Principles of Knowledge Discovery in Data

Fall 2002

<u>Chapter 6</u>: Mining Association Rules

Dr. Osmar R. Zaïane

University of Alberta

Summary of Last Chapter

- What are summarization and generalization?
- What are the methods for descriptive data mining?
- What is the difference with OLAP?
- Can we discriminate between data classes?

Course Content

- Introduction to Data Mining
- Data warehousing and OLAP
- Data cleaning
- Data mining operations
- Data summarization
- Association analysis
- Classification and prediction
- Clustering
- Web Mining
- Spatial and Multimedia Data Mining
- *Other topics if time permits*

Chapter 6 Objectives

Understand association analysis in large datasets and get a brief introduction to the different types of association rule mining

Association Rules Outline

- What is association rule mining?
- How do we mine single-dimensional boolean associations?
- How do we mine multilevel associations?
- How do we mine multidimensional associations?
- Can we constrain the association mining?
- How do we get itemsets without candidate generation?

What Is Association Mining?

- **Association rule mining searches for relationships between items in a dataset**:
  - Finding association, correlation, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
  - Rule form: "**Body → Head [support, confidence]**".

- **Examples:**
  - buys(x, "bread") → buys(x, "milk") [0.6%, 65%]
  - major(x, "CS") ^ takes(x, "DB") → grade(x, "A") [1%, 75%]

Association Rule Mining

mining association rules (Agrawal et. al SIGMOD93)	**Fast algorithm (Agrawal et. al VLDB94)**	**Partitioning (Navathe et. al VLDB95)**
Hash-based (Park et. al SIGMOD95)	**Multilevel A.R. (Han et. al. VLDB95)**	**Generalized A.R. (Srikant et. Al. VLDB95)**
Quantitative A.R. (Srikant et. al SIGMOD96)	**Incremental mining (Cheung et. al ICDE96)**	**Parallel mining (Agrawal et. al TKDE96)**
Distributed mining (Cheung et. al PDIS96)	**Meta-ruleguided mining (Kamber et al. KDD97)**	**Direct Itemset Counting (Brin et. al SIGMOD97)**
N-dimensional A.R. (Lu et. al DMKD'98)	**Constraint A.R. (Ng et. al SIGMOD'98)**	**A.R. with recurrent items (Zaïane et. al ICDE'00)**
FP without Candidate gen. (Han et. al SIGMOD'00)		

And many many others: **Spatial AR; Sequence Associations;AR for multimedia; AR in time series;AR with progressive refinement;** etc.

Basic Concepts

A transaction is a set of items: $T=\{i_a, i_b,\ldots i_t\}$

$T \subset I$, where I is the set of all possible items $\{i_1, i_2,\ldots i_n\}$

D, the task relevant data, is a set of transactions.

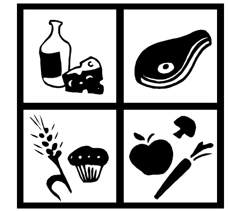
An association rule is of the form:
$P \rightarrow Q$, where $P \subset I$, $Q \subset I$, and $P \cap Q = \emptyset$

Basic Concepts (con't)

P➔Q holds in *D* with <u>support</u> s
and
P➔Q has a <u>confidence</u> c in the transaction set *D*.

Support(P➔Q) = Probability(P∪Q)
Confidence(P➔Q)=Probability(Q/P)

Itemsets



A set of items is referred to as <u>itemset</u>.

An itemset containing k items is called **k-itemset**.

An items set can also be seen as a conjunction of items (or a predicate)

Support and Confidence

- **Support** of $P = P_1 \wedge P_2 \wedge \ldots \wedge P_n$ in ***D***
  - σ(P/*D*) is the percentage of transactions T in *D* satisfying P. (number of T by cardinality of *D*).
- **Confidence** of a rule P $\rightarrow$ Q
  - φ(P $\rightarrow$ Q/ *D*) ratio σ((P $\wedge$ Q)/ *D*) by σ(P/ *D*)
- Thresholds:
  - *minimum support* σ'
  - *minimum confidence* φ'

Strong Rules

- **Frequent (or large) predicate** P in set D
  - support of P larger than minimum support,
- Rule P $\rightarrow$ Q (c%) is **strong**
  - predicate (P $\wedge$ Q) is frequent (or large),
  - c is larger than minimum confidence.

Different Kinds of Association Rules

- **Boolean vs. Quantitative Associations**
  - Association on discrete vs. continuous data
  - Ex. Age(X,30-45) ∧ Income(X, 50K-75K) ➔ Buys(X, SUVcar)

➢ **Boolean Association Rules**

➢ **Quantitative Association Rules**

Different Kinds of Association Rules

- **Single dimension vs. multiple dimensional associations**
  - Based on the dimensions in data involved.
  - One predicate then single dimension. More predicates then multi-dimensions.
  - Ex. Buys(X, bread) ➔ Buys(X, milk)

    Age(X,30-45) ∧ Income(X, 50K-75K) ➔ Buys(X, SUVcar)

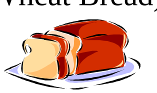
➢ **Single-dimensional Association Rules**

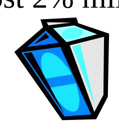
➢ **Multi-dimensional Association Rules**

Different Kinds of Association Rules

- **Single level vs. multiple-level analysis**
  - Based on the level of abstractions involved.
  - Find association rules at different levels of abstraction.
  - Ex. Buys(X, bread) ➔ Buys(X, milk)

    Buys(X, Wheat Bread) ➔ Buys(X, Formost 2% milk)





➢ **Single-level Association Rules**

➢ **Multi-level Association Rules**

Different Kinds of Association Rules

- **Single occurrence vs. multiple occurrences**
  - One item may occur more than once in the transaction.
  - Not the presence of the item is important but its frequency.
  - Ex. Buys(X, bread, 2) ➔ Buys(X, milk, 1)

➢ **Single-occurrence-items Association Rules**

➢ **Recurrent-items Association Rules**

Different Kinds of Association Rules

- **Simple vs. constraint-based**
  - Constraints can be added on the rules to be discovered
- **Association vs. correlation analysis**
  - Association does not necessarily imply correlation.

$$\frac{P(A \wedge B)}{P(A)P(B)} \quad =1? \quad >1? \quad <1?$$

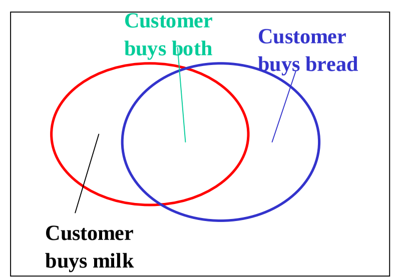
Association Rules Outline

- What is association rule mining?
- How do we mine single-dimensional boolean associations?
- How do we mine multilevel associations?
- How do we mine multidimensional associations?
- Can we constrain the association mining?
- How do we get itemsets without candidate generation?

How do we Mine Association Rules?

- **Input**
  - A database of transactions
  - Each transaction is a list of items (Ex. purchased by a customer in a visit)
- Find all rules that associate the presence of one set of items with that of another set of items.
  - Example: *98% of people who purchase tires and auto accessories also get automotive services done*
  - There are no restrictions on the number of items in the head or body of the rule.

Rule Measures: Support and Confidence



Find all the rules X & $Y \rightarrow Z$ with minimum confidence and support

- support, ***s***, probability that a transaction contains {X, Y, Z}
- confidence, ***c***, conditional probability that a transaction having {X, Y} also contains *Z*.

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Let minimum support 50%, and minimum confidence 50%, we have

- $A \rightarrow C$ (50%, 66.6%)
- $C \rightarrow A$ (50%, 100%)

Mining Association Rules

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

For rule $A \rightarrow C$:

support = support({A, C}) = 50%

confidence = support({A, C})/support({A}) = 66.6%

The Apriori principle:

Any subset of a frequent itemset must be frequent.

Mining Frequent Itemsets: the Key Step

① Find the *frequent itemsets*: the sets of items that have minimum support

- A subset of a frequent itemset must also be a frequent itemset, i.e., if {AB} is a frequent itemset, both {A} and {B} should be frequent itemsets
- Iteratively find frequent itemsets with cardinality from 1 to k (k-itemsets)

② Use the frequent itemsets to generate association rules.

The Apriori Algorithm

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

L_1 = {frequent items};
for (k = 1; L_k !=∅; 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 $\cup_k L_k$;

The Apriori Algorithm -- Example

Database D

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

C_2

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

Scan D

C_2

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

L_2

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

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Note: {1,2,3}{1,2,5} and {1,3,5} not in C_3

Generating Association Rules from Frequent Itemsets

•Only strong association rules are generated.
•Frequent itemsets satisfy minimum support threshold.
•Strong AR satisfy minimum confidence threshold.

•Confidence(A➔B) = Prob(B/A) = $\frac{\text{Support}(A \cup B)}{\text{Support}(A)}$

For each frequent itemset, **f**, generate all non-empty subsets of **f**.
For every non-empty subset **s** of **f** **do**
    output rule **s➔(f-s)** if support(**f**)/support(**s**) ≥ min_confidence
end

Improving efficiency of Apriori

• **Reducing the number of scans**
(there are k DB scans for k-itemsets)

• **Eliminating scans by indexing** (Hashing)

• **Reducing sizes and number of transactions**
(no need for non frequent items)

• **Partitioning**

Optimization: Direct Hash and Pruning

- DHP: Direct Hash and Pruning (Park, Chen and Yu, SIGMOD'95).
  - Reduce the size of candidate sets to minimize the cost
  - Reduce the size of the transaction database as well
- Using a hash table to keep track the counts of 2-itemset. Using the counts to prune C_2 (C_2 is usually the largest)
- An item in transaction *t* can be trimmed if it does not appear in at least *k* of the candidate *k*-itemsets in *t*.

Optimization: The Partitioning Algorithm

- Partition (Savasere, Omiecinski, & Navathe, VLDB'95).
  - Divide database into *n* partitions.
  - A frequent item must be frequent in at least one partition.
  - Process one partition in main memory at a time:
    - For each partition, generate frequent itemsets using the Apriori algorithm
    - also form *tidlist* for all itemsets to facilitate counting in the merge phase
  - After all partitions are processed, the local frequent itemsets are merged into global frequent sets; support can be computed from the *tidlists*.

Optimization: Sampling and Itemset Counting

- Sampling (Toivonen. VLDB'96).
  - A probabilistic approach finds association rules in about one pass.
- Dynamic Itemset Counting (Brin et. al. SIGMOD'97)
  - Reducing the number of scans over the transactions by starting to count itemsets dynamically during scans
  - Using data structure to keep track of counters and reordering items to reduce increment costs

Incremental Update of Discovered Rules

- Partitioned derivation and incremental updating.
- A fast updating algorithm, FUP (Cheung et al.'96)
  - View a database: original $DB \cup$ incremental db.
  - A k-itemset (for any k),
    - **frequent** in $DB \cup db$ if frequent in both DB and db.
    - **non frequent** in $DB \cup db$ if also in both DB and db.
  - For those only frequent in *DB*, merge corresponding counts in *db*.
  - For those only frequent in *db*, search *DB* to update their itemset counts.
- Similar methods can be adopted for data removal and update.
- Principles applicable to distributed/parallel mining.

Parallel and Distributed Mining

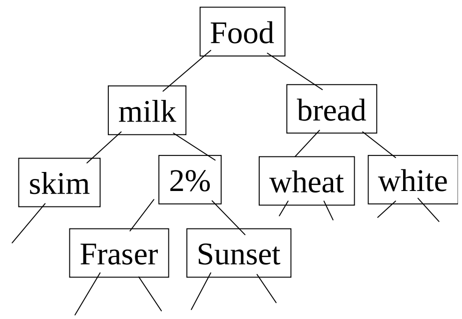
- **PDM** (Park et al.'95):
  - Use a hashing technique (DHP-like) to identify candidate k-itemsets from the local databases.
- **Count Distribution** (Agrawal & Shafer'96):
  - An extension of the Apriori algorithm.
  - May require a lot of messages in count exchange.
- **FDM** (Cheung et al.'96).
  - Observation: If an itemset X is globally large, there exists partition D_i such that X and all its subsets are locally large at D_i.
  - Candidate sets are those which are also local candidates in some component database, plus some message passing optimizations.

Association Rules Outline

- What is association rule mining?
- How do we mine single-dimensional boolean associations?
- How do we mine multilevel associations?
- How do we mine multidimensional associations?
- Can we constrain the association mining?
- How do we get itemsets without candidate generation?

Multiple-Level Association Rules

- Items often form hierarchy.
- Items at the lower level are expected to have lower support.
- Rules regarding itemsets at appropriate levels could be quite useful.
- Transaction database can be encoded based on dimensions and levels
- It is smart to explore shared multi-level mining (Han & Fu,VLDB'95).



TID	Items
T1	{111, 121, 211, 221}
T2	{111, 211, 222, 323}
T3	{112, 122, 221, 411}
T4	{111, 121}
T5	{111, 122, 211, 221, 413}

Mining Multi-Level Associations

- **A top_down, progressive deepening approach.**
  - First find high-level strong rules:
    milk → bread [20%, 60%].
  - Then find their lower-level "weaker" rules:
    2% milk → wheat bread [6%, 50%].
- **Variations at mining multiple-level association rules.**
  - Level-crossed association rules:
    2% *milk* → ***Wonder wheat*** *bread*
  - Association rules with multiple, alternative hierarchies:
    2% *milk* → *Wonder bread*

Multi-Level Mining: Progressive Deepening

- A top-down, progressive deepening approach:
  - First mine high-level frequent items:
    milk (15%), bread (10%)
  - Then mine their lower-level frequent itemsets:
    2% milk (5%), wheat bread (4%)

  When one threshold is set for all levels; if support too high, it is possible to miss meaningful associations at low level; if support too low, it is possible to generate uninteresting rules
- Different minimum support threshold across multi-levels lead to different algorithms.

Approaches to Mining Multi-level Association Rules

- Uniform minimum support for all levels
  - Same support σ for all levels
  - Avoid examining itemsets containing items whose ancestor is not frequent.
  - Simpler, but it is unlikely that lower level items are as frequent as higher level items.

Approaches to Mining Multi-level Association Rules

- Reduced minimum support at lower levels

  Examine only those descendents whose ancestor's support is frequent or non-negligible (controlled).

  – **Level-by-level independent**

  Full depth search

  – **Level-cross filtering by single item**

  A specific association is examined from a more general one => items are examined only if parents are frequent.

  – **Level-cross filtering by k-itemsets**

  Frequency of ancestry examined for k-itemsets and not just items

Association Rules Outline

- What is association rule mining?
- How do we mine single-dimensional boolean associations?
- How do we mine multilevel associations?
- How do we mine multidimensional associations?
- Can we constrain the association mining?
- How do we get itemsets without candidate generation?

Mining Multi-Dimensional Associations

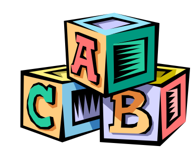
- **Multi-dimensional vs. transaction-based associations**
  - Multi-dimensional (linking different attributes)
    - major(x, "cs") ^ region(x, "oxford") → gpa(x,"good").
  - Transaction-based (linking the same kind of attributes)
    - takes(x, "chemistry") ^ takes (x, "biology") → takes(x, "bio-chemistry").
- **Multi-level association (drilling on any dimension)**
  - Lower levels often adopt lower *min_support* thresholds.
- **Method**:
  - Construct data cube (with count/frequency aggregated)
  - Perform level-wise/dimension-wise search in the data cube (Kamber et al., KDD'97).

Categorical and Quantitative

In a multidimensional context there are:

•Categorical dimensions (attributes)

•Ex. Occupation, Location, etc.



•Quantitative dimensions (attributes)

•Ex. Price, Age, etc.

Apriori, as it is, does not handle quantitative data.

Quantitative Association Rules

RecordID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	Yes	2

Sample Rules	Support	Confidence
<age:30..39> and <married: yes> ==> <numCars:2>	40%	100%
<NumCars: 0..1> ==> <Married: No>	40%	66.70%

Mapping Quantitative to Boolean

- One possible solution is to map the problem to the Boolean association rules:
  - discretizing a non-categorical attribute to intervals
    - Age [20,29], [30,39],...
  - forming Boolean records
    - categorical attributes: each value becomes one item
    - non-categorical attributes: each interval becomes one item

RecordID	Age	Married	NoCars
100	23	No	1
500	38	Yes	2

RecID	Age: 20..29	Age: 30..39	Married: Yes	Married: No	Cars: 0	Cars: 1	Cars: 2
100	1	0	0	1	0	1	0
500	0	1	1	0	0	0	1

Mining Quantitative Association Rules

- Problems with the mapping
  - too few intervals: lost information
  - too low support: too many rules
- Solutions
  - using the supports of an itemset and its generalizations to determine the intervals
  - Binning (equi-width,equi-dept,distance based)
  - using interest measure to control the number of association rules

Association Rules Outline

- What is association rule mining?
- How do we mine single-dimensional boolean associations?
- How do we mine multilevel associations?
- How do we mine multidimensional associations?
- Can we constrain the association mining?
- How do we get itemsets without candidate generation?

Restricting Association Rules

•Useful for interactive and ad-hoc mining
•Reduces the the set of association rules discovered and confines them to more relevant rules.

• **Before mining**
✓ Knowledge type constraints: classification, etc.
✓ Data constraints: SQL-like queries (DMQL)
✓ Dimension/level constraints: relevance to some dimensions and some concept levels.

• **While mining**
✓ Rule constraints: form, size, and content.
✓ Interestingness constraints: support, confidence, correlation.

• **After mining**
✓ Querying association rules

Rule Constraints in Association Mining

- Two kind of rule constraints:
  - Rule form constraints: meta-rule guided mining.
    - P(x, y) ^ Q(x, w) → takes(x, "database systems").
  - Rule content constraint: constraint-based query optimization (where and having clauses)(Ng, et al., SIGMOD'98).
    - sum(LHS) < 100 ^ min(LHS) > 20 ^ count(LHS) > 3 ^ sum(RHS) > 1000
- **1-variable vs. 2-variable constraints** (Lakshmanan, et al. SIGMOD'99):
  - 1-var: A constraint confining only one side (L/R) of the rule, e.g., as shown above.
  - 2-var: A constraint confining both sides (L and R).
    - sum(LHS) < min(RHS) ^ max(RHS) < 5* sum(LHS)

Constrained Association Query Optimization Problem

- Given a set of constraints C, the algorithm should:
  - Find only the frequent sets that satisfy the given constraints C
  - Find all frequent sets that satisfy the given constraints C
- A naïve solution:
  - Apply Apriori for finding all frequent sets, and then test them for constraint satisfaction one by one.
- Better approach:
  - Comprehensive analysis of the properties of constraints and try to **push them as deeply as possible inside** the frequent set computation.

Association Rules Outline

- What is association rule mining?
- How do we mine single-dimensional boolean associations?
- How do we mine multilevel associations?
- How do we mine multidimensional associations?
- Can we constrain the association mining?
- How do we get itemsets without candidate generation?

Problems with Apriori

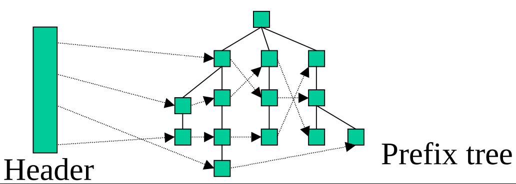
- Generation of candidate itemsets are expensive (Huge candidate sets)
  - 10^4 frequent 1-itemset will generate 10^7 candidate 2-itemsets
  - To discover a frequent pattern of size 100, e.g., $\{a_1, a_2, \ldots, a_{100}\}$, one needs to generate $2^{100} \approx 10^{30}$ candidates.
- High number of data scans

Frequent Pattern Growth

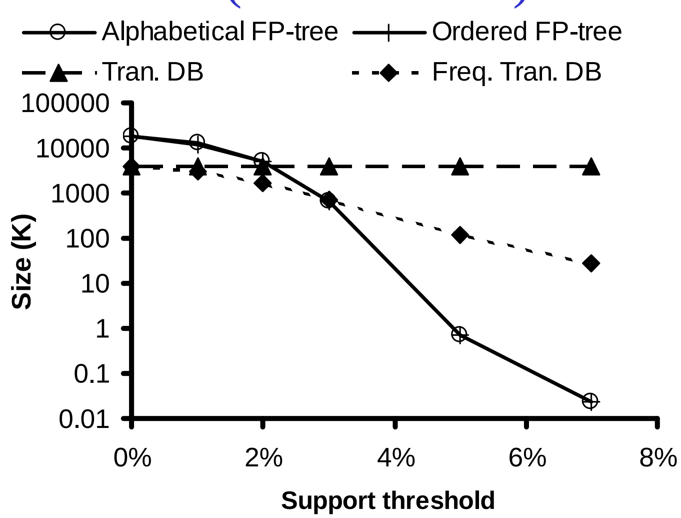
- First algorithm that allows frequent pattern mining without generating candidate sets
- Requires Frequent Pattern Tree

Frequent Pattern Tree

- Prefix tree
- Each node contains the item name, frequency and pointer to another node of the same kind
- Frequent item header that contains item names and pointer to the first node in FP tree



Database Compression Using FP-tree (T10I4D100k)



Building the Frequent Pattern Tree

- Scan the data set to generate the frequency list
- Frequency list is sorted in descending order
- Only items with frequency greater than required support are being kept

Frequent Pattern Tree

F, A, C, D, G, I, M, P
A, B, C, F, L, M, O
B, F, H, J, O
B, C, K, S, P
A, F, C, E, L, P, M, N

Required Support: **3**

F:4, C:4, A:3, B:3, M:3, P:3 ~~D:1 E:1 G:1 H:1 I:1 J:1 K:1 L:1 O:1~~

Frequent Pattern Tree

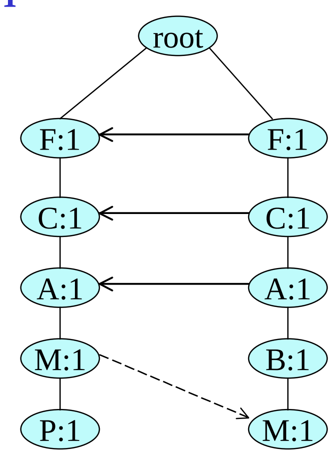
Original Transaction	Ordered frequent items
F, A, C, D, G, I, M, P	F, C, A, M, P
A, B, C, F, L, M, O	F, C, A, B, M
B, F, H, J, O	F, B
B, C, K, S, P	C, B, P
A, F, C, E, L, P, M, N	F, C, A, M, P

Required Support: **3**

F:4, C:4, A:3, B:3, M:3, P:3

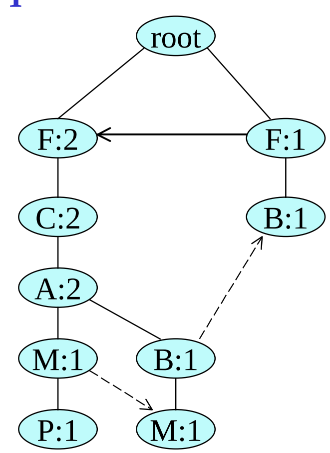
F, C, A, M, P
F, C, A, B, M
F, B
C, B, P
F, C, A, M, P

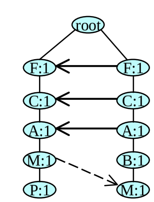
Frequent Pattern Tree



F, C, A, M, P
F, C, A, B, M
F, B
C, B, P
F, C, A, M, P

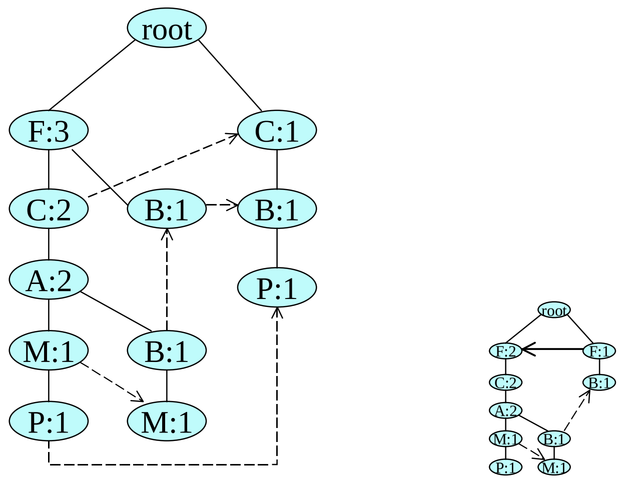
Frequent Pattern Tree





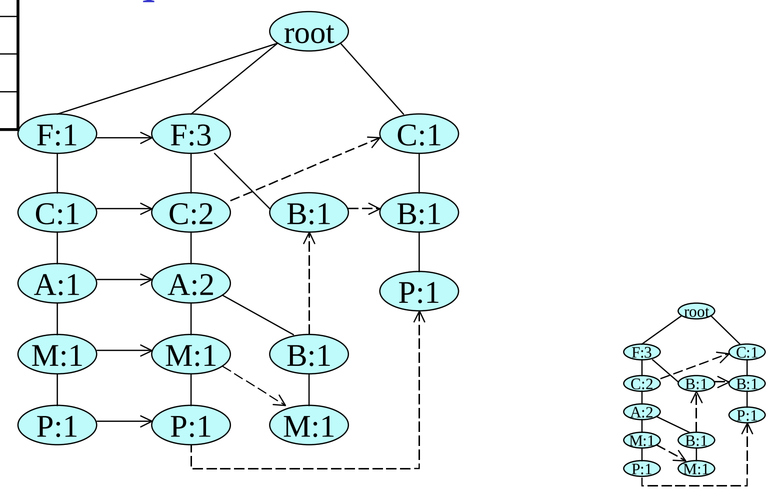
Frequent Pattern Tree

F, C, A, M, P
F, C, A, B, M
F, B
C, B, P
F, C, A, M, P



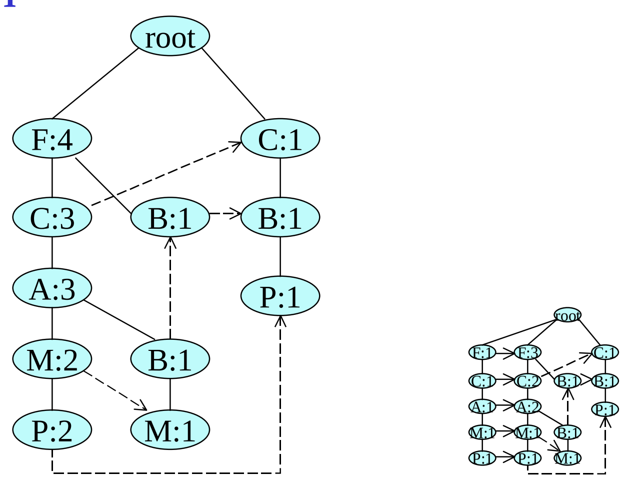
Frequent Pattern Tree

F, C, A, M, P
F, C, A, B, M
F, B
C, B, P
F, C, A, M, P

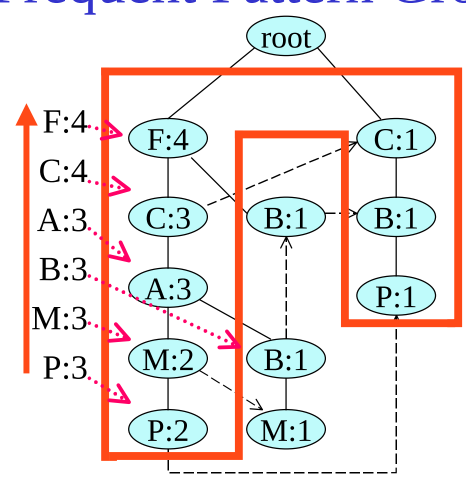


Frequent Pattern Tree

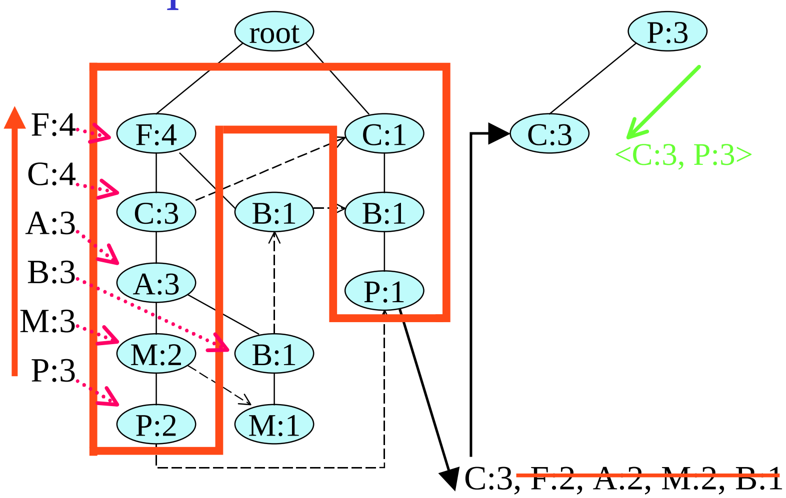
F, C, A, M, P
F, C, A, B, M
F, B
C, B, P
F, C, A, M, P



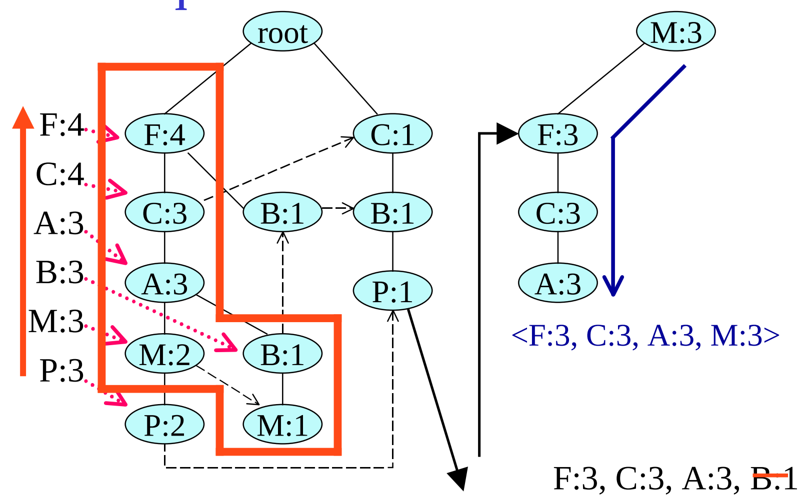
Frequent Pattern Growth



Frequent Pattern Growth



Frequent Pattern Growth

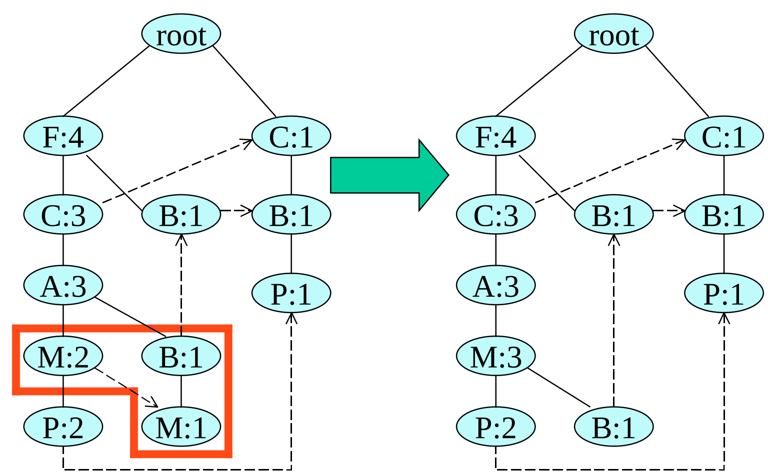


Frequent Pattern Growth

- Is FP growth perfect?
- Not Yet
  – No incremental update
  – Required rebuild of the FP tree when support is decreased

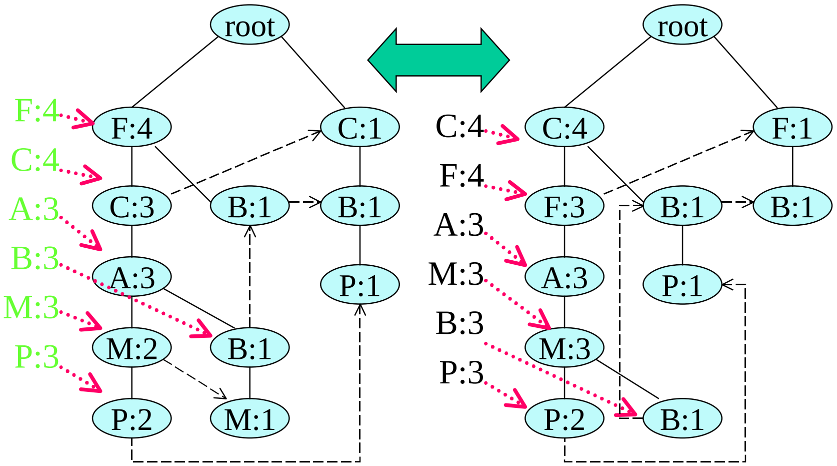
Frequent Pattern Tree/Growth

- FP Tree is not locally optimized



Frequent Pattern Tree/Growth

- FP Tree is sensitive to header order



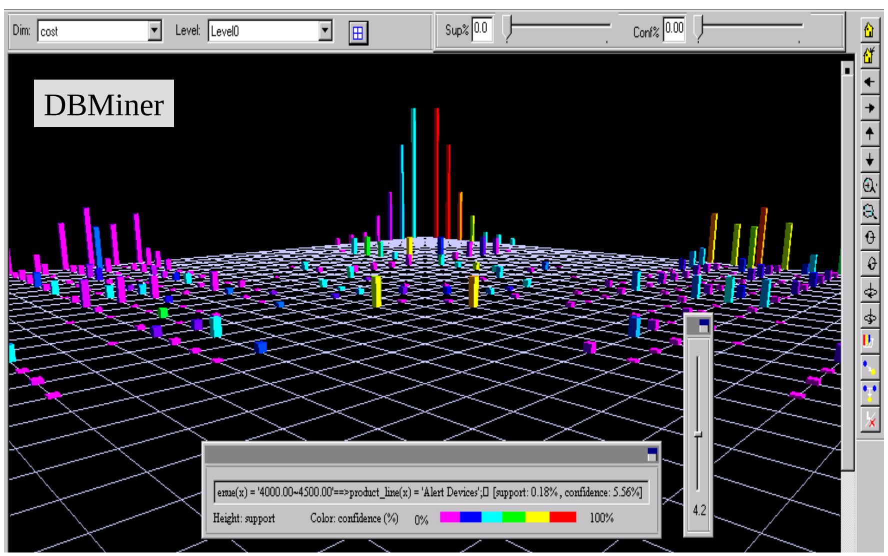
Presentation of Association Rules (Table Form)

	Body	Implies	Head	Supp (%)	Conf (%)	F	G	H	I
1	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00'	28.45	40.4				
2	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00'	20.46	29.05				
3	cost(x) = '0.00~1000.00'	==>	order_qty(x) = '0.00~100.00'	59.17	84.04				
4	cost(x) = '0.00~1000.00'	==>	revenue(x) = '1000.00~1500.00'	10.45	14.84				
5	cost(x) = '0.00~1000.00'	==>	region(x) = 'United States'	22.56	32.04				
6	cost(x) = '1000.00~2000.00'	==>	order_qty(x) = '0.00~100.00'	12.91	69.34				
7	order_qty(x) = '0.00~100.00'	==>	revenue(x) = '0.00~500.00'	28.45	34.54				
8	order_qty(x) = '0.00~100.00'	==>	cost(x) = '1000.00~2000.00'	12.91	15.67				
9	order_qty(x) = '0.00~100.00'	==>	region(x) = 'United States'	25.9	31.45				
10	order_qty(x) = '0.00~100.00'	==>	cost(x) = '0.00~1000.00'	59.17	71.86				
11	order_qty(x) = '0.00~100.00'	==>	product_line(x) = 'Tents'	13.52	16.42				
12	order_qty(x) = '0.00~100.00'	==>	revenue(x) = '500.00~1000.00'	19.67	23.88				
13	product_line(x) = 'Tents'	==>	order_qty(x) = '0.00~100.00'	13.52	98.72				
14	region(x) = 'United States'	==>	order_qty(x) = '0.00~100.00'	25.9	81.94				
15	region(x) = 'United States'	==>	cost(x) = '0.00~1000.00'	22.56	71.39				
16	revenue(x) = '0.00~500.00'	==>	cost(x) = '0.00~1000.00'	28.45	100				
17	revenue(x) = '0.00~500.00'	==>	order_qty(x) = '0.00~100.00'	28.45	100				
18	revenue(x) = '1000.00~1500.00'	==>	cost(x) = '0.00~1000.00'	10.45	96.75				
19	revenue(x) = '500.00~1000.00'	==>	cost(x) = '0.00~1000.00'	20.46	100				
20	revenue(x) = '500.00~1000.00'	==>	order_qty(x) = '0.00~100.00'	19.67	96.14				
21									
22									
23	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.45	40.4				
24	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.45	40.4				
25	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	19.67	27.93				
26	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	19.67	27.93				
27	cost(x) = '0.00~1000.00' AND order_qty(x) = '0.00~100.00'	==>	revenue(x) = '500.00~1000.00'	19.67	33.23				

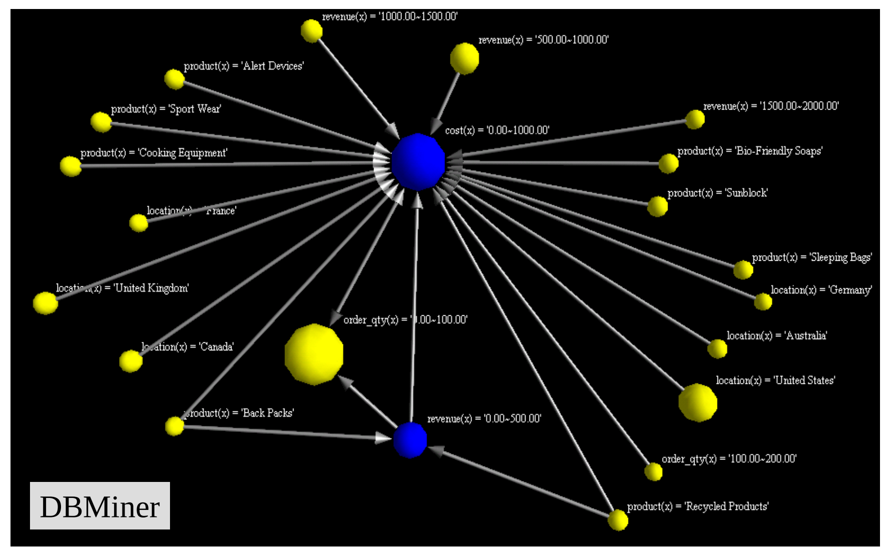
Sheet1

DBMiner

Visualization of Association Rule in Plane Form



Visualization of Association Rule Using Rule Graph



Visualization of Association Rule Using Table Graph (DBMiner Web version)