

UNIT 4

ASSOCIATION RULE MINING

AND

APRIORI ALGORITHM

By :
Priyanka Bhardwaj

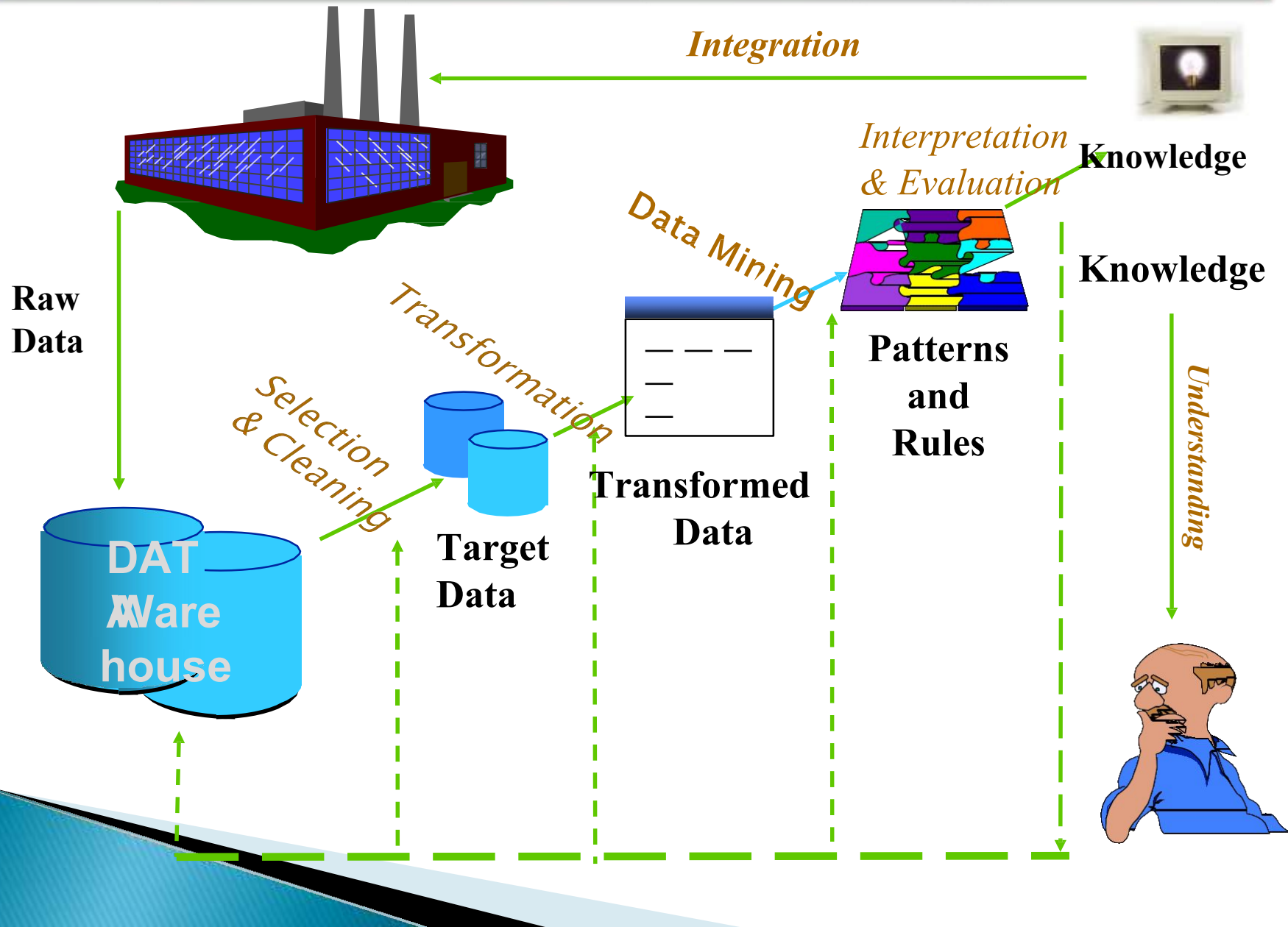
Table of Contents

- Introduction
- Data Mining Process
- Data Mining Techniques
 - Association Rule
- Example
- APRIORI Algorithm

Definition

- Data mining in Data is the non-trivial process of identifying
 - *valid*
 - *novel*
 - potentially *useful*
 - and ultimately *understandable patterns* in data.
- Data Mining extraction of useful pattern from data sources , e.g., databases, texts, web, image.
- Data Mining is also known as Knowledge Discovery in Databases (KDD).

Data Mining Process



Data Mining Process

- Problem formulation
- Data collection
 - subset data: sampling might hurt if highly skewed data
 - feature selection: principal component analysis, heuristic search
- Pre-processing: cleaning
 - name/address cleaning, different meanings (annual, yearly), duplicate removal, supplying missing values
- Transformation:
 - map complex objects e.g. time series data to features e.g. frequency
- Choosing mining task and mining method:
- Result evaluation and Visualization:

Data Mining

- Classification
 - Clustering
 - Regression
 - Association
- Rules

Data Mining

- Classification:
mining patterns that can classify future data into known classes.
- Association rule mining
mining any rule of the form $X \rightarrow Y$, where X and Y are sets of data items.
- Clustering
identifying a set of similarity groups in the data

ASSOCIATION RULE MINING

•

Itemset

- A set of items together is called an itemset.
for eg {bread,butter,milk} is an itemset that contain 3 items
- If any itemset has k-items it is called a k-itemset. An itemset consists of two or more items.
- An itemset that occurs frequently is called a frequent itemset.
- **Thus frequent itemset mining is a data mining technique to identify the items that often occur together.**

What is frequent itemset?

- A set of items is called frequent if it satisfies a minimum threshold value for support and confidence
- For frequent itemset mining method, we consider only those transactions which meet minimum threshold support and confidence requirements.

Frequent Pattern Mining

- The frequent pattern mining algorithm is one of the most important techniques of data mining to discover relationships between different items in a dataset. These relationships are represented in the form of association rules.
- FPM has many applications in the field of data analysis, software bugs, cross-marketing, sale campaign analysis, market basket analysis, etc.
- Association rules apply to supermarket transaction data, that is, to examine the customer behavior

What is market basket analysis?

- Market Basket Analysis is a modelling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items.
- For example, if you are in an English pub and you buy a pint of beer and don't buy a bar meal, you are more likely to buy crisps (US. chips) at the same time than somebody who didn't buy beer.
- Market basket analysis seeks to find relationships between purchases. Typically the relationship will be in the form of a rule:
IF {beer, no bar meal} THEN {crisps}.

Association Rule Mining

- Proposed by **Agrawal et al in 1993**

- *Association Rule Mining is defined as:*

“Let $I = \{\text{bread butter milk}\}$ be a set of ‘n’ binary attributes called items. Let $D = \{ \dots \}$ be set of transaction called database. Each transaction in D has a unique transaction ID and contains a subset of the items in I . A rule is defined as an implication of form $X \rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The set of items X and Y are called antecedent and consequent of the rule respectively.” bread \rightarrow butter

- 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

<i>TID</i>	<i>Items</i>
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

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

Implication means
co-occurrence, not causality!

Example of Association Mining

90% of transactions that purchase bread and butter also purchase milk

“IF” part = **antecedent**

“Item set” = the items (e.g., products) comprising the antecedent or consequent

- Antecedent and consequent are *disjoint* (i.e., have no items in common)

Antecedent: bread and butter

Consequent: milk

Confidence factor: 90%



- **Itemset**

- 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. $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$

- **Support**

- Fraction of transactions that contain an itemset
- E.g. $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$

- **Frequent Itemset**

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

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Association Rule Mining

Support

Every association rule has a support and a confidence.

“The support is the percentage of transactions that demonstrate the rule.”

Example: Database with transactions (customer_# : item_a1, item_a2, ...)

1: 1, 3, 5.

2: 1, 8, 14, 17, 12.

3: 4, 6, 8, 12, 9, 104.

4: 2, 1, 8.

support {8,12} = 2 (,or 50% ~ 2 of 4 customers)

support {1, 5} = 1 (,or 25% ~ 1 of 4 customers)

support {1} = 3 (,or 75% ~ 3 of 4 customers)

Association Rule Mining

Support

An itemset is called **frequent** if its **support** is **equal or greater** than an agreed upon minimal value – **the support threshold**

add to previous example:

if threshold 50%

then itemsets $\{8,12\}$ and $\{1\}$ called **frequent**

Association Rule Mining

Confidence

Every association rule has a support and a confidence.

An association rule is of the form: $X \Rightarrow Y$

- **$X \Rightarrow Y$: if someone buys X, he also buys Y**

The confidence is the **conditional probability** that, given **X** present in a transition, **Y** will also be present.

Confidence measure, by definition:

Confidence($X \Rightarrow Y$) equals $\text{support}(X, Y) / \text{support}(X)$

Association Rule Mining

Confidence

We should only consider **rules** derived from itemsets with **high support**, and that also **have high confidence**.

“A rule with low confidence is not meaningful.”

Rules don't explain anything, they just point out hard facts in data volumes.

Association Rule Mining

- Major steps in association rule mining
 - Frequent itemsets generation
 - Rule derivation
- Use of support and confidence in association mining
 - S for frequent itemsets
 - C for rule derivation

Example

Example: Database with transactions (customer_# : item_a1,
item_a2, ...)

- 1: 3, 5, 8.
- 2: 2, 6, 8.
- 3: 1, 4, 7, 10.
- 4: 3, 8, 10.
- 5: 2, 5, 8.
- 6: 1, 5, 6.
- 7: 4, 5, 6, 8.
- 8: 2, 3, 4.
- 9: 1, 5, 7, 8.
- 10: 3, 8, 9, 10.

Conf ({5} => {8}) ?

$\text{supp}(\{5\}) = 5$, $\text{supp}(\{8\}) = 7$,

$\text{supp}(\{5,8\}) = 4$,

then **conf ({5} => {8}) = $4/5 = 0.8$ or 80%**

Example

Database with transactions (customer_# : item_a1, item_a2, ...)

1: 3, 5, 8.

2: 2, 6, 8.

3: 1, 4, 7, 10.

4: 3, 8, 10.

5: 2, 5, 8.

6: 1, 5, 6.

7: 4, 5, 6, 8.

8: 2, 3, 4.

9: 1, 5, 7, 8.

10 **Conf ({8} \Rightarrow {5})? 80% Done. Conf ({8} \Rightarrow {5})?**

$\text{supp}(\{5\}) = 5$, $\text{supp}(\{8\}) = 7$, $\text{supp}(\{5,8\}) = 4$,

then **conf({8} \Rightarrow {5}) = $4/7 = 0.57$ or 57%**

Example

Database with transactions (customer_# : item_a1, item_a2, ...)

Conf ({5} => {8}) ? 80% Done.

Conf ({8} => {5}) ? 57% Done.

Rule ({5} => {8}) more meaningful then

Rule ({8} => {5})

Example

Database with transactions (customer_# : item_a1, item_a2, ...)

1: 3, 5, 8.

2: 2, 6, 8.

3: 1, 4, 7, 10.

4: 3, 8, 10.

5: 2, 5, 8.

6: 1, 5, 6.

7: 4, 5, 6, 8.

8: 2, 3, 4.

9: 1, 5, 7, 8.

10: 3, 8, 9, 10.

Conf ({9} => {3}) ?

$\text{supp}(\{9\}) = 1$, $\text{supp}(\{3\}) = 1$,

$\text{supp}(\{3,9\}) = 1$,

then $\text{conf}(\{9\} \Rightarrow \{3\}) = 1/1 = 1.0$ or 100%.

OK?

Example

Database with transactions (customer_# : item_a1, item_a2, ...)

Conf({9} \Rightarrow {3}) = 100%. Done.

Notice: High Confidence, Low Support.

-> Rule ({9} \Rightarrow {3}) not meaningful

APRIORI

- **APRIORI** is an efficient algorithm to find association rules (or, actually, **frequent itemsets**).
- This algorithm uses two steps “join” and “prune” to reduce the search space. It is an iterative approach to discover the most frequent itemsets.

APRIORI

Example:

with $k = 3$ (& k -itemsets lexicographically ordered)

$\{3,4,5\}, \{3,4,7\}, \{3,5,6\}, \{3,5,7\}, \{3,5,8\}, \{4,5,6\}, \{4,5,7\}$

generate all possible $(k+1)$ -itemsets, by, for each to sets where we have $\{a_1, a_2, \dots, a_{(k-1)}, X\}$ and $\{a_1, a_2, \dots, a_{(k-1)}, Y\}$, results in candidate $\{a_1, a_2, \dots, a_{(k-1)}, X, Y\}$.

$\{3,4,5,7\}, \{3,5,6,7\}, \{3,5,6,8\}, \{3,5,7,8\}, \{4,5,6,7\}$

Apriori says:

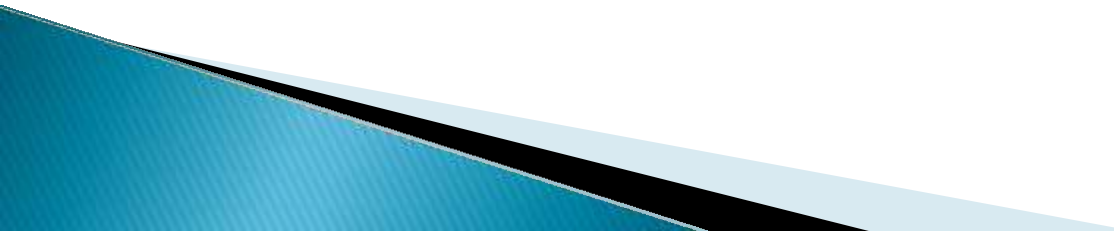
The probability that item I is not frequent is if:

- $P(I) < \text{minimum support threshold}$, then I is not frequent.
- $P(I+A) < \text{minimum support threshold}$, then $I+A$ is not frequent, where A also belongs to itemset.
- If an itemset set has value less than minimum support then all of its supersets will also fall below min support, and thus can be ignored. This property is called the Antimonotone property.

The steps followed in the Apriori Algorithm of data mining are:

Join Step: This step generates $(K+1)$ itemset from K -itemsets by joining each item with itself.

Prune Step: This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.



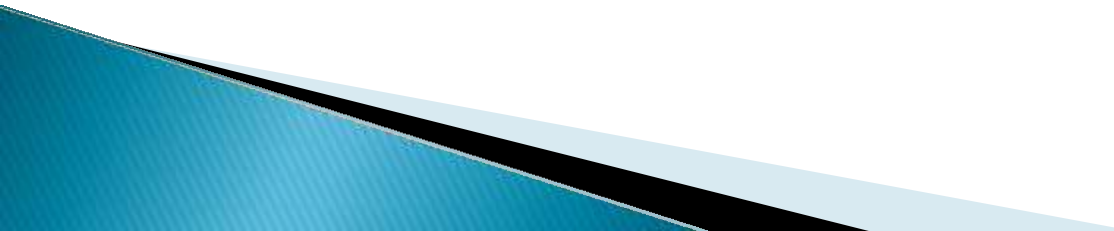
Steps In Apriori

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database. This data mining technique follows the join and the prune steps iteratively until the most frequent itemset is achieved. A minimum support threshold is given in the problem or it is assumed by the user.

#1) In the first iteration of the algorithm, each item is taken as a 1-itemsets candidate. The algorithm will count the occurrences of each item.

#2) Let there be some minimum support, min_sup (eg 2). The set of 1 – itemsets whose occurrence is satisfying the min sup are determined. Only those candidates which count more than or equal to min_sup , are taken ahead for the next iteration and the others are pruned.

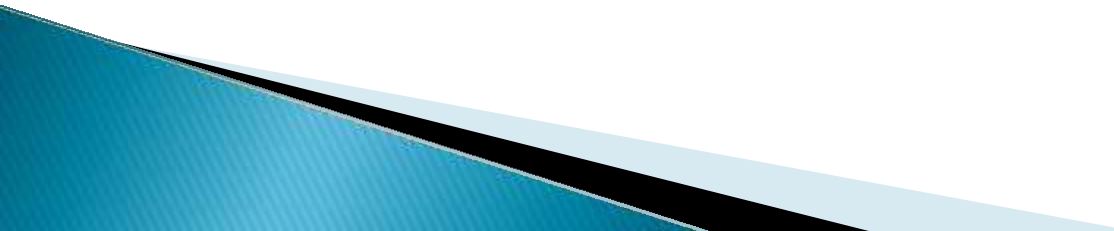
#3) Next, 2-itemset frequent items with min_sup are discovered. For this in the join step, the 2-itemset is generated by forming a group of 2 by combining items with itself.



#4) The 2-itemset candidates are pruned using min-sup threshold value. Now the table will have 2 –itemsets with min-sup only.

#5) The next iteration will form 3 –itemsets using join and prune step. This iteration will follow antimonotone property where the subsets of 3-itemsets, that is the 2 –itemset subsets of each group fall in min_sup. If all 2-itemset subsets are frequent then the superset will be frequent otherwise it is pruned.

#6) Next step will follow making 4-itemset by joining 3-itemset with itself and pruning if its subset does not meet the min_sup criteria. The algorithm is stopped when the most frequent itemset is achieved.



Example of Apriori:

Support threshold=50%, Confidence= 60%

Transaction	List of items
T1	I1,I2,I3
T2	I2,I3,I4
T3	I4,I5
T4	I1,I2,I4
T5	I1,I2,I3,I5
T6	I1,I2,I3,I4

1. Count Of Each Item

Item	Count
I1	4
I2	5
I3	4
I4	4
I5	2

TABLE-2

Support threshold=50% $\Rightarrow 0.5 * 6 = 3 \Rightarrow \text{min_sup}=3$

2. **Prune Step:** **TABLE -2** shows that I5 item does not meet $\text{min_sup}=3$, thus it is deleted, only I1, I2, I3, I4 meet min_sup count.

Item	Count
I1	4
I2	5
I3	4
I4	4

TABLE
-3

3. Join Step: Form 2-itemset. From **TABLE-1** find out the occurrences of 2-itemset.

Item	Count
I1,I2	4
I1,I3	3
I1,I4	2
I2,I3	4
I2,I4	3
I3,I4	2

TABLE
-4

4. Prune Step: TABLE -4 shows that item set {I1, I4} and {I3, I4} does not meet min_sup, thus it is deleted.

Item	Count
I1,I2	4
I1,I3	3
I2,I3	4
I2,I4	3

**TABLE
-5**

5. Join and Prune Step: Form 3-itemset.

From the **TABLE- 1** find out occurrences of 3-itemset.

From **TABLE-5**, find out the 2-itemset subsets which support min_sup.

We can see for itemset {I1, I2, I3} subsets, {I1, I2}, {I1, I3}, {I2, I3}

are occurring in **TABLE-5** thus {I1, I2, I3} is frequent.

We can see for itemset {I1, I2, I4} subsets, {I1, I2}, {I1, I4}, {I2, I4}, {I1, I4} is not frequent, as it is not occurring in **TABLE-5**

thus {I1, I2, I4} is not frequent,

hence it is deleted.

Only {I1, I2, I3} is frequent.

TABLE

5

Item
I1,I2,I3
I1,I2,I4
I2,I3,I4

6. Generate Association Rules: From the frequent itemset discovered above the association could be:

$\{I1, I2\} \Rightarrow \{I3\}$

Confidence = support $\{I1, I2, I3\}$ / support $\{I1, I2\}$ = $(3/4) * 100 = 75\%$

$\{I1, I3\} \Rightarrow \{I2\}$

Confidence = support $\{I1, I2, I3\}$ / support $\{I1, I3\}$ = $(3/3) * 100 = 100\%$

$\{I2, I3\} \Rightarrow \{I1\}$

Confidence = support $\{I1, I2, I3\}$ / support $\{I2, I3\}$ = $(3/4) * 100 = 75\%$

$\{I1\} \Rightarrow \{I2, I3\}$

Confidence = support $\{I1, I2, I3\}$ / support $\{I1\}$ = $(3/4) * 100 = 75\%$

$\{I2\} \Rightarrow \{I1, I3\}$

Confidence = support $\{I1, I2, I3\}$ / support $\{I2\}$ = $(3/5) * 100 = 60\%$

$\{I3\} \Rightarrow \{I1, I2\}$

Confidence = support $\{I1, I2, I3\}$ / support $\{I3\}$ = $(3/4) * 100 = 75\%$

This shows that all the above association rules are strong if minimum confidence threshold is 60%.

The Apriori Algorithm: Pseudo Code

C: Candidate item set of size k

L: Frequent itemset of size k

- Join Step: C_k is generated by joining L_{k-1} with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
- Pseudo-code : C_k : Candidate itemset of size k
 L_k : frequent itemset of size k

$L_1 = \{\text{frequent items}\};$

for ($k = 1; L_k \neq \emptyset; k++$) **do begin**

C_{k+1} = candidates generated from L_k ;

for each transaction t in database **do**

increment the count of all candidates in C_{k+1}
that are contained in t

L_{k+1} = candidates in C_{k+1} with min_support

end

return $\cup_k L_k$;

TID	ITEMS
100	1,3,4
200	2,3,5
300	1,2,3,5
400	2,5

Min SUPPORT=2

MIN CONFIDENCE=60%

APRIORI ALGORITHM EXAMPLE

Database D

Minsup = 0.5

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

Scan D →

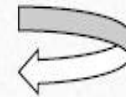
C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

→

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3



L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

Scan D ←

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}



C_3

itemset
{2 3 5}

Scan D →

L_3

itemset	sup
{2 3 5}	2

Original dataset

Transaction ID	Items Bought
T1	{Mango, Onion, Nintendo, Key-chain, Eggs, Yo-yo}
T2	{Doll, Onion, Nintendo, Key-chain, Eggs, Yo-yo}
T3	{Mango, Apple, Key-chain, Eggs}
T4	{Mango, Umbrella, Corn, Key-chain, Yo-yo}
T5	{Corn, Onion, Onion, Key-chain, Ice-cream, Eggs}

Customized dataset

Considering each event with an unique character, we get the database in a short view that given below

Assuming

Mango=M Onion=O Nintendo=N Key-chain=K
Eggs=E Yo-yo=Y Doll=D Apple=A
Umbrella=U Corn=C Ice-cream=I

Transaction ID	Items Bought
T1	{M, O, N, K, E, Y}
T2	{D, O, N, K, E, Y}
T3	{M, A, K, E}
T4	{M, U, C, K, Y}
T5	{C, O, O, K, I, E}

Finding support count

Item	No of transactions
M	3
O	3
N	2
K	5
E	4
Y	3
D	1
A	1
U	1
C	2
I	1

Fig: Result after scanning database first time

Finding I1

Item	Number of transactions
M	3
O	3
K	5
E	4
Y	3

Fig: Result after considering minimum support

Finding c2

Item Pairs	Number of transactions
MO	1
MK	3
ME	2
MY	2
OK	3
OE	3
OY	2
KE	4
KY	3
EY	2

Fig: Result after $L1 * L1$ join step

Finding L2

Item Pairs	Number of transactions
MK	3
OK	3
OE	3
KE	4
KY	3

Fig: Result after pruning step of C2 dataset

Finding C3

Item Set	Number of transactions
OKE	3
KEY	2

Fig: Result after L2*L2 join step

Finding L3

Item Set	Number of transactions
OKE	3

Fig: Result after pruning step of C3 dataset

FP Growth

Frequent-pattern growth(FP-growth):

Finding frequent itemsets without candidate generation.

- First, compress the database representing frequent items into a **frequent-pattern tree, or FP-tree**, which retains the itemset association information.
- Then divide the compressed database into a set of **conditional databases** (a special kind of projected database), each associated with one frequent item or “pattern fragment,” and mines each such database separately.

Frequent-pattern growth(FP-growth): Example

TID	List of item IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

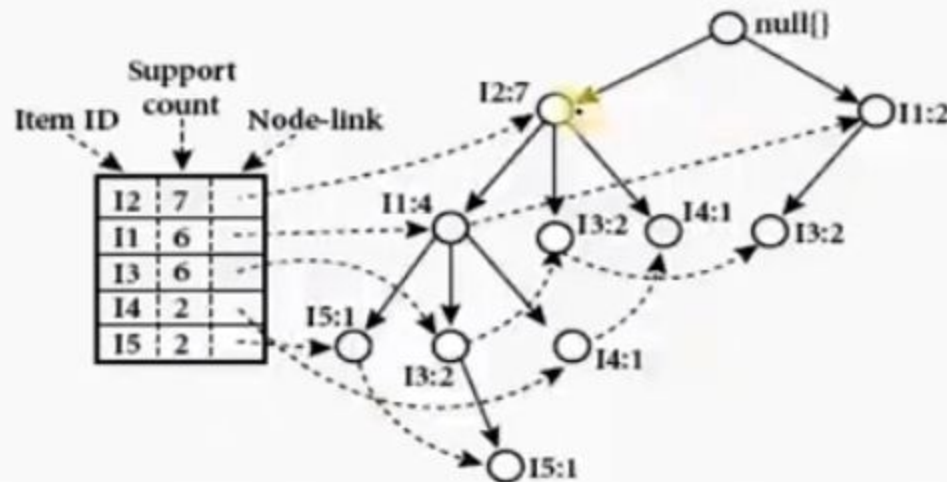
min support =2

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Items	Sup.count
I2	7
I1	6
I3	6
I4	2
I5	2

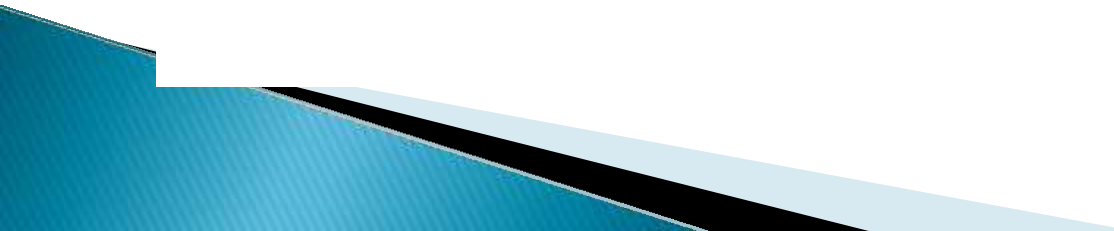
Frequent-pattern growth(FP-growth): Example



Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
15	{ {I2, I1: 1}, {I2, I1, I3: 1} }	$\langle I2: 2, I1: 2 \rangle$	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
14	{ {I2, I1: 1}, {I2: 1} }	$\langle I2: 2 \rangle$	{I2, I4: 2}
13	{ {I2, I1: 2}, {I2: 2}, {I1: 2} }	$\langle I2: 4, I1: 2 \rangle, \langle I1: 2 \rangle$	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
11	{ {I2: 4} }	$\langle I2: 4 \rangle$	{I2, I1: 4}

ECLAT (Equivalence Class Transformation) Vertical Apriori

ECLAT (Equivalence Class Transformation): Vertical Apriori

- Both Apriori and FP-growth use horizontal data format .
 - ECLAT mines frequent itemsets using the Vertical Data Format.
 - It is a depth first search based algorithm.
 - In this, each item is stored together with its T_ID (Transaction ID).
 - It uses intersection based approach to compute the support an itemset.
- 

TID	List of Items
T1	I1, I2, I5
T2	I2, I4
T3	I2, I3
T4	I1, I2, I4
T5	I1, I3
T6	I2, I3
T7	I1, I3
T8	I1, I2, I3, I5
T9	I1, I2, I3

Min Support Count=2,

Confidence =70%

**Generate Association Rule
using ECLAT Algorithm**

Database is in Horizontal Data Format.



TID	List of Items
T1	I1, I2, I5
T2	I2, I4
T3	I2, I3
T4	I1, I2, I4
T5	I1, I3
T6	I2, I3
T7	I1, I3
T8	I1, I2, I3, I5
T9	I1, I2, I3

Step 1: Database is in Vertical Data Format

Itemset	List of Items
I1	T1, T4, T5, T7, T8, T9
I2	T1, T2, T3, T4, T6, T8, T9
I3	T3, T5, T6, T7, T8, T9
I4	T2, T4
I5	T1, T8

Step 2: Itemset generated by intersection of 1 itemset

Itemset	List of Items
I1	T1, T4, T5, T7, T8, T9
I2	T1, T2, T3, T4, T6, T8, T9
I3	T3, T5, T6, T7, T8, T9
I4	T2, T4
I5	T1, T8

Itemset	List of Items
I1, I2	T1,T4,T8,T9
I1, I3	T5,T7, T8, T9
I1, I4	T4
I1, I5	T1,T8
I2, I3	T3, T6, T8, T9
I2, I4	T2, T4
I2, I5	T1, T8
I3, I4	--
I3, I5	T8
I4, I5	--

Itemset	List of Items
I1, I2	T1,T4,T8,T9
I1, I3	T5,T7, T8, T9
I1, I4	T4
I1, I5	T1,T8
I2, I3	T3, T6, T8, T9
I2, I4	T2, T4
I2, I5	T1, T8
I3, I4	--
I3, I5	T8
I4, I5	--

Itemset	List of Items
I1, I2	T1,T4,T8,T9
I1, I3	T5,T7, T8, T9
I1, I5	T1,T8
I2, I3	T3, T6, T8, T9
I2, I4	T2, T4
I2, I5	T1, T8

Min. Support=2

Step 3: Itemset generated by intersection of 2 itemset

Itemset	List of Items
I1, I2	T1,T4,T8,T9
I1, I3	T5,T7, T8, T9
I1, I5	T1,T8
I2, I3	T3, T6, T8, T9
I2, I4	T2, T4
I2, I5	T1, T8

Itemset	List of Items
I1, I2,I3	T8, T9
I1, I2, I5	T1, T8
I1, I3, I5	T8
I2, I3, I4	--
I2, I3, I5	T8
I2, I4, I5	--

Itemset	List of Items
I1, I2,I3	T8, T9
I1, I2, I5	T1, T8

Step 3: Itemset generated by intersection of 3 itemset

Itemset	List of Items
I1, I2, I3	T8, T9
I1, I2, I5	T1, T8

Itemset	List of Items
I1, I2, I3, I5	T8

Table with four itemset is null

Rules can be formed from following three itemset

Itemset	List of Items
I1, I2, I3	T8, T9
I1, I2, I5	T1, T8

We can expand any rule.
For e.g. I1, I2 and I5

Confidence=70%

Association Rule	Confidence	Confidence (%)
$I1 \wedge I2 \rightarrow I5$	$C(I1, I2, I5) / C(I1, I2) = 2/4$	50%
$I1 \wedge I5 \rightarrow I2$	$C(I1, I2, I5) / C(I1, I5) = 2/2$	100%
$I2 \wedge I5 \rightarrow I1$	$C(I1, I2, I5) / C(I2, I5) = 2/2$	100%
$I1 \rightarrow I2 \wedge I5$	$C(I1, I2, I5) / C(I1) = 2/6$	33%
$I2 \rightarrow I1 \wedge I5$	$C(I1, I2, I5) / C(I2) = 2/7$	29%
$I5 \rightarrow I1 \wedge I2$	$C(I1, I2, I5) / C(I5) = 2/2$	100%

TID	List of Items
T1	K, A, D, B
T2	D, A, C, E, B
T3	C, A, B, E
T4	B, A, D

Min Support Count=60%,

Confidence =80%

**Generate Association Rule
using ECLAT Algorithm**

Database is in Horizontal Data Format.

Itemset	List of Items
A	T1, T2, T3, T4
B	T1, T2, T3, T4
C	T2, T3
D	T1, T2, T4
E	T2, T3
K	T1

Itemset	List of Items
A	T1, T2, T3, T4
B	T1, T2, T3, T4
D	T1, T2, T4

Min Support Count=60%

i.e. $(60 \times 4) / 100 = 2.4 = 3$

Step 2: Itemset generated by intersection of 1 itemset

Itemset	List of Items
A	T1, T2, T3, T4
B	T1, T2, T3, T4
D	T1, T2, T4

Itemset	List of Items
A, B	T1, T2, T3, T4
A, D	T1, T2, T4
B, D	T1, T2, T4

Step 3: Itemset generated by intersection of 2 itemset

Itemset	List of Items
A, B, D	T1, T2, T4

Expand any rule.
For e.g. A, B, D

Confidence = 80%

Association Rule	Confidence	Confidence (%)
$A \wedge B \rightarrow C$	$C(A, B, C) / C(A, B) = 3/4$	75%
$A \wedge C \rightarrow B$	$C(A, B, C) / C(A, C) = 3/3$	100%
$B \wedge C \rightarrow A$	$C(A, B, C) / C(B, C) = 3/3$	100%
$A \rightarrow B \wedge C$	$C(A, B, C) / C(A) = 3/4$	75%
$B \rightarrow A \wedge C$	$C(A, B, C) / C(B) = 3/4$	75%
$C \rightarrow A \wedge B$	$C(A, B, C) / C(C) = 3/3$	100%

Advantages of ECLAT:

- It uses less memory than Apriori as its concept is based on depth first search.
- It requires less time for frequent pattern generation than Apriori. Because there is no repeated scanning of the data to compute individual support values.

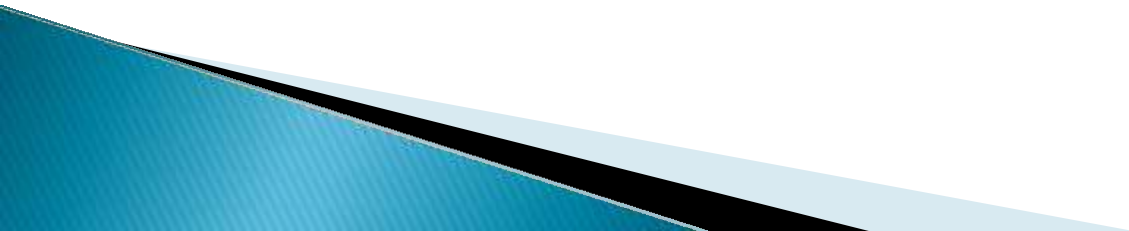
Disadvantages of ECLAT:

- Suitable for small dataset.
- When T_ID list is large, then it takes more time to store candidate set. Also it requires more time for intersection.

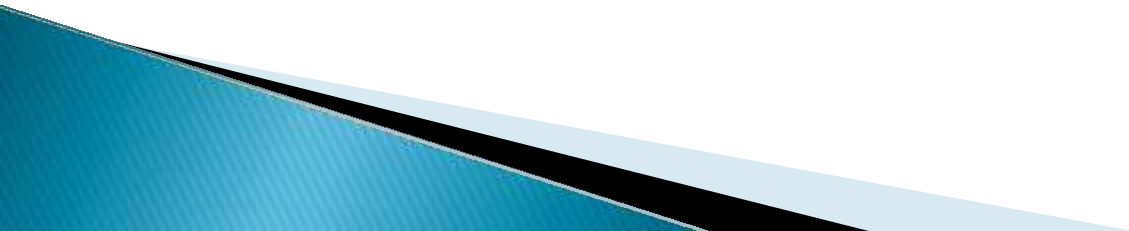


MINING VARIOUS KINDS OF ASSOCIATION RULES

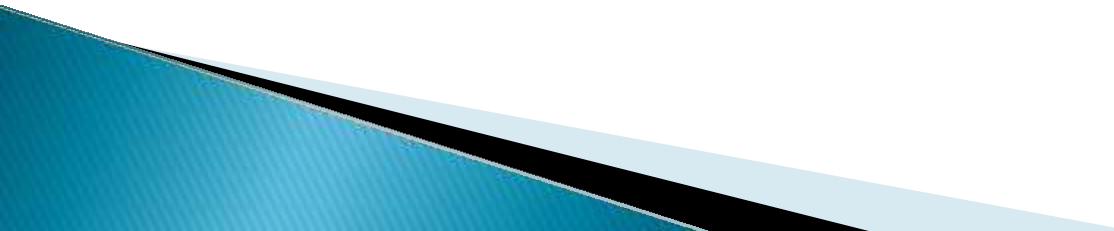
- MULTILEVEL ASSOCIATION RULES MINING
- MULTIDIMENSIONAL ASSOCIATION RULES MINING



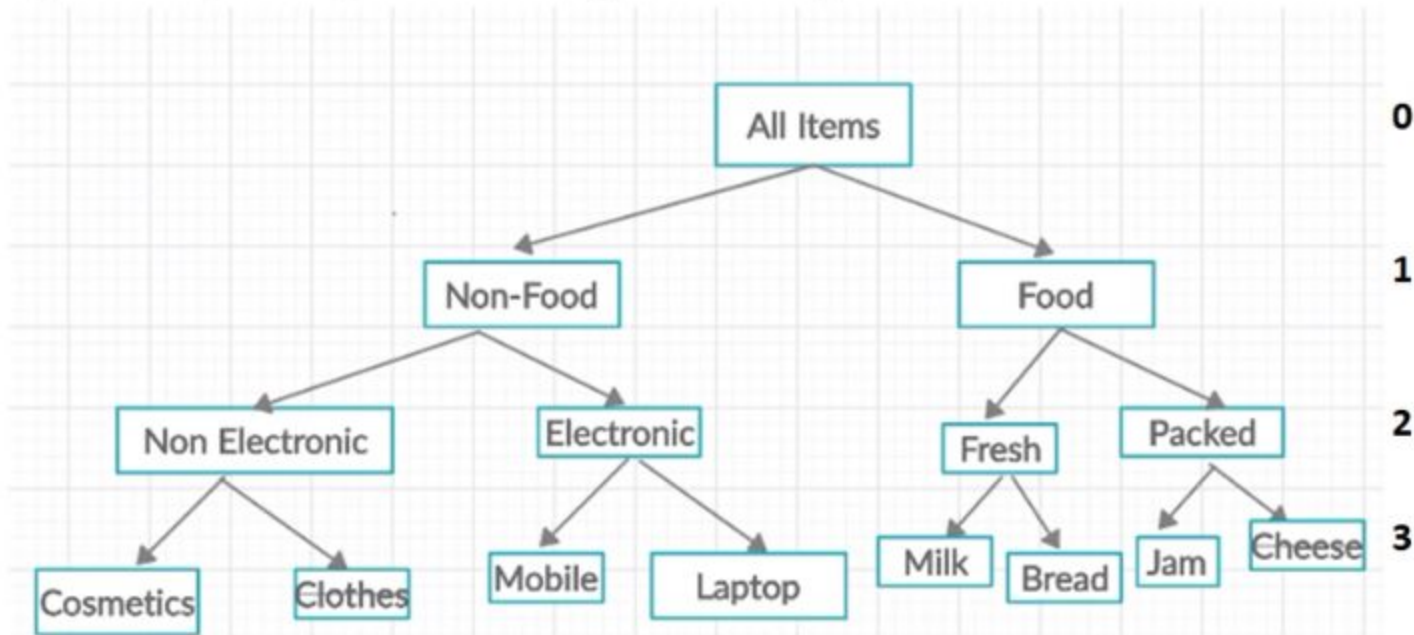
MULTILEVEL ASSOCIATION RULES



Multilevel Association Rules

- When transactions data is taken for link analysis. It is present at the low level of abstraction that is detail form.
 - It is very difficult to form association rules at the low level of abstraction as data scarcity is there. Also resultant rules can not efficiently used.
 - Using concept hierarchies, transaction data can be represented at various levels of abstraction.
 - In Multilevel Association Rules, association rules are generated at multiple levels of abstraction
 - Instead of going at lower level of abstraction, association rules are generated from higher level of abstraction which represents common sense knowledge and be use used efficiently.
- 

Concept Hierarchy: It is a sequence of mappings from a set of low-level concepts to higher level, more general concepts. Below figure has five levels from levels 0 to 4,

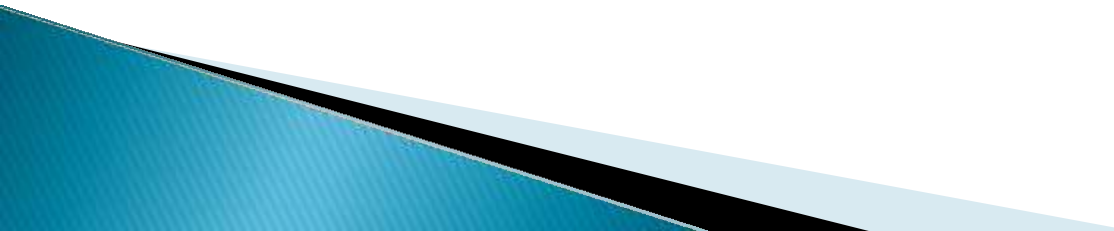


Cosmetics	Clothes	Mobile	Laptop	Milk	Bread	Jam	Cheese
CO1	CL1	MO1	L1	M1	B1	J1	CH1
CO2	CL2	MO2	L2	M2	B2	J2	CH2
CO3	CL3	MO3	L3	M3	B3	J3	CH3
CO4	CL4	MO4	L4	M4	B4	J4	CH4

Need of Multiple-Level Association Rules?

- Sometimes at low data level, data does not show any significant pattern. But there are useful information hiding behind.
- Aim is to find the hidden information in or between levels of abstraction

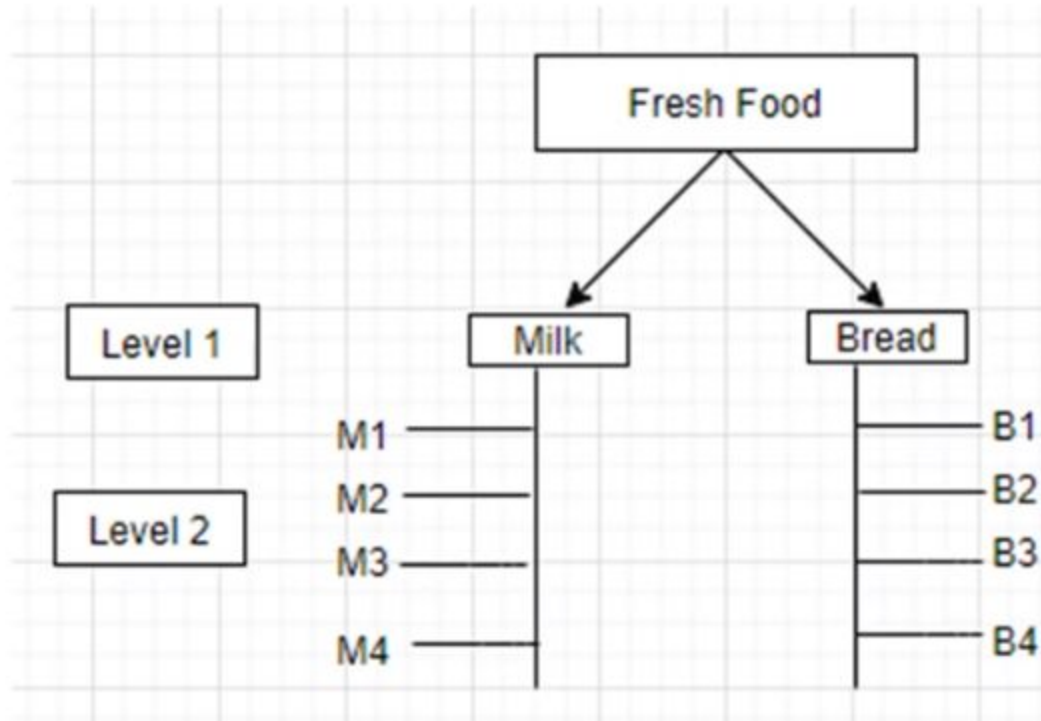
Three ways

- 1) Uniform Support (Using uniform minimum support for all levels)**
 - 2) Reduced support (Using reduced minimum support at lower levels)**
 - 3) Group-based support (Using item or group based minimum support)**
- 

1) Uniform Support (Using uniform minimum support for all levels)

It specifies only one minimum support threshold at all levels.

Simple to implement.



Level 2

TID	Items
T1	M1, B2
T2	M2, B1
T3	B2
T4	M3, B1
T5	M2

Level 1

TID	Items
T1	Milk, Bread
T2	Milk, Bread
T3	Bread
T4	Milk, Bread
T5	Milk

Case 1: Min Sup.=**20%** i.e. **1**

Case 2: Min Sup.=**40%** i.e. **2**

Case 3: Min Sup.=**50%** i.e. **3**

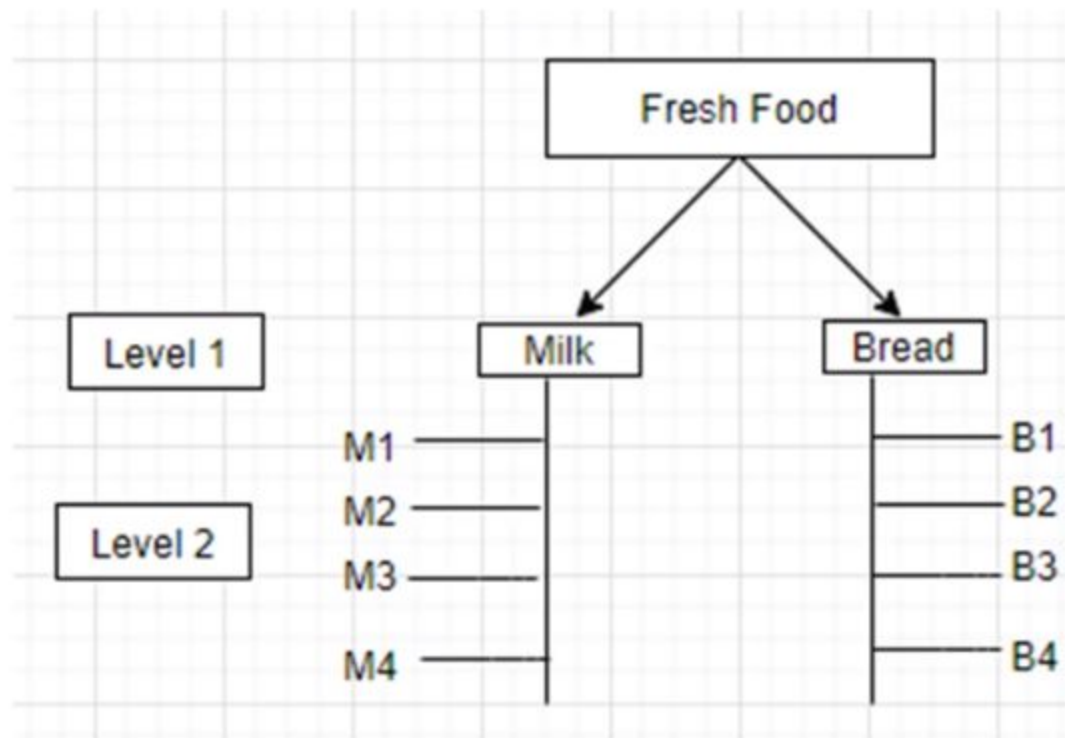
Drawback :

- If the minimum support threshold is set too high, it could miss some meaningful associations occurring at low abstraction levels.
- If the threshold is set too low, it may generate many uninteresting associations occurring at high abstraction levels.



2) Reduced Support (Using reduced minimum support at lower levels)

It specifies different threshold at different level. High threshold at higher level and low threshold at lower level.



Level 2

TID	Items
T1	M1, B2
T2	M2, B1
T3	B2
T4	M2, B1
T5	M2

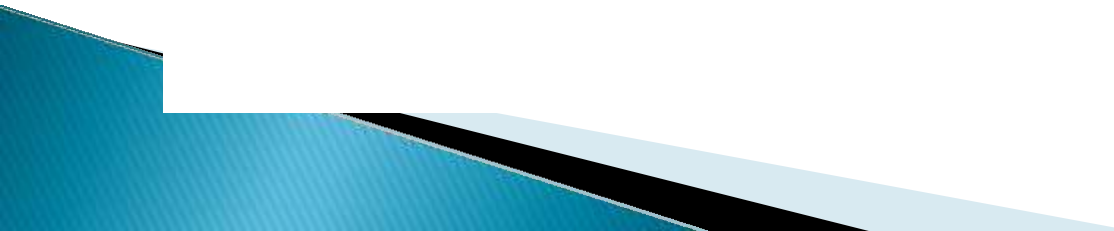
Level 1

TID	Items
T1	Milk, Bread
T2	Milk, Bread
T3	Bread
T4	Milk, Bread
T5	Milk

Level 2: Min Sup.=40% i.e. 2

Level 1: Min Sup.=50% i.e. 3

3) Group-based support (Using item or group based minimum support)

- Group wise threshold value for support and confidence is input by the user or expert.
 - Group is selected based on product price or item of interest. Because often experts have insight as to which groups are more important than others.
 - For e.g. Experts are interested in purchase pattern of laptop and cloths in non-and electronic category. Therefore low support threshold is set for this group to give attention of these items purchase pattern.
- 

MULTIDIMENSIONAL ASSOCIATION RULES

Dimension represents attributes in database.

1) Single dimensional or Intra Dimensional Association Rule

It contains a single distinct predicate (e.g. purchase) with its multiple occurrences.

For e.g. purchase(X, "Milk") \rightarrow purchase(X, "Bread")

TID	Purchase
1	Milk, Bread
2	Bread, Butter
3	Beer
4	Milk, Bread, Butter
5	Bread, Beer
6	Milk, Bread, Butter

Min. Support=3

2) Multi dimensional or Inter Dimensional Association Rule

It contains two or more predicate. Each predicate occurs only once.

e.g.

$\text{Student}(X, \text{"Yes"}) \wedge \text{Credit Rating}(X, \text{"Excellent"}) \rightarrow \text{buys_Laptop}(X, \text{"Yes"})$

$\text{Student}(X, \text{"No"}) \wedge \text{Credit Rating}(X, \text{"Fair"}) \rightarrow \text{buys_Laptop}(X, \text{"No"})$

No.	Student	Credit Rating	Buys_Laptop
1	Yes	Excellent	Yes
2	Yes	Fair	No
3	No	Excellent	Yes
4	No	Excellent	Yes
5	Yes	Fair	No
6	No	Fair	No

Min. Support=1

Attributes	Item
Student=Yes	I1
Student=No	I2
Credit Rating=Excellent	I3
Credit Rating=Fair	I4
Buys_Laptop=Yes	I5
Buys_Laptop=No	I6

P: Present A: Absent

No.	Student	Credit Rating	Buys_Laptop
1	Yes	Excellent	Yes
2	Yes	Fair	No
3	No	Excellent	Yes
4	No	Excellent	Yes
5	Yes	Fair	No
6	No	Fair	No

Attributes	Item
Student=Yes	I1
Student=No	I2
Credit Rating=Excellent	I3
Credit Rating=Fair	I4
Buys_Laptop=Yes	I5
Buys_Laptop=No	I6

C1

I1	I2	I3	I4	I5	I6
P	A	P	A	P	A
P	A	A	P	A	P
A	P	P	A	P	A
A	P	P	A	P	A
P	A	A	P	A	P
A	P	A	P	A	P
3	3	3	3	3	3

Apply Apriori

(2 Itemset generation)

L1

I1	I2	I3	I4	I5	I6
P	A	P	A	P	A
P	A	A	P	A	P
A	P	P	A	P	A
A	P	P	A	P	A
P	A	A	P	A	P
A	P	A	P	A	P
3	3	3	3	3	3

C2

Itemset	Count
(I1, I2)	0
(I1, I3)	1
(I1, I4)	2
(I1, I5)	1
(I1, I6)	2
(I2, I3)	2
(I2, I4)	1
(I2, I5)	2
(I2, I6)	1
(I3, I4)	0
(I3, I5)	3
(I3, I6)	0
(I4, I5)	0
(I4, I6)	3
(I5, I6)	0

(3 Itemset generation)

Join & Prune property:

First elements must be common of two records.

L2

Itemset	Count
(l1, l3)	1
(l1, l4)	2
(l1, l5)	1
(l1, l6)	2
(l2, l3)	2
(l2, l4)	1
(l2, l5)	2
(l2, l6)	1
(l3, l5)	3
(l4, l6)	3

C3

Itemset	Count
(l1, l3, l4)	0
(l1, l3, l5)	1
(l1, l3, l6)	0
(l1, l4, l5)	0
(l1, l4, l6)	2
(l1, l5, l6)	0
(l2, l3, l4)	0
(l2, l3, l5)	2
(l2, l3, l6)	0
(l2, l4, l5)	0
(l2, l4, l6)	1
(l2, l5, l6)	0

L3

Itemset	Count
(I1, I3, I5)	1
(I1, I4, I6)	2
(I2, I3, I5)	2
(I2, I4, I6)	1

Join & Prune property:

4 Itemset generation not possible as first two items are not same.

Select those records having I5 and I6 item as they represent final class label.

Predefined confidence can be used.

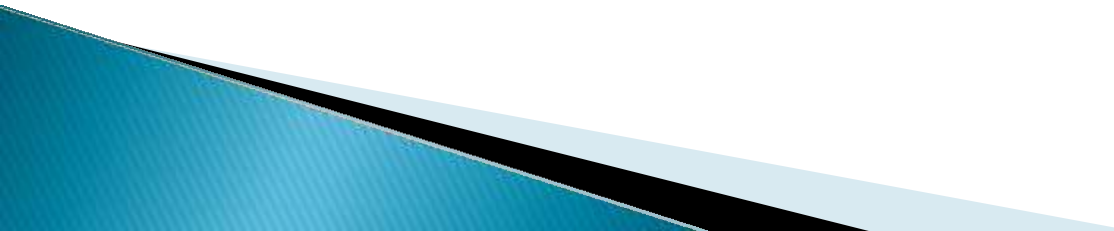
Association Rule	Confidence	Confidence (%)
$I1 \wedge I3 \rightarrow I5$	$C(I1, I3, I5) / C(I1, I3) = 1/1$	100%
$I2 \wedge I3 \rightarrow I5$	$C(I2, I3, I5) / C(I2, I3) = 2/2$	100%
$I1 \wedge I4 \rightarrow I6$	$C(I1, I4, I6) / C(I1, I4) = 2/2$	100%
$I2 \wedge I4 \rightarrow I6$	$C(I2, I4, I6) / C(I2, I4) = 1/1$	100%

Four Multidimensional rules can be obtained

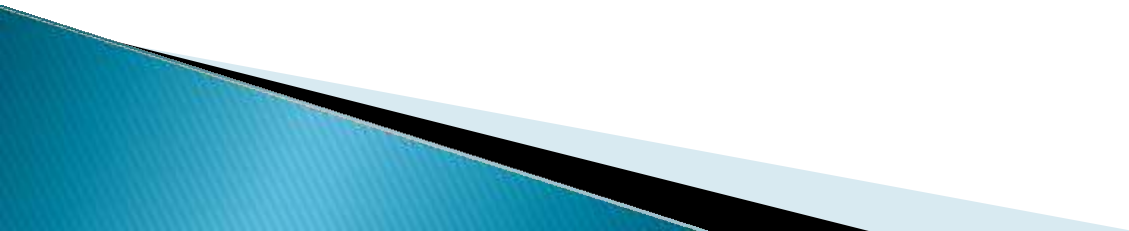
- 1) $\text{Student}(X, \text{"Yes"}) \wedge \text{Credit Rating}(X, \text{"Excellent"}) \rightarrow \text{buys_Laptop}(X, \text{"Yes"})$
- 2) $\text{Student}(X, \text{"No"}) \wedge \text{Credit Rating}(X, \text{"Fair"}) \rightarrow \text{buys_Laptop}(X, \text{"No"})$
- 3) $\text{Student}(X, \text{"Yes"}) \wedge \text{Credit Rating}(X, \text{"Fair"}) \rightarrow \text{buys_Laptop}(X, \text{"No"})$
- 4) $\text{Student}(X, \text{"No"}) \wedge \text{Credit Rating}(X, \text{"Excellent"}) \rightarrow \text{buys_Laptop}(X, \text{"Yes"})$



Interesting measure- Lift and χ^2

- Most association rule mining algorithms employ a support-confidence framework.
 - Often, many interesting rules can be found using low support thresholds.
 - Although minimum support and confidence thresholds *help* weed out or exclude the exploration of a good number of uninteresting rules, many rules so generated are still not interesting to the users
- 

1) Strong Rules Are Not Necessarily Interesting: An Example



Interestingness Measure: Lift

- Measure of dependent/correlated events: **lift**

$$\text{lift}(B, C) = \frac{c(B \rightarrow C)}{s(C)} = \frac{s(B \cap C)}{s(B) \times s(C)}$$

- Lift(B, C) may tell how B and C are correlated

- Lift(B, C) = 1: B and C are independent
- > 1: positively correlated
- < 1: negatively correlated

- For our example, $\text{lift}(B, C) = \frac{400 / 1000}{600 / 1000 \times 750 / 1000} = 0.89$

$$\text{lift}(B, \neg C) = \frac{200 / 1000}{600 / 1000 \times 250 / 1000} = 1.33$$

- Thus, B and C are negatively correlated since $\text{lift}(B, C) < 1$;
- B and $\neg C$ are positively correlated since $\text{lift}(B, \neg C) > 1$

Lift is more telling than s & c

	B	$\neg B$	Σ_{row}
C	400	350	750
$\neg C$	200	50	250
Σ_{col}	600	400	1000

Interestingness Measure: χ^2

- Another measure to test correlated events: χ^2

$$\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}$$

- General rules

- $\chi^2 = 0$: independent

- $\chi^2 > 0$: correlated, either positive or negative, so it needs additional test

- Now, $\chi^2 = \frac{(400 - 450)^2}{400} + \frac{(350 - 300)^2}{350} + \frac{(200 - 150)^2}{200} + \frac{(50 - 100)^2}{50} = 55.89$

- χ^2 shows B and C are negatively correlated since the expected value is 450 but the observed is only 400

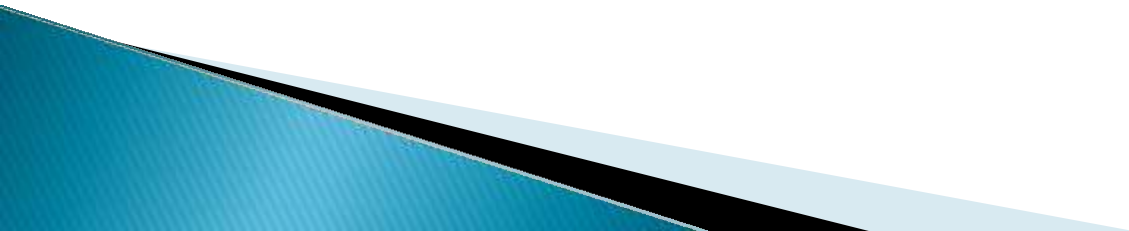
χ^2 tells also better than s & c

	B	$\neg B$	Σ_{row}
C	400 (450)	350 (300)	750
$\neg C$	200 (150)	50 (100)	250
Σ_{col}	600	400	1000

Expected value

Observed value

Constraint- Based Association Mining.



Constraints

While mining, the users specify intuition or expectations as constraints to confine the search space. This strategy is known as constraint-based mining.

The constraints can include the following:

Knowledge type constraints: Specify the type of knowledge to be mined, such as association or correlation.

Data constraints: These specify the set of task-relevant data.

Dimension/level constraints: These specify the desired dimensions (or attributes) of the data, or levels of the concept hierarchies, to be used in mining.

Interestingness constraints: These specify thresholds on statistical measures of rule interestingness, such as *support*, *confidence*, and *correlation*.

Rule constraints: These specify the form of rules to be mined.

Meta rule-Guided Mining Rules



- Metarules allow users to specify the syntactic form of rules that they are interested in mining.
- The rule forms can be used as constraints to help improve the efficiency of the mining process.
- Metarules may be based on the analyst's experience, expectations, or intuition regarding the data or may be automatically generated based on the database schema.

$$P_1(X, Y) \wedge P_2(X, W) \Rightarrow \text{buys}(X, \text{"office software"}), \quad (5.26)$$

$$\text{age}(X, \text{"30...39"}) \wedge \text{income}(X, \text{"41K...60K"}) \Rightarrow \text{buys}(X, \text{"office software"}) \quad (5.27)$$

