ASSOCIATION RULE MINING

BEIYU LIN

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

Market 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

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

REVIEW: SET AND SUBSET

- {a, b, c, d} ⇔ a set (there are one or more than one items)
- Subset ⇔ possible combinations of the items in a set
  - Possible sets:
  - {a}, {b}, {c}, {d}, {a, b}, {a, c}, {a, d}, {b, c}, {b, d}, {c, d},{a, b, c}, {a, b, d}, {a, c, d}, {b, c, d},{a, b, c, d}, {}
  - What is the total number of the subset:

$2^4 = 16$

$2^4 - 1 = 15$

Not Count {}

ASSOCIATION RULE MINING

- **Itemset (set / subset)**
  - 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

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

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

= min_support

DEFINITION: ASSOCIATION RULE

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

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:

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

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

=# of itemset / total # transac

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

= # of itemset of X and Y/ # of

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

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

- 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 expensive / prohibitive!

{a,b,c,d} ➔ # of subset: 2^n

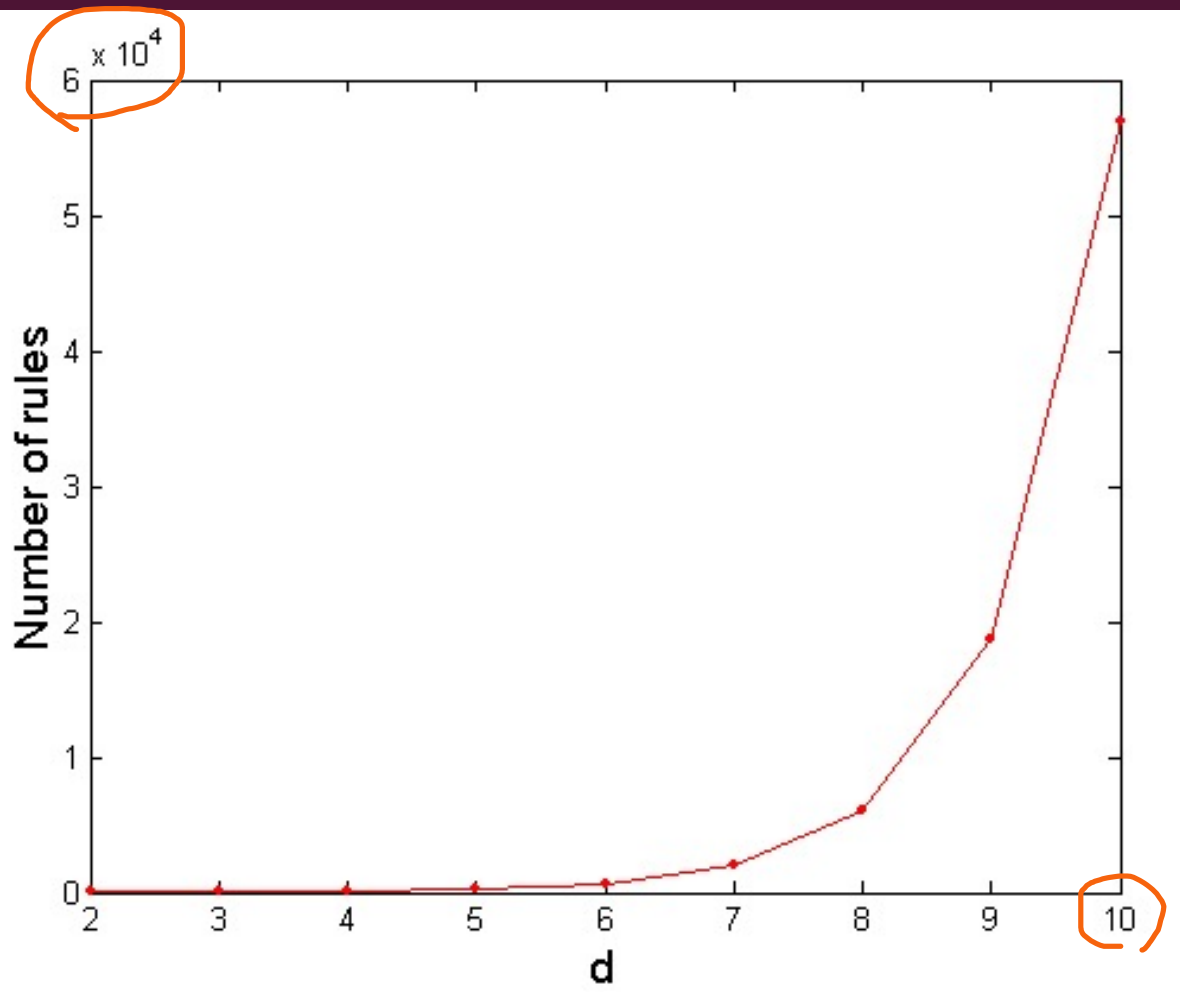
COMPUTATIONAL COMPLEXITY

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

$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

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

If d=6, R = 602 rules



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} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

{a, b, c}
={b, c, a}
={c, a, b}
={c,b,a}

Observations:

X={B}
Y={M,D}

$$S(X \to Y) = \frac{\#\text{of } X \cup Y}{\#\text{ trans}} = \frac{\#\{B, M, D\} = 2}{5} = \frac{2}{5} = 0.4$$

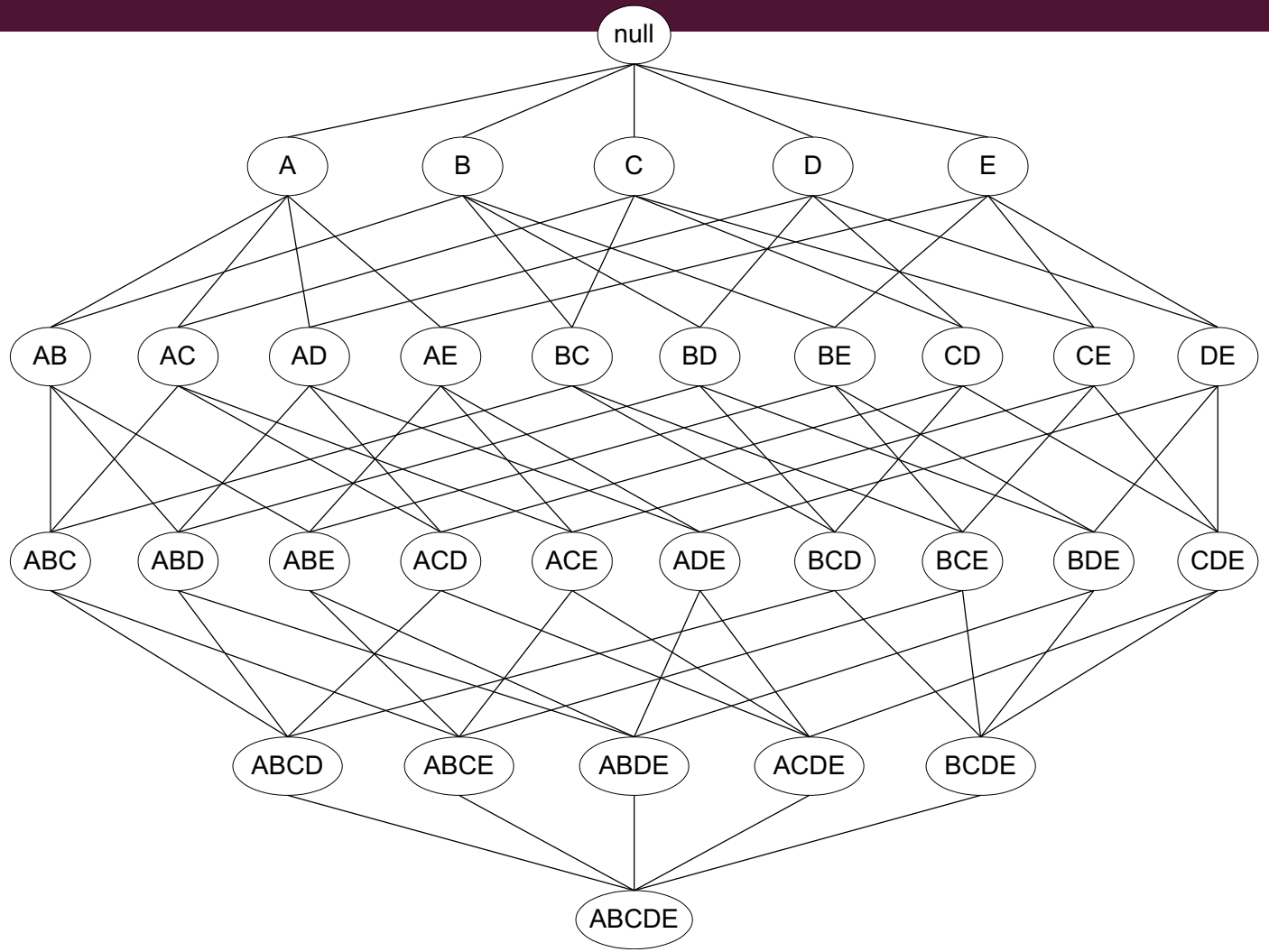
- All the above rules are binary partitions of the same itemset: {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

$$C(X \to Y) = \frac{\#\text{of } X \cup Y}{\#\text{of } X} = \frac{2}{3} = 0.67$$

MINING ASSOCIATION RULES

- **Two-step approach:**
  1. Frequent Itemset Generation
     - Generate all itemsets whose support ≥ 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**

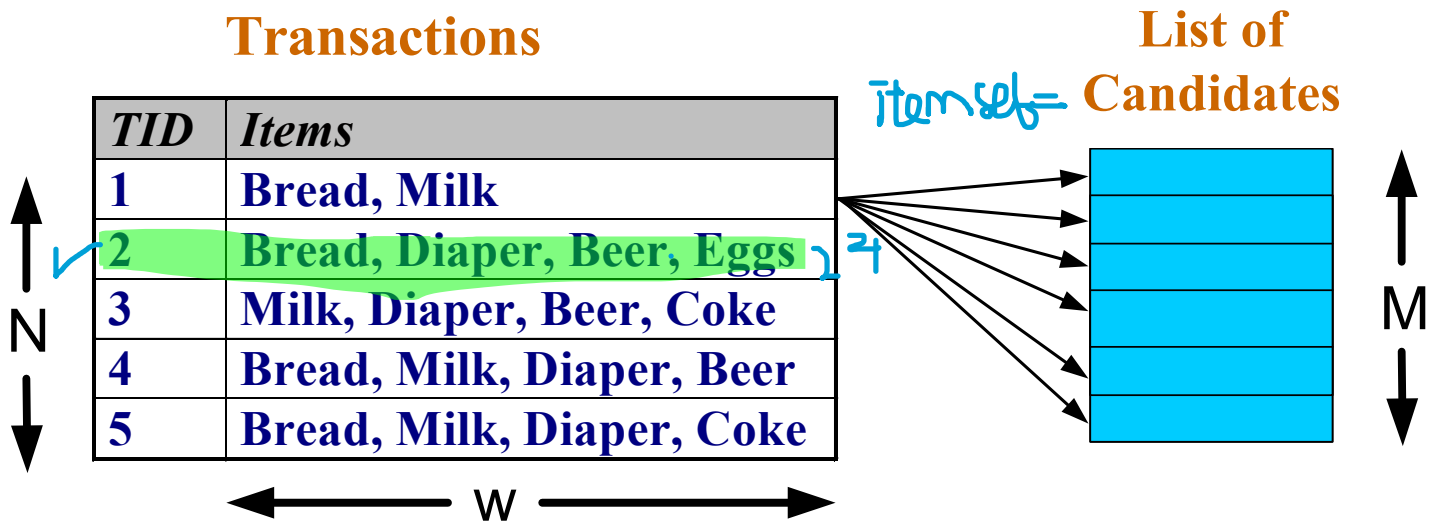
FREQUENT ITEMSET GENERATION



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 ~ 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