

MaxMiner: Mining Max-patterns

- Review!
 - An itemset X is a **max-pattern** if X is frequent and there exists no frequent super-pattern $Y \supset X$
 - i.e., no such a Y
 - Y is a super-pattern of X
 - The support of Y is greater than minSup
 - The support of Y can be smaller than that of X
- MaxMiner is based on the Apriori algorithm
- R. Bayardo. Efficiently mining long patterns from databases. In *SIGMOD'98*



MaxMiner: Mining Max-patterns

- 1st scan: find frequent items and **sort** them (**ascending order**)
 - A, B, C, D, E (E is most frequently occurring)
- 2nd scan: find support for 2-itemsets with max-patterns
 - **AB**, AC, AD, AE, **ABCDE** ←
 - **BC**, BD, BE, **BCDE** ←
 - **CD**, CE, **CDE** ←
 - **DE**

Potential max-patterns

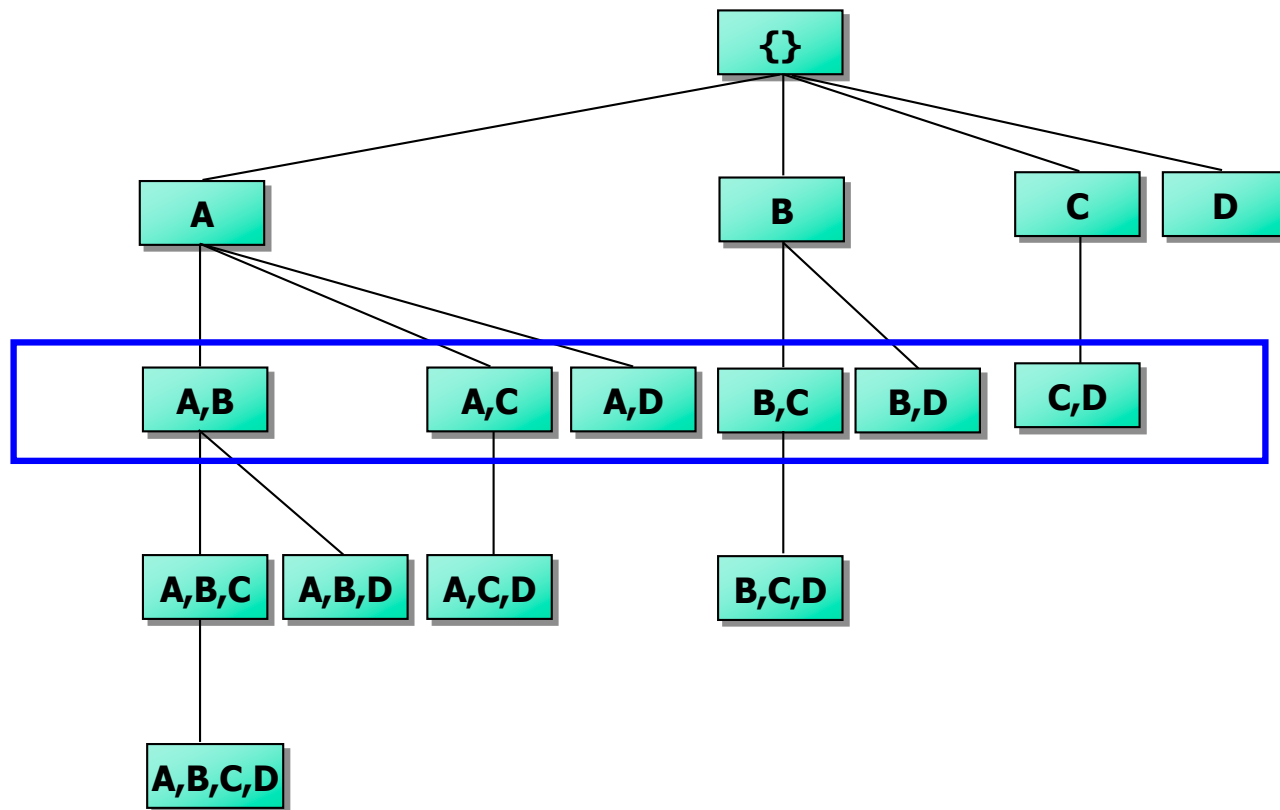
Tid	Items
10	A,B,C,D,E
20	B,C,D,E,
30	A,C,D,F,E

- Reduce a lot of candidates in later stages
 - Since BCDE is a max-pattern, no need to check BCD, BDE, CDE in later scan
 - If AC is infrequent, no need to check ABC in later scans



MaxMiner: Mining Max-patterns

Complete *set-enumeration tree* over four items



Mining Closed Patterns: CLOSET

- Review
 - An itemset X is **closed** if X is *frequent* and there exists *no super-pattern* $Y \supset X$, *with the same support as X*
 - i.e., no such a Y
 - Y is a super-pattern of X
 - The support of Y is should be the same as that of X



Mining Closed Patterns: CLOSET

- Use the FP-tree for finding frequent patterns
- Flist: list of all frequent items in support ascending order
 - Flist: c-e-f-a-d
- Divide search space
 - Patterns having d
 - Patterns having a but no d
 - Patterns having f but no d and a
 - Patterns having e but no d, a, and f
 - Patterns having c but no d, a, f, and e

Min_sup=2

TID	Items
10	a, c, d, e, f
20	a, b, e
30	c, e, f
40	a, c, d, f
50	c, e, f



Mining Closed Patterns: CLOSET

- Naïve approach: Quite costly!
 - To mine a complete set of all frequent itemsets
 - To remove every frequent itemset whose support is the same as that of its superset
- Find only the closed itemsets recursively in an efficient way during the mining process using the FP-tree
 - Key idea: every transaction having d also has $cfa \Rightarrow cfad$ is a frequent closed pattern
 - You can consider details by referring to FP-Growth
- J. Pei, J. Han & R. Mao. CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", DMKD'00



CHARM: Mining by Exploring Vertical Data Format

- Vertical format: $t(AB) = \{T_{11}, T_{25}, \dots\}$
 - tid-list: list of trans.-ids containing an itemset
- Algorithm
 - Transform a horizontally formatted data to a vertically format by scanning the dataset once
 - Easy: # of items is much smaller than that of transactions
 - Starting with $k=1$, construct candidate $(k+1)$ -itemsets from frequent k -itemsets
 - Using the TID-sets intersection and Apriori property
 - Repeat this process with k incremented by 1 until no frequent itemsets can be found



CHARM: Mining by Exploring Vertical Data Format

Table 5.3 The vertical data format of the transaction data set D of Table 5.1.

<i>itemset</i>	<i>TID_set</i>
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}



CHARM: Mining by Exploring Vertical Data Format

Table 5.4 The 2-itemsets in vertical data format.

itemset	TID_set
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}

Table 5.5 The 3-itemsets in vertical data format.

itemset	TID_set
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

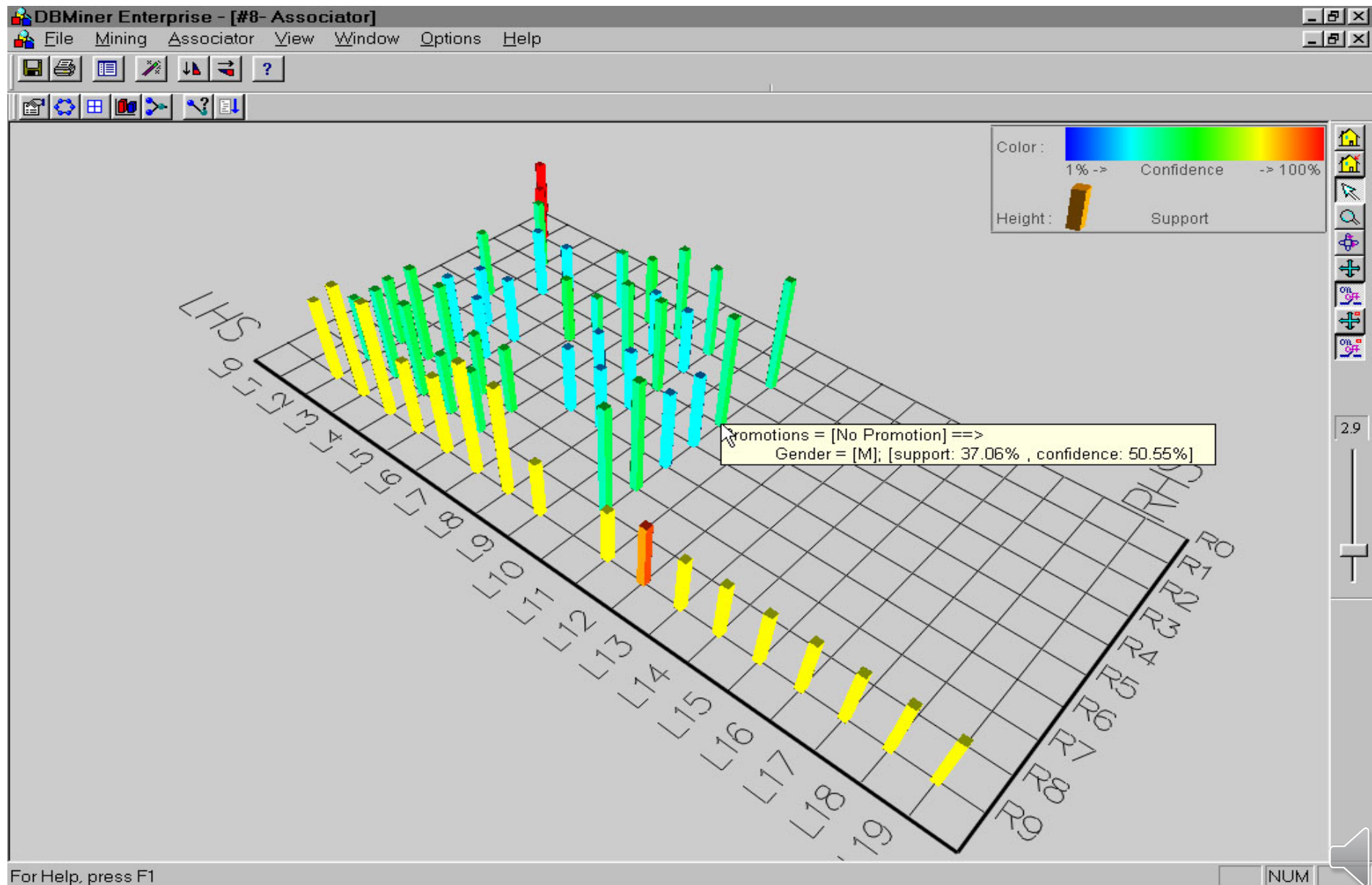


CHARM: Mining by Exploring Vertical Data Format

- No need to scan a database to find the support of $(k+1)$ itemsets
 - TID set of each k -itemset carries sufficient information including a support value
 - But, it is quite long and requires large space for intersection
- Diffset technique
 - Keep track of only the difference of TID sets for $(k+1)$ -itemset and its corresponding k -itemset
 - $\{I1\} = \{t1, t4, t5, t7, t8, t9\}$, $\{I1, I2\} = \{t1, t4, t8, t9\}$
 - Store $\text{Diffset}(\{I1, I2\}, \{I1\}) = \{t5, t7\}$ instead of storing $\{I1, I2\}$
 - Effective with many dense and long patterns

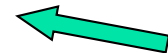


Visualization of Association Rules: Plane Graph



Chapter 5: Mining Frequent Patterns, Association and Correlations

- Basic concepts and a road map
- Efficient and scalable frequent itemset mining methods
- Mining various kinds of association rules
- From association mining to correlation analysis
- Constraint-based association mining
- Summary



Mining Various Kinds of Association Rules

- Mining multilevel association
- Mining multidimensional association
- Mining quantitative association
- Mining interesting correlation patterns



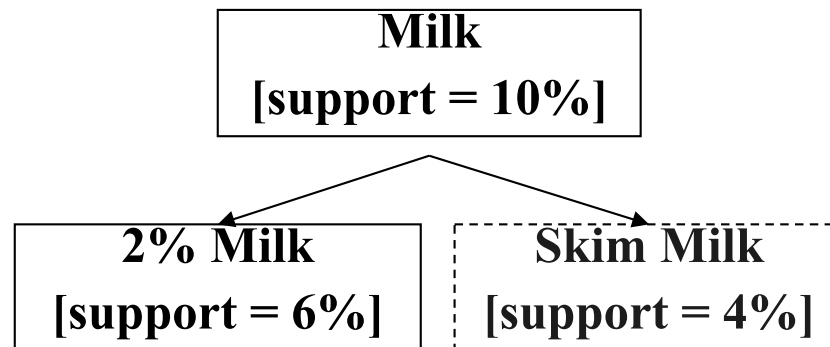
Mining Multiple-Level Association Rules

- Items often form hierarchies
- Flexible support settings
 - Items at the lower level are expected to have lower support
- Exploration of *shared* multi-level mining (Agrawal & Srikant@VLDB'95, Han & Fu@VLDB'95)

uniform support

Level 1
min_sup = 5%

Level 2
min_sup = 5%



reduced support

Level 1
min_sup = 5%

Level 2
min_sup = 3%



Multi-level Association: Redundancy Filtering

- Some rules may be redundant due to “ancestor” relationships between items.
- Example
 - milk \Rightarrow wheat bread [support = 8%, confidence = 70%]
 - 2% milk \Rightarrow wheat bread [support = 2%, confidence = 72%]
- We say the first rule is an ancestor of the second rule.
- A (descendent) rule is redundant if
 - Its support is close to the *“expected” value*, based on the rule’s ancestor
 - Its confidence is close to that of the rule’s ancestor



Mining Multi-Dimensional Association

- Single-dimensional rules: (having a dimension or a predicate)

$\text{buys}(X, \text{"milk"}) \Rightarrow \text{buys}(X, \text{"bread"})$: milk \Rightarrow bread

- Multi-dimensional rules: ≥ 2 dimensions or predicates

- Inter-dimension assoc. rules (*no repeated predicates*)

$\text{age}(X, \text{"19-25"}) \wedge \text{occupation}(X, \text{"student"}) \Rightarrow \text{buys}(X, \text{"coke"})$

- hybrid-dimension assoc. rules (*repeated predicates*)

$\text{age}(X, \text{"19-25"}) \wedge \text{buys}(X, \text{"popcorn"}) \Rightarrow \text{buys}(X, \text{"coke"})$



Attribute Types

- $\text{age}(X, "19-25") \wedge \text{occupation}(X, "student") \Rightarrow \text{buys}(X, "coke")$
attributes
- Categorical Attributes
 - Finite number of possible values, no ordering among values
- Quantitative Attributes
 - Numeric, implicit ordering among values
 - Discretization and clustering approaches required



Mining Quantitative Associations

- Techniques can be categorized by how numerical attributes, such as **age** or **salary** are treated
 1. Static discretization based on predefined concept hierarchies (data cube methods)
 2. Dynamic discretization based on data distribution (quantitative rules, e.g., Agrawal & Srikant@SIGMOD96)
 3. Clustering: Distance-based association (e.g., Yang & Miller@SIGMOD97)



Static Discretization of Quantitative Attributes

- Discretized prior to mining using a concept hierarchy
 - $\text{age}(X, "19-25") \wedge \text{occupation}(X, "student") \Rightarrow \text{buys}(X, "coke")$
 - Numeric values are replaced by ranges (as a categorical value)
- In a relational database, finding all frequent k -predicate sets will require k or $k+1$ table scans.



Quantitative Association Rules

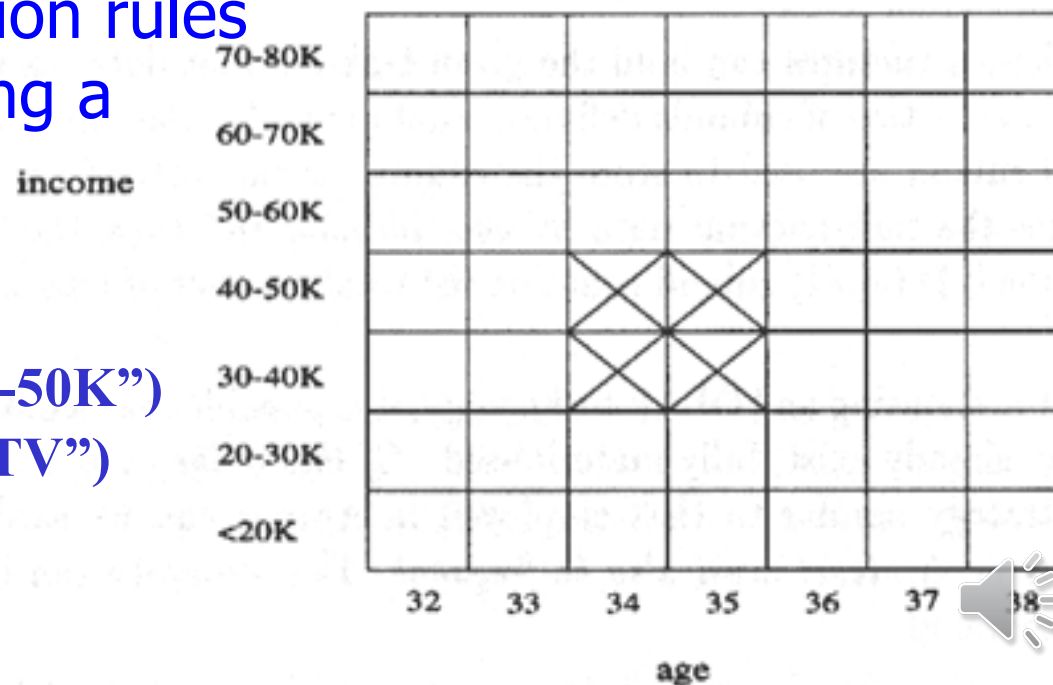
- Proposed by Lent, Swami and Widom ICDE'97
- Numeric attributes are *dynamically* discretized
- 2-D quantitative association rules: $A_{\text{quan1}} \wedge A_{\text{quan2}} \Rightarrow A_{\text{cat}}$
 - The confidence is higher than threshold
 - The support is higher than threshold

- Cluster *adjacent* association rules to form general rules using a 2-D grid

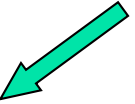
- Example

$\text{age}(X, "34-35") \wedge \text{income}(X, "30-50K")$
 $\Rightarrow \text{buys}(X, "high\ resolution\ TV")$

Note: simplified!



Chapter 5: Mining Frequent Patterns, Association and Correlations

- Basic concepts and a road map
- Efficient and scalable frequent itemset mining methods
- Mining various kinds of association rules
- From association mining to correlation analysis 
- Constraint-based association mining
- Summary



Interestingness Measure: Correlations (Lift)

- *play basketball* \Rightarrow *eat cereal* [40%, 66.7%] is misleading
 - The overall % of students eating cereal is 75% > 66.7%.
- *play basketball* \Rightarrow *not eat cereal* [20%, 33.3%] is more meaningful, although it has lower support and confidence
- Measure of dependent/correlated events: **lift**

Contingency table

$$lift = \frac{P(A \cup B)}{P(A)P(B)}$$

	Basketball	Not basketball	Sum (row)
Cereal	2000	1750	3750
Not cereal	1000	250	1250
Sum(col.)	3000	2000	5000

$$lift(B, C) = \frac{2000 / 5000}{3000 / 5000 * 3750 / 5000} = 0.89 \quad lift(B, \neg C) = \frac{1000 / 5000}{3000 / 5000 * 1250 / 5000} = 1.33$$



Is lift Good Measure of Correlation?

- "*Buy walnuts \Rightarrow buy milk* [1%, 80%]" is misleading
 - if 85% of customers buy milk
- Support and confidence are not good to represent correlations
- So many interestingness measures? (Tan, Kumar, Sritastava @KDD'02)

$$lift = \frac{P(A \cup B)}{P(A)P(B)} \quad \text{Cosine} = \frac{P(A \cup B)}{\sqrt{P(A)P(B)}}$$

$$all_conf = \frac{sup(X)}{max_item_sup(X)}$$

	Milk	No Milk	Sum (row)
Coffee	m, c	~m, c	c
No Coffee	m, ~c	~m, ~c	~c
Sum(col.)	m	~m	Σ

DB	m, c	~m, c	m~c	~m~c	lift	all-conf	coh	χ^2
A2	1000	100	100	10,000	9.26	0.91	0.83	9055
C1	100	1000	1000	100,000	8.44	0.09	0.05	670
C2	1000	100	10000	100,000	9.18	0.09	0.09	8172
B1	1000	1000	1000	1000	1	0.5	0.33	0