation.



# Property of Constraints

110

- Property of constraints
  - anti-monotonicity
  - succinctness

# Anti-monotonicity

111

- anti-monotonicity (downward closed)
  - $S'$  is subset of  $S$  and  $S'$  violates  $C$ , so does  $S$ .
  - Apriori property is also anti-monotonic
  - Prune unqualified itemsets at each iteration
    - e.g.  $\text{sum}(I.\text{price}) \leq \$100$
    - e.g.  $\text{min}(J.\text{price}) \geq 500$
    - e.g.  $\text{avg}(I.\text{price}) \leq 100$  is not anti-monotonic

# Characterization of Anti-Monotonicity Constraints

112

$S \theta v, \theta \in \{=, \leq, \geq\}$	Yes
$v \in S$	no
$S \subseteq V$	no
$S \subseteq V$	yes
$S = V$	partly
$\min(S) \leq v$	no
$\min(S) \geq v$	yes
$\min(S) = v$	partly
$\max(S) \leq v$	yes
$\max(S) \geq v$	no
$\max(S) = v$	partly
$\text{count}(S) \leq v$	yes
$\text{count}(S) \geq v$	no
$\text{count}(S) = v$	partly
$\text{sum}(S) \leq v$	yes
$\text{sum}(S) \geq v$	no
$\text{sum}(S) = v$	partly
$\text{avg}(S) \theta v, \theta \in \{=, \leq, \geq\}$	no
(frequent constraint)	(yes)

# Succinctness

113

- **Succinctness:**
  - For any set  $S_1$  and  $S_2$  satisfying  $C$ ,  $S_1 \cup S_2$  satisfies  $C$
  - Given  $A_1$  is the sets of size 1 satisfying  $C$ , then any set  $S$  satisfying  $C$  are based on  $A_1$ , i.e., it contains a subset belongs to  $A_1$ ,
- **Example :**
  - $\min(S.\text{Price}) \leq v$  is succinct
  - $\sum(S.\text{Price}) \geq v$  is not succinct
    - May have other itemsets not in  $A_1$  with their sum being greater than  $v$
- **Optimization:**
  - If  $C$  is succinct, then  $C$  is pre-counting prunable. The satisfaction of the constraint alone is not affected by the iterative support counting.

# Characterization of Constraints by Succinctness

114

$S \theta v, \theta \in \{=, \leq, \geq\}$	Yes
$v \in S$	yes
$S \subseteq V$	yes
$S \subseteq V$	yes
$S = V$	yes
$\min(S) \leq v$	yes
$\min(S) \geq v$	yes
$\min(S) = v$	yes
$\max(S) \leq v$	yes
$\max(S) \geq v$	yes
$\max(S) = v$	yes
$\text{count}(S) \leq v$	weakly
$\text{count}(S) \geq v$	weakly
$\text{count}(S) = v$	weakly
$\text{sum}(S) \leq v$	no
$\text{sum}(S) \geq v$	no
$\text{sum}(S) = v$	no
$\text{avg}(S) \theta v, \theta \in \{=, \leq, \geq\}$	no
(frequent constraint)	(no)



# The Apriori Algorithm — Example

115

Database D

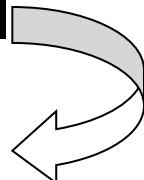
TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$C_1$   
Scan D

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

$L_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3



$L_2$

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

itemset
{2 3 5}

Scan D

$L_3$

itemset	sup
{2 3 5}	2



# Naïve Algorithm: Apriori + Constraint

116

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

Sum{S.price < 5}

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

$L_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

$C_2$

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

←

Scan D

Scan D

$L_3$

itemset	sup
{2 3 5}	2

→



# The Constrained Apriori Algorithm: Push an Anti-monotone Constraint Deep

117

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$C_1$   
Scan D

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

$L_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

$L_2$

itemset	sup.
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

itemset	sup.
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

Scan D

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

itemset
{2 3 5}

Scan D

itemset	sup.
{2 3 5}	2

Constraint:

Sum{S.price < 5}



# The Constrained Apriori Algorithm: Push a Succinct Constraint Deep

118

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$C_1$   
Scan D

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

$L_1$

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

$L_2$

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

$C_2$

Scan D

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

$C_3$

itemset
{2 3 5}

Scan D

itemset	sup
{2 3 5}	2

**Constraint:**  
 $\min\{S.\text{price} \leq 1\}$

