Association Rule Mining

Fernando Calderon
Department of Computer Science and Information Engineering
Fu Jen Catholic University
fhcalderon87@gmail.com

Association Rule Mining

- Basic concept
  - **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

$\{Diaper\} \rightarrow \{Beer\}$

$\{Milk, Bread\} \rightarrow \{Eggs, Coke\}$

$\{Beer, Bread\} \rightarrow \{Milk\}$

➢ Implication means co-occurrence, not causality!

Definition: Frequent Itemset

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

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

Market-Basket transactions

TID	Items
1	A, C, D
2	B, C, E
3	A, B, C, E
4	B, E

Minimum Support = 0.5

- Frequent Itemsets:
  - {A} (2/4), {B} (3/4), {C} (3/4), {E}(3/4)
  - {A,C}(2/4), {B,C}(2/4), {B,E}(3/4), {C,E}(2/4)
  - {B,C,E}(2/4)

Definition: Association Rule

- Association Rule
  - X $\rightarrow$ Y
    - X and Y are itemsets
  - E.g., {Milk, Diaper} $\rightarrow$ {Beer}
- Rule Evaluation Metrics
  - Support (s)
    - Fraction of transactions containing both X and Y
  - Confidence (c)
    - How often items in Y contain X

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:

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

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

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

Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *minsup* threshold
  - confidence ≥ *minconf* threshold

- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the *minsup* and *minconf* thresholds

  ⇒ Computationally prohibitive!

Example

Market-Basket transactions

TID	Items
1	A, C, D
2	B, C, E
3	A, B, C, E
4	B, E

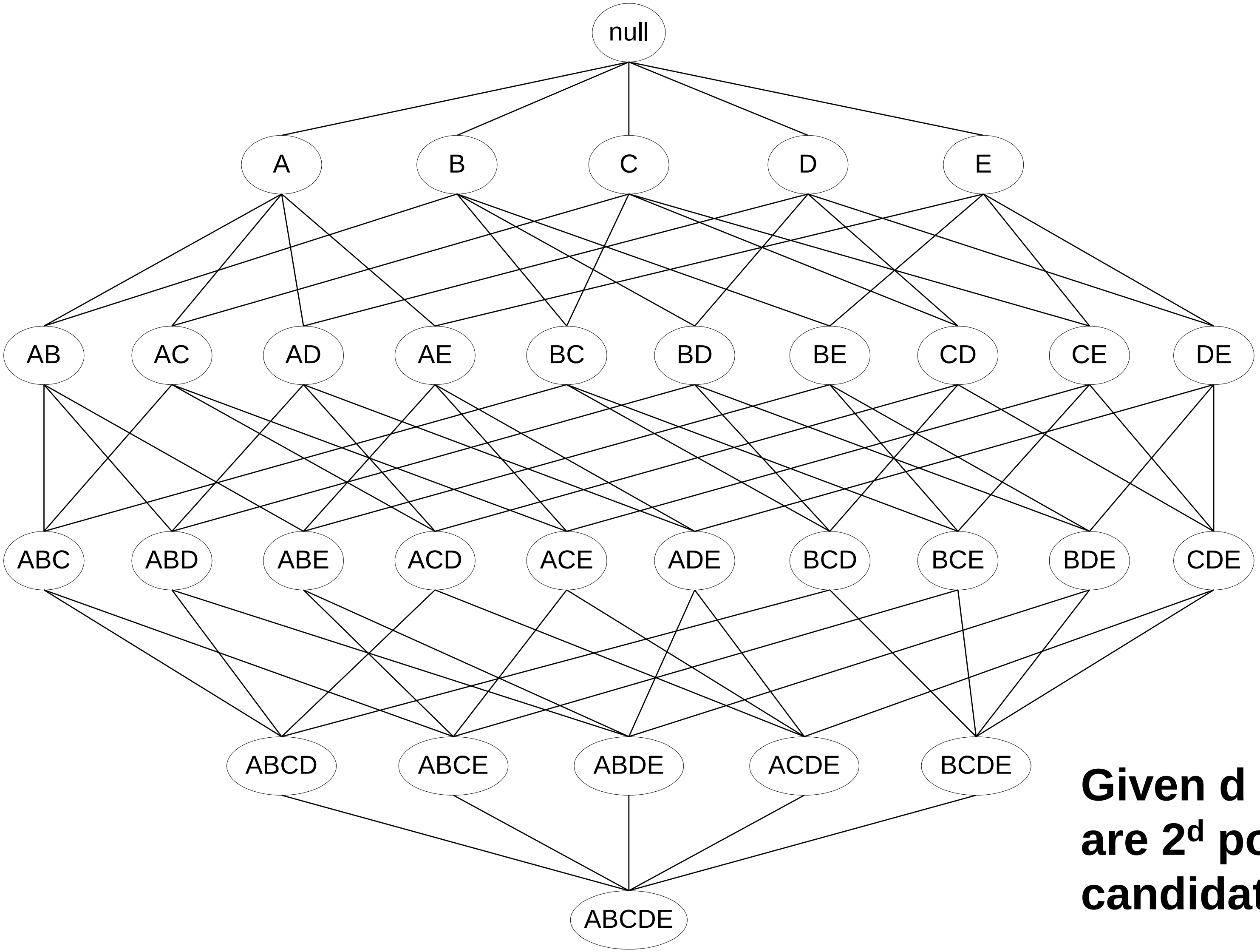
Minimum Support = 0.5
Minimum Confidence = 2/3

- Frequent Itemsets:
  - {A} (2/4), {B} (3/4), {C} (3/4), {E}(3/4), {A,C}(2/4), {B,C}(2/4), {B,E}(3/4), {C,E}(2/4), {B,C,E}(2/4)
- Rules:
  - {A} →{C} (2/2), {B} →{C} (2/3), {B} →{E} (2/3), {B} →{C,E} (2/3)
  - {C} →{E} (2/3), {E} →{C} (2/3), {E} →{B} (3/3)
  - {B,C} →{E} (2/2), {B,E} →{C} (2/3), {C,E} →{B} (2/2)

Mining Association Rules

- Observations:
  - All the above rules are binary partitions of the same itemset
  - Rules originating from the same itemset have identical support but can have different confidence
  - Thus, we may decouple the support and confidence requirements
- Two-step approach:
  - Frequent Itemset Generation
    - Generate all itemsets whose support $\geq$ minsup
  - Rule Generation
    - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset
- Frequent itemset generation is still computationally expensive

Frequent Itemset Generation



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

Frequent Itemset Generation (Contd.)

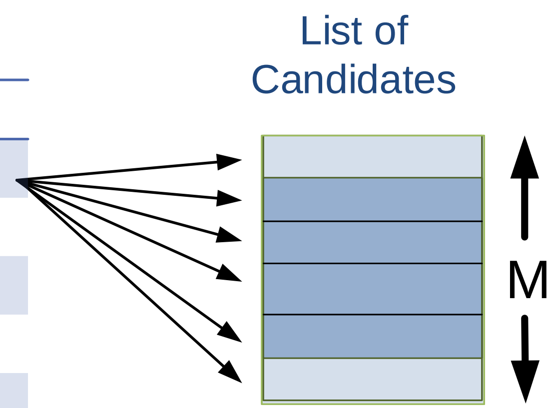
- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database

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

N

W



  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => Expensive since $M = 2^d$!!!

Apriori Algorithm

[redacted]

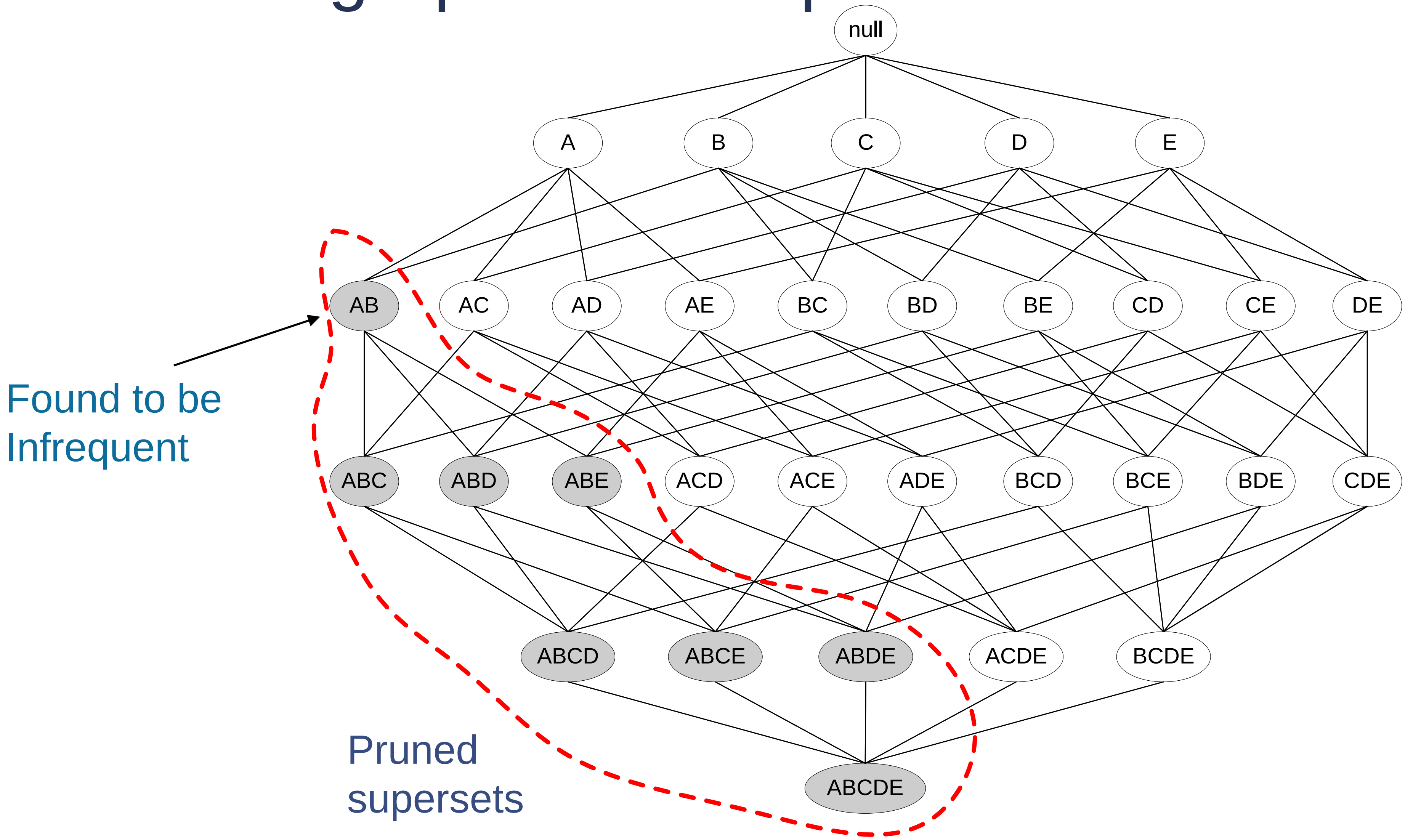
Apriori Principle

- If an itemset is frequent
  - Then all of its subsets must also be frequent
- Apriori principle holds:

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

  - 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

Items (1-itemsets)

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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

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

Minimum Support Count= 3

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3

If every subset is considered,

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

With support-based pruning,

$$C_1^6 + C_2^4 + 1 = 13$$

Apriori Algorithm

- Let k=1
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
  - Generate candidate (k+1)-itemsets from frequent k-itemsets
  - Prune candidate k-itemsets that are infrequent
  - Count the support of each candidate by scanning the DB
  - Eliminate candidates that are infrequent

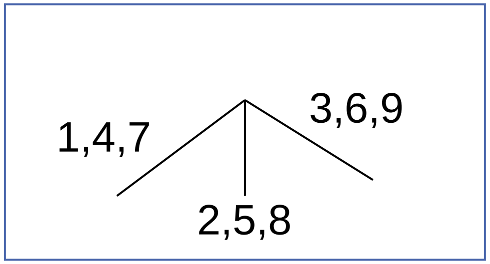
Count Supports of Candidates

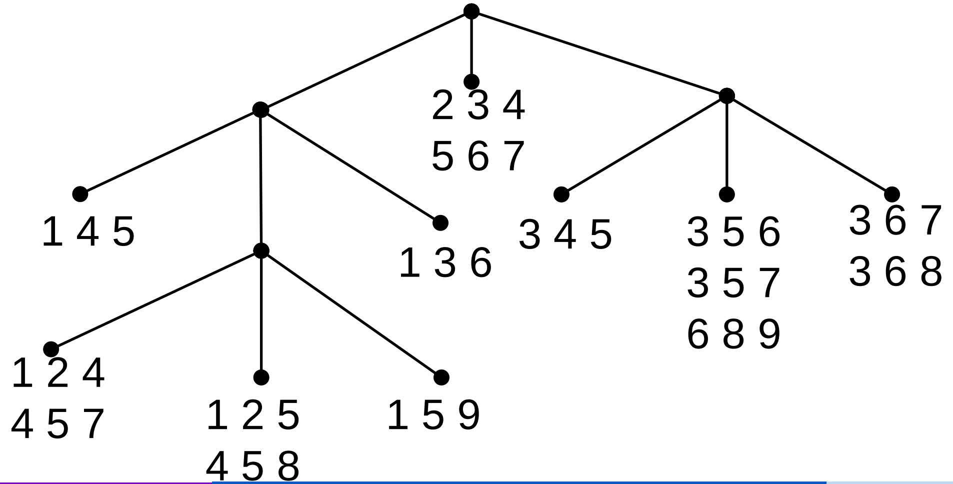
- Why counting supports of candidates a problem?
  - The total number of candidates can be very huge
  - One transaction may contain many candidates
- Possible methods:
  - Candidate itemsets are stored in a *hash-tree*
  - *Leaf* node of hash-tree contains a list of itemsets and counts
  - *Interior*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

Generate Hash Tree

- 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:
h(p)=p mod 3





Redundant Rules

- For the same support and confidence, if we have a rule {a,d}=>{c,e,f,g}, we have
  - {a,d}=>{c,e,f}
  - {a}=>{c,e,f,g}
  - {a,d,c}=>{e,f,g}
  - {a}=>{d,c,e,f,g}

Improvement of Apriori Algorithm

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

Partition: Scan Database Only Twice

- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association in large databases. In *VLDB'95*
- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
  - Scan 1: partition database and find local frequent patterns
  - Scan 2: consolidate global frequent patterns

DHP(Direct Hashing & Pruning)

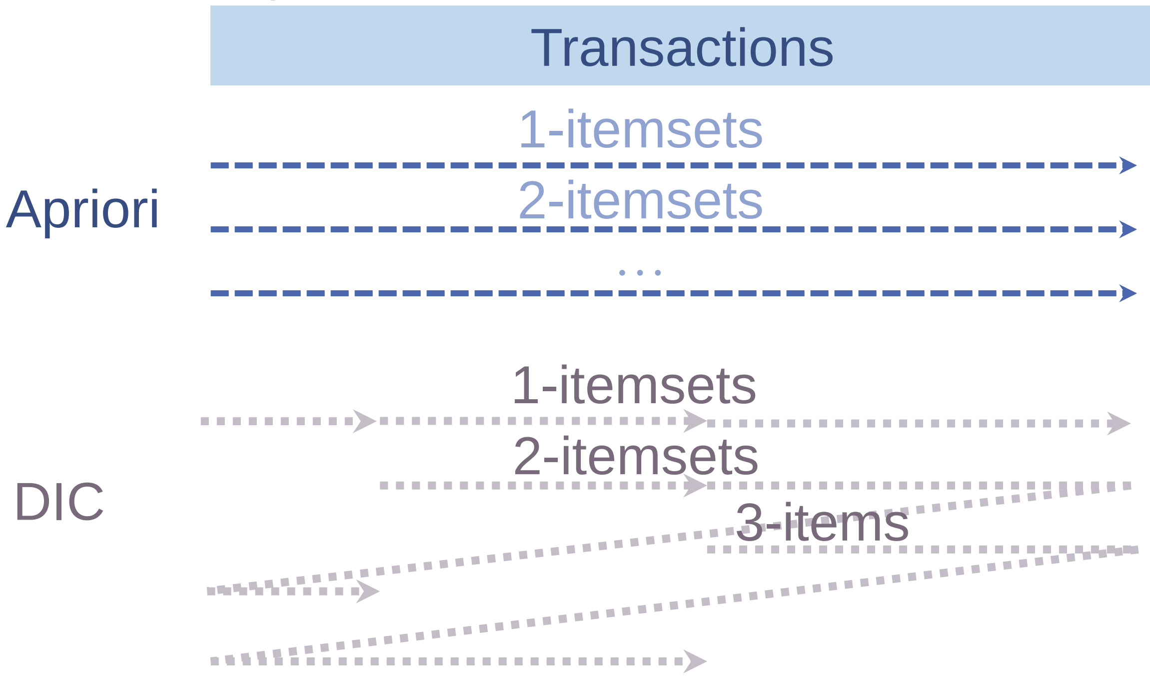
- J. Park, M. Chen, and P. Yu. An effective hash-based algorithm for mining association rules. In *SIGMOD'95*
- Reduce the Number of Candidates
- A *k*-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent
  - Candidates: a, b, c, d, e
  - Hash entries: {*ab, ad, ae*} {*bd, be, de*} …
  - Frequent 1-itemset: *a, b, d, e*
  - *ab* is not a candidate 2-itemset if the sum of count of {*ab, ad, ae*} is below support threshold

Sampling for Frequent Patterns

- H. Toivonen. Sampling large databases for association rules. In *VLDB'96*
- Select a sample of original database, mine frequent patterns within sample using Apriori
- Scan database once to verify frequent itemsets found in sample, only *borders* of closure of frequent patterns are checked
  - Example: check *abcd* instead of *ab, ac, …, etc.*
- Scan database again to find missed frequent patterns

Dynamic Itemset Counting

- Reduce Number of Scans
- S. Brin R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In SIGMOD'97
- The counting of $\{x_1,x_2,x_3,..x_k\}$ only begins, once all length-{k-1} subsets of are determined frequent



Bottleneck of Frequent-pattern Mining

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

Compact Representation of Frequent Itemsets

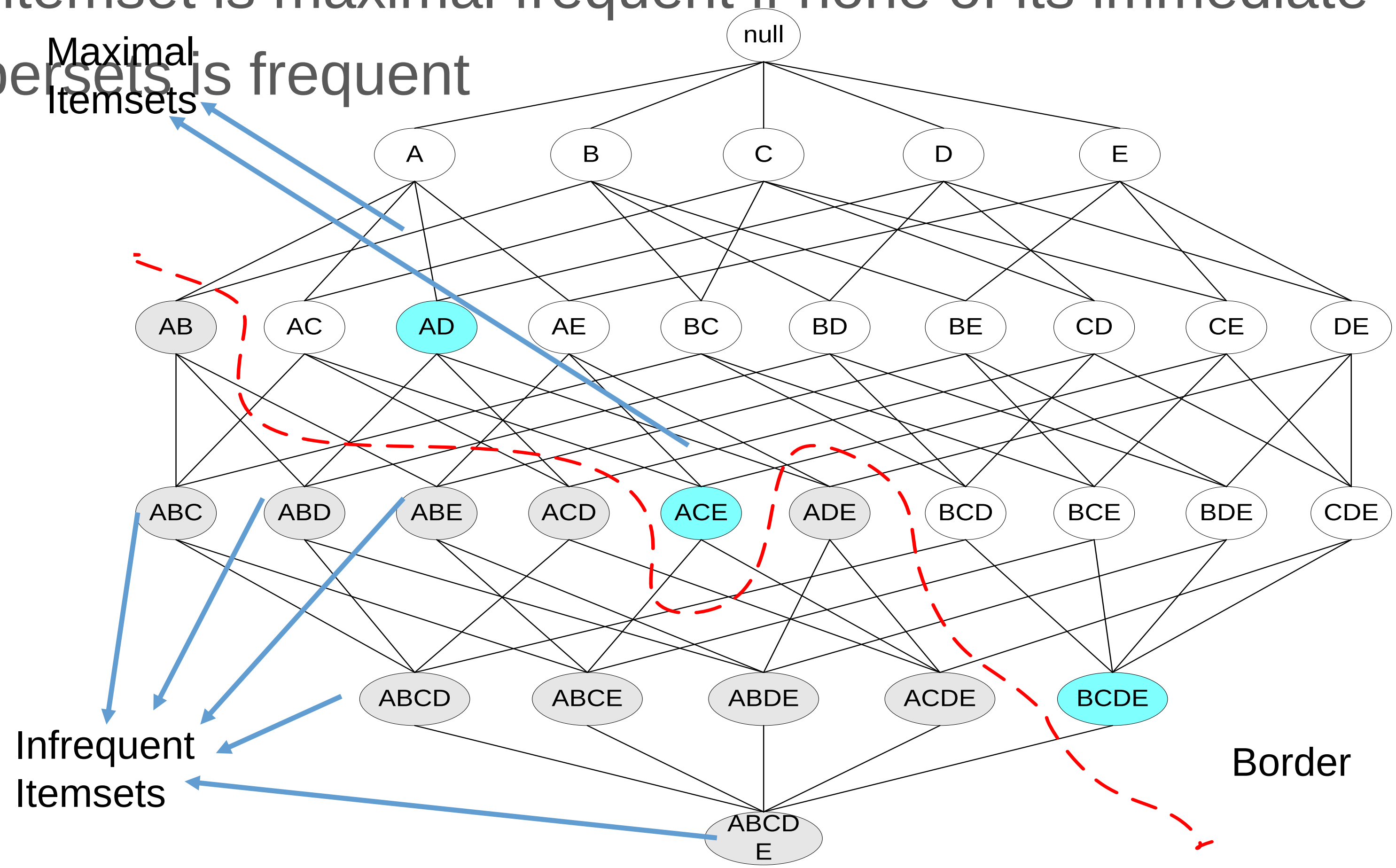
- Some itemsets are redundant because they have identical support as their supersets

TID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

- Number of frequent itemsets $= 3 \times \sum_{k=1}^{10} \binom{10}{k}$
- Need a compact representation

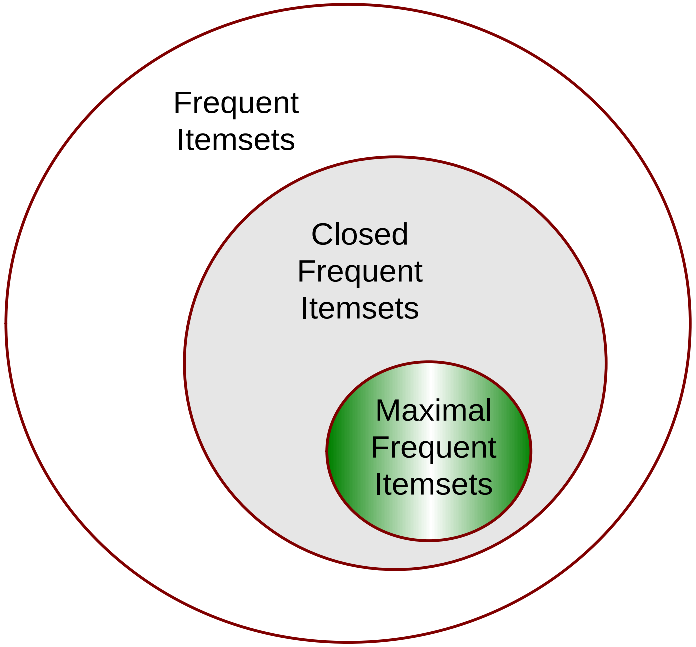
Maximal Frequent Itemset

- 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
  - It provides a minimal representation of itemsets without losing their support information



Closed Association Rules

- **Association rule on frequent closed itemsets:**
- Rule $X \Rightarrow Y$ is an association rule on frequent closed itemsets if

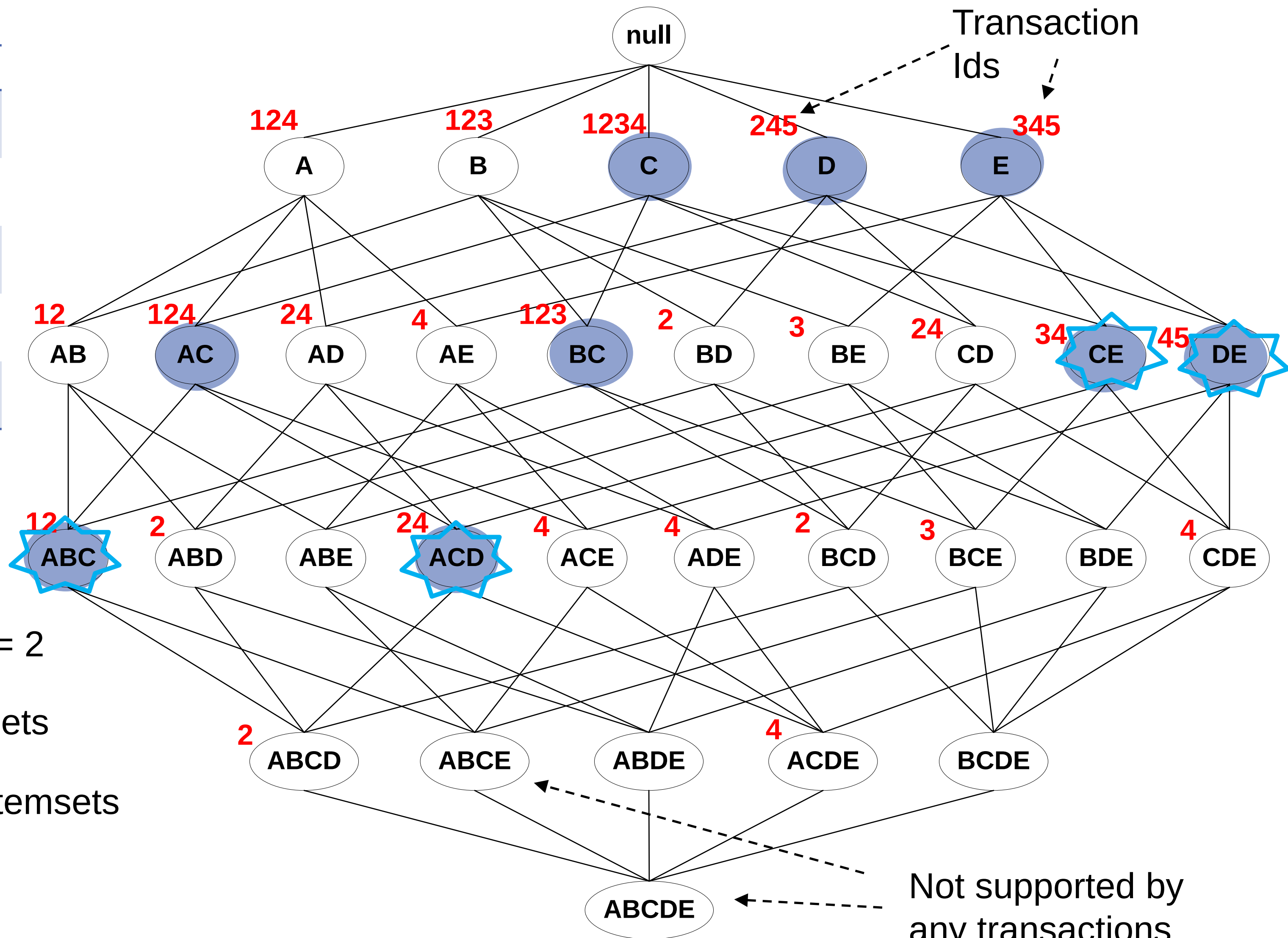
  (1) both X and $X \cup Y$ are frequent closed itemsets.

  (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

Maximal vs Closed Itemsets

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



Minimum support = 2

Closed Itemsets

Maximal Itemsets

Frequent Pattern Growth

[redacted]

FP-Growth (Frequent Pattern Growth)

- 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.

- 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

FP-Growth (Contd)

- 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

- 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

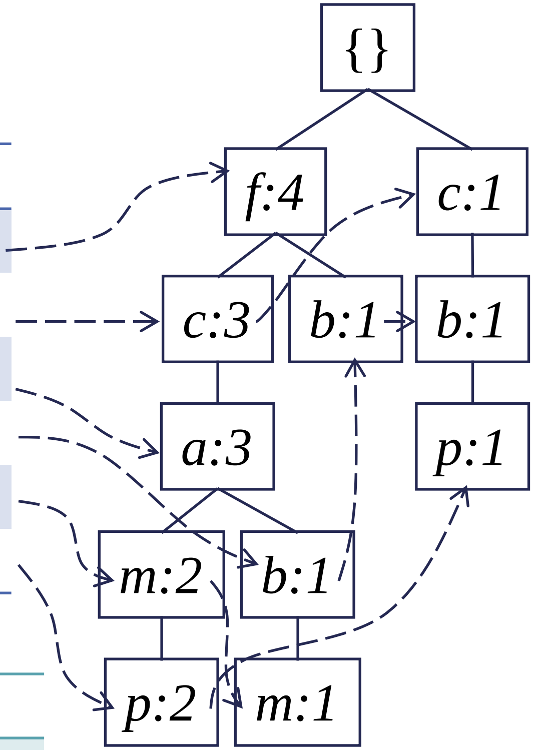
f	4
c	4
a	3
b	3
m	3
p	3

FP-Trees Construction (Contd.)

- Step 2: Scan DB and construct the FP-tree

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

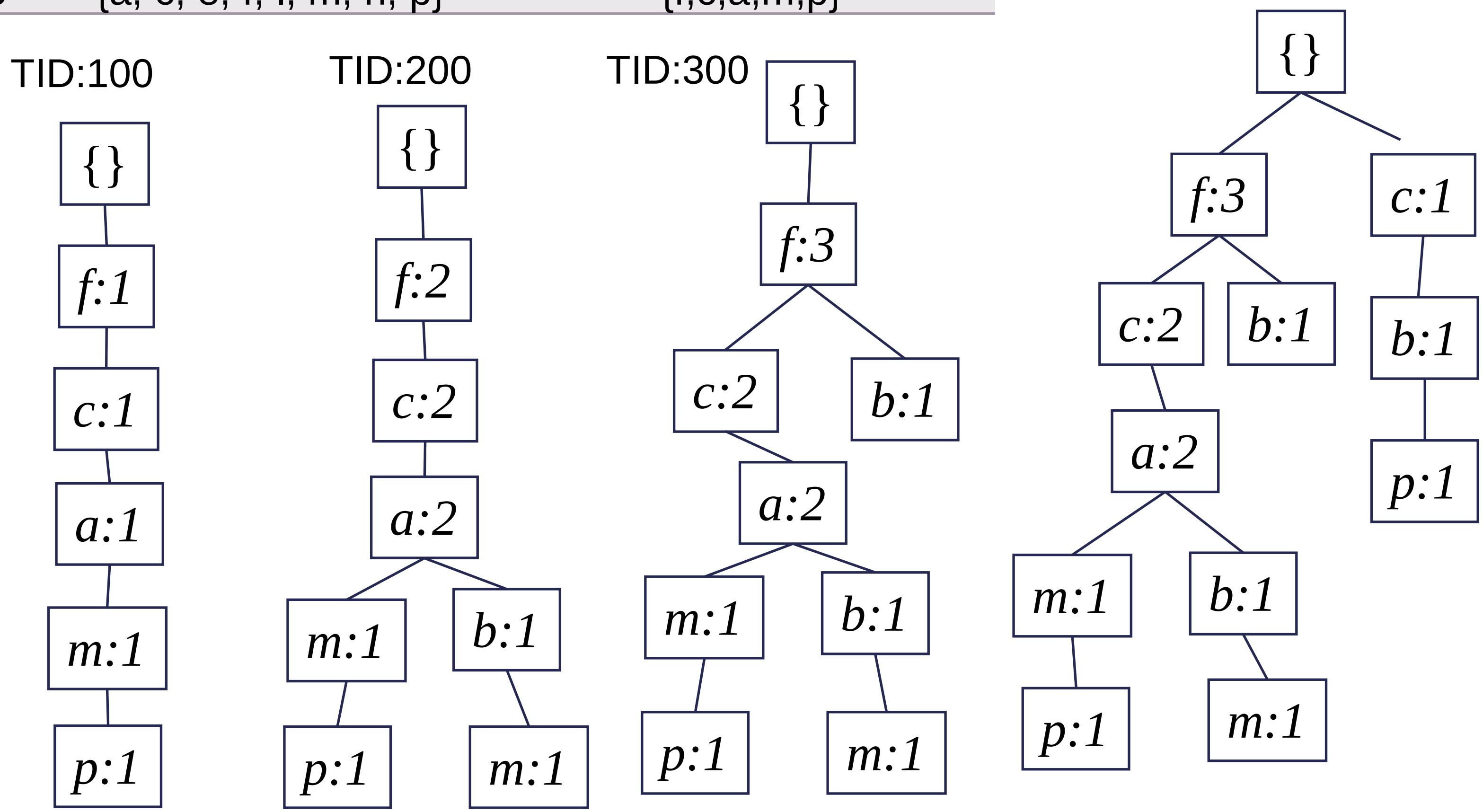
Item	Frequency	Head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



TID	Items bought	Order
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}

FP-Trees Construction (Contd.)

TID	Items bought	Order
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}



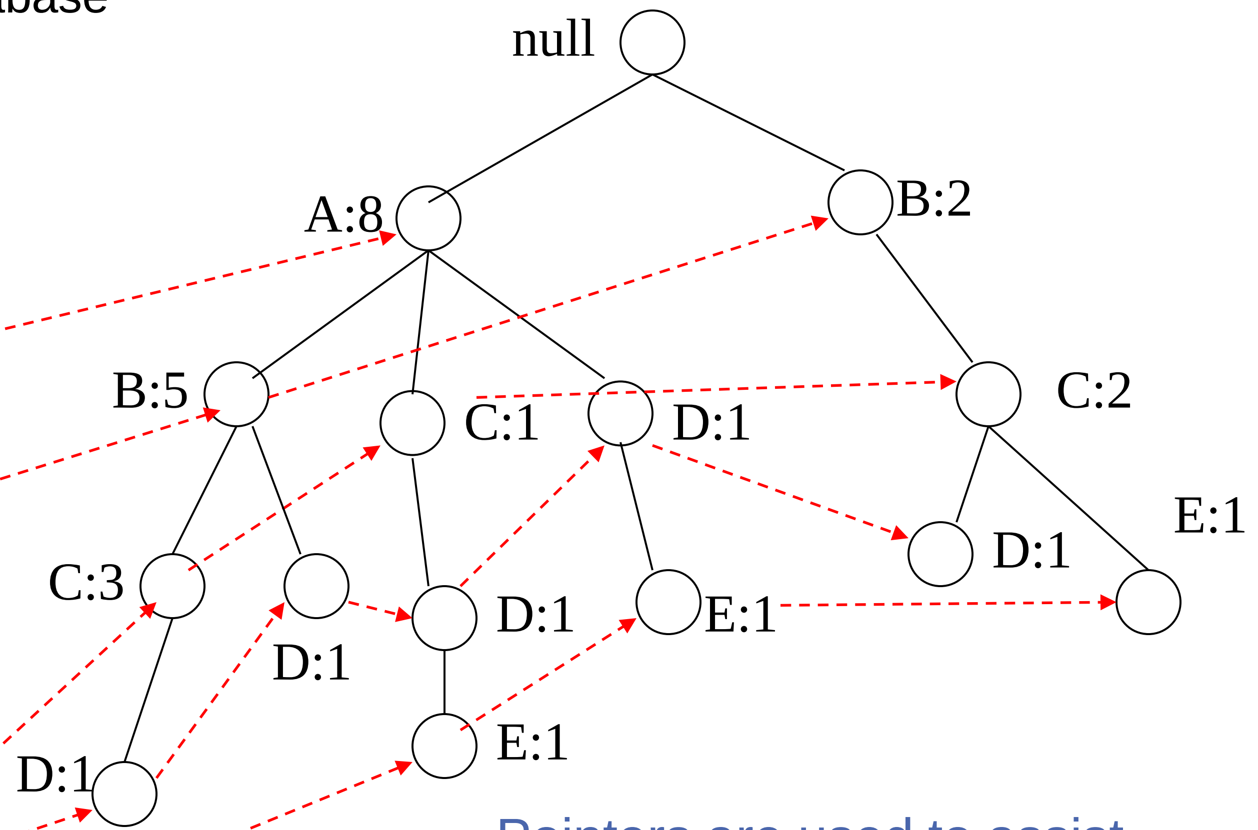
Another FP-Tree Construction Example

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{A}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Transaction Database

Header table

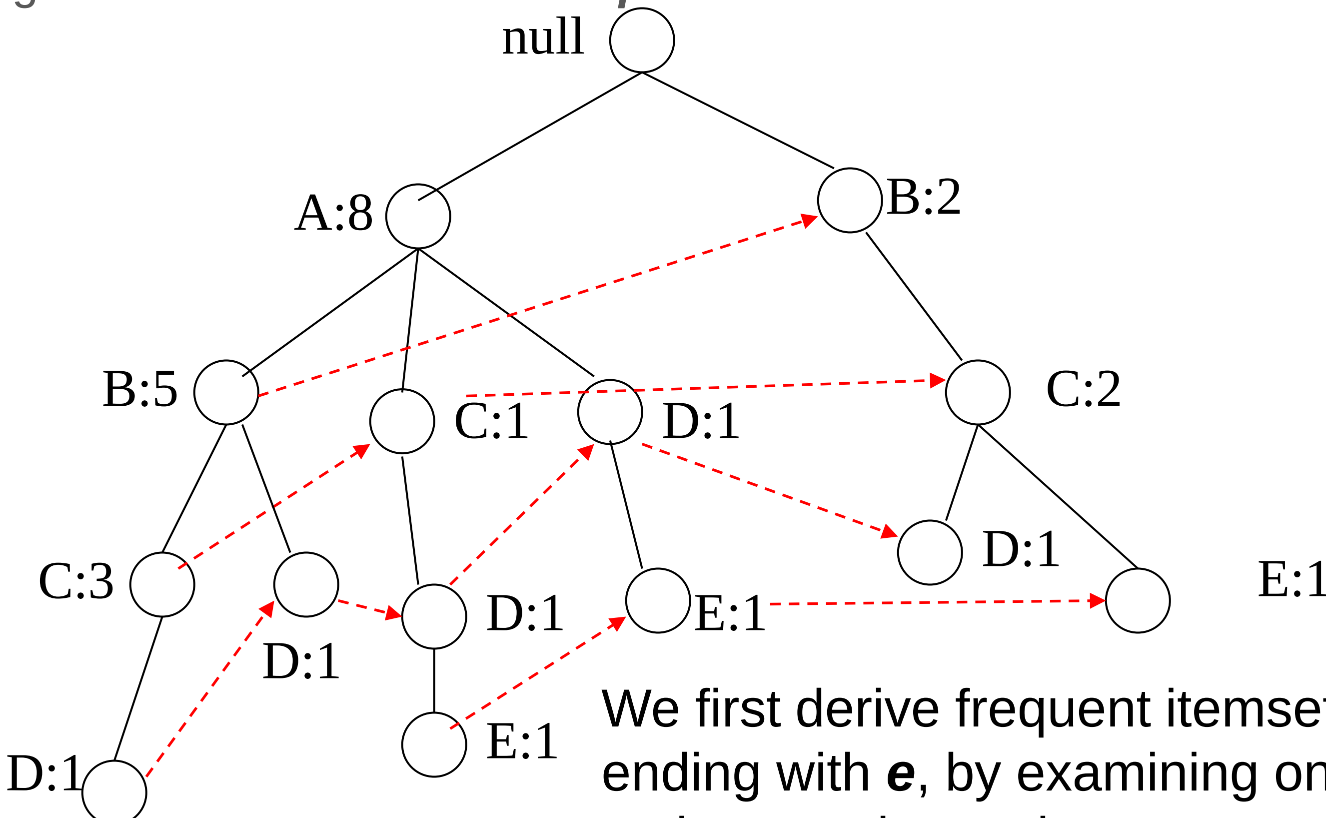
Item	Pointer
A	
B	
C	
D	
E	



Pointers are used to assist frequent itemset generation

Frequent Itemset Generation in FP-Growth Algorithm

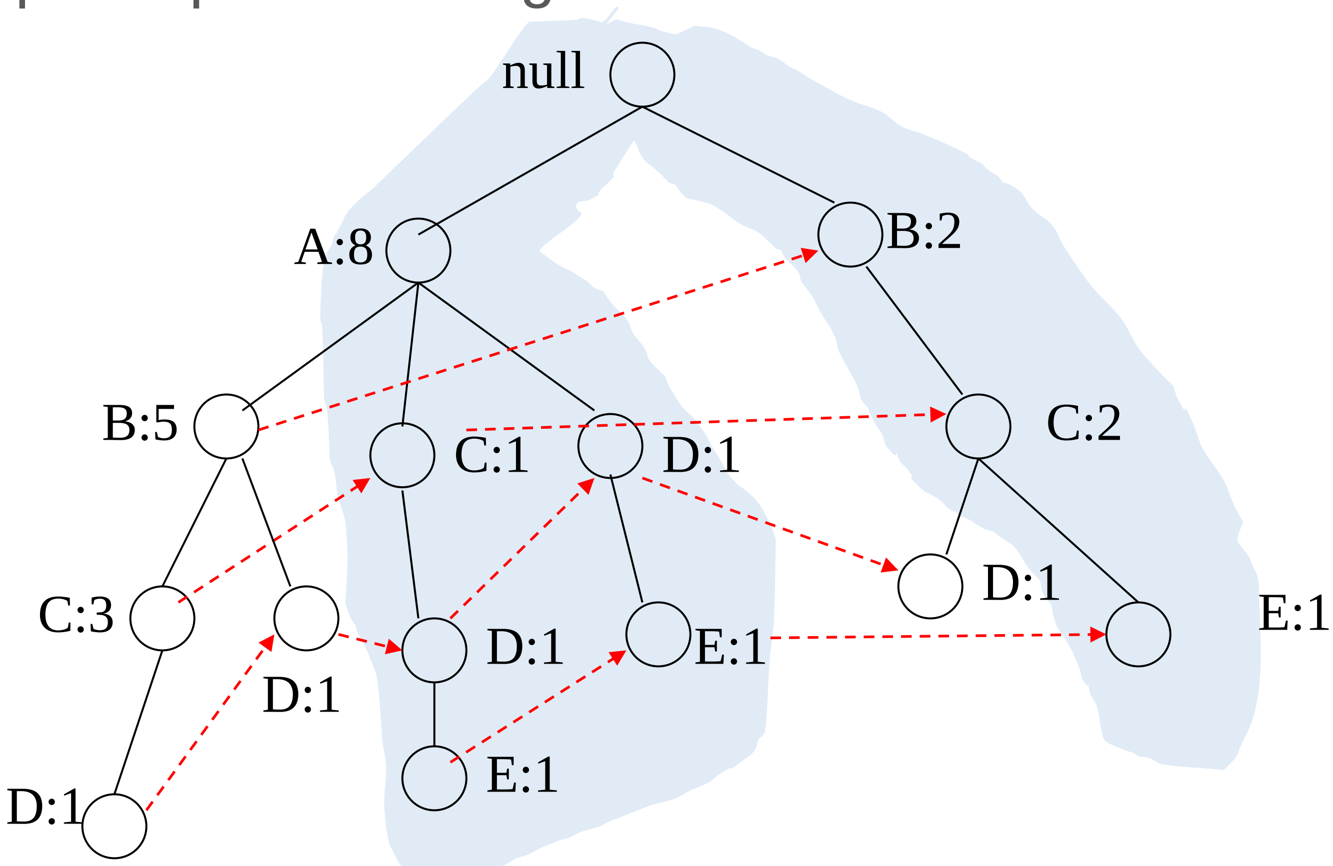
- FP-growth generates frequent itemsets by
  - Exploring the FP-Tree in a ***bottom-up*** fashion



We first derive frequent itemsets ending with ***e***, by examining only the paths contains node *e*

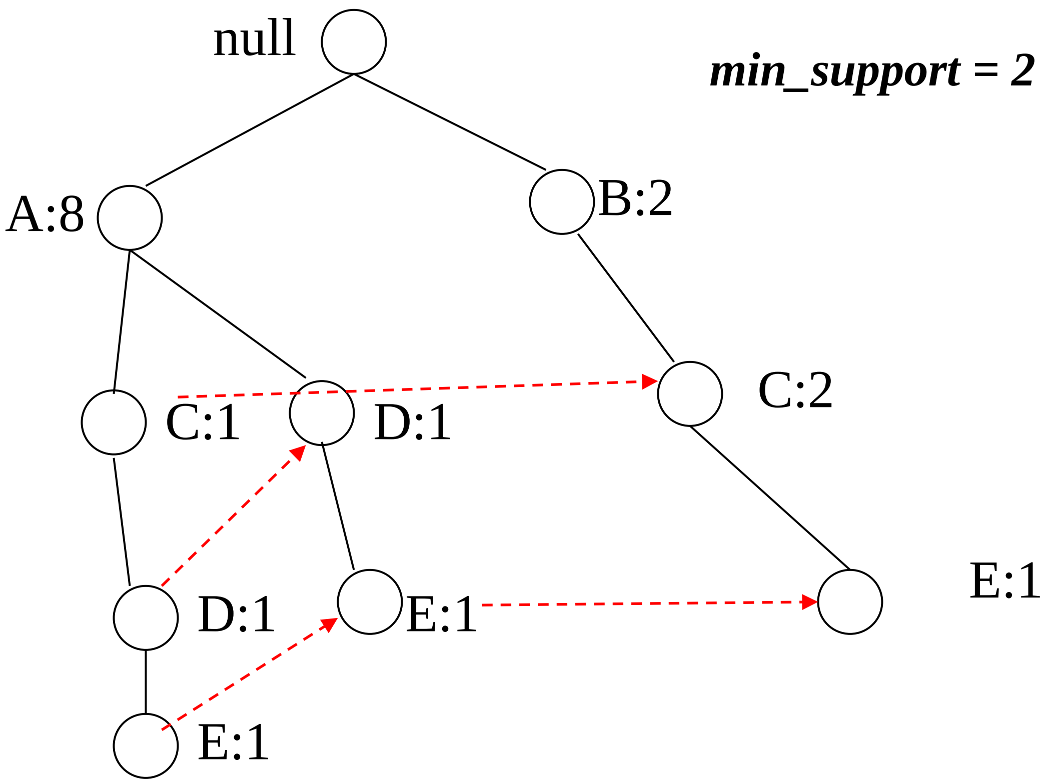
Generating Conditional FP-Tree for *e*

- Find the prefix paths ending in e first



Generating Conditional FP-Tree for ***e***

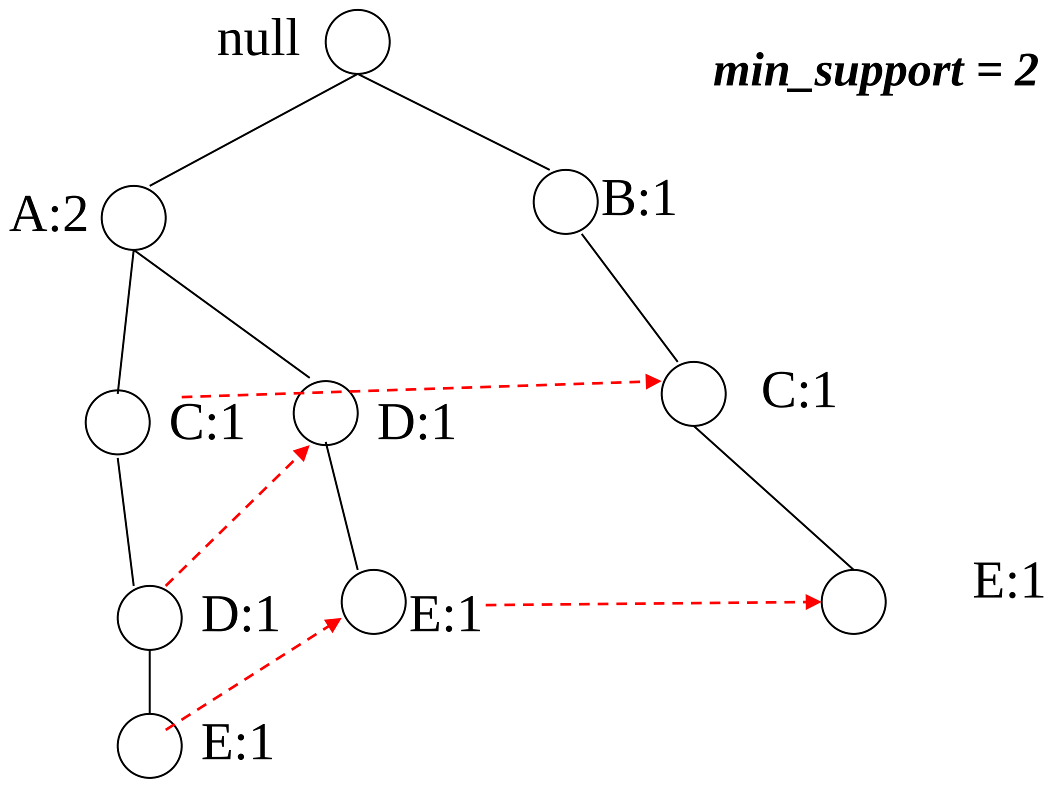
- {e} support count=3 → {e} is declared as frequent itemset



- Solve the suproblems of finding frequent itemsets ending in {de},{ce},{be}, and {ae}

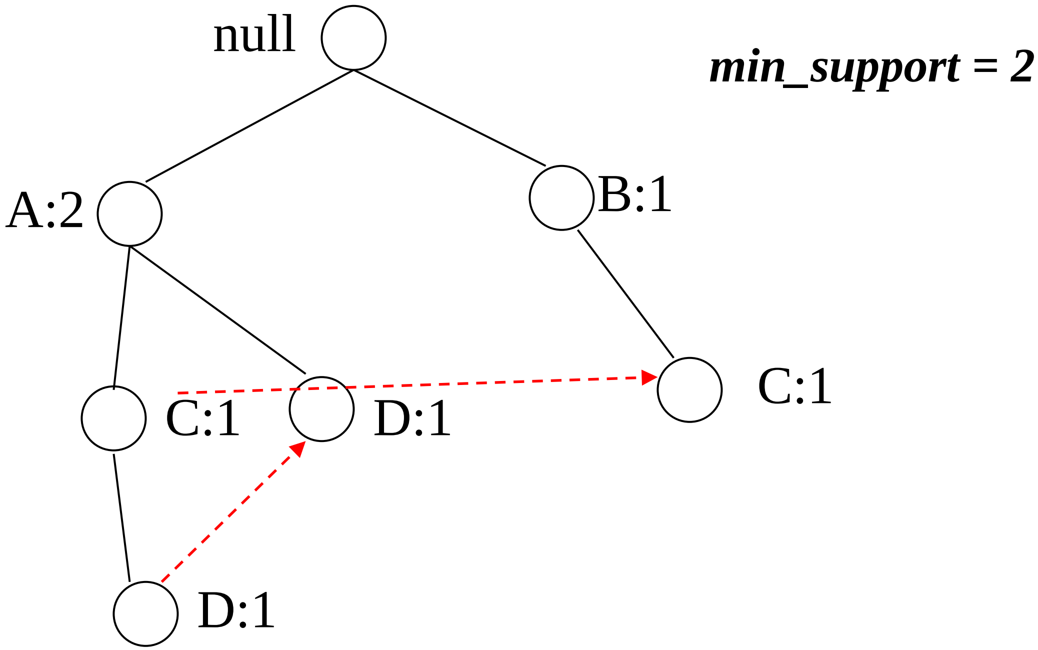
Generating Conditional FP-Tree for ***e***

- Update the support counts along the prefix path



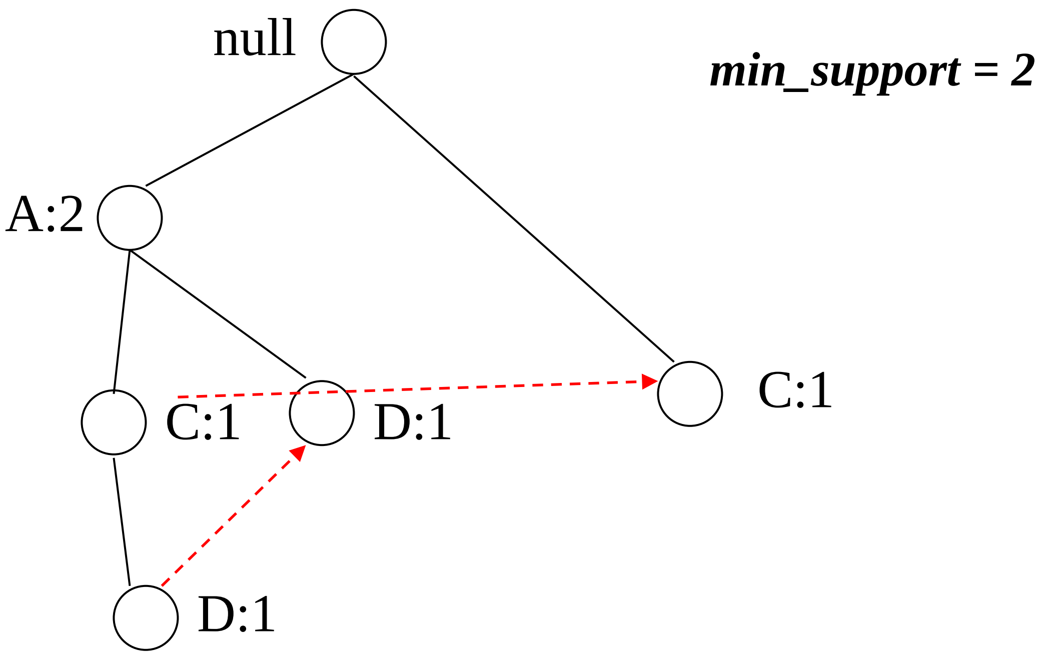
Generating Conditional FP-Tree for ***e***

- Truncate the prefix paths by removing the nodes for “e”



Generating Conditional FP-Tree for ***e***

- Safely remove the infrequent item



- Recursively using the same approach to find frequent itemsets ending in {de},{ce}, and {ae}
- Continually generating conditional FP-Trees for other item

Principles of Frequent Pattern Growth

- Pattern growth property
  - Let α be a frequent itemset in DB, B be α's conditional pattern base, and β be an itemset in B.
  - Then $\alpha \cup \beta$ is a frequent itemset in DB iff β 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 "

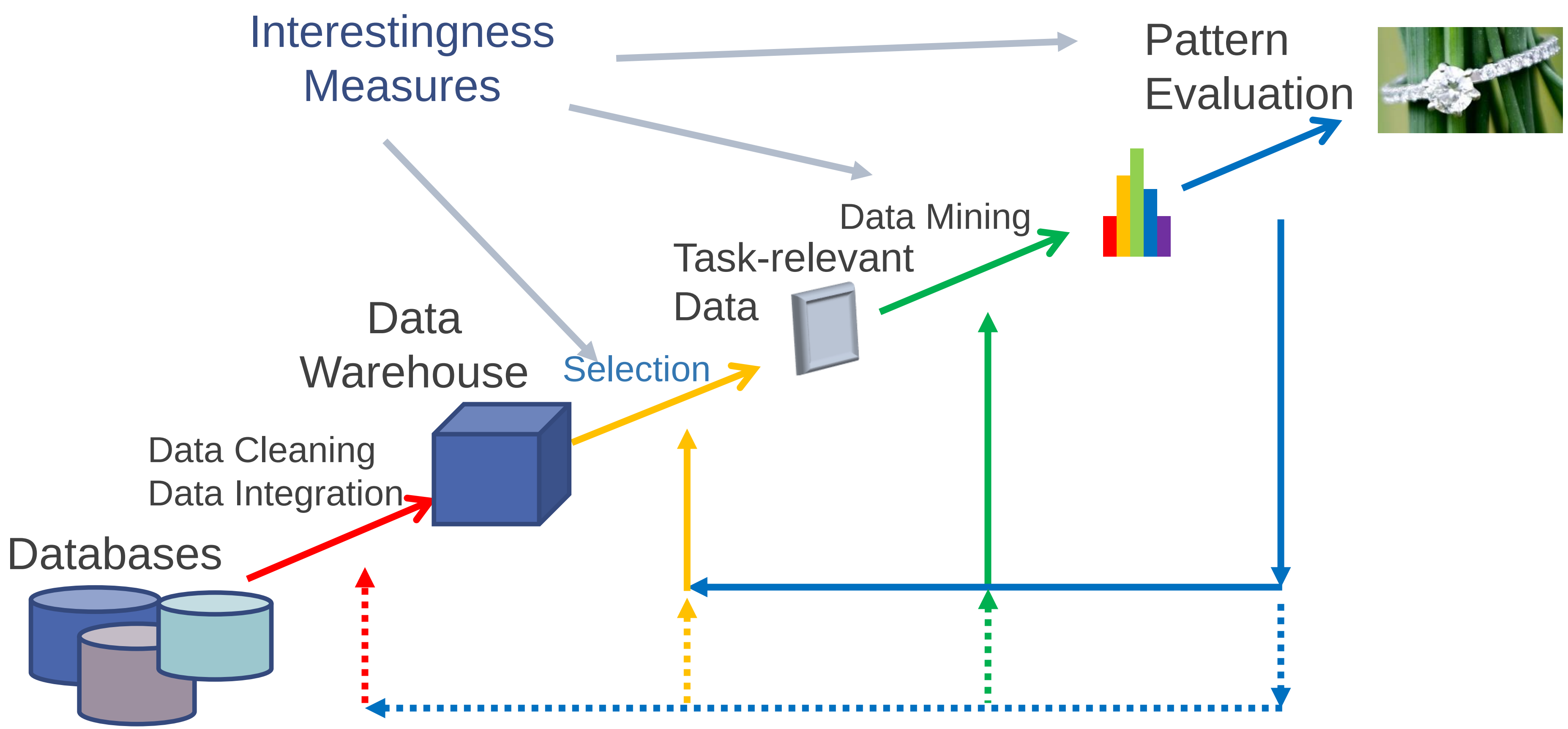
Why Is FP-Growth the Winner?

- Divide-and-conquer:
  - Decompose both the mining task and DB according to the frequent patterns obtained
  - 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

Pattern Evaluation

- Association rule algorithms tend to produce too many rules
  - Many of them are uninteresting or redundant
    - Redundant if $\{A,B,C\} \rightarrow \{D\}$ and $\{A,B\} \rightarrow \{D\}$ have same support & confidence
- In the original formulation of association rules, support & confidence are the only measures used
- Interestingness measures can be used to prune/rank the derived patterns

Application of Interestingness Measure



Computing Interestingness Measure

- Obtained from a contingency table

Contingency table for $X \rightarrow Y$

	Y	$\bar{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\bar{X}$	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	$\|T\|$

$f_{11}: support\ of\ X\ and\ Y$
$f_{01}: support\ of\ \bar{X}\ and\ Y$
$f_{10}: support\ of\ X\ and\ \bar{Y}$
$f_{00}: support\ of\ \bar{X}\ and\ \bar{Y}$

Used to define various measures

- E.g., support, confidence, lift, Gini, J-measure

Drawback of Confidence

	$Coffee$	$\overline{Coffee}$	
Tea	15	5	20
$\overline{Tea}$	75	5	80
	90	10	100

Association Rule: Tea $\rightarrow$ Coffee

Confidence: $P(Coffee|Tea) = 0.75$

but $P(Coffee) = 0.9$

$\Rightarrow$ Although confidence is high, rule is misleading

$\Rightarrow P(Coffee|\overline{Tea}) = 0.9375$

Statistical Independence

- Population of 1000 students
  - 600 students know how to swim (S)
  - 700 students know how to bike (B)
  - 420 students know how to swim and bike (S,B)

  - $P(S \wedge B) = 420/1000 = 0.42$
  - $P(S) \times P(B) = 0.6 \times 0.7 = 0.42$

  - $P(S \wedge B) = P(S) \times P(B)$ => Statistical independence
  - $P(S \wedge B) > P(S) \times P(B)$ => Positively correlated
  - $P(S \wedge B) < P(S) \times P(B)$ => Negatively correlated

Statistical-based Measures

- Measures that take into account statistical dependence

$$Lift = \frac{P(Y|X)}{P(Y)}$$

$$Interest = \frac{P(X,Y)}{P(X)P(Y)}$$

$$PS = P(X,Y) - P(X)P(Y)$$

$$\varphi - coefficient = \frac{P(X,Y) - P(X)P(Y)}{\sqrt{P(X)[1-P(X)]P(Y)[1-P(Y)]}}$$

Example: Interest Factor

	$Coffee$	$\overline{Coffee}$	
Tea	15	5	20
$\overline{Tea}$	75	5	80
	90	10	100

Association Rule: Tea $\rightarrow$ Coffee

Confidence: $P(Coffee|Tea) = 0.75$

$P(Coffee) = 0.9$, $P(Tea) = 0.2$

$\Rightarrow \text{Interest} = {}^{0.15}/_{(0.9\times0.2)} = 0.83$

< 1, therefore is negatively associated

Different Propose Measures

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i),-\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),\ P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]-P(B)^2-P(\overline{B})^2,\ P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A),P(A\|B))$
12	Laplace (L)	$\max\Big(\frac{NP(A,B)+1}{NP(A)+2},\frac{NP(A,B)+1}{NP(B)+2}\Big)$
13	Conviction (V)	$\max\Big(\frac{P(A)P(\overline{B})}{P(A\overline{B})},\frac{P(B)P(\overline{A})}{P(B\overline{A})}\Big)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\Big(\frac{P(B\|A)-P(B)}{1-P(B)},\frac{P(A\|B)-P(A)}{1-P(A)}\Big)$
18	Added Value (AV)	$\max(P(B\|A)-P(B),P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B),P(A\|B)-P(A))$

Some measures are good for certain applications, but not for others

Comparing Different Measures

10 examples of contingency tables:

Example	f_{11}	f_{10}	f_{01}	f_{00}
E1	8123	83	424	1370
E2	8330	2	622	1046
E3	9481	94	127	298
E4	3954	3080	5	2961
E5	2886	1363	1320	4431
E6	1500	2000	500	6000
E7	4000	2000	1000	3000
E8	4000	2000	2000	2000
E9	1720	7121	5	1154
E10	61	2483	4	7452

Rankings of contingency tables using various measures:

#	ϕ	λ	α	Q	Y	κ	M	J	G	s	c	L	V	I	IS	PS	F	AV	S	ζ	K
E1	1	1	3	3	3	1	2	2	1	3	5	5	4	6	2	2	4	6	1	2	5
E2	2	2	1	1	1	2	1	3	2	2	1	1	1	8	3	5	1	8	2	3	6
E3	3	3	4	4	4	3	3	8	7	1	4	4	6	10	1	8	6	10	3	1	10
E4	4	7	2	2	2	5	4	1	3	6	2	2	2	4	4	1	2	3	4	5	1
E5	5	4	8	8	8	4	7	5	4	7	9	9	9	3	6	3	9	4	5	6	3
E6	6	6	7	7	7	7	6	4	6	9	8	8	7	2	8	6	7	2	7	8	2
E7	7	5	9	9	9	6	8	6	5	4	7	7	8	5	5	4	8	5	6	4	4
E8	8	9	10	10	10	8	10	10	8	4	10	10	10	9	7	7	10	9	8	7	9
E9	9	9	5	5	5	9	9	7	9	8	3	3	3	7	9	9	3	7	9	9	8
E10	10	8	6	6	6	10	5	9	10	10	6	6	5	1	10	10	5	1	10	10	7

Properties of A Good Measure

- Piatetsky-Shapiro:

  3 properties a good measure M must satisfy:

  - M(A,B) = 0 if A and B are statistically independent
  - M(A,B) increase monotonically with P(A,B) when P(A) and P(B) remain unchanged
  - M(A,B) decreases monotonically with P(A) [or P(B)] when P(A,B) and P(B) [or P(A)] remain unchanged

Property under Variable Permutation

	B	$\bar{B}$
A	p	q
$\bar{A}$	r	s

	A	$\bar{A}$
B	p	q
$\bar{B}$	r	s

- Does M(A,B) = M(B,A)?

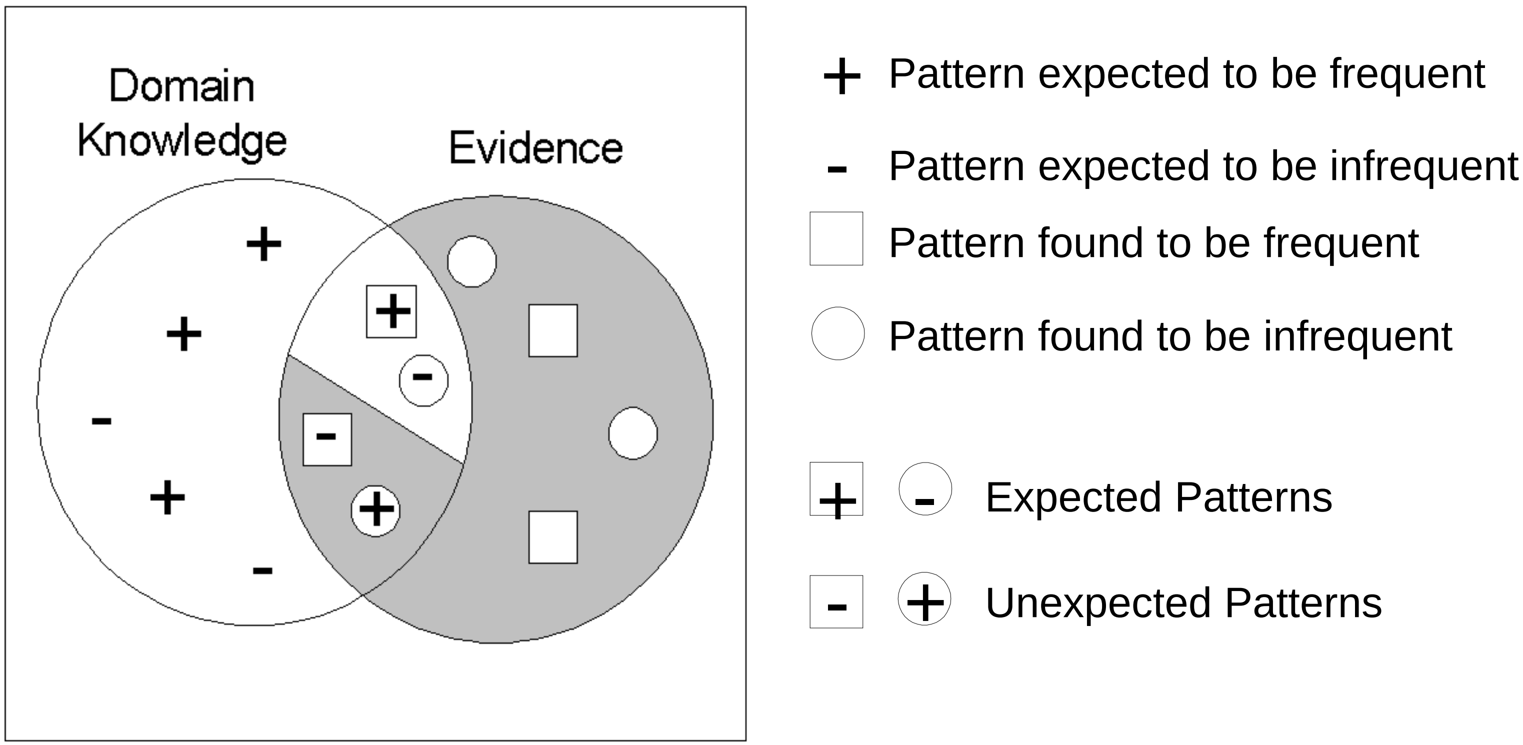
- Symmetric measures:
  - Support, lift, collective strength, cosine, Jaccard
- Asymmetric measures:
  - Confidence, conviction, Laplace, J-measure

Subjective Interestingness Measure

- Objective measure:
  - Rank patterns based on statistics computed from data
  - e.g., 21 measures of association (support, confidence, Laplace, Gini, mutual information, Jaccard, etc).
- Subjective measure:
  - Rank patterns according to user's interpretation
    - A pattern is subjectively interesting if it contradicts the expectation of a user (Silberschatz & Tuzhilin)
    - A pattern is subjectively interesting if it is actionable (Silberschatz & Tuzhilin)

Interestingness via Unexpectedness

- Need to model expectation of users (domain knowledge)



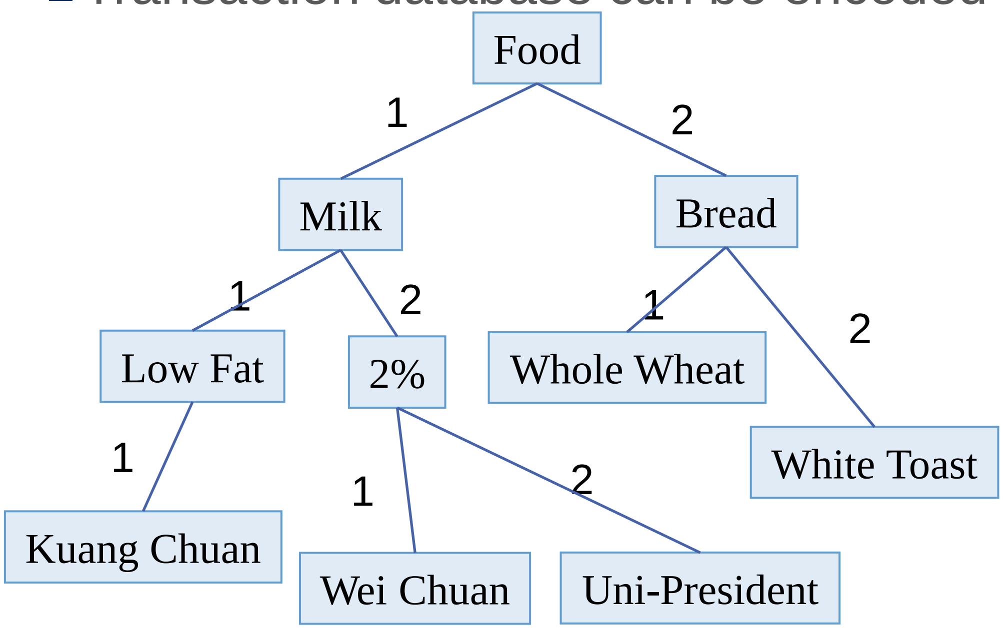
- Need to combine expectation of users with evidence from data (i.e., extracted patterns)

Multilevel Association Rules

[redacted]

Multiple-Level Association Rules

- Items often form hierarchy
- Items at the lower level are expected to have lower support
- Rules regarding itemsets at appropriate levels could be 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

- A top down, progressive deepening approach:
  - First find high-level strong rules:
    - milk $\rightarrow$ bread [20%, 60%]
  - Then find their lower-level "weaker" rules:
    - 2% milk $\rightarrow$ wheat bread [6%, 50%]

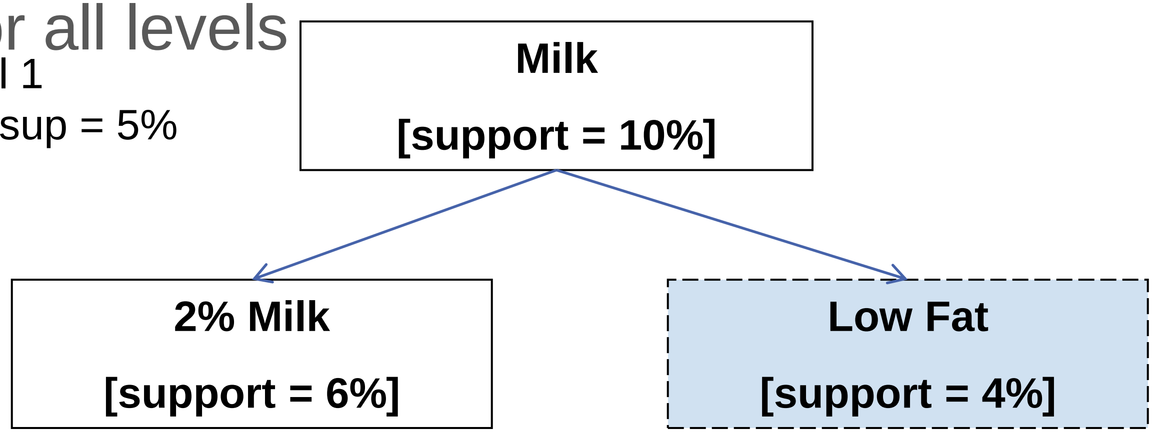
- Variations at mining multiple-level association rules
  - Level-crossed association rules:
    - *2% milk* $\rightarrow$ *wheat bread*
  - Association rules with multiple, alternative hierarchies:
    - *2% milk* $\rightarrow$ *bread*

Uniform Support

- Multi-level mining with uniform support: the same minimum support for all levels

Level 1
min_sup = 5%

Level 2
min_sup = 5%

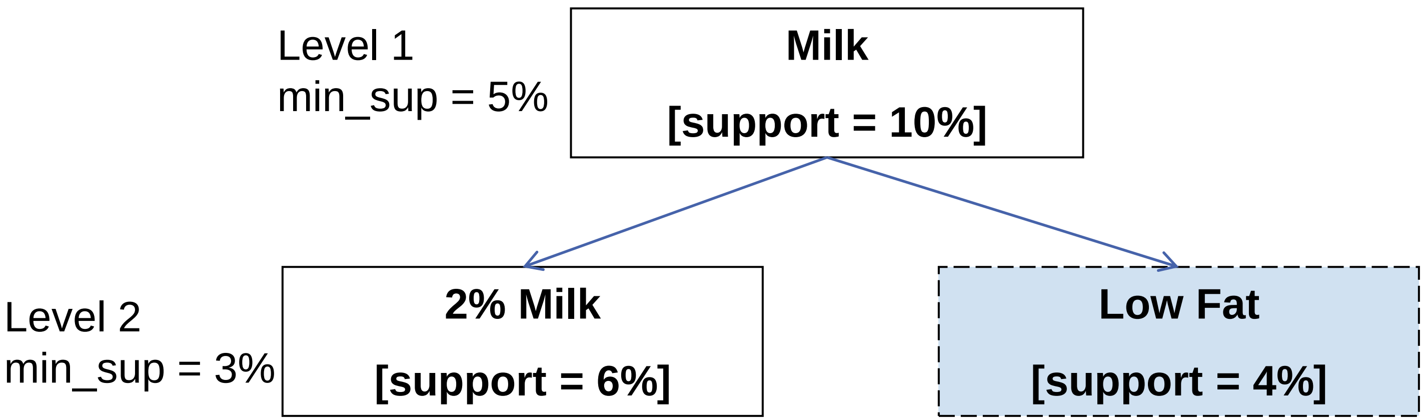


☺ No need to examine itemsets containing any item whose ancestors do not have minimum support

☹ Lower level items do not occur as frequently. If support threshold
- too high ⇒ miss low level associations
- too low ⇒ generate too many high level associations

Reduced Support

- Reduce minimum support at lower levels



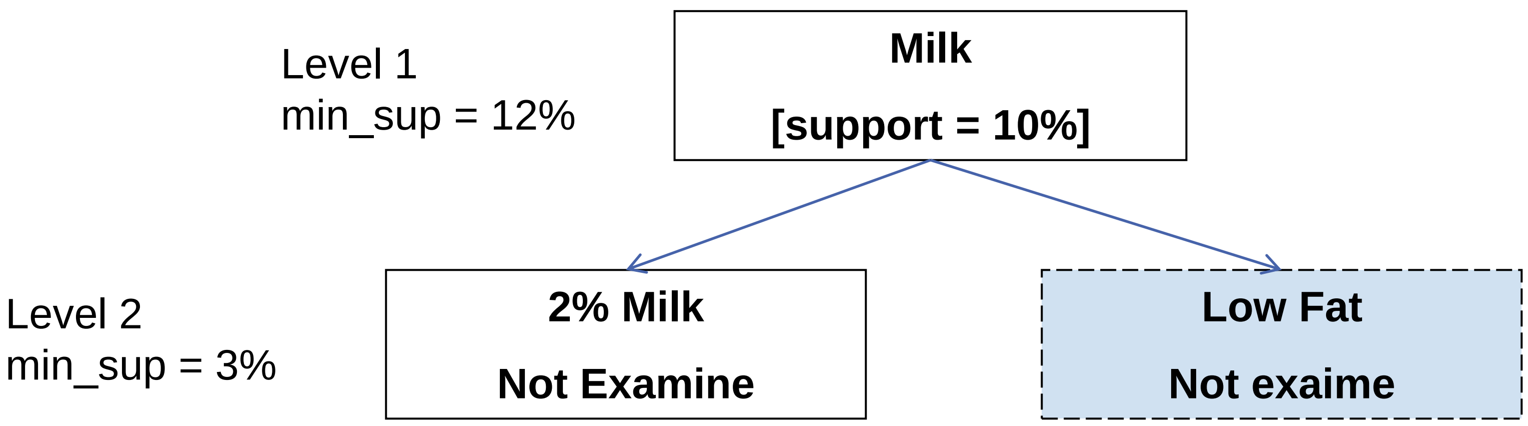
- 4 search strategies:
  - Level-by-level independent
  - Level-cross filtering by single item
  - Level-cross filtering by k-itemset
  - Controlled level-cross filtering by single item

Level by Level Independent

- Full breadth search
- No background knowledge of frequent itemsets is used to pruning
- 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

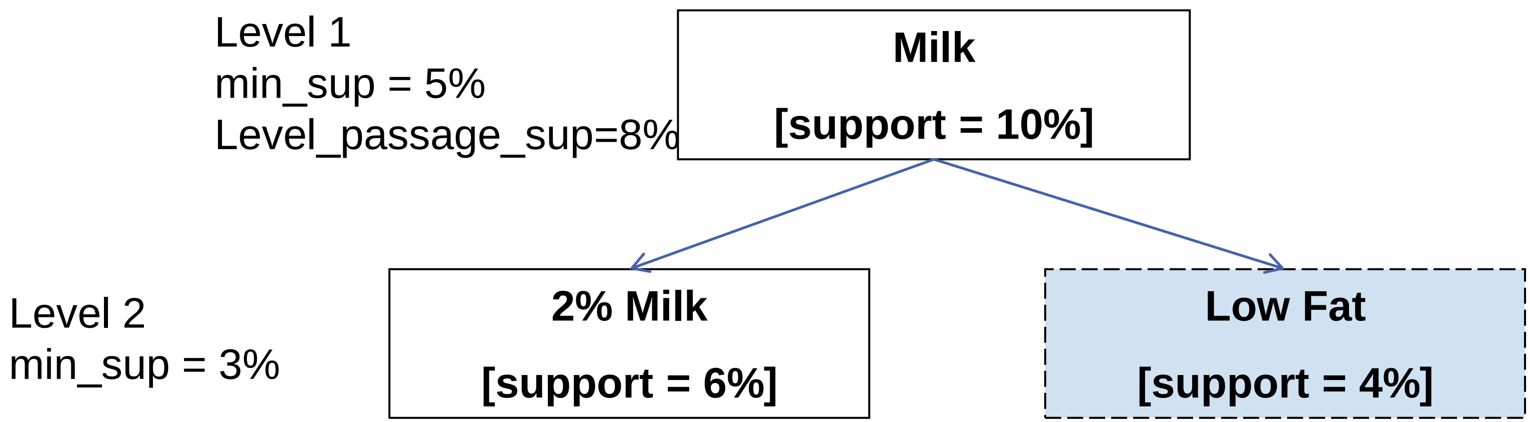


Level-cross Filtering by K-itemset

- A k-itemset at the i-th level is examined iff its corresponding parent k-itemset at the (i-1)-th level is frequent
  - Prune a k-pattern if the corresponding k-pattern at the upper level is infrequent

Controlled Level-cross Filtering by Single Item

- Consider *subfrequent* items passing a passage threshold



ML Associations with Flexible Support Constraints

- 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

Multidimensional Association Rules

- Single dimensional association rule
  - E.g.: buys(bread) $\wedge$ buys(milk) $\Rightarrow$ buys(butter)
- Multidimensional association rule
  - E.g.: age(34-35) $\wedge$ income(30K-50K) $\Rightarrow$ buys(HDTV)

- Attributes types
  - Categorical
    - Finite number of possible values, no ordering among values
  - Numerical
    - Numeric, implicit ordering among values

Example of Quantitative Association Rules

TID	Age	Married	#Cars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	yes	2

TID	Age:20-29 **(A)**	Age:30-40 **(B)**	Married: Yes **(C)**	Married: No **(D)**	#Cars:0-1 **(E)**	#Cars: 2 **(F)**
100	1	0	0	1	1	0
200	1	0	1	0	1	0
300	1	0	0	1	1	0
400	0	1	1	0	0	1
500	0	1	1	0	0	1

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

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

Discretization Issues

- Size of the discretized intervals affect support & confidence

$\{Refund = No, (Income = \$51{,}250)\} \rightarrow \{Cheat = No\}$

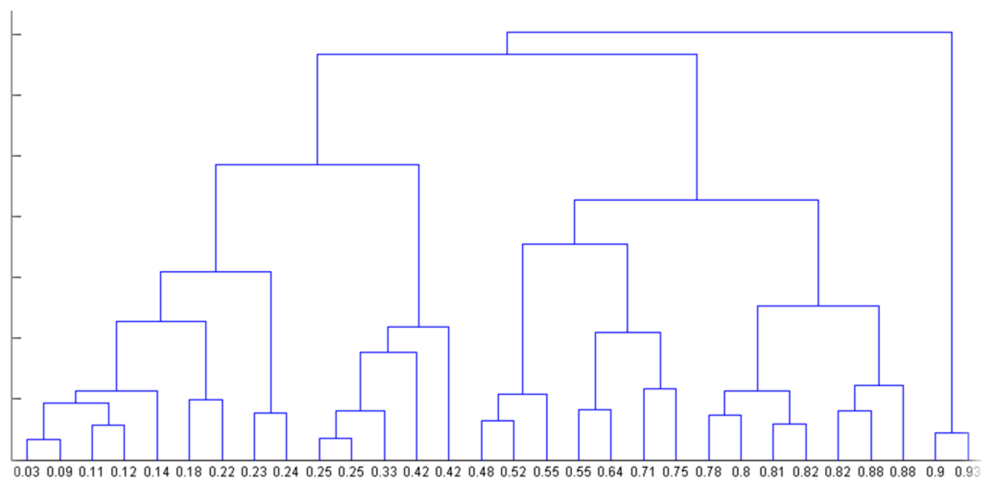
$\{Refund = No, (60K \leq Income \leq 80K)\} \rightarrow \{Cheat = No\}$

$\{Refund = No, (0K \leq Income \leq 1B)\} \rightarrow \{Cheat = No\}$

  - If intervals too small
    - May not have enough support
  - If intervals too large
    - May not have enough confidence
- Potential solution: use all possible intervals

Discretization Issues (Contd.)

- Execution time
  - If intervals contain n values, there are on average $O(n^2)$ possible ranges
- Too many rules



{Refund = No, (Income = \$51,250)} $\rightarrow$ {Cheat = No}

{Refund = No, (51K $\leq$ Income $\leq$ 52K)} $\rightarrow$ {Cheat = No}

{Refund = No, (50K $\leq$ Income $\leq$ 60K)} $\rightarrow$ {Cheat = No}

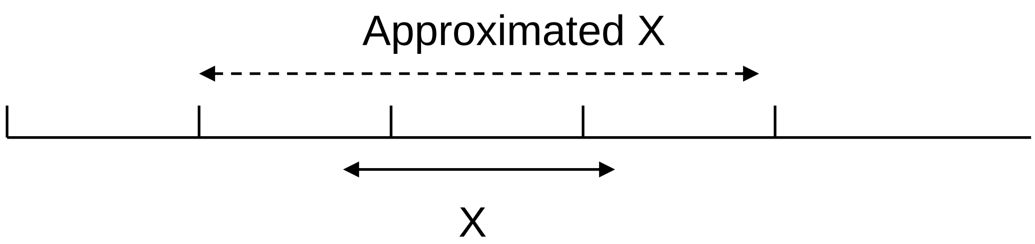
Approach by Srikant & Agrawal

- R. Srikant and R. Agrawal, "Mining Quantitative Association Rules in Large Relational Tables". ACM SIGMOD96

- Preprocess the data
  - Discretize attribute using equi-depth partitioning
    - Use *partial completeness measure* to determine number of partitions
    - Merge adjacent intervals as long as support is less than max-support
- Apply existing association rule mining algorithms
- Determine interesting rules in the output

Partial Completeness Measure

- Discretization will lose information



- Use *partial completeness measure* to determine how much information is lost
  - K-Complete to measure the lost

Partial Completeness

- R : rules obtained before partition
- R’: rules obtained after partition
- Partial Completeness measures the maximum distance between a rule in R and its closest generalization in R’

- $\hat{X}$ is a generalization of itemset X: if

$$\forall x \in attributes(X)[< x, l, u > \in X \wedge < x, l', u' > \in \hat{X} \Rightarrow l' \leq l \leq u \leq u']$$

- The distance is defined by the ratio of support

K-Complete

- C : the set of frequent itemsets
- For any K ≥ 1, P is K-complete with regards to C if:
  - $P \subseteq C$
  - For any itemset X (or its subset) in C, there exists a generalization whose support is no more than K times that of X (or its subset)
- The smaller K is, the less the information lost

K-Complete Example

Number	Itemset	Support
1	{<Age: 20 ..30>}	5%
2	{<Age: 20 ..40>}	6%
3	{<Age: 20 ..50>}	8%
4	{<Cars: 1 ..2>}	5%
5	{<Cars: 1 ..2>}	6%
6	{<Age: 20 ..30>,<Cars: 1 ..2>}	4%
7	{<Age: 20 ..40>,<Cars: 1 ..3>}	5%

1.2times

1.3times

1.2times

1.25times

Itemsets 2,3,5, and 7 form a 1.5-complete set

Interestingness Measure

<Age: 20 .. 30> → <Cars: 1..2> (8% sup., 70% conf.)

~~<Age: 20 .. 25> → <Cars: 1..2> (2% sup., 70% conf.)~~

- Given an itemset: $Z = \{z_1, z_2, \ldots, z_k\}$ & its generalization $Z' = \{z_1', z_2', \ldots, z_k'\}$

  P(Z): support of Z

  $E_{Z'}(Z)$: expected support of Z based on Z'

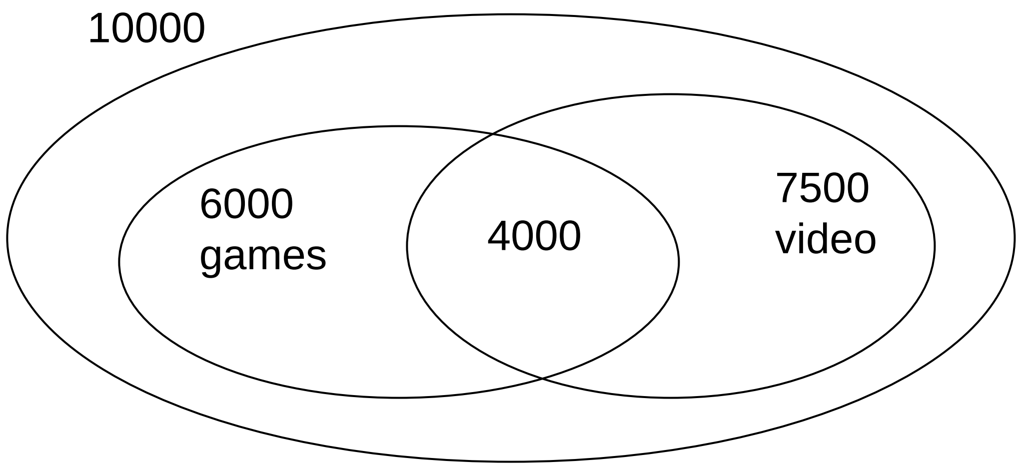
$$E_{Z'}(Z) = \frac{P(z_1)}{P(z_1')} \times \frac{P(z_2)}{P(z_2')} \times \cdots \times \frac{P(z_k)}{P(z_k')} \times P(Z')$$

- Z is R-interesting with regards to Z' if $P(Z) \geq R \times E_{Z'}(Z)$

From Association Mining to Correlation Analysis

[redacted]

Strong Rules & Interesting



- Corr(A,B)=P(AUB)/(P(A)P(B))
  - Corr(A, B)=1, A & B are independent
  - Corr(A, B)<1, occurrence of A is negatively correlated with B
  - Corr(A, B)>1, occurrence of A is positively correlated with B

- E.g. Corr(games, videos)=0.4/(0.6*0.75)=0.89
  - In fact, games & videos are negatively associated
  - Purchase of one actually decrease the likelihood of purchasing the other

Association Rules with Weighted Items

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

- 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)