DATA WAREHOUSING AND MINING

T.E. CSE-DS, Sem V

Association Rule Mining: Improving Efficiency Of Apriori Algorithm, Mining Multilevel Association Rules, FP tree, FP growth example

Poonam Pangarkar

Improving Efficiency of Apriori Algorithm

Hash-based technique (hashing itemsets into corresponding buckets):

When scanning each transaction in the database to generate the frequent 1-itemsets, L1, we can generate all the 2-itemsets for each transaction, map them into the different buckets of a hash table structure, and increase the corresponding bucket counts.

Improving Efficiency of Apriori Algorithm

Transaction reduction (reducing the number of transactions scanned in future iterations):

A transaction that does not contain any frequent k-itemsets cannot contain any frequent (k + 1)-itemsets.

Therefore, such a transaction can be marked or removed from further consideration because subsequent database scans will not need to consider such transactions

Improving Efficiency of Apriori Algorithm

Partitioning (partitioning the data to find candidate itemsets): A partitioning technique can be used that requires just two database scans to mine the frequent itemsets.

It consists of two phases.

In phase I, the algorithm divides the transactions of D into n nonoverlapping partitions. If the minimum relative support threshold for transactions in D is minSup, then the minimum support count for a partition is **minSup × the number of transactions in that partition.**

For each partition, all the local frequent itemsets (i.e., the itemsets frequent within the partition) are found.

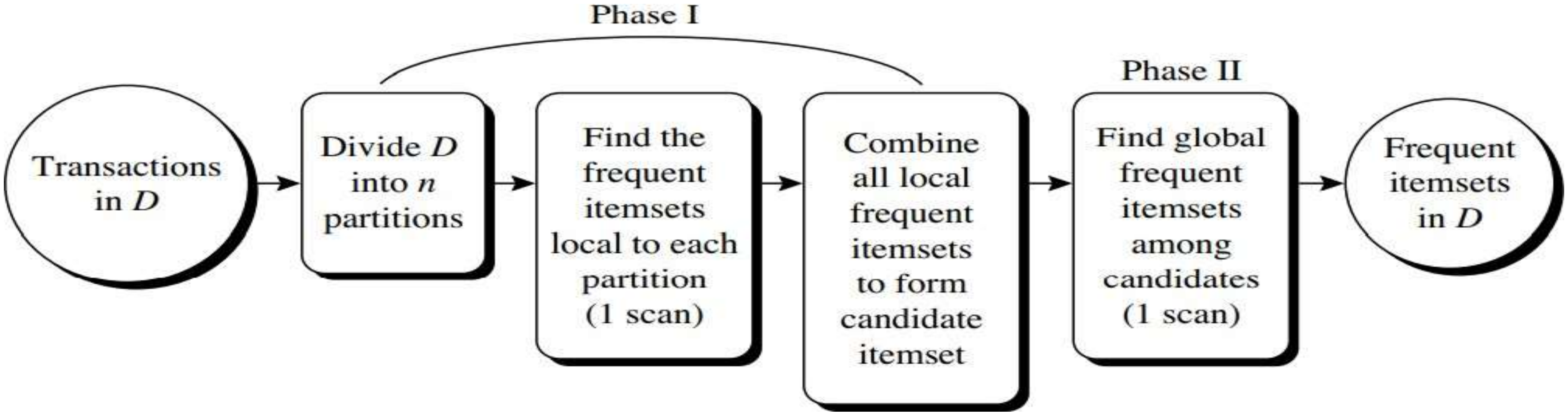
All local frequent itemsets are candidate itemsets with respect to D.

The collection of frequent itemsets from all partitions forms the global candidate itemsets with respect to D.

Improving Efficiency of Apriori Algorithm

In phase II, a second scan of D is conducted in which the actual support of each candidate is assessed to determine the global frequent itemsets.

Partition size and the number of partitions are set so that each partition can fit into main memory and therefore be read only once in each phase.


Phase I
Phase II
Transactions in *D*
Divide *D* into *n* partitions
Find the frequent itemsets local to each partition (1 scan)
Combine all local frequent itemsets to form candidate itemset
Find global frequent itemsets among candidates (1 scan)
Frequent itemsets in *D*

Improving Efficiency of Apriori Algorithm

Sampling (mining on a subset of the given data):

The basic idea of the sampling approach is to pick a random sample S of the given data D, and then search for frequent itemsets in S instead of D. In this way, we trade off some degree of accuracy against efficiency.

The S sample size is such that the search for frequent itemsets in S can be done in main memory, and so only one scan of the transactions in S is required overall. Because we are searching for frequent itemsets in S rather than in D, it is possible that we will miss some of the global frequent itemsets.

Multilevel Association Rules

- Generalized association rule allow rules at different levels.
- Association rules could be generalised for any and all levels of hierarchy.
- A generalized association rule X → Y, is defined like a regular association rule with the restriction that no item in Y may be above any item in X.

Multilevel Association Rules

- Variation of Generalized association rules are multilevel Association rule.
- Itemsets may occur from any level in the hierarchy.
- Concept hierarchy may be traversed in top down fashionand large itemsets are generated.
- When large item sets are generated at level i , Large itemsets are generated at level i+1.

Using uniform minimum support at all levels.

Consider the same minimum support for all levels hierarchy.

If very high support is considered then many low level association rules may be missed.

If very low support is considered then many high level association rules are generated.

Using reduced minimum support at lower level

- Consider separate minimum support at each hierarchy.
- Strategies could be :
- Children of only frequent nodes are checked for minSup

Mining Multidimensional Rules

Single Dimensional Rules : The rules contains only one distinct predicate.

Butter → Milk

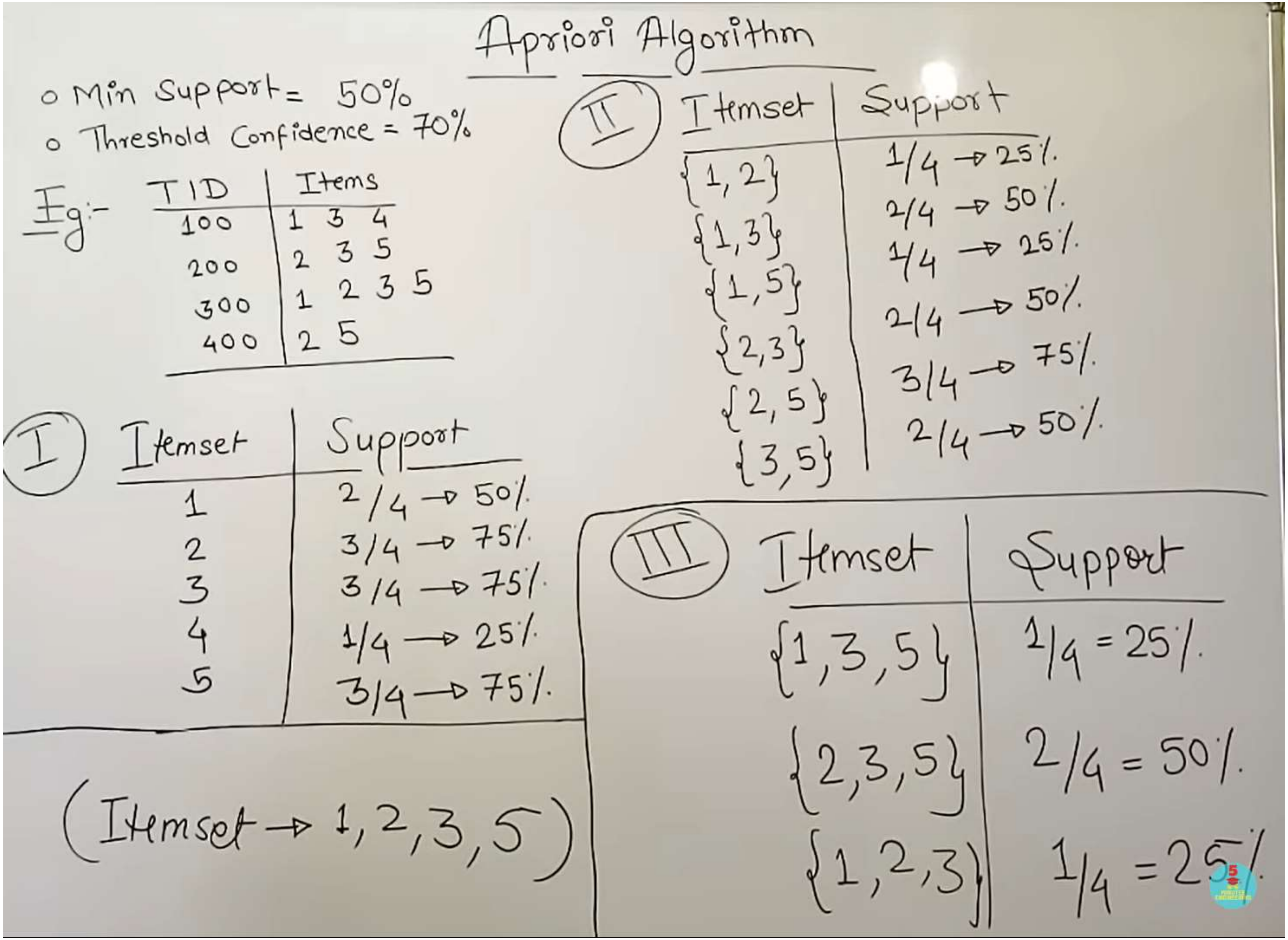
Multidimnesion : The rule contains two or more dimensions

Interdimension Rule : this rule doesn’t have any repeated predicate.

Gender(X, ‘Male’) and Salary(X, High) → buys(X, Computer)

Hybrid Dimension association rules: Many occurrences of the same predicate

Gender(male) and buys(TV) → buys(DVD)



Apriori Algorithm

- Min Support = 50%
- Threshold Confidence = 70%

Eg:-

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

I

Itemset	Support
1	2/4 → 50%
2	3/4 → 75%
3	3/4 → 75%
4	1/4 → 25%
5	3/4 → 75%

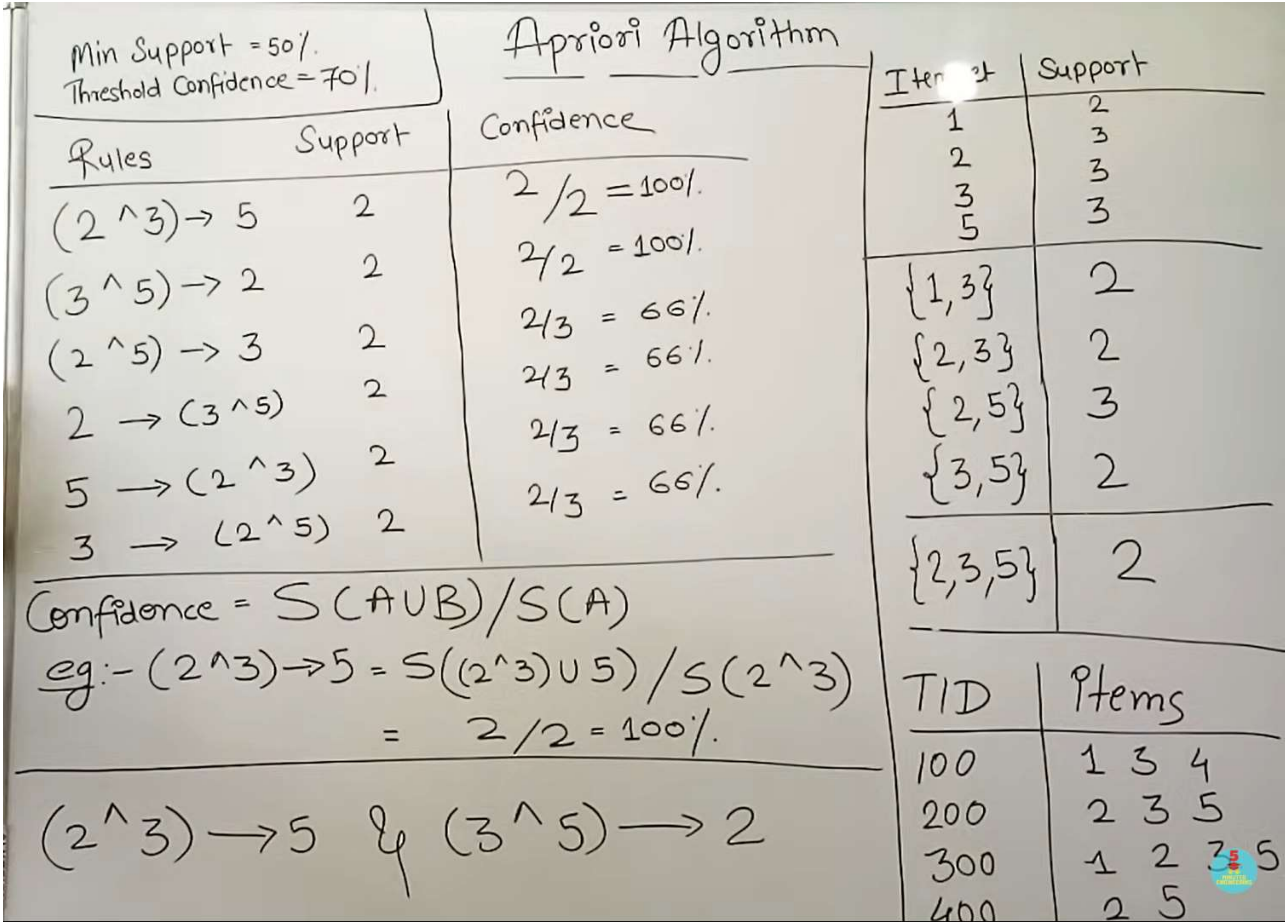
(Itemset → 1, 2, 3, 5)

II

Itemset	Support
{1, 2}	1/4 → 25%
{1, 3}	2/4 → 50%
{1, 5}	1/4 → 25%
{2, 3}	2/4 → 50%
{2, 5}	3/4 → 75%
{3, 5}	2/4 → 50%

III

Itemset	Support
{1, 3, 5}	1/4 = 25%
{2, 3, 5}	2/4 = 50%
{1, 2, 3}	1/4 = 25%



Min Support = 50%
Threshold Confidence = 70%

Apriori Algorithm

Rules	Support	Confidence
$(2 \wedge 3) \rightarrow 5$	2	2/2 = 100%
$(3 \wedge 5) \rightarrow 2$	2	2/2 = 100%
$(2 \wedge 5) \rightarrow 3$	2	2/3 = 66%
$2 \rightarrow (3 \wedge 5)$	2	2/3 = 66%
$5 \rightarrow (2 \wedge 3)$	2	2/3 = 66%
$3 \rightarrow (2 \wedge 5)$	2	2/3 = 66%

$$\text{Confidence} = S(A \cup B)/S(A)$$

eg:- $(2 \wedge 3) \rightarrow 5 = S((2 \wedge 3) \cup 5)/S(2 \wedge 3)$

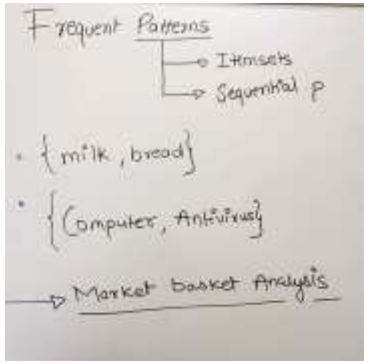
$= 2/2 = 100\%$

$(2 \wedge 3) \longrightarrow 5$ & $(3 \wedge 5) \longrightarrow 2$

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

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

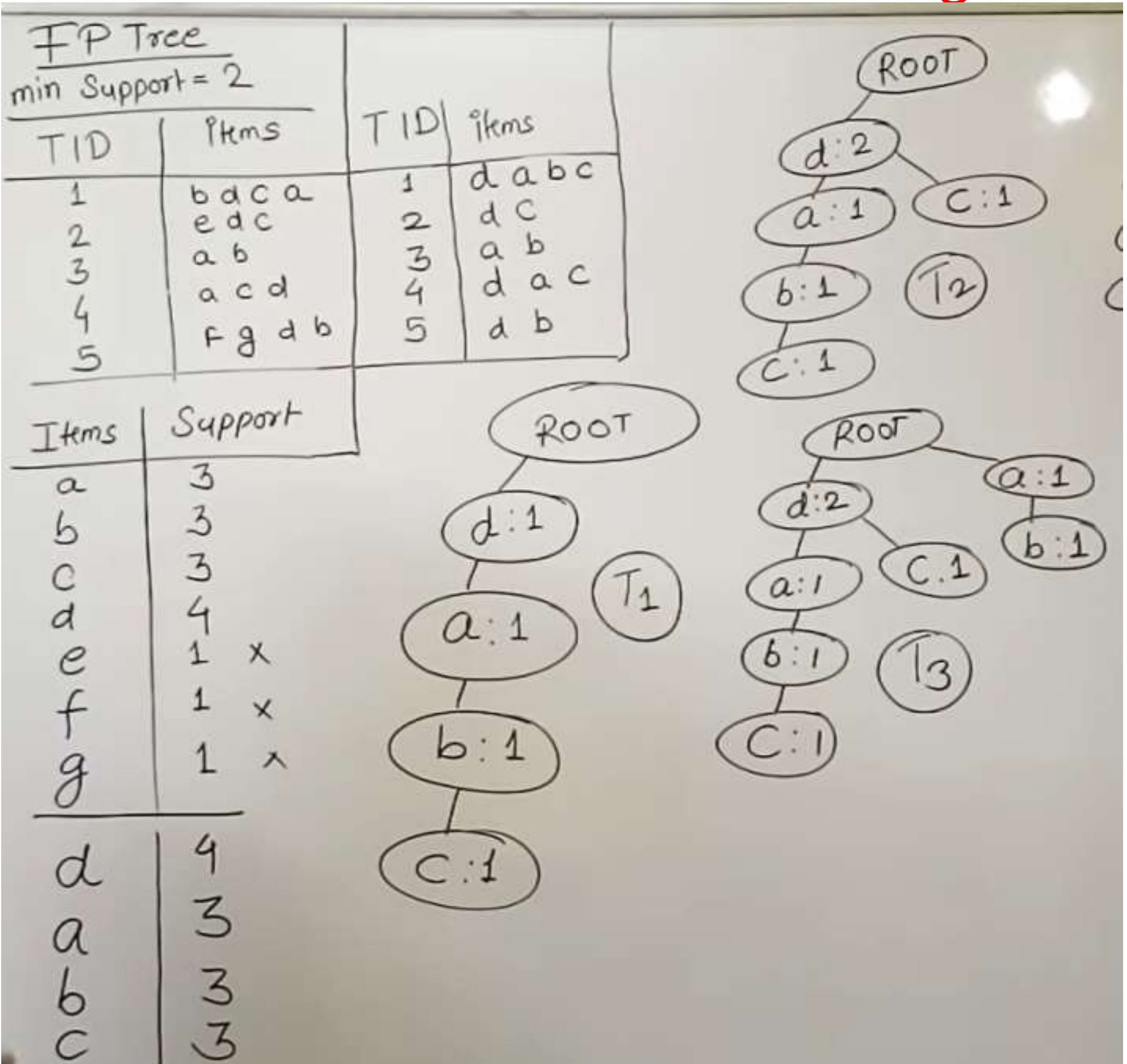
FP tree generation



Frequent Patterns
- Itemsets
- Sequential P

- {milk, bread}
- {Computer, Antivirus}

→ Market basket Analysis



FP Tree

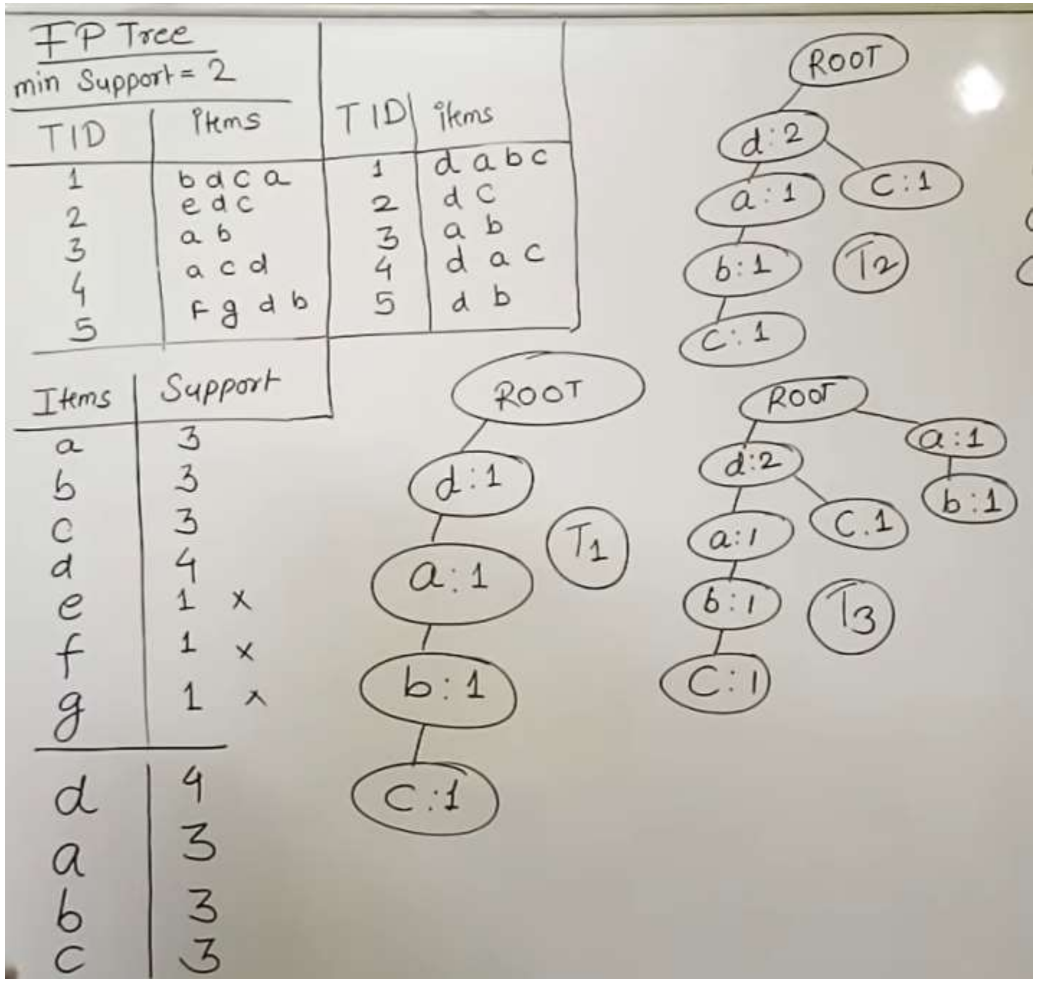
min Support = 2

TID	items
1	b d c a
2	e d c
3	a b
4	a c d
5	f g d b

TID	items
1	d a b c
2	d c
3	a b
4	d a c
5	d b

Items	Support
a	3
b	3
c	3
d	4
e	1 x
f	1 x
g	1 x

d	4
a	3
b	3
c	3



FP Tree

min Support = 2

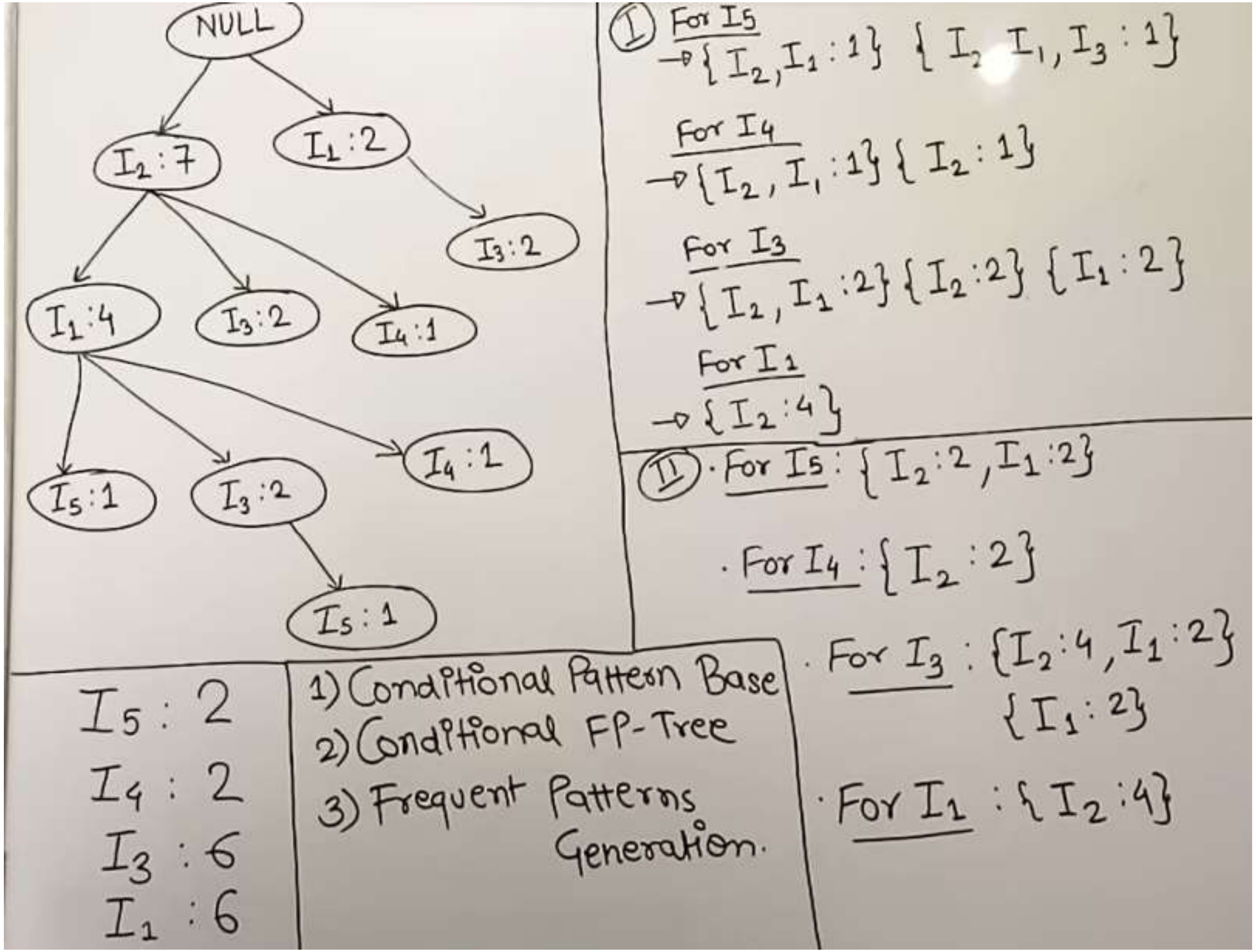
TID	items
1	b d c a
2	e d c
3	a b
4	a c d
5	f g d b

TID	items
1	d a b c
2	d c
3	a b
4	d a c
5	d b

Items	Support
a	3
b	3
c	3
d	4
e	1 x
f	1 x
g	1 x

d	4
a	3
b	3
c	3

FP Growth Example



I For I_5

→ $\{I_2, I_1 : 1\}$ $\{I_2\ I_1, I_3 : 1\}$

For I_4

→ $\{I_2, I_1 : 1\}$ $\{I_2 : 1\}$

For I_3

→ $\{I_2, I_1 : 2\}$ $\{I_2 : 2\}$ $\{I_1 : 2\}$

For I_1

→ $\{I_2 : 4\}$

II · For I_5 : $\{I_2 : 2, I_1 : 2\}$

· For I_4 : $\{I_2 : 2\}$

· For I_3 : $\{I_2 : 4, I_1 : 2\}$ $\{I_1 : 2\}$

· For I_1 : $\{I_2 : 4\}$

I_5	:	2
I_4	:	2
I_3	:	6
I_1	:	6

1) Conditional Pattern Base
2) Conditional FP-Tree
3) Frequent Patterns Generation.

Frequent Patterns Generations

- For I_5

$\{I_2, I_5 : 2\}$ $\{I_1, I_5 : 2\}$

$\{I_2, I_1, I_5 : 2\}$

- For I_4

$\{I_2, I_4 : 2\}$

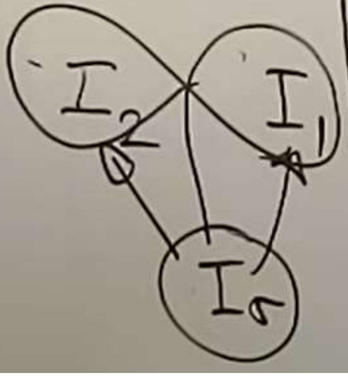
- For I_3

$\{I_2, I_3 : 4\}$ $\{I_1, I_3 : 4\}$

$\{I_2, I_1, I_3 : 2\}$

- For I_1

$\{I_2, I_1 : 4\}$



(I) For I_5

→ $\{I_2, I_1 : 1\}$ $\{I_2, I_1, I_3 : 1\}$

For I_4

→ $\{I_2, I_1 : 1\}$ $\{I_2 : 1\}$

For I_3

→ $\{I_2, I_1 : 2\}$ $\{I_2 : 2\}$ $\{I_1 : 2\}$

MS = 2

For I_1

→ $\{I_2 : 4\}$

(II) · For I_5 : $\{I_2 : 2, I_1 : 2\}$

· For I_4 : $\{I_2 : 2\}$

For I_3 : $\{I_2 : 4, I_1 : 2\}$

$\{I_1 : 2\}$

For I_1 : $\{I_2 : 4\}$