Association Mining

I. Motivation

1. Why is Frequent Pattern Mining Important?

Frequent pattern mining is crucial for discovering hidden relationships in large datasets, which can lead to valuable insights and improved decision-making.

2. Practical Examples

→ Supermarket Shelf Management (Market-Basket Model):

- Goal: Identify items frequently bought together.
- Example Rule: "If someone buys diapers and milk, they are likely to buy beer"
- Use cases:
  - ↳ Optimize store layouts by placing commonly co-purchased items closer
  - ↳ Implement promotions, e.g. discount on diapers while increasing beer prices

→ Applications in E-commerce:

- Amazon's recommendation system: "People who bought X also bought Y"

→ Plagiarism Detection:

- Items: Sentences from documents
- Frequent patterns of sentence overlaps may indicate plagiarism.

→ Healthcare
. Items : Drugs and side effects
. Detect combinations of drugs that may lead to specific side effects
. Note : Require observing both the presence and absence of items.

→ Graph-based Analysis
. Discovering communities in networks like Twitter

3. In General
. A general many-to-many mapping (association) between two kinds of things
↳ But we ask about connections among "items", not "baskets".

II. Definition

1. Frequent Itemsets

. Definition :
A set of items that appear together frequently in a dataset.
. Measure
→ Support : The nb. of basket (Transactions) that contain the specified item(s)
→ Formula :

$$\text{Support}(I) = \frac{\text{Nb of baskets containing } I}{\text{Total nb. of baskets}}$$

→ Frequent Itemset : An itemset with support greater than or equal to a predefined threshold.

Example: Find Frequent Itemsets

Transactions:

TID	Items
1	Milk, Coke, Beer
2	Milk, Beer
3	Milk, Pepsi, Beer
4	Coke, Beer, Juice
5	Milk, Pepsi, Juice
6	Coke, Juice
7	Milk, Coke, Beer, Juice
8	Beer, Coke

Support Threshold = 3 baskets

Step 1: Count the support of each item and remove those that have support count lower then the threshold (3)

Supp (Milk) = 5 ✓ Supp (Beer) = 6 ✓

Supp (Coke) = 5 ✓ Supp (Juice) = 4 ✓

Supp (Pepsi) = 2 ✗

L_1 = {Milk}, {Coke}, {Beer}, {Juice}

Step 2: Count the support of each pair of the remaining items

Supp {Milk, Coke} = 2 ✗ Supp {Coke, Beer} = 4 ✓

Supp {Milk, Beer} = 4 ✓ Supp {Coke, Juice} = 3 ✓

Supp {Milk, Juice} = 2 ✗ Supp {Beer, Juice} = 2 ✗

$L_2 = \{Milk, Beer\}, \{Coke, Beer\}, \{Coke, Juice\}$

Step 3: Count the support of set of 3 items

Supp {Milk, Beer, Coke} = 2 ✗
Supp {Milk, Beer, Juice} = 1 ✗
Supp {Coke, Beer, Juice} = 2 ✗
Supp {Milk, Coke, Juice} = 1 ✗

=> Frequent Itemset:
{Milk}, {Coke}, {Beer}, {Juice}, {Milk, Beer}
{Coke, Beer}, {Coke, Juice}

2. Association Rules

Definition: IF-then rules about the contents of baskets.
IF $\{i_1, i_2, i_3, \ldots, i_k\} \rightarrow j$. it means:
"IF a basket contains $\{i_1, i_2, i_3, \ldots, i_k\}$, it is likely to contain j."

Confidence: The probability that j is in the basket, given $I = \{i_1, i_2, i_3, \ldots, i_k\}$ is present.

→ Formula:

$$conf(I \rightarrow j) = \frac{Support\ (I \cup j)}{Support\ (I)}$$

3. Interesting Association Rules

Not all high-confidence rules are interesting, the rule $X \rightarrow$ milk may have high confidence for many itemsets X, because milk is just

purchased very often (independent of X) and the confidence will be high. So we need to find the interest of an association rule $I \rightarrow j$

- **Interest**: Difference between the confidence of the rule and the overall probability of j appearing in any basket
  - Formula:

$$\text{Interest}(I \rightarrow j) = \text{Confidence}(I \rightarrow j) - \Pr(j)$$

  - Interesting rules are those with high positive or negative interest values (usually above 0.5)

- **Example**: Based on the previous transactions table

$$\text{Confidence}\{\text{Milk}\} \rightarrow \{\text{Beer}\} = \frac{\text{Support}\{\text{Milk, Beer}\}}{\text{Support}\{\text{Milk}\}}$$

$$= 4/5 = 0.8$$

$$\text{Confidence}\{\text{Milk, Beer}\} \rightarrow \{\text{Coke}\} = \frac{\text{Suppor}\{M,B,C\}}{\text{Support}\{M,B\}}$$

$$= 2/4 = 0.5$$

$$\text{Interes}\{\text{Milk, Beer}\} \rightarrow \{\text{coke}\} = \text{Confidence}\{M,B\} \rightarrow \{C\} - \Pr\{\text{Coke}\}$$

$$= |0.5 - 5/8| = 0.125 < 0.5$$

$\Rightarrow$ So the rule $\{\text{Milk, Beer}\} \rightarrow \{\text{Coke}\}$ is not very interesting.

4. Finding Association Rules - Overview

- **Problem:** Find all association rules with support $\geq s$ and confidence $\geq c$
- **Process:**

1) **Finding Frequent Itemsets I:** Find all itemsets that appear frequently enough to meet the support threshold
2) **Generating Association Rules:** For each frequent itemset generate potential association rules:
   For every subset A of I, generate a rule $A \rightarrow I \setminus A$
   ↳ For example, from {m, b}, we can generate rules like {m} → {b} and {b} → {m}
3) **Pruning and Confidence Calculation:** For each rule, calculate its confidence.
   ↳ Rules that don't meet the confidence threshold are discarded.
4) **Filtering Interesting Rules:** Finaly, rules with high positive or negative interest values are selected as the most interesting.

- **Example:**

Baskets	Items
B1	m, c, b
B2	m, p, j
B3	m, c, b, n
B4	c, j
B5	m, p, b
B6	m, c, b, j
B7	c, b, j
B8	b, c

Support Threshold = 3
Confidence Threshold = 0.75

1) Find Frequent Itemsets

s = 3

Itemset	Supp
m	5
c	6
b	6
n	1
p	2
j	4

C1

Itemset	Supp
m	5
c	6
b	6
j	4

L1

Itemset	Supp
m.c	3
m.b	4
m.j	2
c.b	5
c.j	3
b.j	2

C2

Itemset	Supp
m.c	3
m.b	4
c.b	5
c.j	3

L2

Itemset	Supp
m.c.b	3
m.c.j	1
m.b.j	1
c.b.j	2

C3

Itemset	Supp
m.c.b	3

L3

Frequent itemsets: {m.c}, {m.b}, {c.b}, {c.j}, {m.c.b}

c = 0.75

2) Generating Association Rules

Table of rules with confidence

Rule	Confidence
m → c	3/5 = 0.6
c → m	3/6 = 0.5
m → b	4/5 = 0.8
b → m	4/6 = 0.66
c → b	5/6 = 0.83
b → c	5/6 = 0.83
c → j	3/6 = 0.5
j → c	3/4 = 0.75

Rule	Confidence
m → cb	3/5 = 0.6
c → mb	3/6 = 0.5
b → mc	3/6 = 0.5
mc → b	3/3 = 1
mb → c	3/4 = 0.75
cb → m	3/5 = 0.6

Association Rules:

$m \rightarrow b$, $c \rightarrow b$, $b \rightarrow c$, $j \rightarrow c$

$mc \rightarrow b$, $mb \rightarrow c$

3) Filtering: Interesting rules by computing interest for each one

Interest >0.5

Interest $\{m \rightarrow b\} = |0.8 - 6/8| = 1/20 = 0.05$ ✗

Interest $\{c \rightarrow b\} = |0.83 - 6/8| = 2/25 = 0.08$ ✗

Interest $\{b \rightarrow c\} = |0.83 - 6/8| = 2/25 = 0.08$ ✗

Interest $\{j \rightarrow c\} = |0.75 - 6/8| = 0$ ✗

Interest $\{mc \rightarrow b\} = |1 - 6/8| = 1/4 = 0.25$ ✗

Interest $\{mb \rightarrow c\} = |0.75 - 6/8| = 0$ ✗

So we don't have any interesting rule.

III. Algorithms

1. A-Priori Algorithm

→ Key Idea: Monotonicity

. IF an itemset I is Frequent (i.e., appears in at least s baskets), then all subsets of I are also Frequent.

. Conversely, if an item i is not Frequent, then no larger itemsets containing i can be Frequent.

→ Steps of the Apriori Algorithm

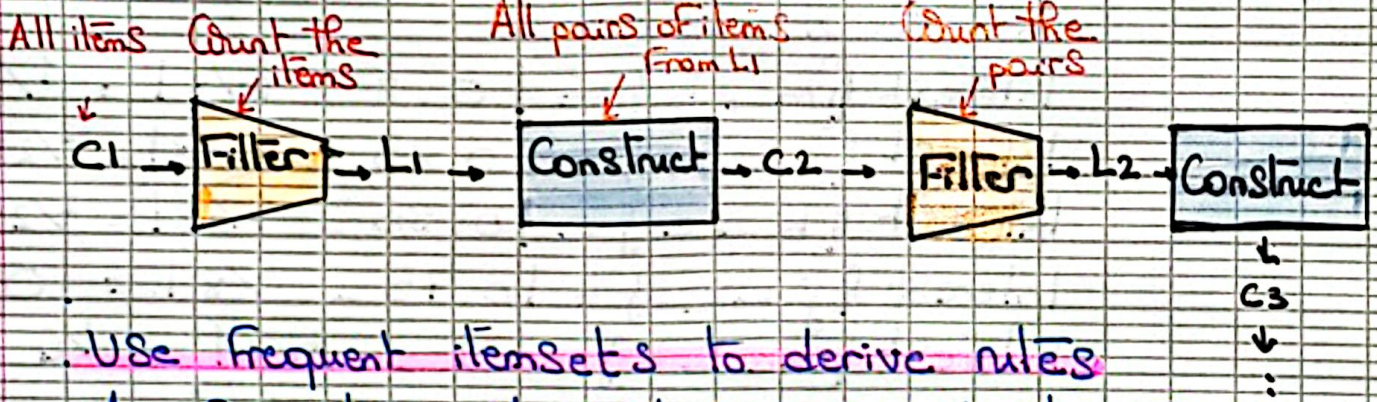
. Iteratively find Frequent K-itemsets by:

1) Candidate Generation (CK): Generate

K-item candidate using Frequent (K-1)-itemsets.

2) Support Pruning: Retain only those candidates whose support meets or exceeds the minimum support threshold.

↳ We repeat until no new frequent itemsets are found



. Use frequent itemsets to derive rules $A \rightarrow B$ where A and B are subsets of the itemset, such that:

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)} \geq \text{minConf}$$

→ Example

1. Generation of frequent itemsets

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

Sup(A) = 2/4 = 0.5 x

Sup(B) = 3/4 = 0.75 ✓

Sup(C) = 3/4 = 0.75 ✓

Sup(D) = 1/4 = 0.25 x

Sup(E) = 0.75 ✓

L1 = {B}, {C}, {E}

Supp (BC) = 2/4 = 0.5 x

Supp (BE) = 3/4 = 0.75 ✓

Supp (EC) = 2/4 = 0.5 x

L2 = {BE}

⇒ Frequent Itemsets: B, C, E, BE

2. Generate Rules

$B \rightarrow E$: $conf(B \rightarrow E) = sup(B \rightarrow E) / sup(B)$
$= 0.75 / 0.75 = 1$

$E \rightarrow B$: $conf(E \rightarrow B) = supp(B \rightarrow E) / sup(E)$
$= 0.75 / 0.75 = 1$

2. FP-Growth Algorithm:

. The FP-Growth (Frequent Pattern Growth) Algorithm is an efficient method for mining frequent patterns from large datasets without generating candidate sets, as seen in the Apriori algorithm

→ Key Concepts

1) Compress the Database

. Convert the large database into a compact, Frequent-Pattern tree (FP-tree)

: This tree preserves the itemset information, avoids costly database scans, and organizes data to facilitate frequent pattern discovery

2) Divide-and-Conquer Approach:

. The algorithm recursively decomposes mining tasks into smaller subtasks

. Frequent patterns are mined directly from the FP-tree, avoiding the generation and testing of candidate sets.

→ Steps of the FP_Growth Algorithm

Step 1: First Scan - Count and Sort Items
- Count the support of each item in the database
- Retain only items with support ≥ minSup
- Create a frequent item list L, sorted in descending order of support

Step 2: Second Scan - Construct the FP_Tree
- Process each Transaction
1. Retain only frequent items sorted by L's order
2. Traverse the FP_Tree :
   - If prefix nodes exist, increment their count
   - If not, create new nodes
3. Maintain a header table linking all occurences of an item in the tree

Step 3: Conditional Pattern Base and Tree
- For each frequent item:
1. Conditional Pattern Base: Extract all paths from the FP_Tree leading to the given item
2. Conditional FP_Tree: Construct a smaller FP_tree using the conditional pattern base
3. Drop infrequent items from the conditional tree

Step 4: Generate Frequent Patterns
- Combine the frequent 1_itemset with the frequent patterns from the conditional FP_trees to generate the complete set of frequent patterns.

Step 5: Generate Association Rules

- From the Frequent patterns:
  - Form rules $A \rightarrow B$ such that:

$$\text{Confidence}(A \rightarrow B) = \frac{\text{Support}(A \cup B)}{\text{Support}(A)} \geq \text{minConf}$$

  - Retain rules that meet the confidence threshold

→ Example:

Transactional Database

TID	Items
T_1	E, K, M, N, O, Y
T_2	D, E, K, N, O, Y
T_3	A, E, K, M
T_4	C, K, M, U, Y
T_5	C, E, I, K, O, O

Parameters:
- minSup = 3
- minConf = 0.8

Step 1: First Scan - Count and Sort

- Count the support of each item

Sup(K) = 5 ✓ Sup(E) = 4 ✓ Sup(M) = 3 ✓
Sup(N) = 2 × Sup(O) = 3 ✓ Sup(Y) = 3 ✓
Sup(D) = 1 × Sup(A) = 1 × Sup(C) = 2 ×
Sup(I) = 1 × Sup(U) = 1 ×

- Frequent Items L:

Retain items with Support ≥ 3, sorted by desc. Support

L = {K, E, M, O, Y}

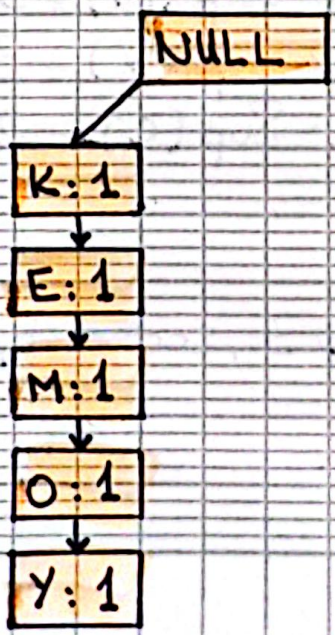
Step 2 : Second Scan - Build FP-tree

- Construct the FP-Tree by inserting Transactions in L's order

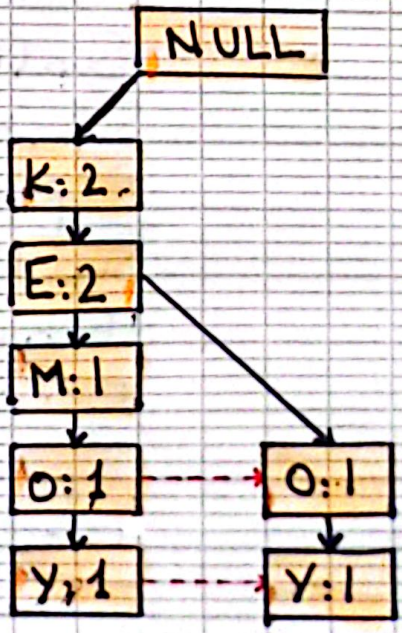
$T_1 = \{K, E, M, O, Y\}$
$T_2 = \{K, E, O, Y\}$
$T_3 = \{K, E, M\}$
$T_4 = \{K, M, Y\}$
$T_5 = \{K, E, O\}$

- First we create the root, label it as "Null"
- For each Transaction :
  - we'll increase nodes count by 1 if the prefix exist
  - we'll create a new nodes and set count to 1 otherwise.
- Nodes with same item-name are linked in sequence via nodes-links

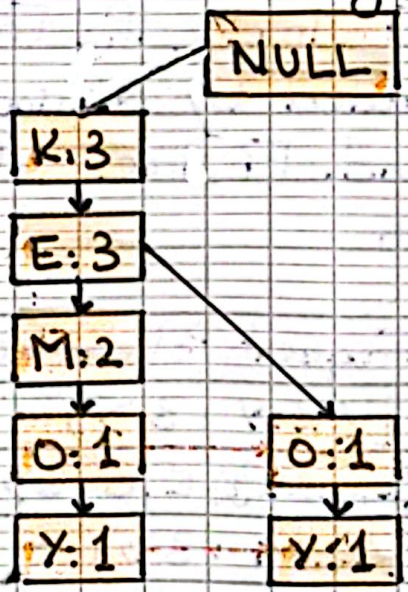
- After Inserting $\{K, E, M, O, Y\}$



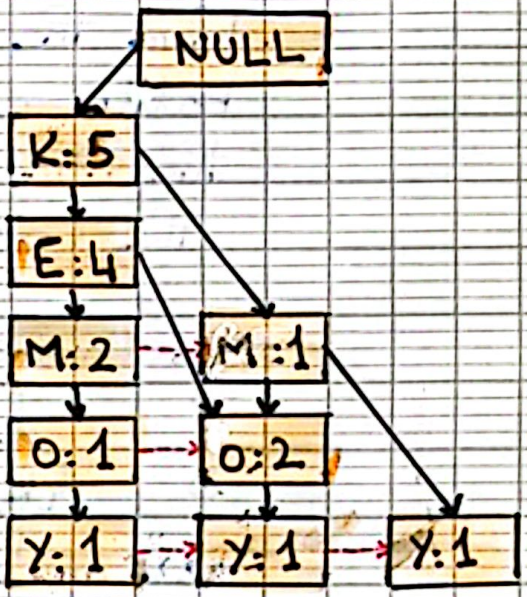
- After Inserting $\{K, E, O, Y\}$



↳ After Inserting {K, E, M}



↳ Final Tree



Step 3 : Conditional Pattern Base and FP-Trees

↳ For each item, the conditional pattern base is computed which is path labels of all the paths which lead to any node of the given item in the frequent pattern tree.

↳ Then for each item the conditional frequent pattern tree is built, by accumulating the count of the items in the conditional pattern base.

Item	Conditional Pattern Base	Conditional FP-Tree
Y	{K,E,M,O:1}; {K,E,O:1}; {K,M:1}	{K:3, E:2, M:2, O:2}
O	{K,E,M:1} ; {K,E:2}	{K:3, E:3, M:3}
M	{K,E:2} ; {K:1}	{K:3, E:2}
E	{K:4}	{K:4}
K	(Root - no base)	(Root - no base)

MinSup = 3

Step 4 : Generate Frequent Patterns

From the conditional FP_tree, the Frequent Pattern rules are generated by pairing the items of the cond. FP_tree set to the corresponding item

Items	Conditional FP-tree	Frequent Pattern Generated
Y	{K:3}	{<K,Y:3>}
O	{K,E:3}	{<K,O:3> <E,O:3> <E,K,O 3>}
M	{K:3}	{<K,M:3>}
E	{K:4}	{<K,E:4>}
K		

Step 5: Generate Rules

For min Conf = 0.8, evaluate rules

Rule	Confidence
K → Y	3/5 = 0.6
Y → K	3/3 = 1
K → O	3/5 = 0.6
O → K	3/3 = 1
E → O	3/4 = 0.75
O → E	3/3 = 1
EK → O	3/4 = 0.75
KO → E	3/3 = 1
EO → K	3/3 = 1
E → KO	3/4 = 0.75
K → EO	3/5 = 0.6

Rule	Confidence
O → EK	3/3 = 1
K → M	3/5 = 0.6
M → K	3/3 = 1
K → E	4/5 = 0.8
E → K	4/4 = 1

IV. Evaluation Metrics for Association Rules

1. Support

- Support $(X \rightarrow Y)$ = Support $(X \cup Y)$
- Indicates how frequently the rule $X \rightarrow Y$ appears in the dataset

2. Confidence

Confidence $(X \rightarrow Y)$ = Support $(X \cup Y)$ / Support (X)

- Measures how often Y occurs when X occurs

V. Interest Measures

1. Lift (Correlation)

range: $[0, \infty]$

- Measures how much more X and Y occur together compared to if they were independent:

$$\text{Lift}(X \rightarrow Y) = \text{Support}(X \rightarrow Y) / \text{Support}(X).\text{Support}(Y)$$
$$= \text{Confidence}(X \rightarrow Y) / \text{Support}(Y)$$

- Interpretation:
  - Lift ≈ 1: No Correlation
  - Lift > 1: Positive Correlation (likely a rlt. between X and Y)
  - Lift < 1: Negative correlation

2. Leverage

range: $[-1, 1]$

- Measures the difference between observed and expected under independence

$$\text{Leverage}(X \rightarrow Y) = \text{Support}(X \rightarrow Y) - \text{Support}(A) \times \text{Support}(C)$$

- Interpretation:
  - Leverage = 0: Independent Items
  - Leverage > 0: positive corr with higher values indicating stronger rlt.

range:
$[0, \infty]$

3. Conviction

- Ratio of the expected frequency of X occuring without Y:

$$\text{Conviction}(X \to Y) = 1 - \text{Support}(Y) / 1 - \text{Confidence}(X \to Y)$$

- Interpretation:
  - Conviction = 1 : Independence
  - Higher conviction indicates stronger dependency of Y on X

→ Example : From the previous table

1. Rule : $Y \to K$, Support(Y) = 3/5 . Support(K) = 1
   Confidence $(Y \to K) = 1$
   . Lift $(Y \to K)$ = . Confidence $(Y \to K)$ / Support(K)
   $= 1 / 1 = 1$
   - occurence of K has no effect on the occurence of Y
   . Leverage $(Y \to K)$ = Supp $(Y \to K)$ - Sup.(Y) x Sup(K)
   $= 3/5 - 3/5 \times 5/5 = 0$
   - Similar observation as lift
   . Conviction $(Y \to K)$ = $(1 - \text{Sup}(K)) / (1 - \text{Conf}(Y \to K))$
   $= (1 - 1) / (1 - 1) \sim \text{inf}$
   - K is highly dependent on Y (Y does not occur without K)