Association Analysis

Association Analysis

- Let's go shopping!

Milk, eggs, sugar, bread

Customer1

Milk, eggs, cereal, bread

Customer2

Eggs, sugar

Customer3

- What do my customer buy? Which product are bought together?
- **Aim:** Find associations and correlations between the different items that customers place in their shopping basket

Example

TID	Items
1	Bread, Peanuts, Milk, Fruit, Jam
2	Bread, Jam, Soda, Chips, Milk, Fruit
3	Steak, Jam, Soda, Chips, Bread
4	Jam, Soda, Peanuts, Milk, Fruit
5	Jam, Soda, Chips, Milk, Bread
6	Fruit, Soda, Chips, Milk
7	Fruit, Soda, Peanuts, Milk
8	Fruit, Peanuts, Cheese, Yogurt

Examples

$\{bread\} \Rightarrow \{milk\}$

$\{soda\} \Rightarrow \{chips\}$

$\{bread\} \Rightarrow \{jam\}$

- 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

What is an Association Rule?

- Implication of the form X ⇒ Y, where X and Y are itemsets
- Example, {bread} ⇒ {milk}
- Rule Evaluation Metrics, Suppor & Confidence
- Support (s)
  - Fraction of transactions that contain both X and Y

$$s = \frac{\sigma(\{\text{Bread}, \text{Milk}\})}{\#\text{ of transactions}} = 0.38$$

- Confidence (c)
  - Measures how often items in Y appear in transactions that contain X

$$c = \frac{\sigma(\{\text{Bread}, \text{Milk}\})}{\sigma(\{\text{Bread}\})} = 0.75$$

Naïve Solution

- 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

- Brute-force approach is computationally prohibitive!

Alternative Solution: 2-step approach

$\{Bread, Jam\} \Rightarrow \{Milk\}$ s=0.4 c=0.75

$\{Milk, Jam\} \Rightarrow \{Bread\}$ s=0.4 c=0.75

$\{Bread\} \Rightarrow \{Milk, Jam\}$ s=0.4 c=0.75

$\{Jam\} \Rightarrow \{Bread, Milk\}$ s=0.4 c=0.6

$\{Milk\} \Rightarrow \{Bread, Jam\}$ s=0.4 c=0.5

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

$\{Milk, Bread, Jam\}$

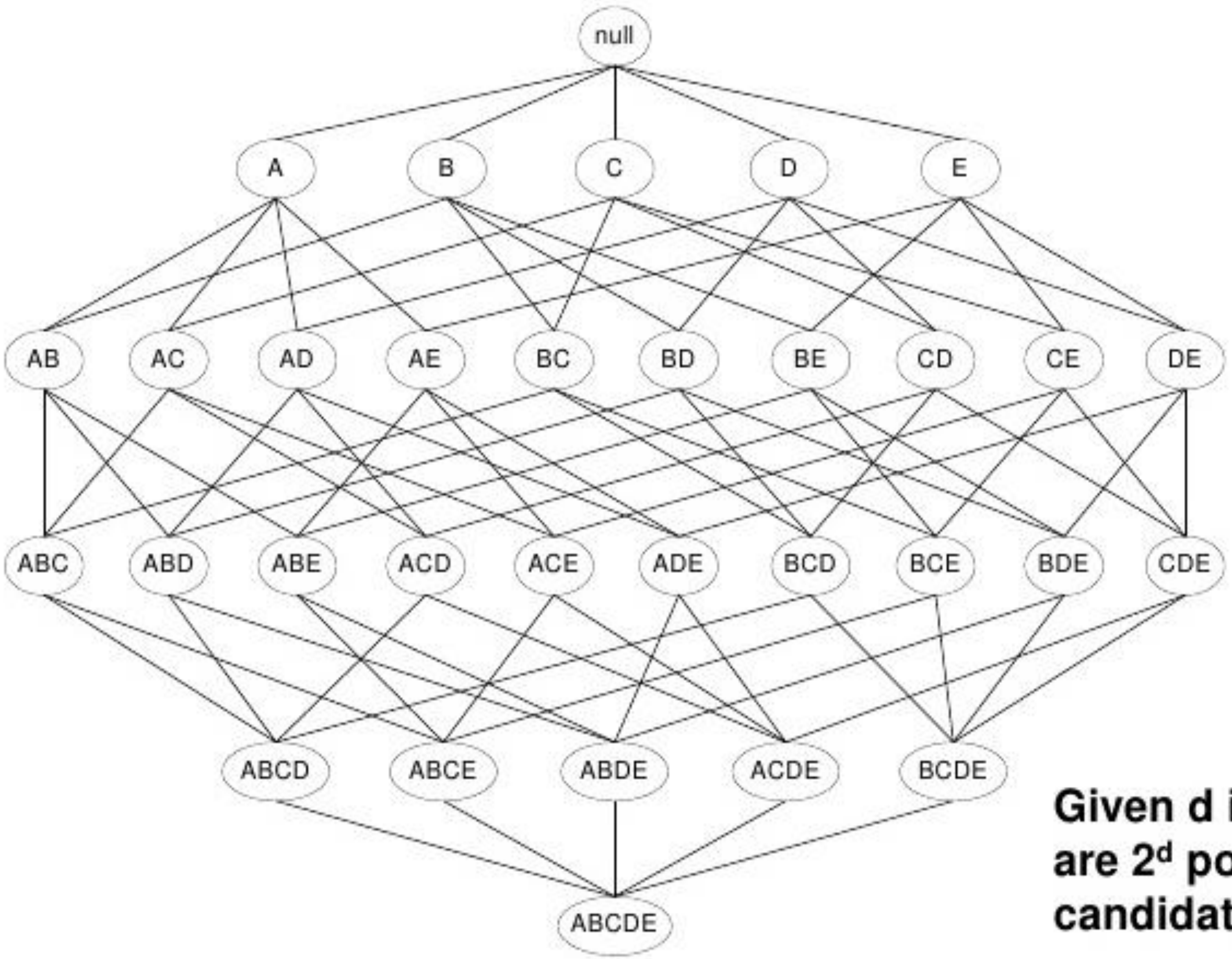
- Rules originating from the same itemset have identical support but can have different confidence
- We can decouple the support and confidence requirements!

Alternative Solution: 2-step approach

- Frequent Itemset Generation
  - Generate all itemsets whose support $\geq$ minsup
- Rule Generation
  - Generate high confidence rules from frequent itemset
  - Each rule is a binary partitioning of a frequent itemset
- Frequent itemset generation is computationally expensive

Frequent Itemset Generation

Frequent Itemset Generation



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

Reducing the number of Frequent Itemsets

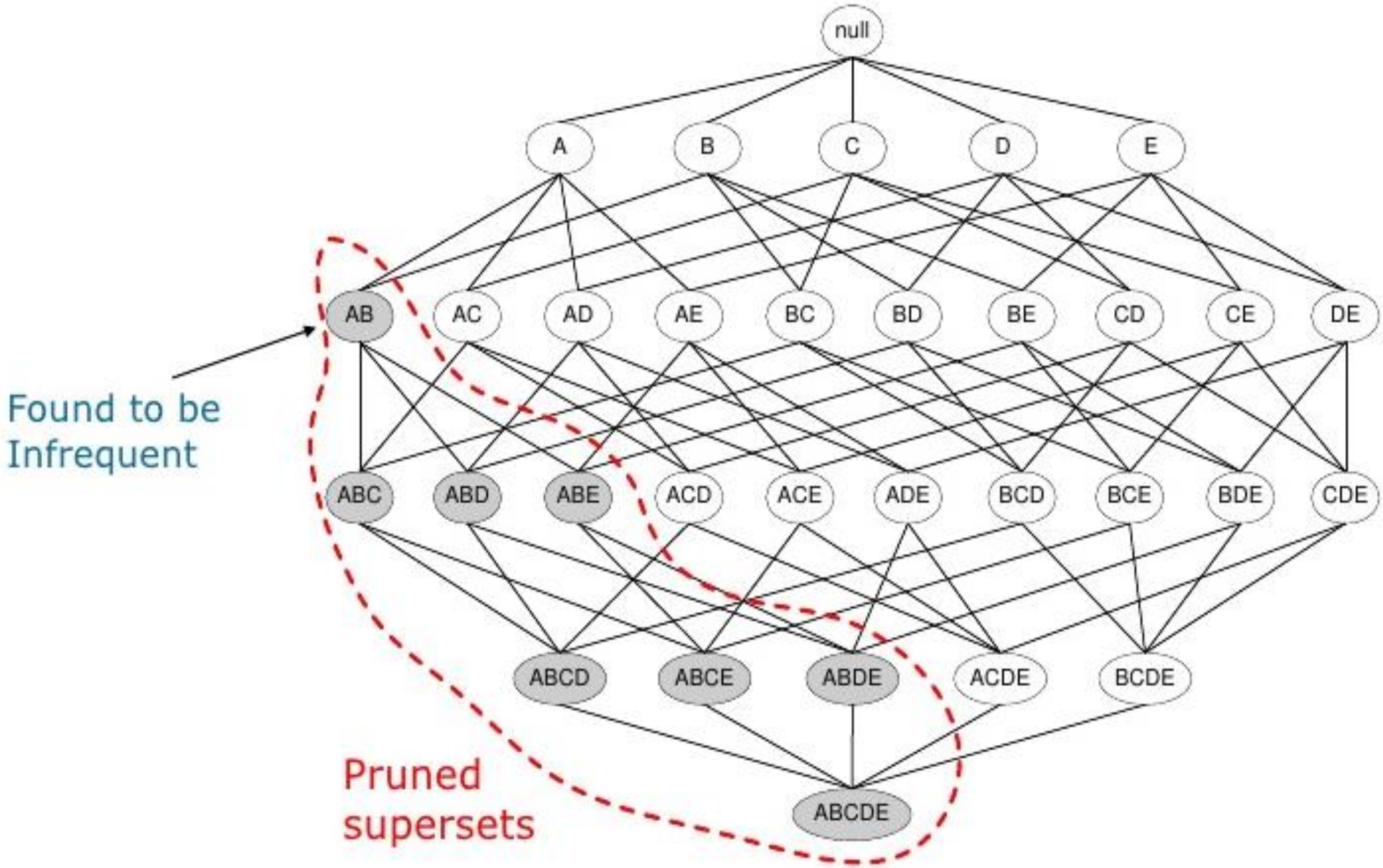
- Apriori principle
  - If an itemset is frequent, then all of its subsets must also be frequent

- 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

Illustrating Apriori Principle



Applying Apriori Principle

Items (1-itemsets)

Item	Count
Bread	4
Peanuts	4
Milk	6
Fruit	6
Jam	5
Soda	6
Chips	4
Steak	1
Cheese	1
Yogurt	1

Minimum Support = 4

2-itemsets

2-Itemset	Count
Bread, Jam	4
Peanuts, Fruit	4
Milk, Fruit	5
Milk, Jam	4
Milk, Soda	5
Fruit, Soda	4
Jam, Soda	4
Soda, Chips	4

3-itemsets

3-Itemset	Count
Milk, Fruit, Soda	4

Apriori Algorithm

- 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
  - Prune candidate itemsets containing subsets of length k that are infrequent
  - Count the support of each candidate by scanning the DB
  - Eliminate candidates that are infrequent, leaving only those that are frequent

Rule Generation

Naïve Approach

- Given a frequent itemset L, find all non-empty subsets $f \subset L$ such that $f \to L - f$ satisfies the minimum confidence requirement
- If {A,B,C,D} is a frequent itemset, candidate rules:

  ABC $\to$D, ABD $\to$C, ACD $\to$B, BCD $\to$A, A $\to$BCD, B $\to$ACD, C $\to$ABD, D $\to$ABC, AB $\to$CD, AC $\to$ BD, AD $\to$ BC, BC $\to$AD, BD $\to$AC, CD $\to$AB

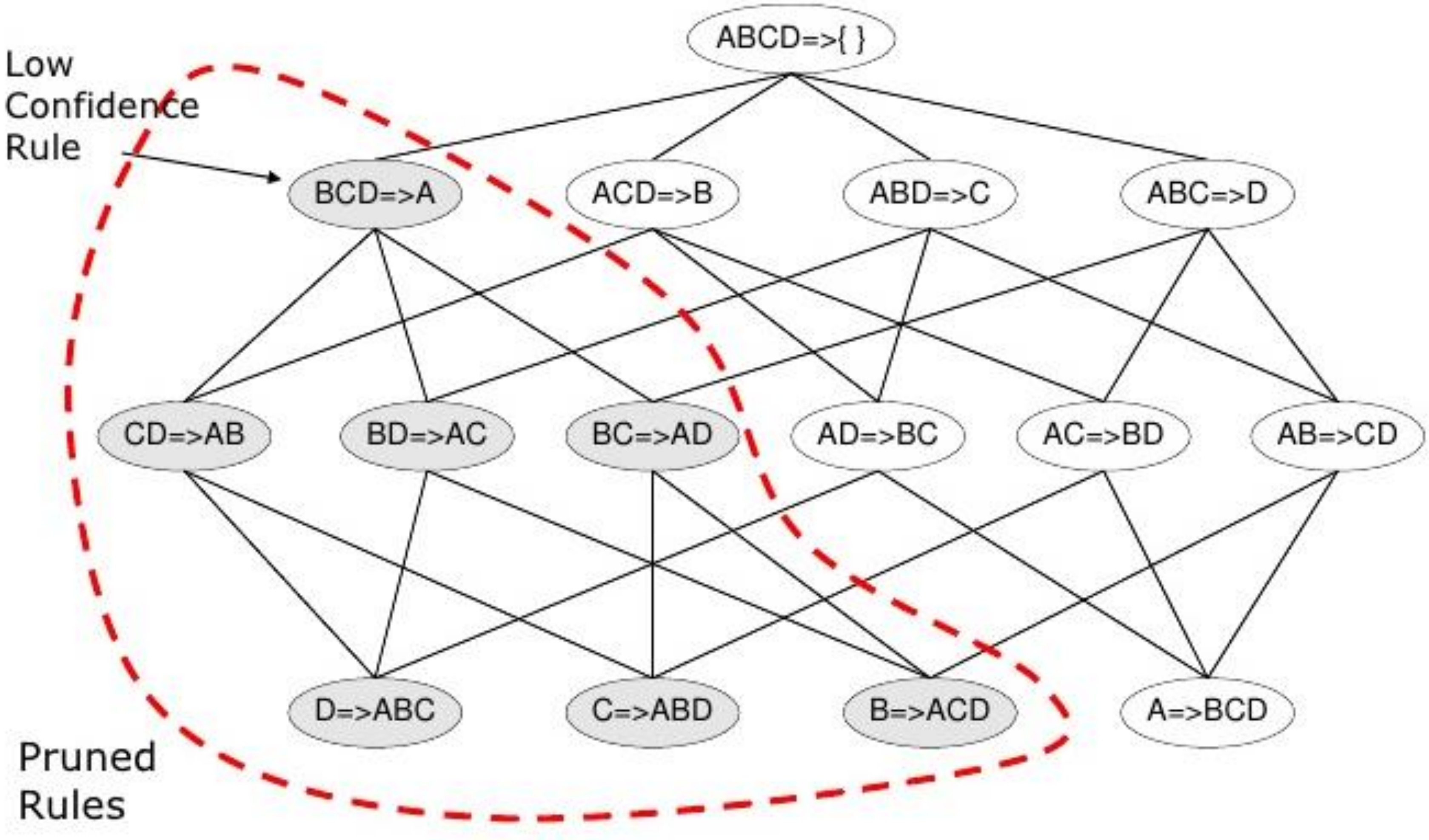
- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \to \emptyset$ and $\emptyset \to L$)

Efficient Approach

- $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$

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

Efficient Approach



Rule Generation

Generating Rules: example

Trans-ID	Items
1	ACD
2	BCE
3	ABCE
4	BE
5	ABCE

Min_support: 60%
Min_confidence: 75%

Frequent Itemset	Support
{BCE},{AC}	60%
{BC},{CE},{A}	60%
{BE},{B},{C},{E}	80%

Rule	Conf.
{BC} =>{E}	100%
{BE} =>{C}	75%
{CE} =>{B}	100%
{B} =>{CE}	75%
{C} =>{BE}	75%
{E} =>{BC}	75%