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

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

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

$\{a,b,c\}=\{a,c,b\}$
$=\{b,c,a\}$

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 $= \frac{\#\{\}}{5}$

Frequent Itemset

An itemset whose support is greater than or equal to a *minsup* threshold

$\frac{\#\{\}}{5} = minsup = 40\%$ $\quad \sigma_m = 2$

$s \geq minsup$

$\sigma \geq \sigma_m$

DEFINITION: ASSOCIATION RULE

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

- Rule Evaluation Metrics
  - Support (s) $= \frac{\# X \cup Y}{5} = \frac{2}{5}$
    - Fraction of transactions that contain both X and Y
  - Confidence (c) $= \frac{\# X \cup Y}{\# X} = \frac{2}{3}$
    - Measures how often items in Y appear in transactions that contain X

σ

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:

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

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|T|} = \frac{2}{5} = 0.4$$ =# of itemset / total # transac

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$ = # of itemset of X and Y/ # of

FREQUENT ITEMSET GENERATION

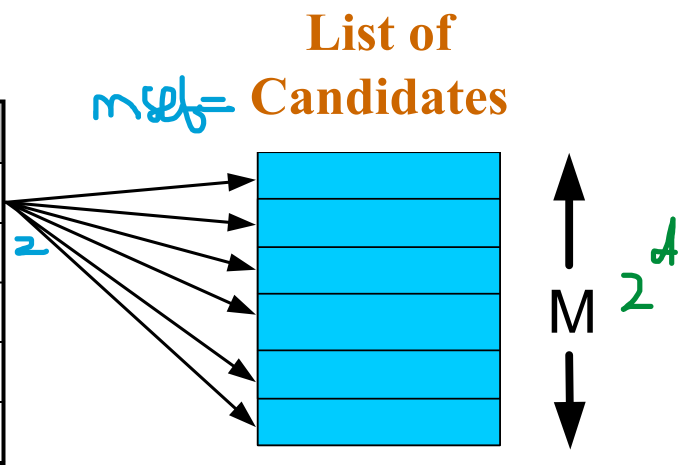
- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database

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

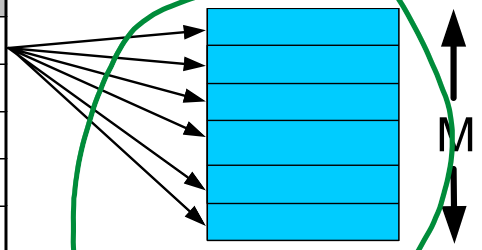
Transactions

N

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

w

List of Candidates



$$C = \frac{\#X \cup Y}{\#X}$$

F; 6

REDUCING NUMBER OF CANDIDATES

A → B ⇔ NOT B ⇒ NOT A

- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent
    If the subset of an itemset is not frequent, then the itemset is not 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

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

$\overbrace{\text{s(\{Milk, Bread, Diaper\})}}^{Y}$ = 2/5

Subset: {milk, bread}

s({milk, break}) = $\underbrace{\text{\#\{milk, bread\}}}_{X}$ /5 $= \frac{3}{5}$

$\frac{3}{5} > \frac{2}{5}$

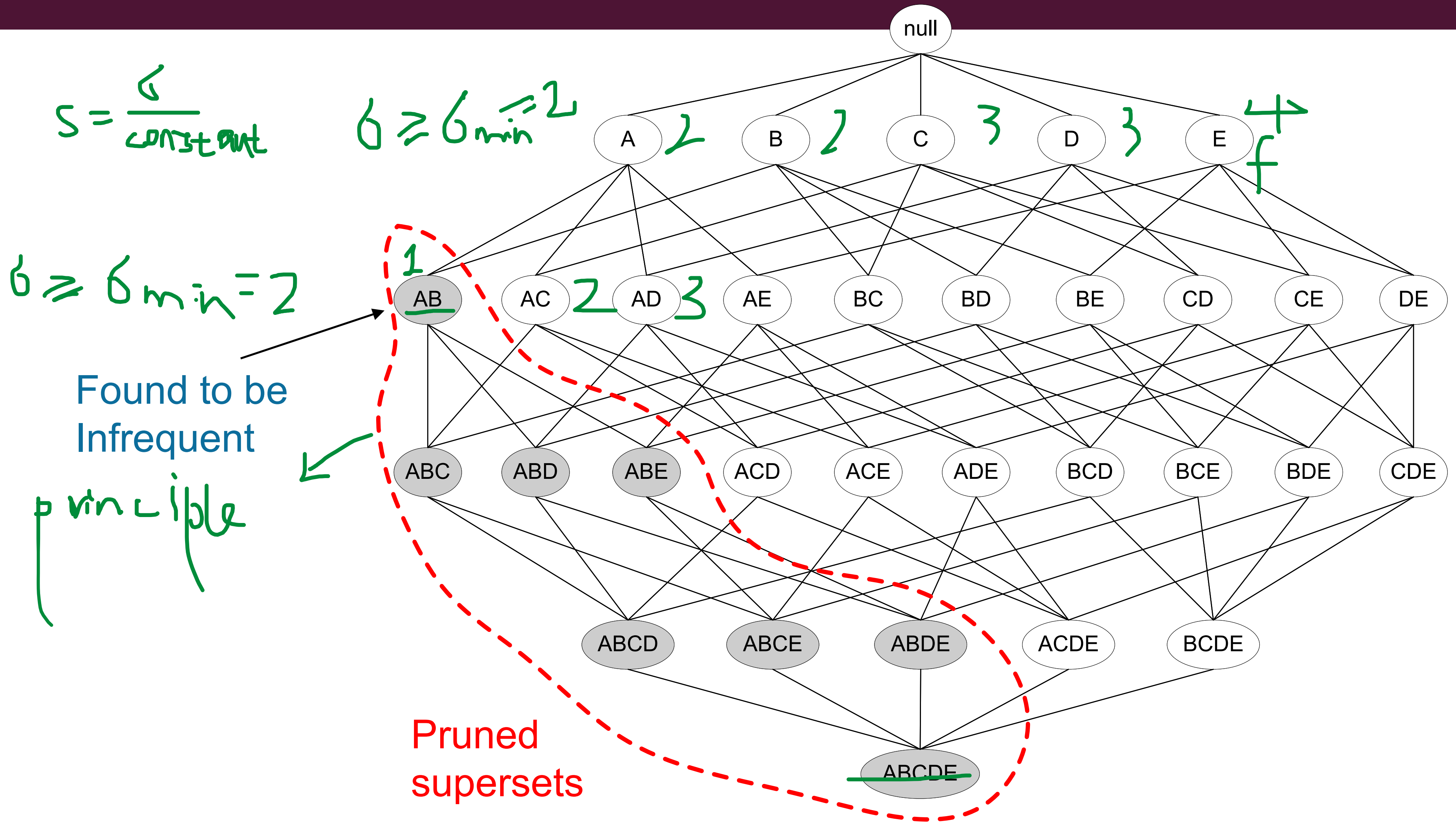
$s(X) \geq s(Y)$

$X \subseteq Y$

ILLUSTRATING APRIORI PRINCIPLE

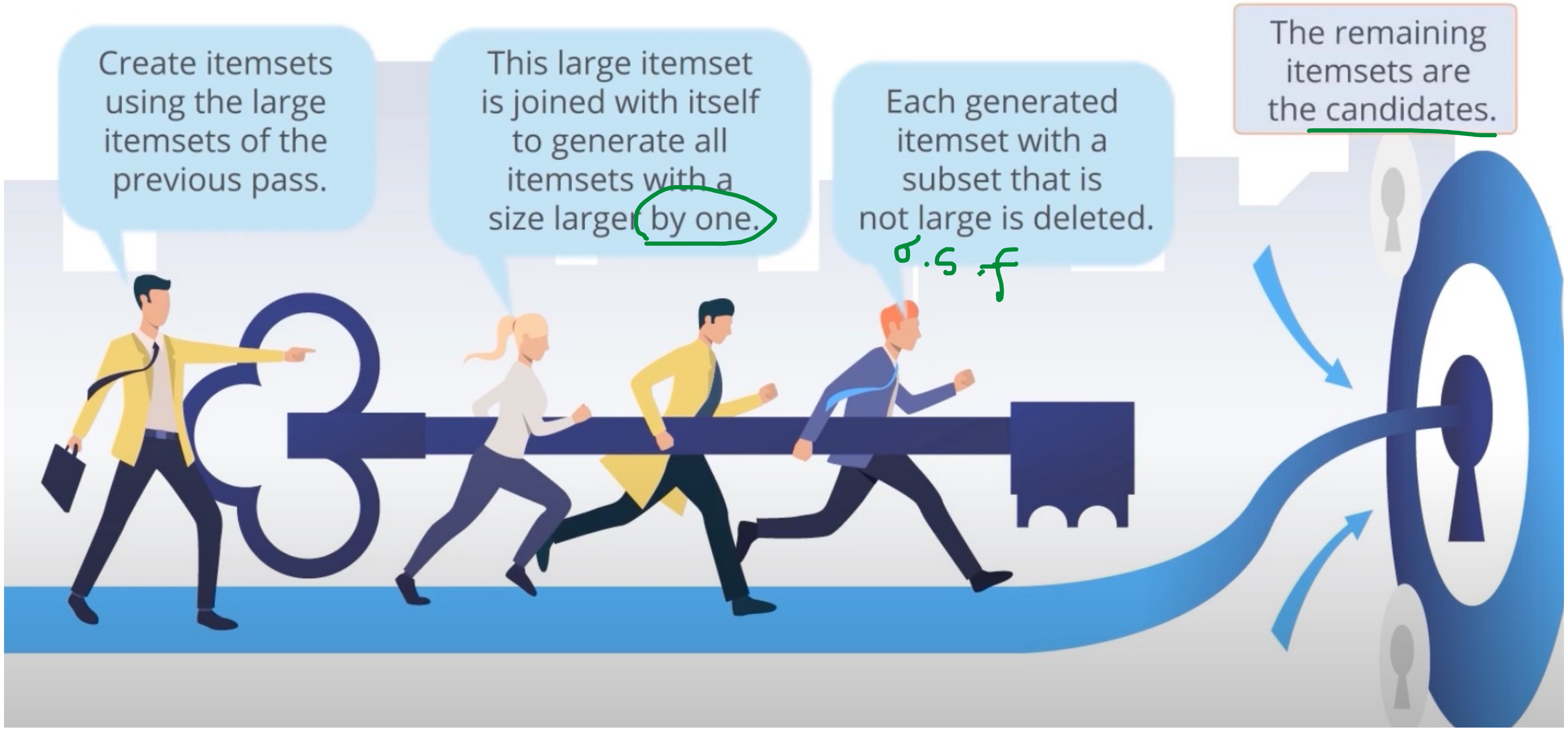
$s = \frac{\sigma}{constant}$ $\sigma \geq \sigma_{min} = 2$

$\sigma \geq \sigma_{min} = 2$



principle

APRIORI PRINCIPLE



APRIORI PRINCIPLE

- Uses frequent itemsets to generate association rules

count $s(A \Rightarrow B) = \frac{\sigma}{(\ \ )}$

- Support ~~value~~ of frequent itemsets is greater than the threshold value

The algorithm reduces the number of candidates being considered by only exploring the itemsets whose support count is greater than the minimum support count.

APRIORI PRINCIPLE EXAMPLE

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5
500	1 3 5

CI1

Itemset	Support
{1}	3 ≥ 2 ✓
{2}	3 ≥ 2 ✓
{3}	4
~~{4}~~	~~1~~
{5}	4

IF

$\sigma \geq \sigma_{min} = 2 \quad (s_{min} = \frac{\ }{5} =$ %$)$

FI1

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5
500	1 3 5

APRIORI PRINCIPLE EXAMPLE

The length of the itemset is extended with 1 (k = k+1).

F11

Itemset	Support
{1}	3
{2}	3
{3}	4
{5}	4

CI2

Itemset	Support
~~{1,2}~~	~~1~~
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

FI2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

$\sigma \geq \sigma_{min} = 2$

APRIORI PRINCIPLE EXAMPLE

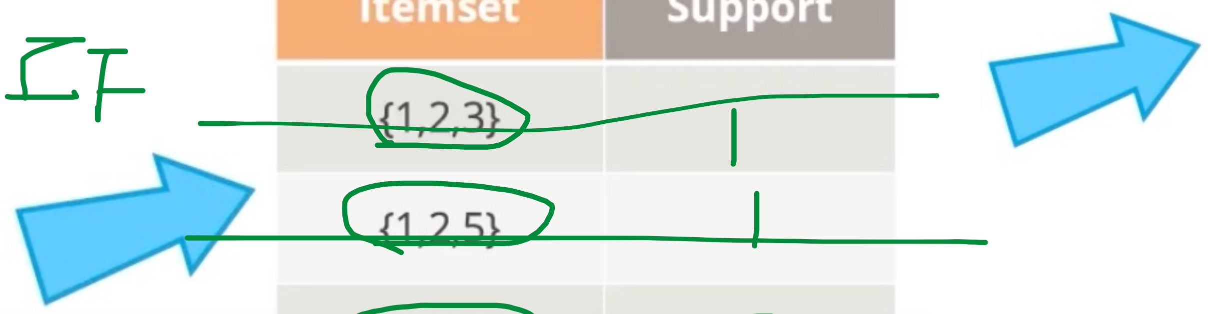
The length of the itemset is extended with 1 (k = k+1).

FI2

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

{1,2,3,5}

IF



CI3

k=3

Itemset	Support
~~{1,2,3}~~	1
~~{1,2,5}~~	1
{1,3,5}	2
{2,3,5}	2

$\sigma \geq \sigma_{min} = 2$

Itemset	Support
{1,3,5}	2
{2,3,5}	2

APRIORI PRINCIPLE EXAMPLE

Divide your itemset to check if there are any other subsets whose support you haven't calculated yet.

CI3

	Itemset	Support
	{1,2,3}	1
	{1,2,5}	1
F	{1,3,5}	2
F	{2,3,5}	2

	Itemset	In FI2?
IF	{1,2}, {1,3}, {2,3}, {1}, {2},{3}	IF
IF	{1,2},{1,5},{2,5}, {1},{2},{5}	IF
	{1,3}, {1,5},{3,5}, {1},{3},{5}	F

Itemset	Support
{1,3}	3
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

APRIORI PRINCIPLE EXAMPLE

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5
500	1 3 5

K=3

Itemset	Support
{1,3,5}	2
{2,3,5}	2

Null

1 2 3 4 5

prune

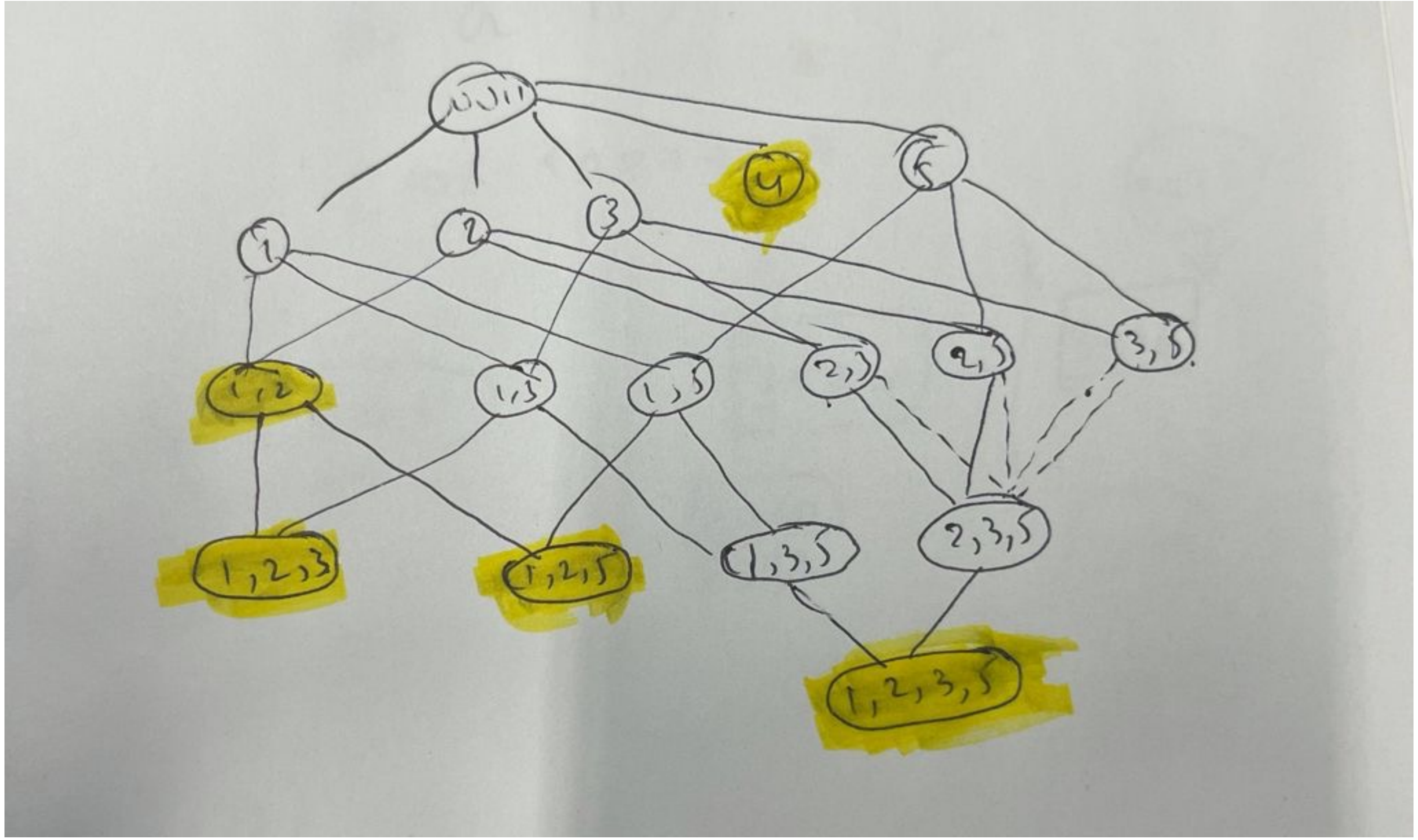
APRIORI PRINCIPLE EXAMPLE

The length of the itemset is extended with 1 (k = k+1).

Itemset	Support
{1,3,5}	2
{2,3,5}	2

C4

Itemset	Support
{1,2,3,5}	1


null
1
2
3
4
5
1,2
1,3
1,5
2,3
2,5
3,5
1,2,3
1,2,5
1,3,5
2,3,5
1,2,3,5

APRIORI ALGORITHM

- F_k: frequent k-itemsets
- L_k: candidate k-itemsets

- Algorithm
  - Let k=1
  - Generate F_1 = {frequent 1-itemsets}
  - Repeat until F_k is empty
    - **Candidate Generation**: Generate L_{k+1} from F_k
    - **Candidate Pruning**: Prune candidate itemsets in L_{k+1} containing subsets of length k that are infrequent
    - **Support Counting**: Count the support of each candidate in L_{k+1} by scanning the DB
    - **Candidate Elimination**: Eliminate candidates in L_{k+1} that are infrequent, leaving only those that are frequent => F_{k+1}