Association Analysis

顏安孜

azyen@nycu.edu.tw

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

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

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

- **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

Definition: Association Rule

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**

- Association Rule
  - An implication expression of the form X → Y, where X and Y are itemsets
  - Example:
    {Milk, Diaper} → {Beer}
- Rule Evaluation Metrics

$$\text{Support, } s(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{N};$$

$$\text{Confidence, } c(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}.$$

Example:

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

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

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

Exercise

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

- Frequent itemset: let minsup = 50%
  - Freq. 1-itemset: Beer: 3, Nuts: 3, Diaper: 4, Eggs: 3
  - Freq. 2-itemset: {Beer, Diaper}: 3
- Association rules: let minconf = 50%
  - Beer → Diaper {support = 60%, confidence = 100%}
  - Diaper → Beer {support = 60%, confidence = 75%}

Association Rule Mining Task

- Given a set of transactions, 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

  $\Rightarrow$ Computationally prohibitive!

Mining Association Rules

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:

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

Observations:

- All the above rules are binary partitions of the same itemset:
  {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

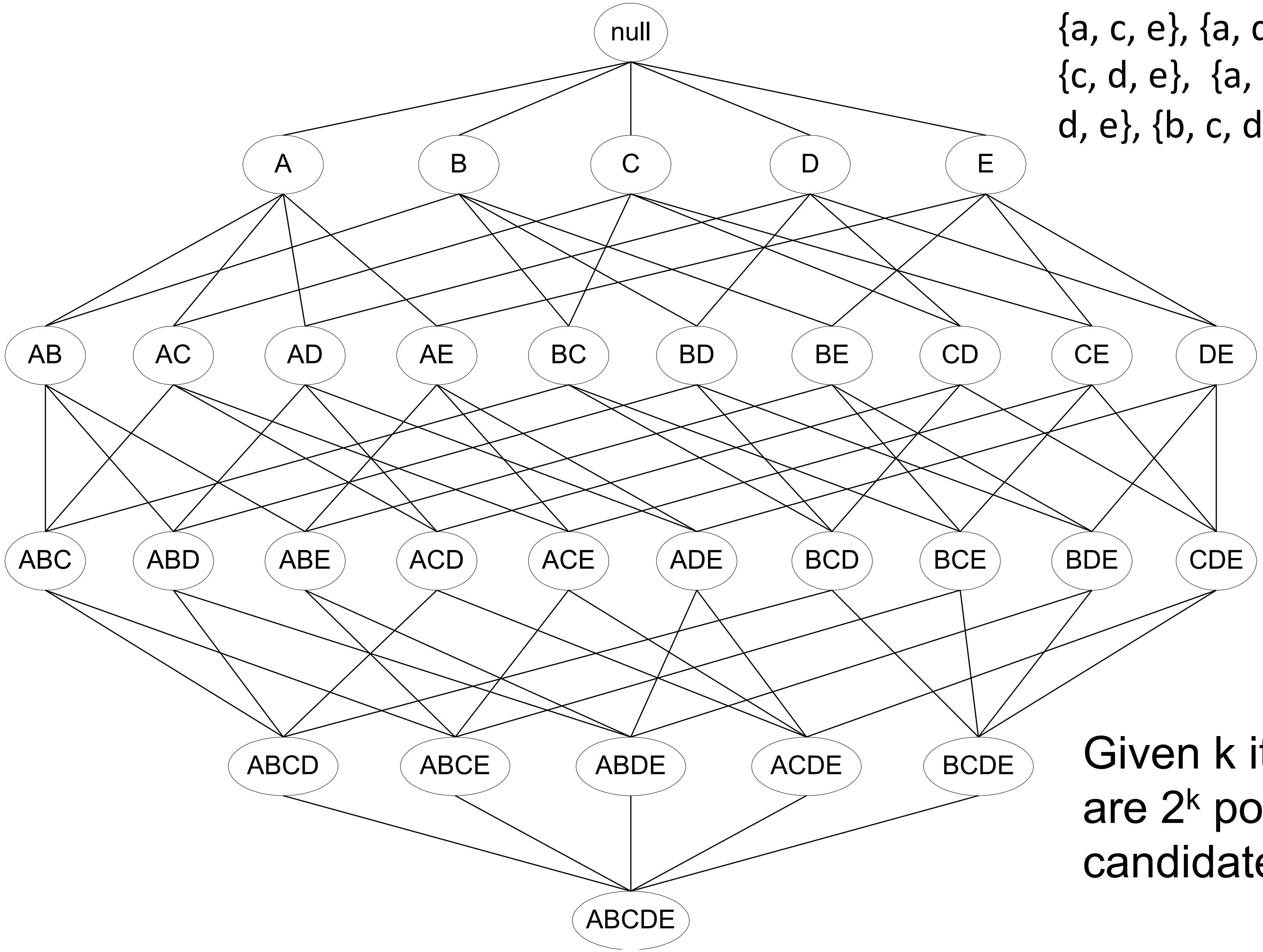
- Two-step approach:
  1. Frequent Itemset Generation
     – Generate all itemsets whose support $\geq$ minsup

  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

All possible itemsets: {a}, {b}, {c}, {d}, {e}, {a, b}, {a, c}, {a, d}, {a, e}, {b, c}, {b, d}, {b, e}, {c, d}, {c, e}, {d, e}, {a, b, c}, {a, b, d}, {a, b, e}, {a, c, d}, {a, c, e}, {a, d, e}, {b, c, d}, {b, c, e}, {b, d, e}, {c, d, e}, {a, b, c, d}, {a, b, c, e}, {a, b, d, e}, {a, c, d, e}, {b, c, d, e}, and {a, b, c, d, e}



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

Frequent Itemset Generation

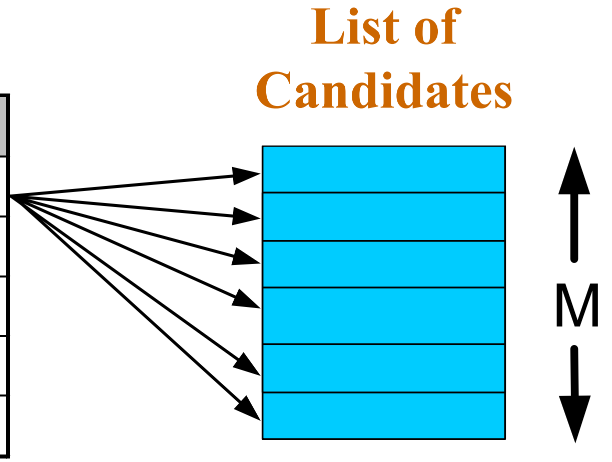
- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database

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 $M = 2^d$!!!

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

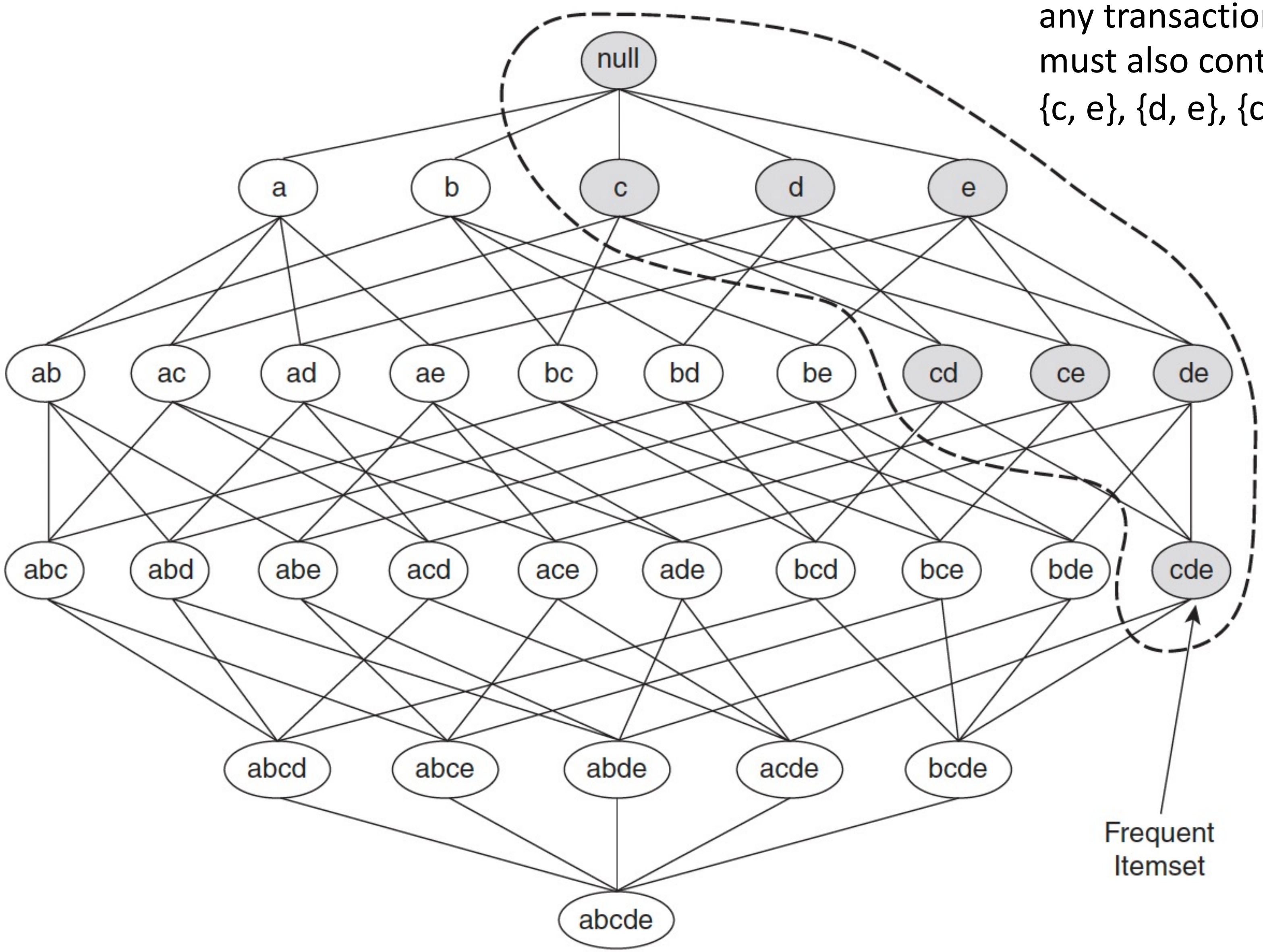
- Apriori principle holds due to the following property of the support measure:

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

  - Support of an itemset never exceeds the support of its subsets

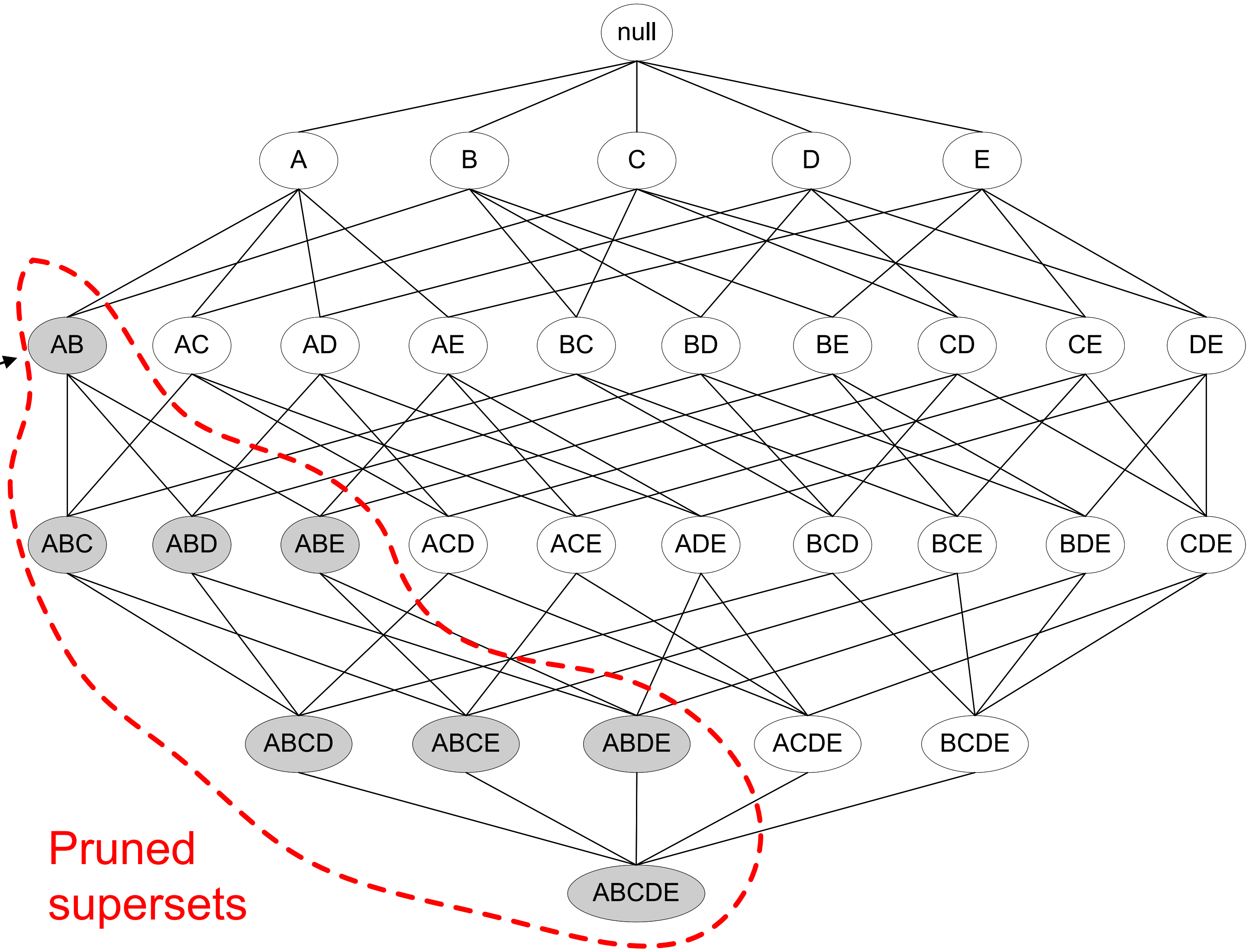
An illustration of the Apriori principle

any transaction that contains {c, d, e} must also contain its subsets, {c, d}, {c, e}, {d, e}, {c}, {d}, and {e}.



{a, b} is infrequent, then all of its supersets must be infrequent too

Found to be Infrequent



The Apriori Algorithm (Pseudo-Code)

C_k: Candidate itemset of size k

F_k : Frequent itemset of size k

k := 1;
F_k := {frequent items}; // frequent 1-itemset
While (F_k != ∅) **do {** // when F_k is non-empty
  C_{k+1} := candidates generated from F_k; // candidate generation
  Derive F_{k+1} by counting candidates in C_{k+1} with respect to *TDB* at minsup;
  k := k + 1
  }
return $\cup_k F_k$ // return F_k generated at each level

The Apriori Algorithm—An Example

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

minsup=2

C1

itemset	support
{1}	3
{2}	3
{3}	4
{4}	1
{5}	4

F1

itemset	support
{1}	3
{2}	3
{3}	4
{5}	4

Only items present in F1

C2

itemset	support
{1,2}	1
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

F2

itemset	support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

TID	Items
T1	1 3 4
T2	2 3 5
T3	1 2 3 5
T4	2 5
T5	1 3 5

minsup=2

F2

itemset	support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

C3

itemset	In F1?
{1,2,3}, {1,2}, {1,3}, {2,3}	No
{1,2,5}, {1,2}, {1,5}, {2,5}	No
{1,3,5}, {1,5}, {1,3}, {3,5}	Yes
{2,3,5}, {2,3}, {2,5}, {3,5}	Yes

F3

itemset	support
{1,3,5}	2
{2,3,5}	2

C4

itemset	support
{1,2,3,5}	1

Subset Creation

- **For I = {1,3,5}**, subsets are {1,3}, {1,5}, {3,5}, {1}, {3}, {5}
  For I = {2,3,5}, subsets are {2,3}, {2,5}, {3,5}, {2}, {3}, {5}

- For every subsets S of I, you output the rule
- S -> (I-S) (means S recommends I-S)
- if **support(I) / support(S) >= min_conf value**

Applying Rules

itemset	support
{1,3,5}	2
{2,3,5}	2

itemset	support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

Applying rules to itemset F3

- **{1,3,5}**
  - **Rule 1:** {1,3} -> ({1,3,5} — {1,3}) means 1 & 3 -> 5

    Confidence = support(1,3,5)/support(1,3) = 2/3 = **66.66% > 60%**

    Hence Rule 1 is **Selected**
  - **Rule 2:** {1,5} -> ({1,3,5} — {1,5}) means 1 & 5 -> 3

    Confidence = support(1,3,5)/support(1,5) = 2/2 = **100% > 60%**

    Rule 2 is **Selected**
  - **Rule 3:** {3,5} -> ({1,3,5} — {3,5}) means 3 & 5 -> 1

    Confidence = support(1,3,5)/support(3,5) = 2/3 = **66.66% > 60%**

    Rule 3 is **Selected**

Applying Rules

itemset	support
{1,3,5}	2
{2,3,5}	2

itemset	support
{1}	3
{2}	3
{3}	4
{5}	4

Applying rules to itemset F3

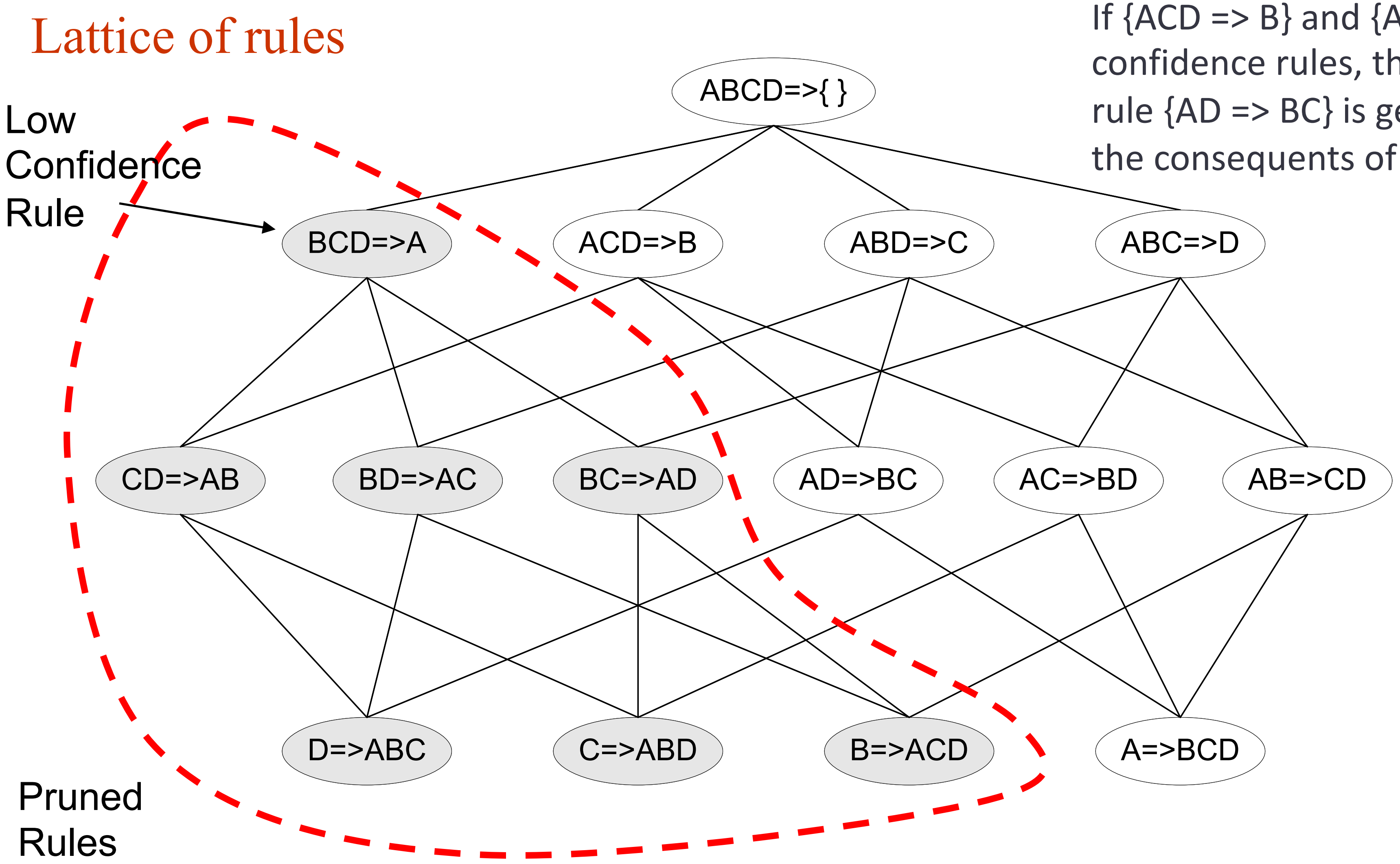
- {1,3,5}
  - **Rule 4:** {1} -> ({1,3,5} — {1}) means 1 -> 3 & 5
    Confidence = support(1,3,5)/support(1) = 2/3 = **66.66% > 60%**
    Rule 4 is **Selected**
  - **Rule 5:** {3} -> ({1,3,5} — {3}) means 3 -> 1 & 5
    Confidence = support(1,3,5)/support(3) = 2/4 = **50% <60%**
    Rule 5 is **Rejected**
  - **Rule 6:** {5} -> ({1,3,5} — {5}) means 5 -> 1 & 3
    Confidence = support(1,3,5)/support(5) = 2/4 = 50% < 60%
    Rule 6 is **Rejected**

Compact Representation of Frequent Itemsets

- Frequent itemsets can be very numerous in practice.
- Identifying a small representative set of frequent itemsets is useful.
- Maximal and closed frequent itemsets are two ways to represent frequent itemsets more compactly.

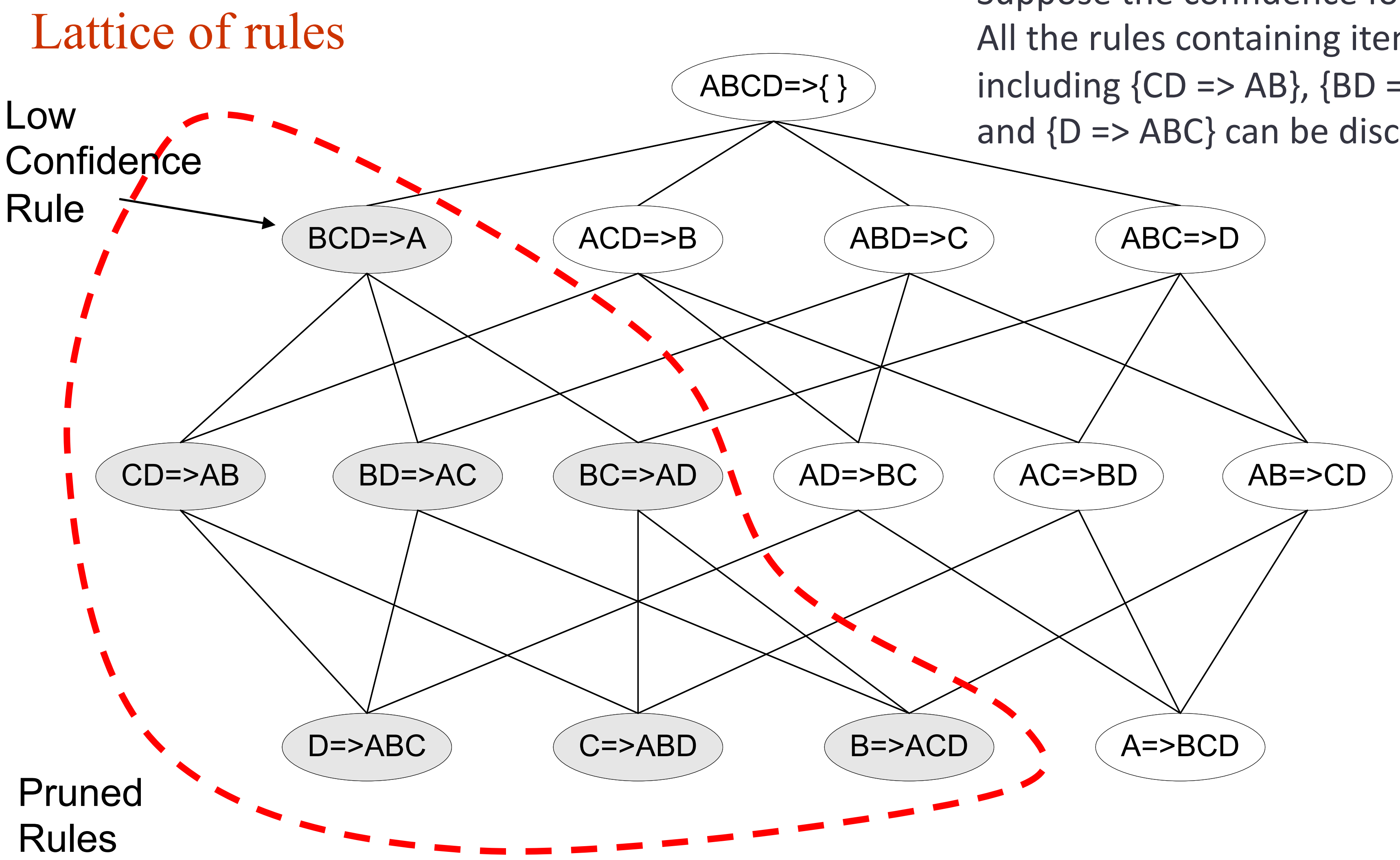
Rule Generation for Apriori Algorithm

If {ACD => B} and {ABD => C} are high confidence rules, then the candidate rule {AD => BC} is generated by merging the consequents of both rules.



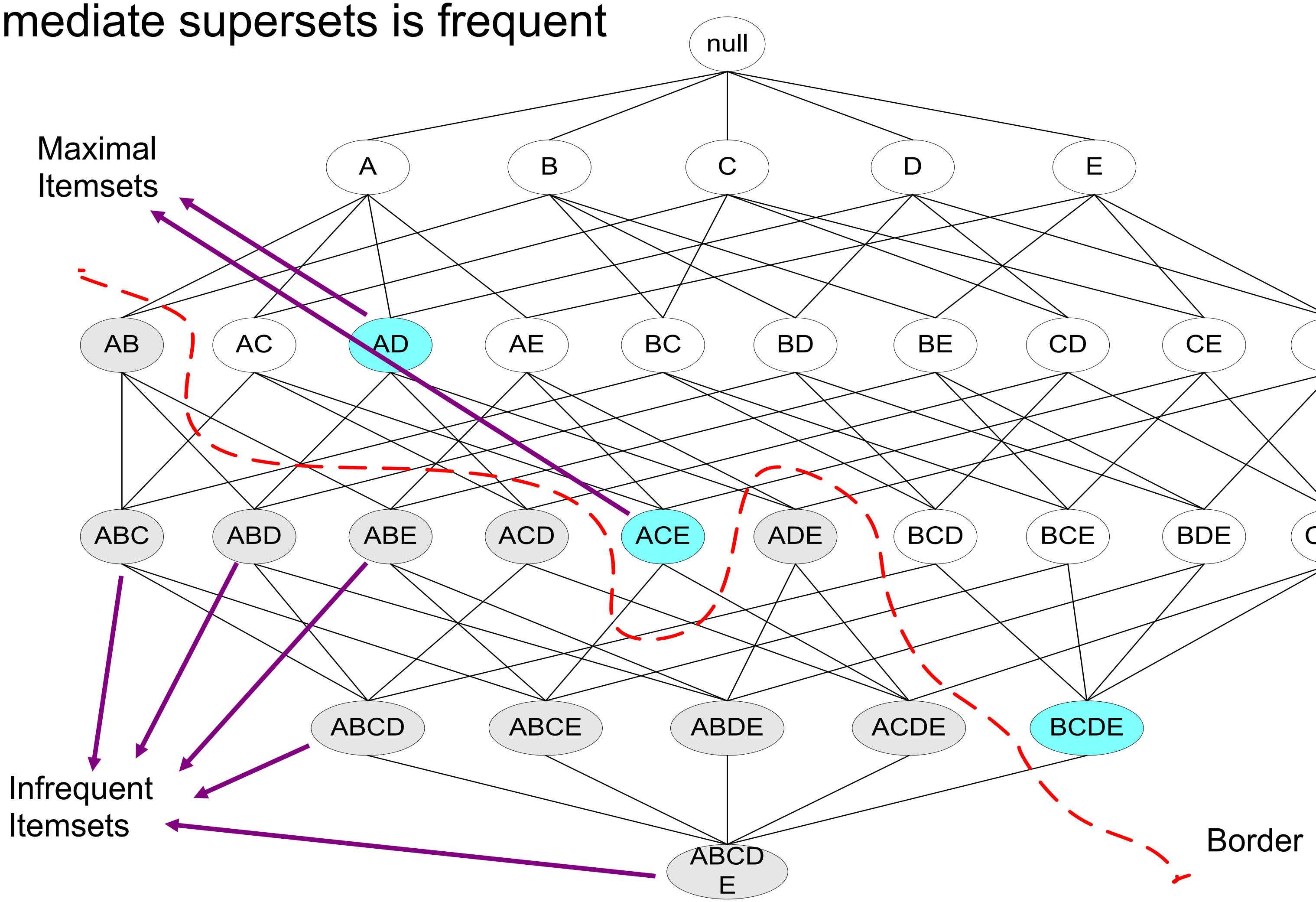
Rule Generation for Apriori Algorithm

Suppose the confidence for {BCD => A} is low. All the rules containing item a in its consequent, including {CD => AB}, {BD => AC}, {BC to AD}, and {D => ABC} can be discarded.



Maximal Frequent Itemset

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

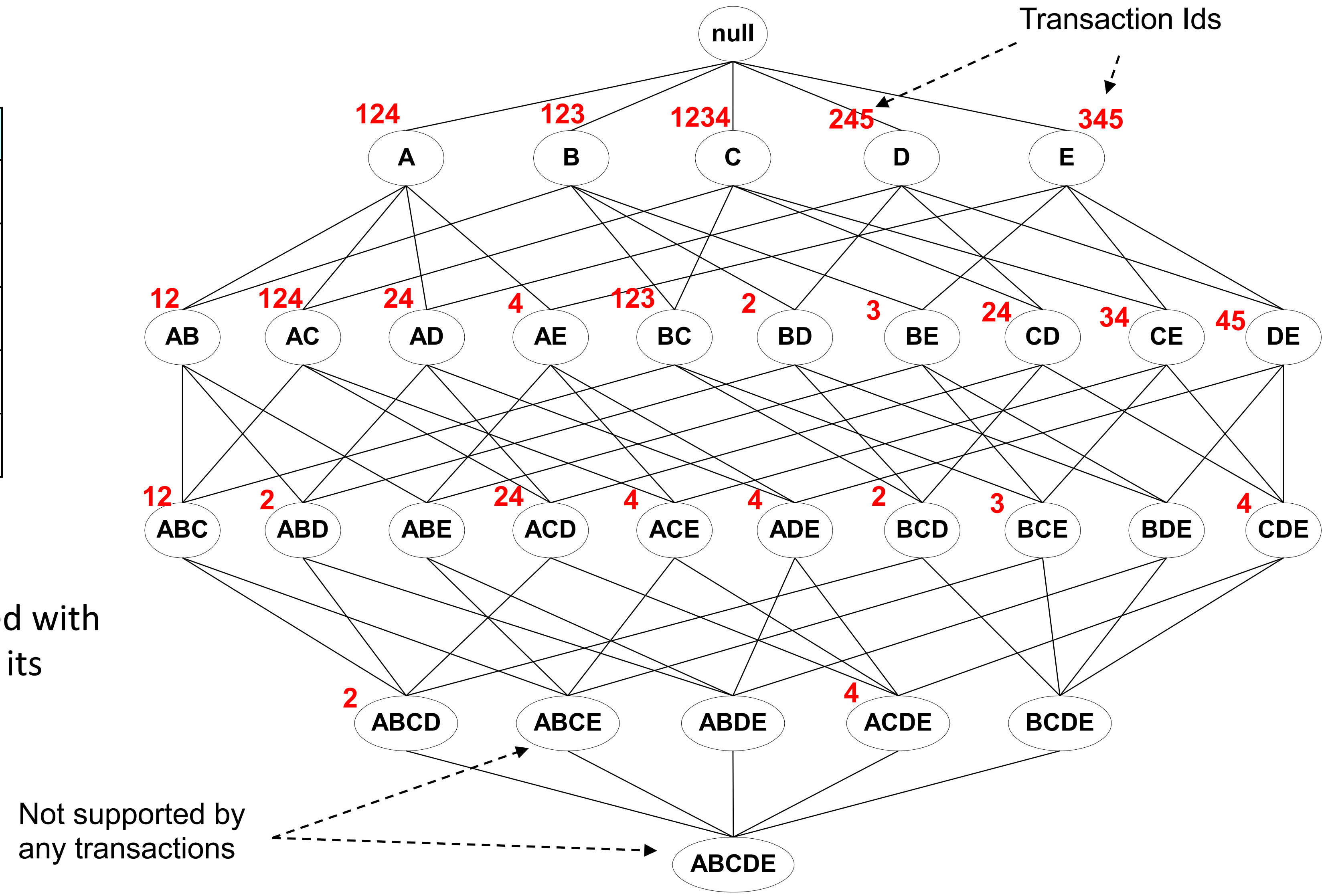


Closed Itemset

- An itemset X is closed if none of its immediate supersets has the same support as the itemset X.
- X is not closed if at least one of its immediate supersets has support count as X.

Maximal vs Closed Frequent Itemsets

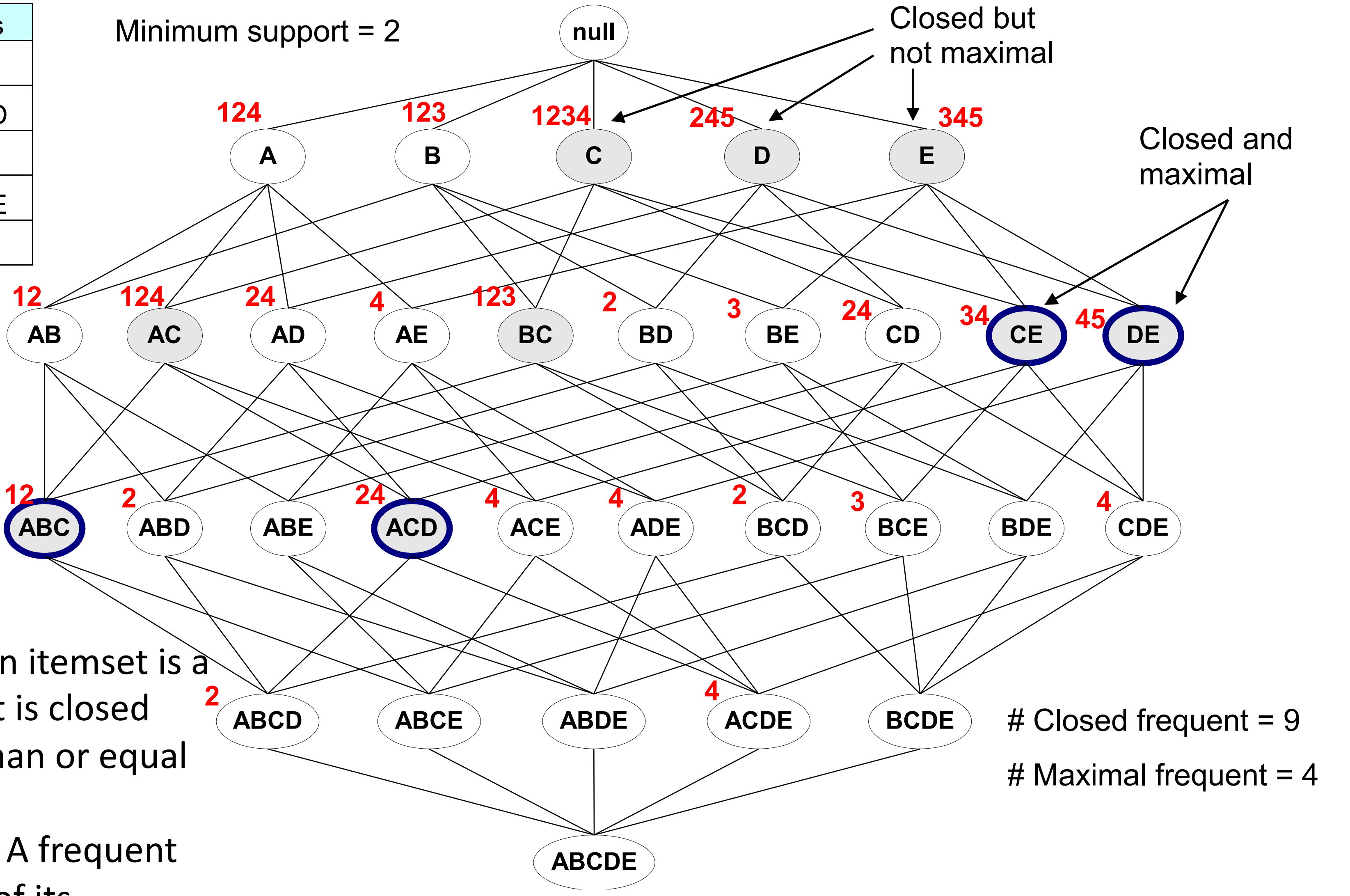
TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE



The node {b, c} is associated with transaction IDs 1, 2, and 3, its support count is 3.

Maximal vs Closed Frequent Itemsets

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



Closed Frequent Itemset: An itemset is a closed frequent itemset if it is closed and its support is greater than or equal to minsup

Maximal Frequent Itemset: A frequent itemset is maximal if none of its immediate supersets are frequent

\# Closed frequent = 9

\# Maximal frequent = 4

Maximal vs Closed Itemsets

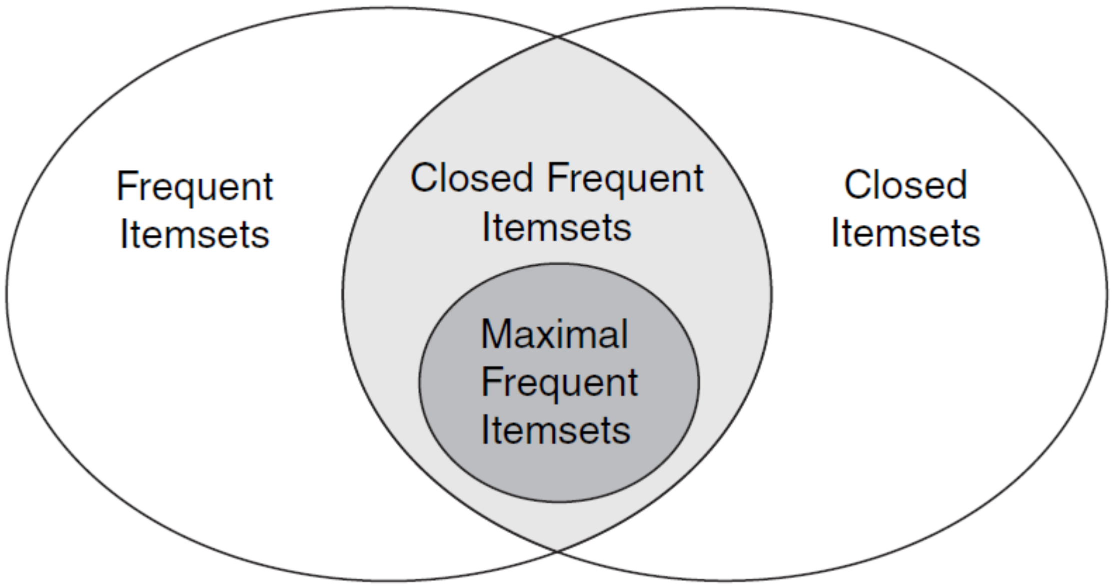


Figure 5.18. Relationships among frequent, closed, closed frequent, and maximal frequent itemsets.

Pattern Evaluation

- Association rule algorithms can produce large number of rules
- Not all patterns/rules are interesting
- Interestingness measures can be used to prune/rank the patterns
    - Objective interestingness measures
        - Support, confidence, correlation, …
    - Subject interestingness measures
        - Query-based, user’s knowledge base, visualization

Computing Interestingness Measure

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

Contingency table

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

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

Used to define various measures

- support, confidence, Gini, entropy, etc.

Limitation of the Support-Confidence Framework

- Are *s* and *c* interesting in association rules: "A ⇒ B" [*s*, *c*]?
- Example: suppose one school may have the following statistics on #of students who may play basketball and/or eat cereal:

2-way contingency table	play-basketball	¬play-basketball	sum(row)
eat-cereal	400	350	750
¬eat-cereal	200	50	250
sum(col.)	600	400	1000

- Association rule mining may generate the following:
  - play-basketball ⇒ eat-cereal [40%, 66.7%] (higher s & c)
- But this strong association rules is misleading: the overall % of students eating cereal is 75% > 66.7%, a more telling rule:
  - ¬play-basketball ⇒ eat-cereal [35%, 87.5%] (higher s & c)

Interestingness Measures: Lift

- Measure of dependent/correlated event: list

$$lift(B,C) = \frac{c(B \to C)}{s(C)} = \frac{s(B \cup C)}{s(B) \times s(C)}$$

	B	¬B	Σ_{row}
C	400	350	750
¬C	200	50	250
Σ_{col}	600	400	1000

- $lift(B,C)$ may tell how B and C are correlated
  - $lift(B,C)$ = 1: B and C are independent
  - > 1: positively correlated
  - < 1: negatively correlated
- For example,

$$lift(B,C) = \frac{\frac{400}{1000}}{\frac{600}{1000} \times \frac{750}{1000}} = 0.89 \quad lift(B,\neg C) = \frac{\frac{200}{1000}}{\frac{600}{1000} \times \frac{250}{1000}} = 1.33$$

- Thus B and C are negatively correlated since $lift(B,C)$ < 1;
- B and ¬C are positively correlated since $lift(B,\neg C)$ > 1

Interestingness Measures: χ^2

- Another measure to test correlated events: χ^2

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Excepted}$$

Observed value

Expected value

	B	¬B	Σ_{row}
C	400 (450)	350 (300)	750
¬C	200 (150)	50 (100)	250
Σ_{col}	600	400	1000

- General rules
  - χ^2=0: independent
  - χ^2>0: correlated, either positive or negative, so it needs additional test
  - Now, $\chi^2 = \frac{(400-450)^2}{450} + \frac{(350-300)^2}{300} + \frac{(200-150)^2}{150} + \frac{(50-100)^2}{100} = 55.56$
  - χ^2 shows B and C are negatively correlated since the expected value is 450 but the observed is only 400

Lift and χ^2: Are they always good measures?

- Null transactions: transactions that contain neither B nor C
- Let's examine the dataset D
  - BC(100) is much rarer than B¬C(1000) and ¬BC (1000), but there are many ¬B¬C (100000)
  - Unlikely B & C will happen together
- But lift(B, C) = 8.44 >> 1 (Lift shows B and C are strongly positively correlated)
- χ^2=670: Observed (BC) >> expected value (11.85)

	B	¬B	Σ_{row}
C	100	1000	1100
¬C	1000	100000	101000
Σ_{col}	1100	101000	102100

	B	¬B	Σ_{row}
C	100 (11.85)	1000	1100
¬C	1000 (1088.15)	100000	101000
Σ_{col}	1100	101000	102100

Null Invariance Measures

- Null invariance: value does not change with the # of null-transactions
- A few interestingness measure; some are null invariant

Measure	Definition	Range	Null-Invariant
$\chi^2(A,B)$	$\sum_{i,j=0,1} \frac{(e(a_i b_j)-o(a_i b_j))^2}{e(a_i b_j)}$	$[0,\infty]$	No
$Lift(A,B)$	$\frac{s(A \cup B)}{s(A)\times s(B)}$	$[0,\infty]$	No
All Confidence $AllConf(A,B)$	$\frac{s(A \cup B)}{max\{s(A),s(B)\}}$	$[0,1]$	Yes
$Jaccard(A,B)$	$\frac{s(A \cup B)}{s(A)+s(B)-s(A \cup B)}$	$[0,1]$	Yes
$Cosine(A,B)$	$\frac{s(A \cup B)}{\sqrt{s(A)\times s(B)}}$	$[0,1]$	Yes
$Kulczynski(A,B)$	$\frac{1}{2}\left(\frac{s(A \cup B)}{s(A)}+\frac{s(A \cup B)}{s(B)}\right)$	$[0,1]$	Yes
Max Confidence $MaxConf(A,B)$	$max\{\frac{s(A)}{s(A \cup B)},\frac{s(B)}{s(A \cup B)}\}$	$[0,1]$	Yes

χ^2 and lift are not null-invariant

AllConf, Jaccard, consine, Kulczynski, and MaxConf are null-invariant measure

Null Invariance: An important property

- Why is null invariance crucial for the analysis of massive transaction data?
  - Many transactions may contain neither milk nor coffee

	milk	¬milk	Σ_{row}
coffee	mc	¬mc	c
¬coffee	m¬c	¬m¬c	¬c
Σ_{col}	m	¬m	Σ

- Life and χ^2 are not null-invariant: not good to evaluate data that contain too many or too few null transactions
- Many measures are not null-invariant

Data set	mc	$\neg mc$	$m\neg c$	$\neg m\neg c$	χ^2	$Lift$
D_1	10,000	1,000	1,000	100,000	90557	9.26
D_2	10,000	1,000	1,000	100	0	1
D_3	100	1,000	1,000	100,000	670	8.44
D_4	1,000	1,000	1,000	100,000	24740	25.75
D_5	1,000	100	10,000	100,000	8173	9.18
D_6	1,000	10	100,000	100,000	965	1.97

Comparison of Null-Invariant Measures

- Not all null-invariant measures are created equal

	milk	¬milk	Σ_{row}
coffee	mc	¬mc	c
¬coffee	m¬c	¬m¬c	¬c
Σ_{col}	m	¬m	Σ

Null-invariant

Data set	mc	$\neg mc$	$m \neg c$	$\neg m \neg c$	$AllConf$	Jaccard	$Cosine$	$Kulc$	$MaxConf$
D_1	10,000	1,000	1,000	100,000	0.91	0.83	0.91	0.91	0.91
D_2	10,000	1,000	1,000	100	0.91	0.83	0.91	0.91	0.91
D_3	100	1,000	1,000	100,000	0.09	0.05	0.09	0.09	0.09
D_4	1,000	1,000	1,000	100,000	0.5	0.33	0.5	0.5	0.5
D_5	1,000	100	10,000	100,000	0.09	0.09	0.29	0.5	0.91
D_6	1,000	10	100,000	100,000	0.01	0.01	0.10	0.5	0.99

Disagree on these cases

Imbalance Ration with Kulczynski Measure

- Imbalance Ration (IR): measure the imbalance of two itemsets A and B in rule implications

$$IR(A,B) = \frac{|s(A) - s(B)|}{s(A) + s(B) - s(A \cup B)}$$

Data set	mc	$\neg mc$	$m\neg c$	$\neg m \neg c$	Jaccard	Cosine	Kulc	IR
D_1	10,000	1,000	1,000	100,000	0.83	0.91	0.91	0
D_2	10,000	1,000	1,000	100	0.83	0.91	0.91	0
D_3	100	1,000	1,000	100,000	0.05	0.09	0.09	0
D_4	1,000	1,000	1,000	100,000	0.33	0.5	0.5	0
D_5	1,000	100	10,000	100,000	0.09	0.29	0.5	0.89
D_6	1,000	10	100,000	100,000	0.01	0.10	0.5	0.99

- D_4 is neutral & balanced; D_5 is neutral but imbalanced
- D_6 is neutral but very imbalanced

Reference

- Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. Introduction to data mining. Pearson Education India, 2016.

- Lecture Slides from Introduction to Data Mining by Prof. Jiawei Han
- http://hanj.cs.illinois.edu/cs412/bk3_slides/06FPBasic.pdf

- Example of Apriori Algorithm
- https://medium.com/edureka/apriori-algorithm-d7cc648d4f1e