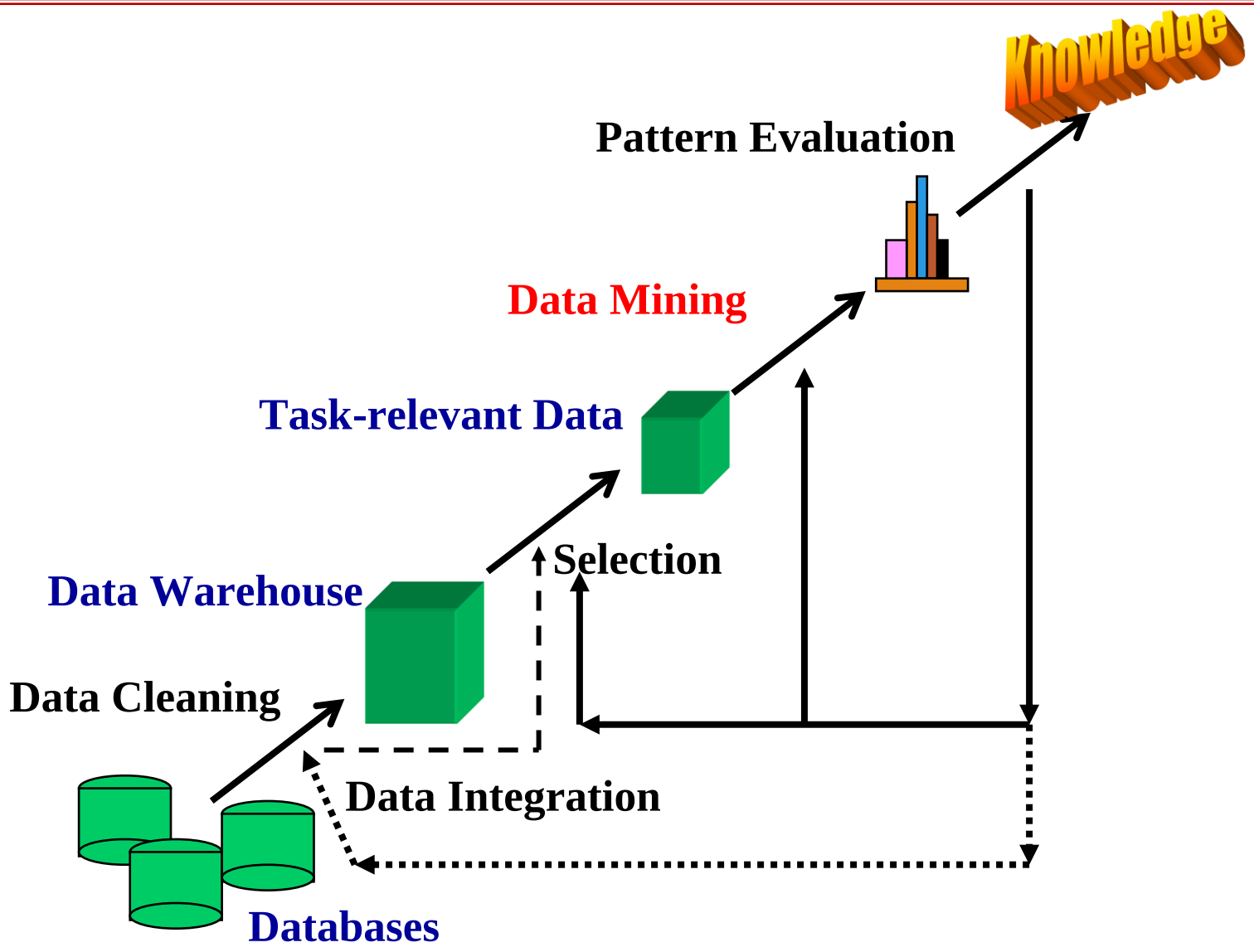
Data Science – Semester 4 – 2022/2023

INTRODUCTION TO Machine Learning

Lecture 4
Frequent Pattern Mining

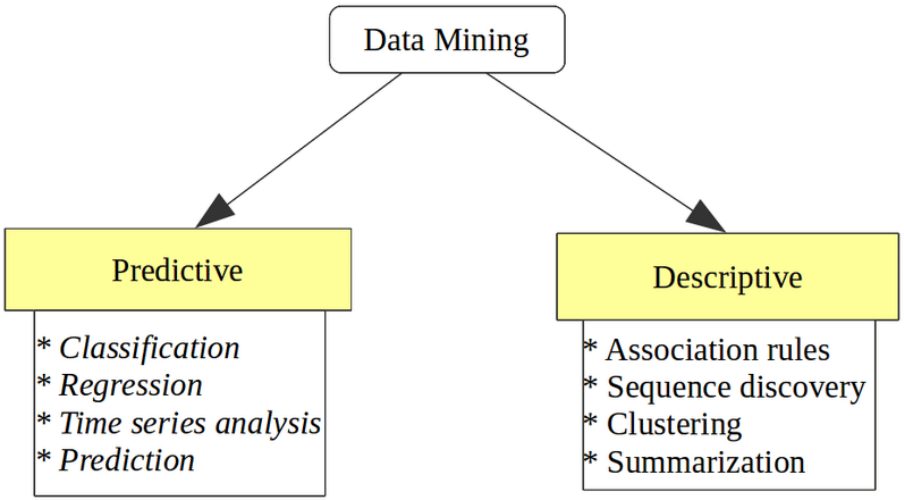
Modified slides from: Mining of Massive Datasets
Jure Leskovec, Anand Rajaraman, Jeff Ullman Stanford University
http://www.mmds.org

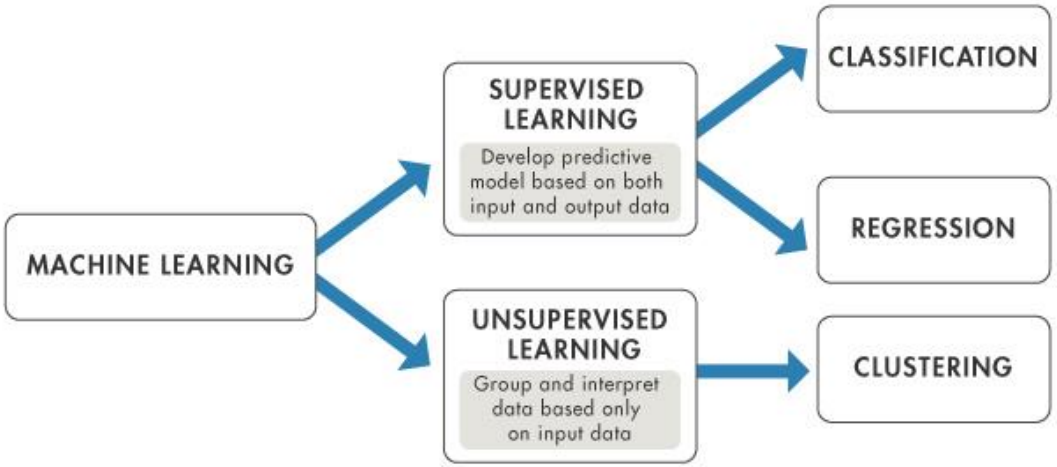
Recall


Knowledge
Pattern Evaluation
Data Mining
Task-relevant Data
Selection
Data Warehouse
Data Cleaning
Data Integration
Databases

Recall

- **Covered data mining techniques**
  - **Association Rule Mining**
  - **Classification/Regression**
  - **Cluster Analysis**





Outline

- Motivation
- Part I (Definition):
  - **Frequent itemsets**
  - **Association rules**
  - **Measure the quality of Association Rule:**
    - Confidence, Support, Interestingness
- Part II (Algorithms):
  - **A-Priori algorithm**
  - **Frequent pattern growth (FPgrowth)**

Association Rule Discovery

Supermarket shelf management – Market-basket model:

- **Goal:** Identify items that are bought together by many customers
- **Approach:** Process the sales data collected with barcode scanners to find dependencies among items
- **A classic rule:**
  - If someone buys diaper and milk, then he/she is likely to buy beer

The Market-Basket Model

- A large set of **items**
  - e.g., things sold in a supermarket
- A **large set** of **baskets**
- Each basket is a **small subset of items**
  - e.g., the things one customer buys on one day
- Want to discover **association rules**
  - People who bought {x,y,z} tend to buy {v,w}
    - Amazon!

Input:

TID	*Items*
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Output:

Rules Discovered:

{Milk} --> {Coke}

{Diaper, Milk} --> {Beer}

Applications – (1)

- **Items** = products; **Baskets** = sets of products someone bought in one trip to the store
- **Real market baskets:** Chain stores keep TBs of data about what customers buy together
  - Tells how typical customers navigate stores, lets them position tempting items
  - Suggests tie-in "tricks", e.g., run sale on diapers and raise the price of beer
  - Need the rule to occur frequently, or no $$'s
- **Amazon's people who bought *X* also bought *Y***

Applications – (2)

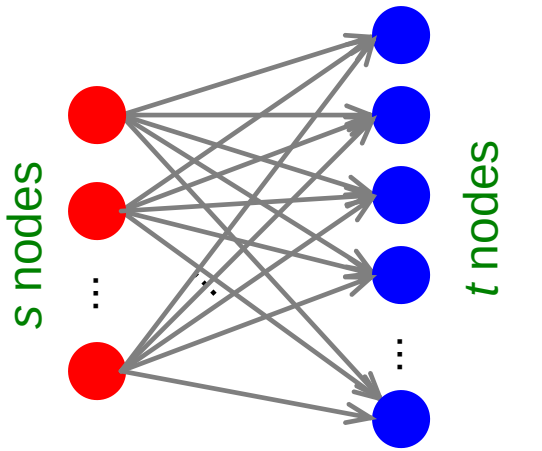
- **Baskets** = sentences; **Items** = documents containing those sentences
  - Items that appear together too often could represent plagiarism
  - Notice items do not have to be "in" baskets
- **Baskets** = patients; **Items** = drugs & side-effects
  - Has been used to detect combinations of drugs that result in particular side-effects
  - **But requires extension:** Absence of an item needs to be observed as well as presence

More generally

- **A general many-to-many mapping (association) between two kinds of things**
  - But we ask about connections among "items", not "baskets"
- **For example:**
  - Finding communities in graphs (e.g., Twitter)

Example:

- **Finding communities in graphs (e.g., Twitter)**
- **Baskets** = nodes; **Items** = outgoing neighbors
  - Searching for complete bipartite subgraphs $K_{s,t}$ of a big graph



A dense 2-layer graph

- **How?**
  - $K_{s,t}$ = a set Y of size t that occurs in s baskets B_i
  - Looking for $K_{s,t}$ → set of support s and look at layer t – all frequent sets of size t

Outline

- Motivation
- Part I (Definition):
  - **Frequent itemsets**
  - **Association rules**
  - **Measure the quality of Association Rule:**
    - Confidence, Support, Interestingness
- Part II (Algorithms):
  - **A-Priori algorithm**
  - **Frequent pattern growth (FPgrowth)**

Frequent Itemsets

- **Simplest question:** Find sets of items that appear together "frequently" in baskets
- ***Support*** for itemset I: Number of baskets containing all items in I
  - (Often expressed as a fraction of the total number of baskets)
- Given a ***support threshold*** s, then sets of items that appear in at least s baskets are called ***frequent itemsets***

TID	Items
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Support of {Beer, Bread} = 2

Example: Frequent Itemsets

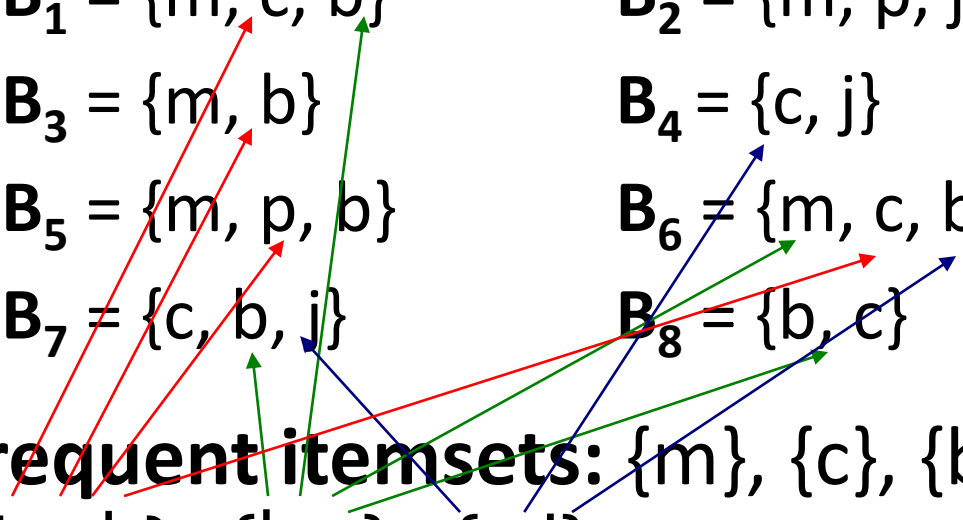
- **Items** = {milk, coke, pepsi, beer, juice}
- **Support threshold** = 3 baskets

B_1 = {m, c, b} B_2 = {m, p, j}

B_3 = {m, b} B_4 = {c, j}

B_5 = {m, p, b} B_6 = {m, c, b, j}

B_7 = {c, b, j} B_8 = {b, c}



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

Association Rules

- **Association Rules:**
  If-then rules about the contents of baskets
- $\{i_1, i_2, \ldots, i_k\} \rightarrow j$ means: "if a basket contains all of $i_1, \ldots, i_k$ then it is ***likely*** to contain j"
- **In practice there are many rules, want to find significant/interesting ones!**
- ***Confidence*** of this association rule is the probability of j given $I = \{i_1, \ldots, i_k\}$

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

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
- **Interest** of an association rule $I \rightarrow j$: difference between its confidence and the fraction of baskets that contain j

$$\text{Interest}(I \rightarrow j) = \text{conf}(I \rightarrow j) - \Pr[j]$$

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

Example: Confidence and Interest

B_1 = {m, c, b} B_2 = {m, p, j}

B_3 = {m, b} B_4= {c, j}

B_5 = {m, p, b} B_6 = {m, c, b, j}

B_7 = {c, b, j} B_8 = {b, c}

- {m,b}: m → b, b → m
- {m,b,c} : frequent itemset: m, c → b
- B,c → m, c → b, m
- Association rule: {m, b} →c
  - Confidence = 2/4 = 0.5
  - Interest = |0.5 – 5/8| = 1/8
    - Item *c* appears in 5/8 of the baskets
    - Rule is not very interesting!

Finding Association Rules

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

- **Hard part:** **Finding the frequent itemsets!**
  - If $\{i_1, i_2, \ldots, i_k\} \rightarrow j$ has high support and confidence, then both $\{i_1, i_2, \ldots, i_k\}$ and $\{i_1, i_2, \ldots, i_k, j\}$ will be "frequent"

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

Mining Association Rules

- **Step 1:** Find all frequent itemsets I
  - (we will explain this next)
- **Step 2:** **Rule generation**
  - For every subset A of I, generate a rule $A \rightarrow I \setminus A$
    - Since I is frequent, A is also frequent
    - **Variant 1:** Single pass to compute the rule confidence
      - confidence(***A,B*→*C,D***) = support(**A,B,C,D**) / support(**A,B**)
    - **Variant 2:**
      - **Observation:** If **A,B,C→D** is below confidence, so is **A,B→C,D**
      - Can generate "bigger" rules from smaller ones!
  - **Output the rules above the confidence threshold**

Example

B_1 = {m, c, b} B_2 = {m, p, j}

B_3 = {m, c, b, n} B_4= {c, j}

B_5 = {m, p, b} B_6 = {m, c, b, j}

B_7 = {c, b, j} B_8 = {b, c}

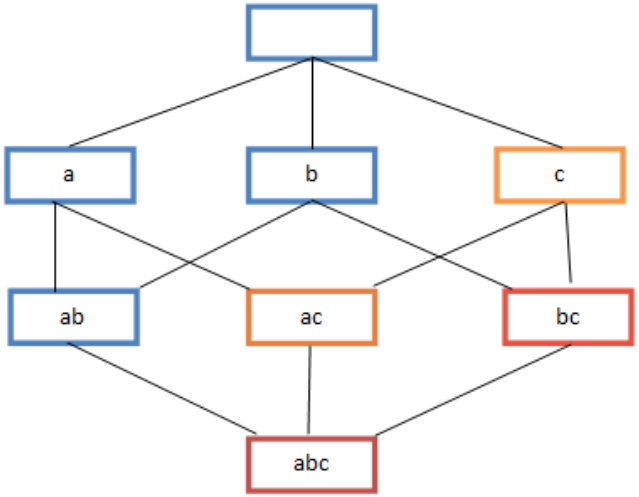
- **Support threshold $s = 3$, confidence $c = 0.75$**
- **1) Frequent itemsets:**
  - {b,m} {b,c} {c,m} {c,j} {m,c,b}
- **2) Generate rules:**
  - ~~b→m: c=4/6~~ b→c: c=5/6 ~~b,c→m: c=3/5~~
  - m→b: c=4/5 … b,m→c: c=3/4
  - ~~b→c,m: c=3/6~~

Outline

- Motivation
- Part I (Definition):
  - **Frequent itemsets**
  - **Association rules**
  - **Measure the quality of Association Rule:**
    - Confidence, Support, Interestingness
- Part II (Algorithms):
  - **A-Priori algorithm**
  - **Frequent pattern growth (FPgrowth)**

A-Priori Algorithm – (1)

- **Key idea: *monotonicity***
  - If a set of items I appears at least s times, so does every **subset** J of I

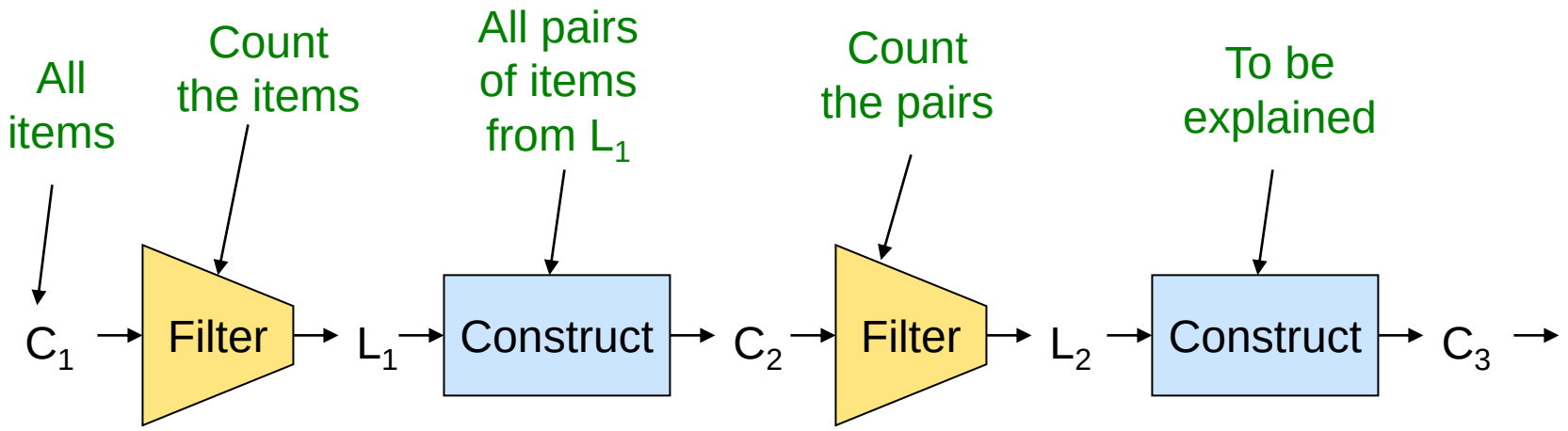


- **Contrapositive for pairs:**
  If item i does not appear in s baskets, then no pair including i can appear in s baskets

- **So, how does A-Priori find freq. pairs?**

Frequent Triples, Etc.

- **For each *k*, we construct two sets of *k-tuples*** (sets of size *k*):
  - C_k = ***candidate k-tuples*** = those that might be frequent sets (support $\geq$ **s**) based on information from the pass for ***k*–1**
  - L_k = the set of truly frequent ***k***-tuples



Example

1. Given the transactional database below, generate all frequent itemsets with minSup =0.5 using Apriori algorithm.
2. Generate all association rules from the frequent itemsets having a confidence = 0.8

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

Example

1- generation of frequent itemsets
Sup(A) = 2/4 = 0.5
Sup(B) = 3/4 = 0.75 --
Sup(C) = 3/4 = 0.75 --
Sup(D) = ¼ = 0.25
Sup(E) = 0.75 –

L1={{B},{C},{E}}
Sup(BC) = 2/4 = 0.5
Sup(BE) = ¾ = 0.75 --
Sup(CE) = 2/4 = 0.5

L2 = { {BE} }

Frequent itemsets: B, C, E, BE

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

Example

Frequent itemsets: B, C, E, BE

B → E : conf(B→E) = sup(B→E)/sup(B) = 0.75/ 0.75 = 1

E → B: conf(E→B) = sup(E→B) / sup(E) = 0.75/0.75 = 1

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

FP-Growth algorithm

- Mining Frequent Patterns Without Candidate Generation
- Compress a large database into a compact, Frequent-Pattern tree (FP-tree) structure
  - highly condensed, but complete for frequent pattern mining
  - avoid costly database scans
- Develop an efficient, FP-tree-based frequent pattern mining method
  - A divide-and-conquer methodology: decompose mining tasks into smaller ones
  - Avoid candidate generation: sub-database test only!

FP-tree construction

Example

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

Step 1

- First Scan: count and sort
  - count the support of each item
  - collect length-1 frequent items, then **sort them in support descending order into *L***, *frequent item list*.

Assume minSup=3

Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
U	1
Y	3

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

Step 2

- Second Scan: create the tree and header table
  - create the root, label it as "*null*"
  - for each transaction *Trans*, do
    - select and sort the frequent items in *Trans*
    - increase nodes count or create new nodes

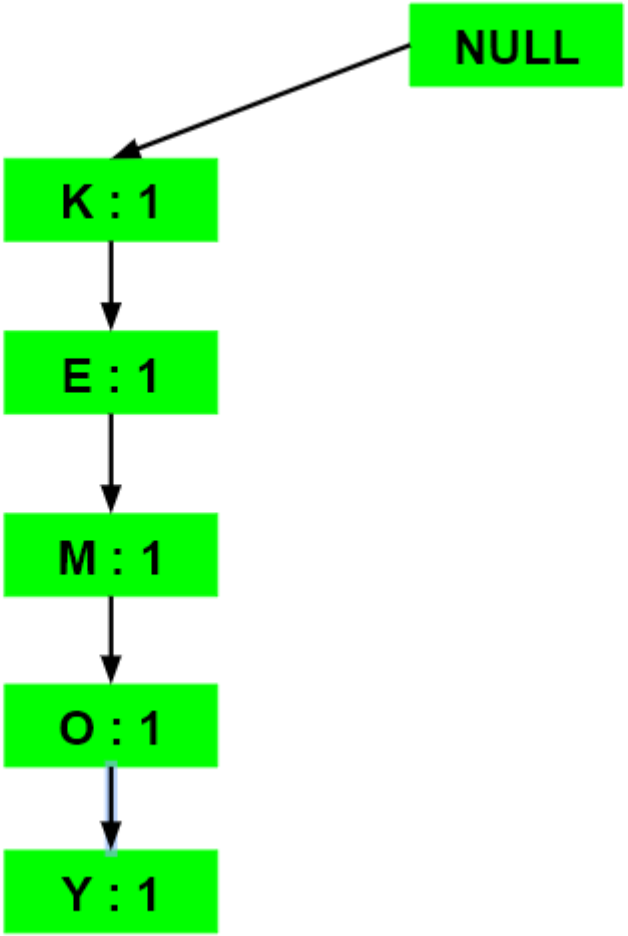
      If prefix nodes already exist, increase their counts by 1;

      If no prefix nodes, create it and set count to 1.
  - build the item header table
    - nodes with the same item-name are linked in sequence via node-links

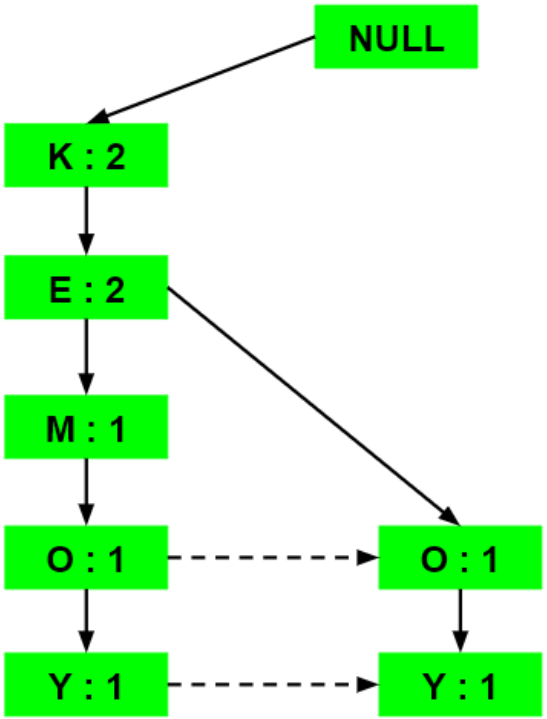
Step 2

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

After inserting {K,E,M,O,Y}



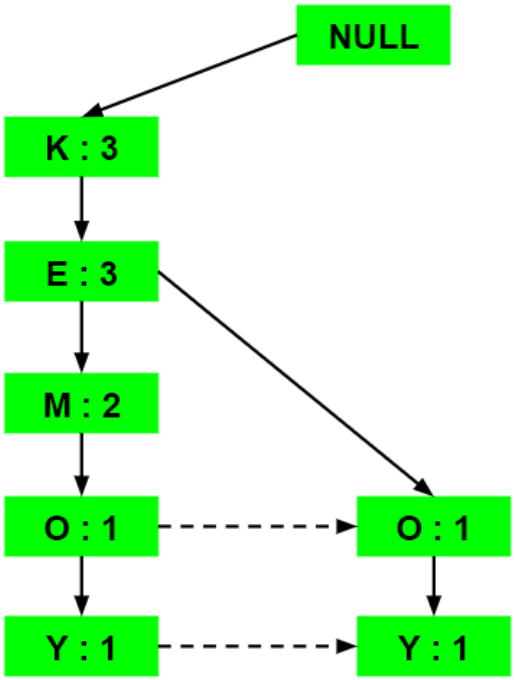
After inserting {K,E,O,Y}



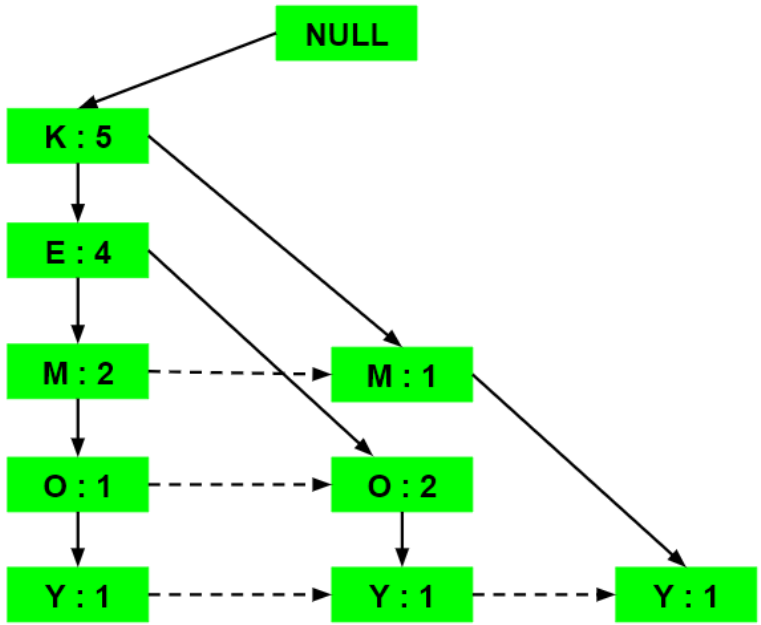
Step 2

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

After inserting {K,E,M}



Final tree



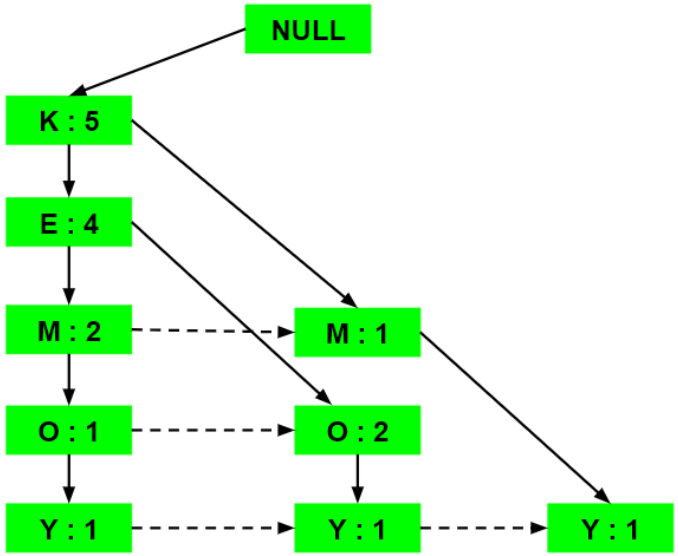
Mining Frequent Patterns using FP-Tree

- Input:
  - FP-tree constructed earlier
  - minimum support threshold
- Output:
  - The complete set of frequent patterns
- Main algorithm:
  - Call FP-growth(*FP-tree*, *null*)

FP-Growth algorithm

- **Step 1**: 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.

Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	



FP-Growth algorithm

- **Step 2**: for each item the **Conditional Frequent Pattern Tree is built.** It is done by accumulating the count of the items in the conditional pattern base. Infrequent items are dropped.

FP-Growth algorithm

- **Step 2**:

items	Conditional pattern base	Conditional frequent pattern tree
Y	{K,E,M,O:1}, {K,E,O:1}, {K,M:1}	{K:3,~~E2:M:2,O:2~~}, {K:3,~~E:2,O:2~~}, {K:3,~~M2~~}
O	{K,E,M:1}, {K,E:2}	{K:3, E:3, ~~M:1~~}, {K:3,E:3}
M	{K:E:2}, {K:1}	{K:3, ~~E2~~}, {k:3}
E	{K:4}	{K:4}
K		

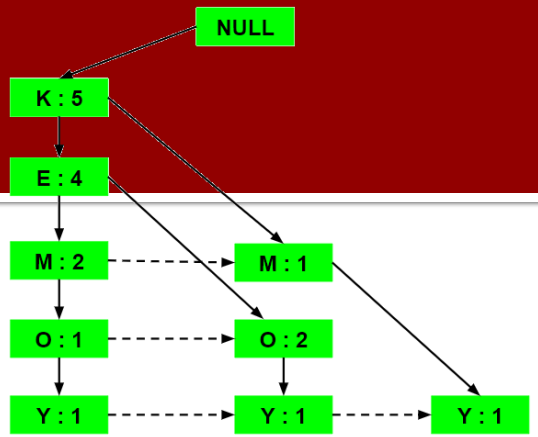
FP-Growth algorithm

- **Step 3:** From the Conditional Frequent Pattern tree, the **Frequent Pattern rules** are generated by pairing the items of the Conditional Frequent Pattern Tree set to the corresponding item. (Add the 1-length frequent itemsets to the set of frequent patterns)

Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

Frequent Pattern Generated
{<K,Y : 3>}
{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
{<K,M : 3>}
{<E,K : 3>}

FP-Growth algorithm

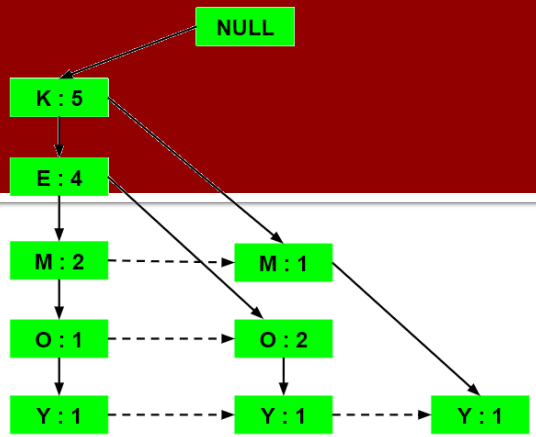


- **Step 4:** generate rules
  - Suppose minConf = 0.8

- Possible rules for first row:
  - K → Y : conf=sup(K→Y)/sup(k) = 3/5 = 0.6
  - Y → K: conf = sup(Y→K)/sup(Y) = 3/3 = 1

Frequent Pattern Generated
{<K,Y : 3>}
{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
{<K,M : 3>}
{<E,K : 3>}

FP-Growth algorithm



- **Step 3:** Suppose minConf = 0.8
- Possible rules for second row:
  - K→ O: conf = 3/5 = 0.6
  - O → K: conf = 3/3 = 1
  - E → O: conf = ¾ = 0.75
  - O → E: conf = 3/3 = 1
  - E, K → O: conf = ¾ =0.75
  - K, O → E: conf = 3/3 = 1
  - E, O → K: conf = 3/3 = 1
  - E → K, O: conf = ¾=0.75
  - K → E, O: conf = 3/5=0.6
  - O → E, K: conf = 3/3=1

Frequent Pattern Generated
{<K,Y : 3>}
{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
{<K,M : 3>}
{<E,K : 3>}

- Try the same example with Apriori, you should get the same result

Final notes on rules evaluation metrics

$$\text{support}(X \rightarrow Y) = \text{support}(X \cup Y)$$

How frequent is this combination ?

$$\text{confidence}(X \rightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)}$$

How often is this combination true ? Or, how likely is it that Y happens when X happens.

Possible extensions: interest measures

- **Lift (Correlation)**

$$\text{lift}(X \rightarrow Y) = \frac{\text{support}(X \rightarrow Y)}{\text{support}(X) \times \text{support}(Y)} = \frac{\text{confidence}(X \rightarrow Y)}{\text{support}(Y)} \quad \text{range: } [0, \infty]$$

- Lift measures how often the antecedent and consequent of a rule X → Y occur together than we would expect if they were statistically independent
- **Lift ~= 1** indicates that the occurrence of Y has **almost no effect** on the occurrence of X
- **lift > 1** → The higher this value, the more likely that the existence of X and Y together in a transaction is not just a random occurrence, but because of some **relationship between them**.
- **lift < 1** → X and Y **appear less often together** than expected

Possible extensions: interest measures

- **Leverage**

$$\text{levarage}(A \rightarrow C) = \text{support}(A \rightarrow C) - \text{support}(A) \times \text{support}(C)$$

$$\text{range: } [-1, 1]$$

  - Similar to lift
  - measure the relation between the probability of a given rule to occur (support(A→C)) and its expected probability if the items were independent (support(A)*support(C)) of each other.
  - The only difference is that lift computes the ratio and leverage computes the difference
  - lift may find very strong associations for less frequent items, while leverage tends to prioritize items with higher frequencies/support in the dataset.
  - **A leverage value of 0 indicates independence.**
  - **values above 0 are desirable**

Possible extensions: interest measures

- **Conviction**

$$\text{conviction}(A \to C) = \frac{1 - \text{support}(C)}{1 - \text{confidence}(A \to C)}, \quad \text{range: } [0, \infty]$$

- Can be interpreted as the ratio of the expected frequency that A occurs without C
- For instance, in the case of a perfect confidence score, the denominator becomes 0 (due to 1 - 1) for which the conviction score is defined as 'inf'.
- Similar to lift, if items are independent, the conviction is 1.

Example

- **Rules with minSup = 0.5 and minConf = 0.8**
  - Y → K: conf = 1

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

- Lift(Y → K) = conf(Y → K) / sup(K) = 1/1 = 1
  - Occurrence of K has no effect on the occurrence of Y
- Leverage(Y → K) = sup(Y→K) – sup(Y)*sup(K) = 3/5 – 3/5*5/5 = 0
  - Similar observation as lift
- Conviction(Y→K) = (1-sup(K)) / (1-conf(Y→K)) = (1- 1)/(1-1) ~ inf
  - k is highly dependent on Y (Y does not occur without K)

Example

- **Rules with minSup = 0.5 and minConf = 0.8**
  - O→E: conf = 1

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

- Lift(O → E) = conf(O → E) / sup(E) = 1/(4/5)) = 1.25
  - >1 → O and E are correlated
- Leverage(O → E) = sup(O→E) – sup(O)*sup(E) = 3/5 - 3/5*4/5 = 0.12
  - > 0 → similar observation as lift
- Conviction(O → E) = (1-sup(E)) / (1-conf(O → E)) = (1- 4/5)/(1-1) ~ inf
  - E is highly dependent on O (O does not occur without E)

Possible extensions

- **Many possible extensions:**
  - Association rules with intervals:
    - For example: Men over 65 have 2 cars
  - Association rules when items are in a taxonomy
    - Bread, Butter $\longrightarrow$ FruitJam
    - BakedGoods, MilkProduct $\longrightarrow$ PreservedGoods
  - Mining rare association rules
  - Sequential Pattern Mining
  - Mining negative association rules
    - in addition consider negated items (i.e. absent from transactions)