

---

# **Association Rule Mining**

Dr. Faisal Kamiran

# Association Rule Mining

- 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

<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 of Association Rules

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\},$   
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\},$   
 $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\},$

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 ( $\sigma$ )**

- 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

- **Association Rule**

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

<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

- **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  $X$

Example:

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

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

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

# Association Rule Mining Task

---

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

# Mining Association Rules

<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 of Rules:

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

## Observations:

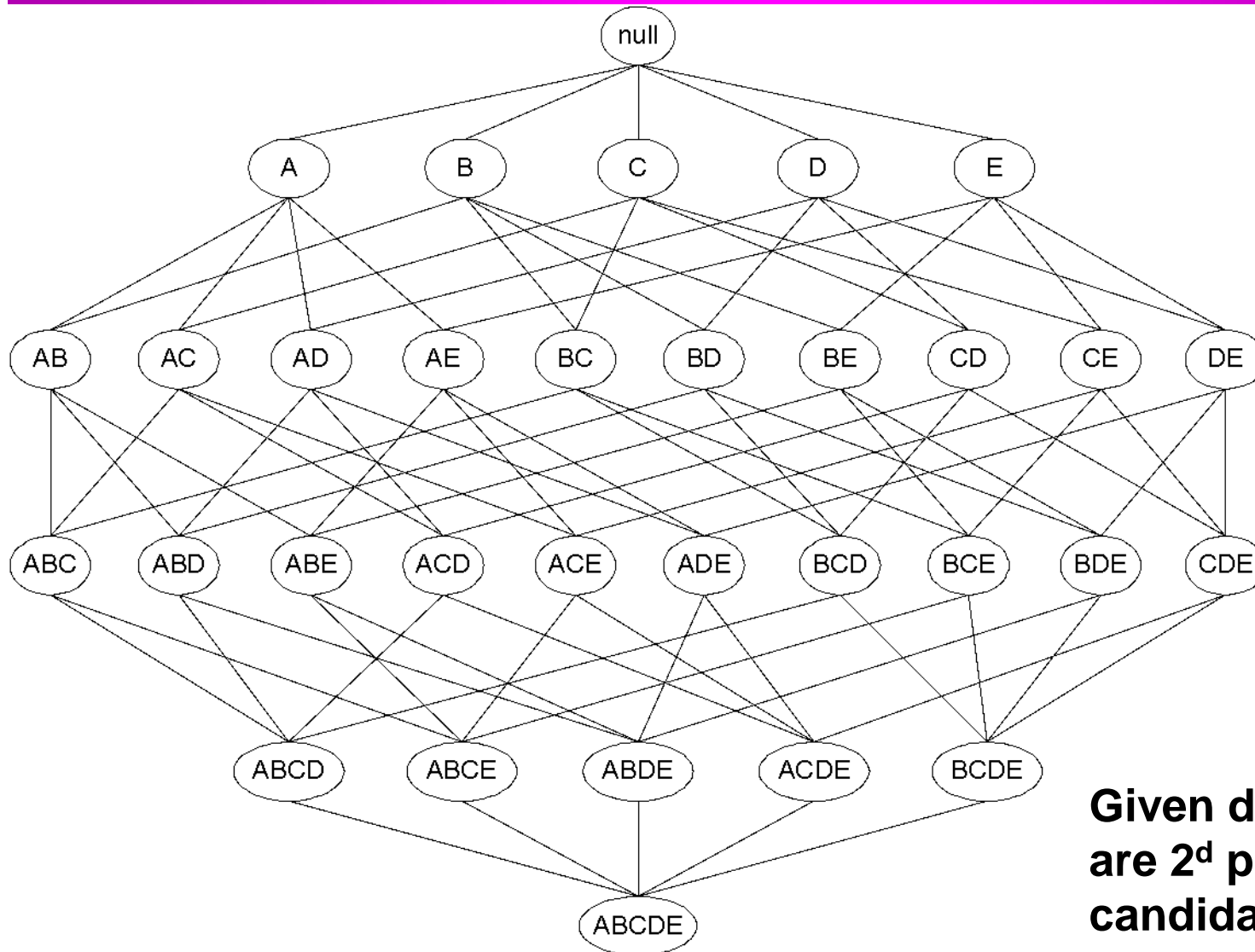
- All the above rules are binary partitions of the same itemset:  
 $\{\text{Milk, Diaper, Beer}\}$
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

# Mining Association Rules

---

- Two-step approach:
  1. Frequent Itemset Generation
    - Generate all itemsets whose support  $\geq$  minsup
  2. 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

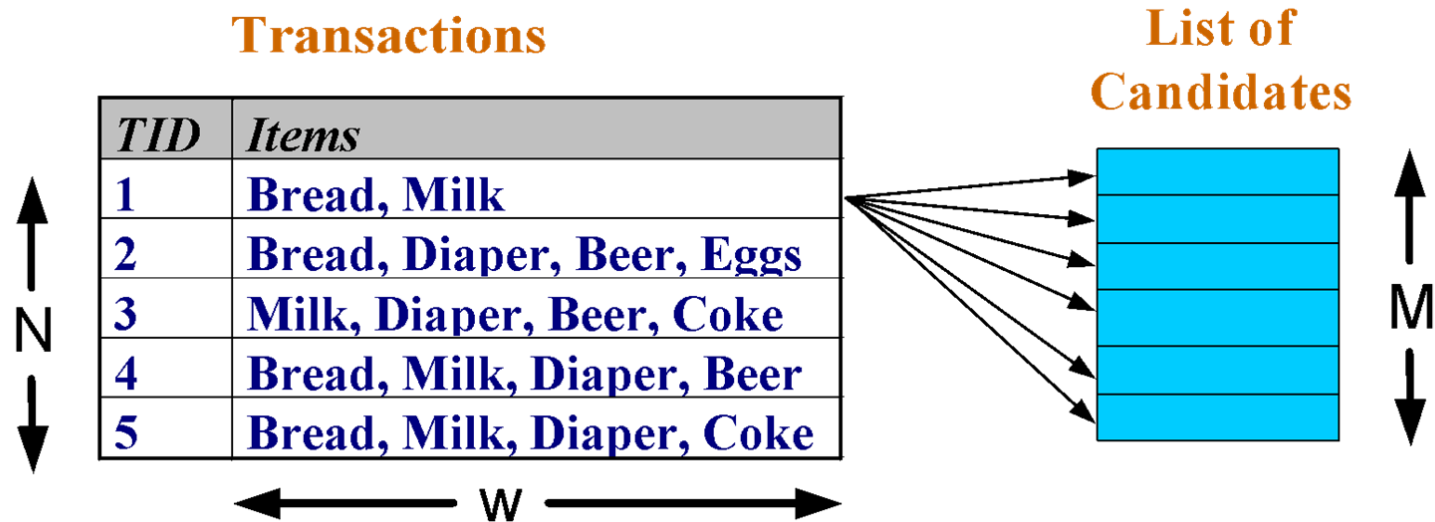


**Given d items, there are  $2^d$  possible candidate itemsets**



# Frequent Itemset Generation

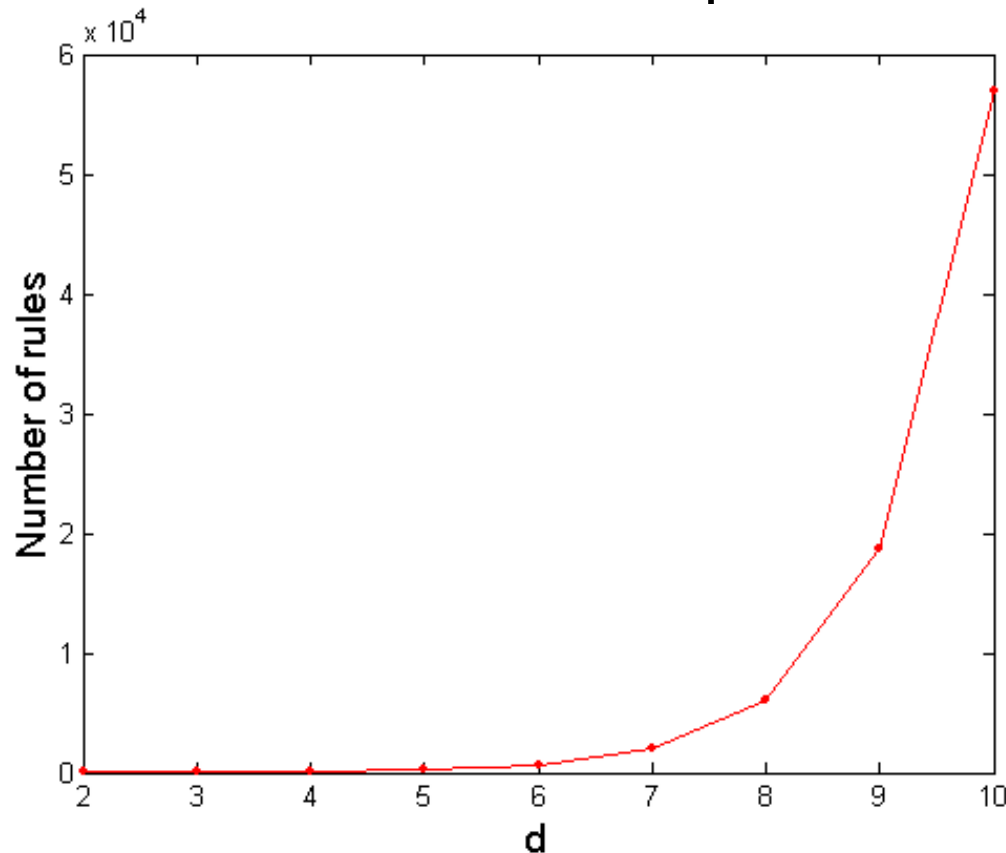
- Brute-force approach:
  - Each itemset in the lattice is a **candidate** frequent itemset
  - Count the support of each candidate by scanning the database



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

# Computational Complexity

- Given  $d$  unique items:
  - Total number of itemsets =  $2^d$
  - Total number of possible association rules:



$$= 3^d - 2^{d+1} + 1$$

**If  $d=6$ ,  $R = 602$  rules**

# Frequent Itemset Generation Strategies

---

- 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

# Reducing Number of Candidates

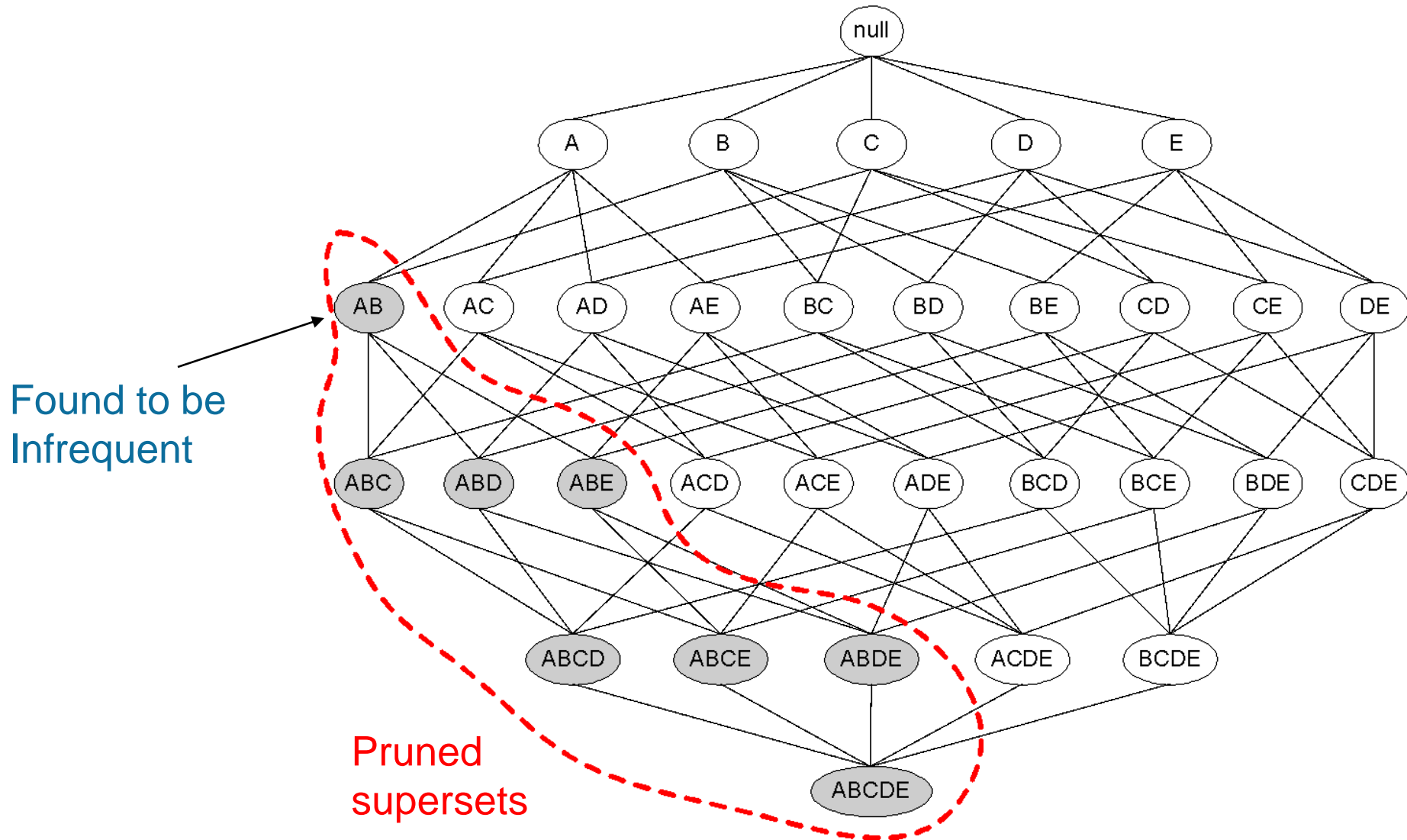
---

- **Apriori principle:**
  - If an itemset is frequent, then all of its subsets 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

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

What about {**Beer**, Diaper, **Milk**} ?  
 And, {Bread, **Milk**, **Beer**}?  
 And, {**Bread**, Diaper, **Beer**}?



# Apriori Algorithm

---

- Method:
  - Let  $k=1$
  - Generate frequent itemsets of length 1
  - Repeat until no new frequent itemsets are identified
    - ◆ Generate length  $(k+1)$  candidate itemsets from length  $k$  frequent itemsets
    - ◆ Count the support of each candidate by scanning the DB
    - ◆ Eliminate candidates that are infrequent, leaving only those that are frequent

# The Apriori Algorithm—An Example

---

$\text{Sup}_{\min} = 2$

Database TDB

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



# The Apriori Algorithm—An Example

$\text{Sup}_{\min} = 2$

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

# The Apriori Algorithm—An Example

$\text{Sup}_{\min} = 2$

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

# The Apriori Algorithm—An Example

$\text{Sup}_{\min} = 2$

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

$C_2$

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

# The Apriori Algorithm—An Example

$\text{Sup}_{\min} = 2$

Database TDB

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

1<sup>st</sup> scan

$C_1$

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

$L_1$

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

$C_2$

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

2<sup>nd</sup> scan

$C_2$

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

# The Apriori Algorithm—An Example

$\text{Sup}_{\min} = 2$

Database TDB

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

1<sup>st</sup> scan

$C_1$

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

$L_1$

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

$C_2$

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

2<sup>nd</sup> scan

$C_2$

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

# The Apriori Algorithm—An Example

$\text{Sup}_{\min} = 2$

Database TDB

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

1<sup>st</sup> scan

$C_1$

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

$L_1$

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

$C_2$

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

2<sup>nd</sup> scan

$C_2$

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

$C_3$

Itemset
{B, C, E}

# The Apriori Algorithm—An Example

$\text{Sup}_{\min} = 2$

Database TDB

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

1<sup>st</sup> scan

$C_1$

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

$L_1$

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

$C_2$

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

2<sup>nd</sup> scan

$C_2$

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

$C_3$

Itemset
{B, C, E}

3<sup>rd</sup> scan

$L_3$

Itemset	sup
{B, C, E}	2

# ECLAT

- For each item, store a list of transaction ids (tids)

Horizontal  
Data Layout

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

Vertical Data Layout

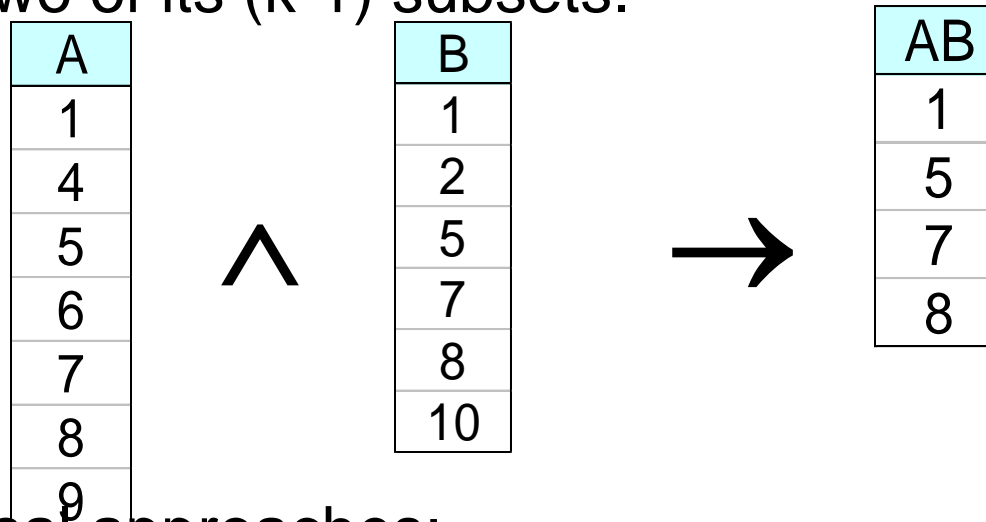
A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				





# ECLAT

- Determine support of any k-itemset by intersecting tid-lists of two of its (k-1) subsets.



- 3 traversal approaches:
  - top-down, bottom-up and hybrid
- Advantage: very fast support counting
- Disadvantage: intermediate tid-lists may become too large for memory

# 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  $\emptyset \rightarrow L$ )

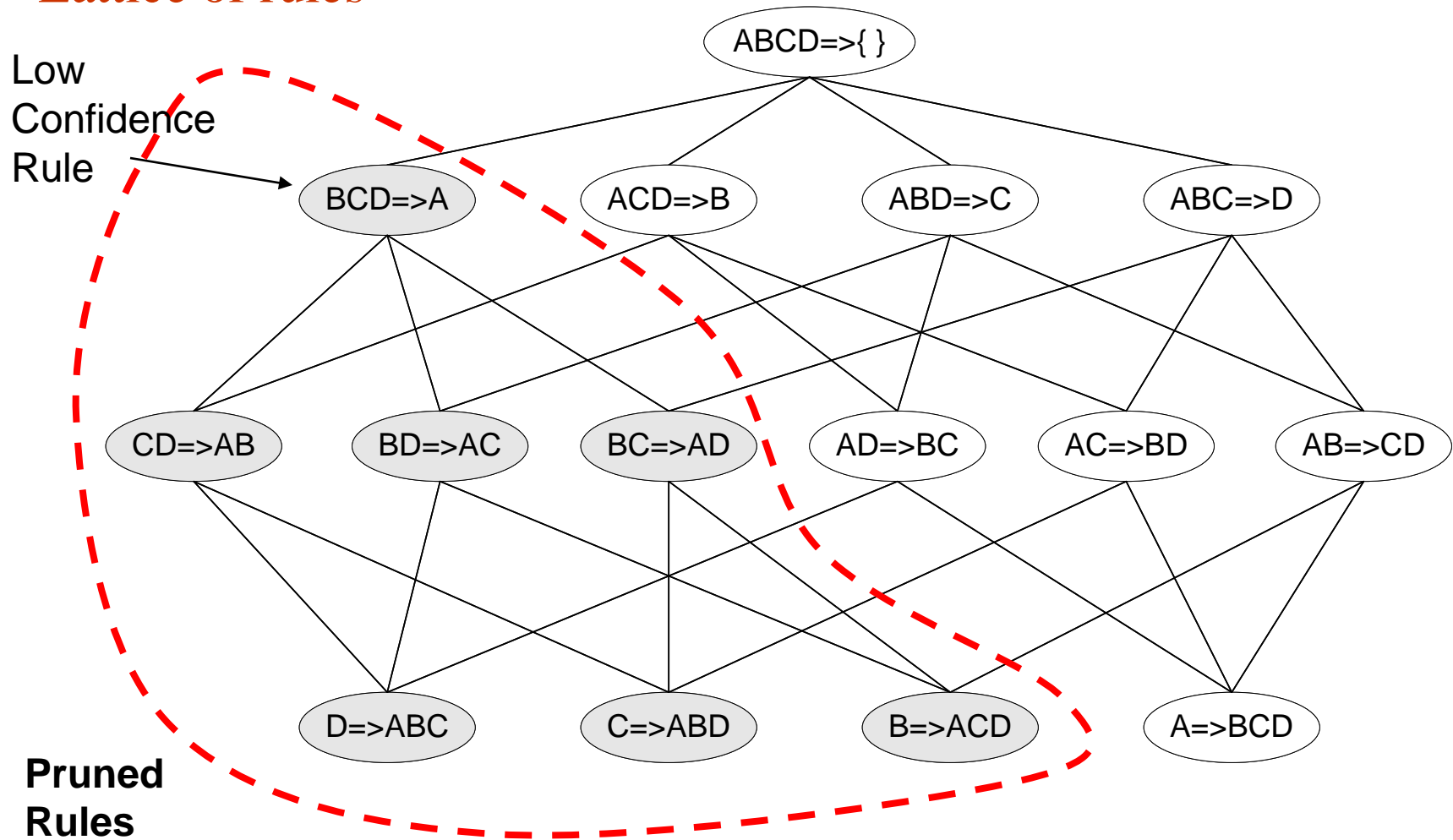
# Rule Generation

- 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.,  $L = \{A, B, C, D\}$ :

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

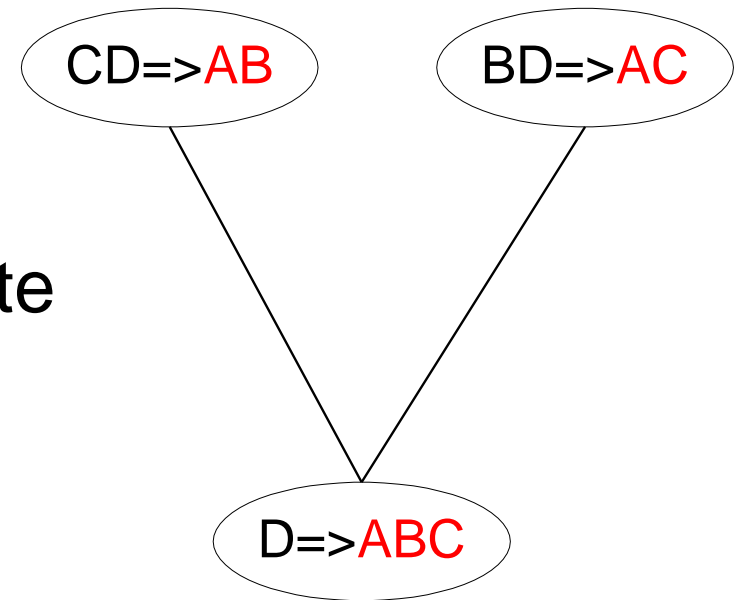
# Rule Generation for Apriori Algorithm

## Lattice of rules



# Rule Generation for Apriori Algorithm

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



# 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
- Interestingness measures can be used to prune/rank the derived patterns
- In the original formulation of association rules, support & confidence are the only measures used

# Computing Interestingness Measure

- Given a rule  $X \rightarrow Y$ , information needed to compute rule interestingness can be obtained from a contingency table

Contingency table for  $X \rightarrow Y$

	Y	$\overline{Y}$	
X	$f_{11}$	$f_{10}$	$f_{1+}$
$\overline{X}$	$f_{01}$	$f_{00}$	$f_{0+}$
	$f_{+1}$	$f_{+0}$	$ T $

$f_{11}$ : support of X and Y

$f_{10}$ : support of  $\underline{X}$  and  $\overline{Y}$

$f_{01}$ : support of  $\overline{X}$  and  $\underline{Y}$

$f_{00}$ : support of  $\overline{X}$  and  $\overline{Y}$

Used to define various measures

□ support, confidence, lift, Gini, J-measure, etc.

# Drawback of Confidence

	Coffee	<u>Coffee</u>	
Tea	15	5	20
<u>Tea</u>	75	5	80
	90	10	100

Association Rule: Tea  $\rightarrow$  Coffee

- Confidence =  $P(\text{Coffee}|\text{Tea})$
- but  $P(\text{Coffee})$
- Although confidence is high, rule is misleading
- $P(\text{Coffee}|\overline{\text{Tea}})$  :



# Statistical-based Measures

---

- Measures that take into account statistical dependence

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

# Example: Lift/Interest

	Coffee	<u>Coffee</u>	
Tea	15	5	20
<u>Tea</u>	75	5	80
	90	10	100

Association Rule: Tea  $\rightarrow$  Coffee

Confidence =  $P(\text{Coffee}|\text{Tea}) = 0.75$

but  $P(\text{Coffee}) = 0.9$

$\Rightarrow \text{Lift} = 0.75/0.9 = 0.8333 (< 1, \text{ therefore is negatively associated})$