Session 54

Association Rules

Table of Content

What will We Learn Today?

1. Basic concept
2. Frequent itemset
3. Apriori algorithm

Basic Concept

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, Vegetables, Fruits, Eggs
3	Milk, Vegetables, Fruits, Coke
4	Bread, Milk, Vegetables, Fruits
5	Bread, Milk, Vegetables, Coke

Example of Association Rules

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

Implication means co-occurrence, not causality!

Frequent Itemset

Definition: Frequent Itemset

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

TID	*Items*
1	Bread, Milk
2	Bread, Vegetables, Fruits, Eggs
3	Milk, Vegetables, Fruits, Coke
4	Bread, Milk, Vegetables, Fruits,
5	Bread, Milk, Vegetables, Coke

Definition: Association Rule

- **Association Rule**
  - An implication expression of the form $X \to Y$, where X and Y are itemsets
  - X = antecedent, Y= consequence
  - Antecedent (if) and a consequent (then).
  - An antecedent is an item found within the data.
  - A consequent is an item found in combination with the antecedent.
  - Example:
    {Milk, Vegetables} $\to$ {Fruits}

TID	*Items*
1	Bread, Milk
2	Bread, Vegetables, Fruits, Eggs
3	Milk, Vegetables, Fruits, Coke
4	Bread, Milk, Vegetables, Fruits
5	Bread, Milk, Vegetables, Coke

Example:

$$\{\text{Milk}, \text{Vegetables}\} \Rightarrow \textit{Fruits}$$

Definition: Association Rule

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
  - Indicates the number of times the if-then statements are found true.
- Lift
  - How many times an if-then statement is expected to be found true

$$\text{support}(A \rightarrow C) = \text{support}(A \cup C), \quad \text{range: } [0, 1]$$

$$\text{confidence}(A \rightarrow C) = \frac{\text{support}(A \rightarrow C)}{\text{support}(A)}, \quad \text{range: } [0, 1]$$

$$\text{lift}(A \rightarrow C) = \frac{\text{confidence}(A \rightarrow C)}{\text{support}(C)}, \quad \text{range: } [0, \infty]$$

Definition: Association Rule

TID	*Items*
1	Bread, Milk
2	Bread, Vegetables, Fruits, Eggs
3	Milk, Vegetables, Fruits, Coke
4	Bread, Milk, Vegetables, Fruits
5	Bread, Milk, Vegetables, Coke

Example:

$$\{\text{Milk}, \text{Vegetables}\} \Rightarrow \textit{Fruits}$$

$$\{X\} \Rightarrow Y$$

$$s = \frac{\sigma(\text{Milk}, \text{Vegetables}, \text{Fruits})}{|\mathrm{T}|} = \frac{2}{5} = 0.4$$

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

$$l = \frac{\sigma(\text{Milk}, \text{Vegetables}, \text{Fruits})}{\sigma(\text{Milk}, \text{Vegetables}) * \sigma(\textit{Fruits})} = \frac{2}{(3 * 3)} = 0.22$$

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!

Mining Association Rules

TID	*Items*
1	Bread, Milk
2	Bread, Vegetables, Fruits, Eggs
3	Milk, Vegetables, Fruits, Coke
4	Bread, Milk, Vegetables, Fruits
5	Bread, Milk, Vegetables, Coke

Example of Rules:

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

Observations:

- All the above rules are binary partitions of the same itemset:
  {Milk, Vegetables, Fruits}
- 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

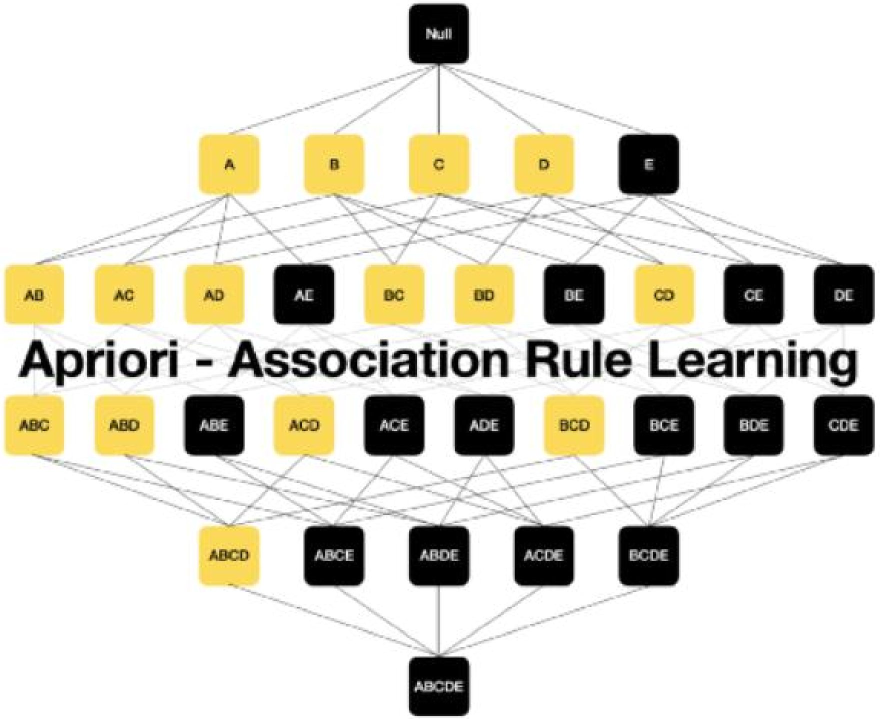
2. Rule Generation
   – Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

- Frequent itemset generation is still computationally expensive

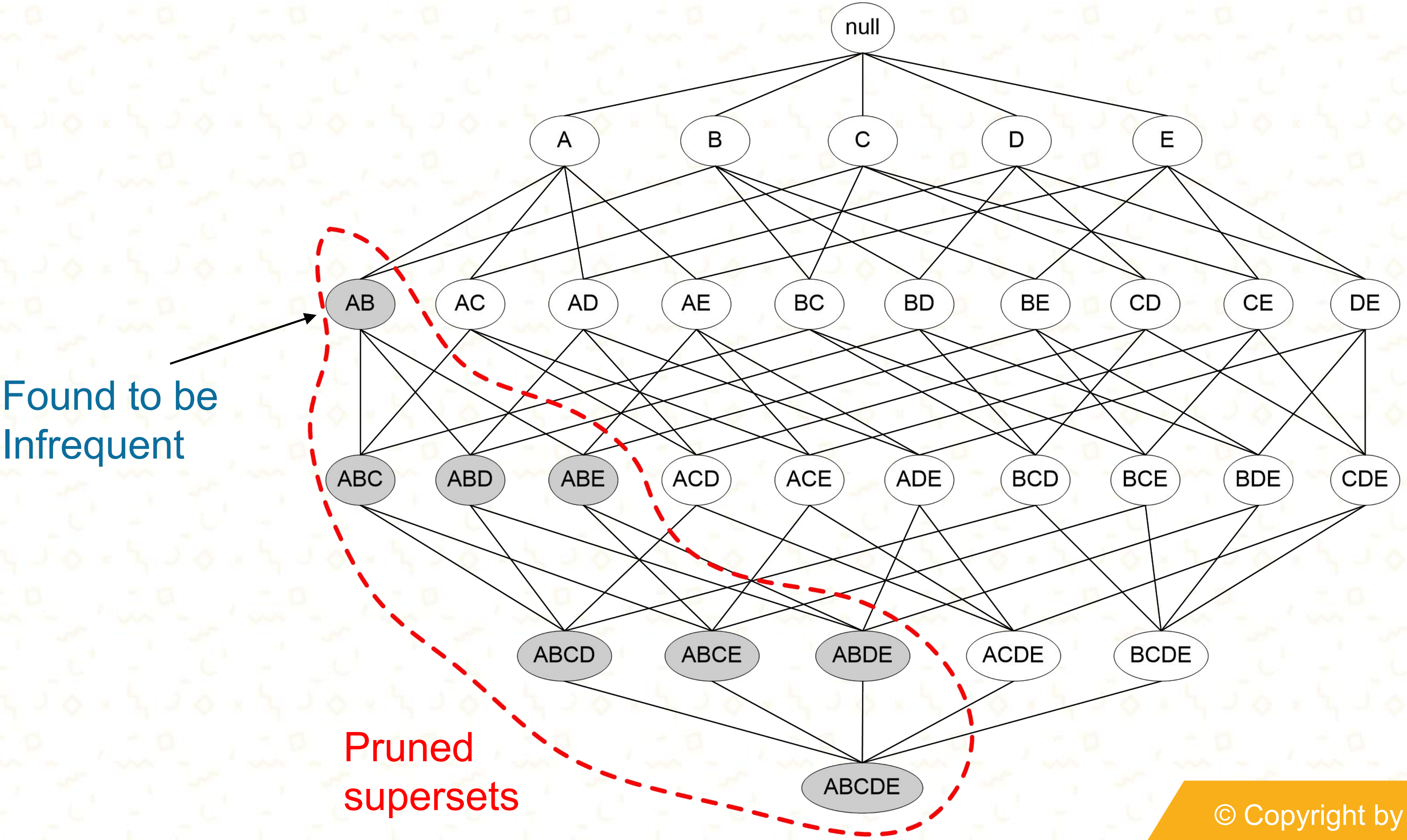
Apriori

Reducing Number of Candidates

- Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data.
- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent
  - Support of an itemset never exceeds the support of its subsets



Illustrating Apriori Principle



Illustrating Apriori Principle

$Sup_{min} = 2$

Database TDB

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

1st 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
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan

C_2

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

L_2

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

C_3

Itemset
{B, C, E}

3rd scan

L_3

Itemset	sup
{B, C, E}	2

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
    - 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

Lets Coding!

DigitalSkola

Uncover The World Of Digital Skills With Us

Thank YOU