All-Confidence, Kulczynski and Cosine Measures

- These 3 measures are independent of the number of transactions in the dataset.
- The range of values for the measures is: [0, 1].
- A value close to 1 means high positive correlation.
- These measures are preferred to Lift or Chi-square, when there are many transactions that don’t include items in A or B.

Summary

- Basic concepts: association rules, support-confident framework, closed and max-patterns
- Scalable frequent pattern mining methods
  - Apriori (Candidate generation & test)
  - Projection-based (FPgrowth, CLOSET+, ...)
  - Vertical format approach (ECLAT, CHARM, ...)
- Which patterns are interesting?
- Pattern evaluation methods

Chapter 7: Advanced Frequent Pattern Mining

6.1.3 Association Rule Mining: A Road Map

Market basket analysis is just one form of association rule mining. In fact, there are many kinds of association rules. Association rules can be classified in various ways, based on the following criteria:

Based on the *types of values* handled in the rule: If a rule concerns associations between the presence or absence of items, it is a **Boolean association rule**. For example, Rule (6.1) above is a Boolean association rule obtained from market basket analysis.

If a rule describes associations between quantitative items or attributes, then it is a **quantitative association rule**. In these rules, quantitative values for items or attributes are partitioned into intervals. The following rule is an example of a quantitative association rule, where X is a variable representing a customer:

$$age(X, \text{"30 ... 39"}) \wedge income(X, \text{"42K ... 48K"}) \Rightarrow buys(X, \textit{high resolution TV}) \quad (6.4)$$

Note that the quantitative attributes, *age* and *income*, have been discretized.

Based on the *dimensions* of data involved in the rule: If the items or attributes in an association rule reference only one dimension, then it is a **single-dimensional association rule.** Note that Rule (6.1) could be rewritten as

$$buys(X, \text{"}computer\text{"}) \Rightarrow buys(X, \text{"}financial\_management\_software\text{"}) \quad (6.5)$$

Rule (6.1) is a single-dimensional association rule since it refers to only one dimension, *buys*.[4] If a rule references two or more dimensions, such as the dimensions *buys, time_of_transaction*, and *customer_category*, then it is a **multidimensional association rule.** Rule (6.4) is considered a multidimensional association rule since it involves three dimensions: *age, income*, and *buys*.

Based on the *levels of abstractions* involved in the rule set: Some methods for association rule mining can find rules at differing levels of abstraction. For example, suppose that a set of association rules mined includes the following rules:

$$age(X, \text{"30 \ldots 39"}) \Rightarrow buys(X, \text{"}laptop\ computer\text{"}) \quad (6.6)$$

$$age(X, \text{"30 \ldots 39"}) \Rightarrow buys(X, \text{"}computer\text{"}) \quad (6.7)$$

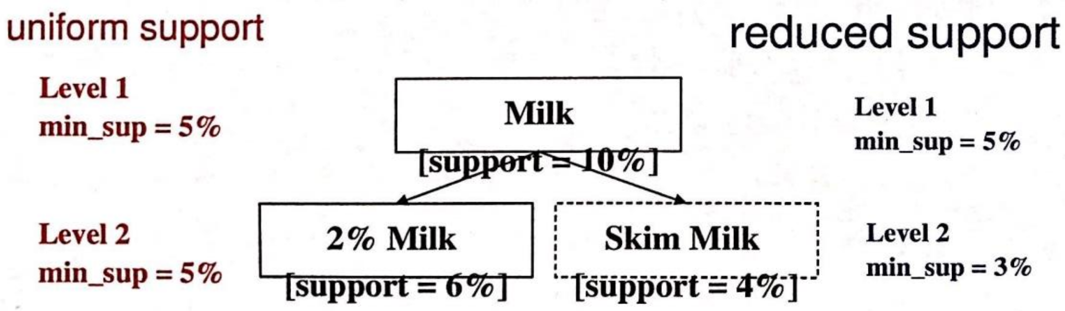
In Rules (6.6) and (6.7), the items bought are referenced at different levels of abstraction. (e.g., "*computer*" is a higher-level abstraction of "*laptop computer*".) We refer to the rule set mined as consisting of **multilevel association rules.** If, instead, the rules within a given set do not reference items or attributes at different levels of abstraction, then the set contains **single-level association rules.**

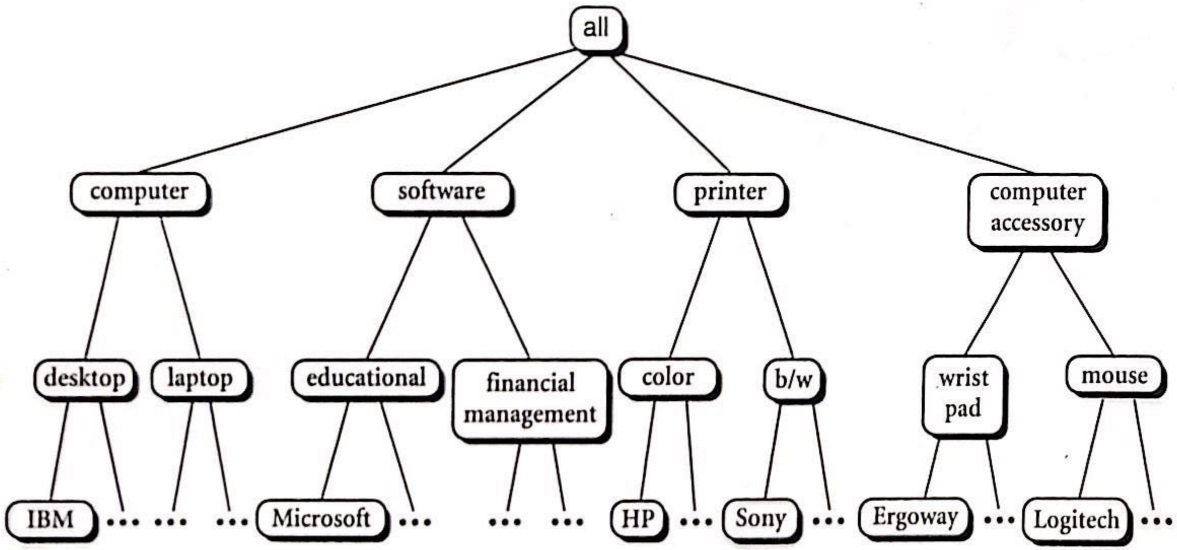
Mining Various Kinds of Association Rules

- Mining multilevel association
- Miming 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@VLB'95, Han & Fu@VLDB'95)





Level
0
1
2
3

Figure ~~6.11~~ 5.10 A concept hierarchy for *AllElectronics* computer items.

Multi-level Association: Redundancy Filtering

- Some rules may be redundant due to "ancestor" relationships between items.
- Example
  - milk ⇒ wheat bread [support = 8%, confidence = 70%]
  - 2% milk ⇒ wheat bread [support = 2%, confidence = 72%]
- We say the first rule is an ancestor of the second rule.
- A rule is redundant if its support is close to the "expected" value, based on the rule's ancestor.

2nd p. 254 (5.10) (5.11)

3rd p. 287

Mining Multi-Dimensional Association

- Single-dimensional rules:

  buys(X, "milk") $\Rightarrow$ buys(X, "bread")

- Multi-dimensional rules: $\geq$ 2 dimensions or predicates
  - Inter-dimension assoc. rules (*no repeated predicates*)

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

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

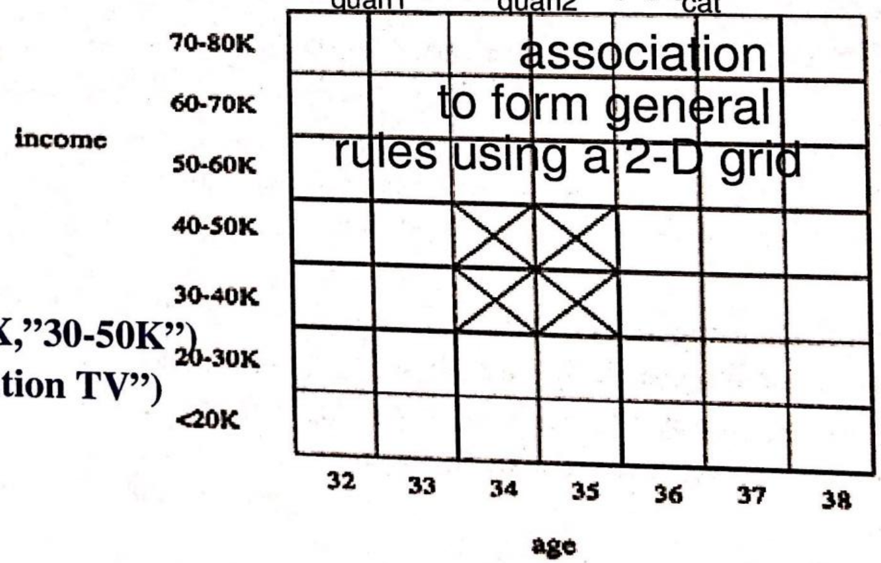
    age(X,"19-25") $\wedge$ buys(X, "popcorn") $\Rightarrow$ buys(X, "coke")

- Categorical Attributes: finite number of possible values, no ordering among values—data cube approach
- Quantitative Attributes: numeric, implicit ordering among values—discretization, clustering, and gradient approaches

Quantitative Association Rules

- Proposed by Lent, Swami and Widom ICDE'97
- Numeric attributes are *dynamically* discretized
  - Such that the confidence or compactness of the rules mined is maximized
- 2-D quantitative association rules: $A_{quan1} \wedge A_{quan2} \Rightarrow A_{cat}$
- Cluster *adjacent* association rules to form general rules using a 2-D grid
- Example

age(X,"34-35") ∧ income(X,"30-50K")
⇒ buys(X,"high resolution TV")



Constraint-based (Query-Directed) Mining

- Finding all the patterns in a database autonomously? — unrealistic!
  - The patterns could be too many but not focused!
- Data mining should be an interactive process
  - User directs what to be mined using a data mining query language (or a graphical user interface)
- Constraint-based mining
  - User flexibility: provides constraints on what to be mined
  - System optimization: explores such constraints for efficient mining—constraint-based mining

Constraints in Data Mining

- Knowledge type constraint:
  - classification, association, etc.
- Data constraint — using SQL-like queries
  - find product pairs sold together in stores in Chicago in Dec.'02
- Dimension/level constraint
  - in relevance to region, price, brand, customer category
- <u>Rule (or pattern) constraint</u>
  - small sales (price $< \$10$) triggers big sales (sum $> \$200$)
- Interestingness constraint
  - strong rules: min_support $\geq$ 3%, min_confidence $\geq$ 60%

Constraint-Based Mining—A General Picture

Constraint	Antimonotone	Monotone	Succinct
$v \in S$	no	yes	yes
$S \supseteq V$	no	yes	yes
$S \subseteq V$	yes	no	yes
$min(S) \leq v$	no	yes	yes
$min(S) \geq v$	yes	no	yes
$max(S) \leq v$	yes	no	yes
$max(S) \geq v$	no	yes	yes
$count(S) \leq v$	yes	no	weakly
$count(S) \geq v$	no	yes	weakly
$sum(S) \leq v$ ($a \in S, a \geq 0$)	yes	no	no
$sum(S) \geq v$ ($a \in S, a \geq 0$)	no	yes	no
$range(S) \leq v$	yes	no	no
$range(S) \geq v$	no	yes	no
$avg(S)\ \theta\ v, \theta \in \{=, \leq, \geq\}$	convertible	convertible	no
$support(S) \geq \xi$	yes	no	no
$support(S) \leq \xi$	no	yes	no

Constraint-Based Frequent Pattern Mining

- Pattern space pruning constraints
  - Anti-monotonic: If constraint c is violated, its further mining can be terminated
  - Monotonic: If c is satisfied, no need to check c again
  - Succinct: c must be satisfied, so one can start with the data sets satisfying c
  - Convertible: c is not monotonic nor anti-monotonic, but it can be converted into it if items in the transaction can be properly ordered
- Data space pruning constraint
  - Data succinct: Data space can be pruned at the initial pattern mining process
  - Data anti-monotonic: If a transaction t does not satisfy c, t can be pruned from its further mining

Pattern Space Pruning with Anti-Monotonicity Constraints

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

- A constraint C is *anti-monotone* if the super pattern satisfies C, all of its sub-patterns do so too
- In other words, *anti-monotonicity:* If an itemset S **violates** the constraint, so does any of its superset

Ex. 1. $sum(S.price) \leq v$ is anti-monotone

Ex. 2. range(S.profit) $\leq$ 15 is anti-monotone

Itemset *ab* violates C

So does every superset of *ab*

Ex. 3. $sum(S.Price) \geq v$ is not anti-monotone

Ex. 4. *support count* is anti-monotone: core property used in Apriori

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Naïve Algorithm: Apriori + Constraint

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~~

Constraint:

Sum{S.price} < 5

Pattern Space Pruning with Monotonicity Constraints

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

- A constraint C is *monotone* if the pattern satisfies C, we do not need to check C in subsequent mining
- Alternatively, monotonicity: *If an itemset S* ***satisfies*** *the constraint, so does any of its superset*

Ex. 1. $sum(S.Price) \geq v$ is monotone

Ex. 2. $min(S.Price) \leq v$ is monotone

Ex. 3. C: range(S.profit) $\geq$ 15

Itemset *ab* satisfies C

So does every superset of *ab*

Correlation Rules

{Bread, Butter} – freq. itemset

Support

{Bread, Butter} – itemset correlated

$\frac{P(A,B)}{P(A) \cdot P(B)}$

Interest

Lift of assoc. rule $A \Rightarrow B$

Lift of assoc. rule $B \Rightarrow A$

Association Mining - Meta Patterns

Example 4.2 A user studying the buying habits of *AllElectronics* customers may choose to mine *association rules* of the form

$$P(X : customer, W) \wedge Q(X, Y) \Rightarrow buys(X, Z)$$

where X is a key of the *customer* relation; P and Q are **predicate variables** that can be instantiated to the relevant attributes or dimensions specified as part of the task-relevant data, and W, Y, and Z are **object variables** that can take on the values of their respective predicates for customers X.

The search for association rules is confined to those matching the given metarule, such as

$$age(X, \text{“}30\ldots39\text{”}) \wedge income(X, \text{“}40K\ldots49K\text{”}) \Rightarrow buys(X, \text{“}VCR\text{”}) \quad [2.2\%, 60\%] \tag{4.1}$$

and

$$occupation(X, \text{“}student\text{”}) \wedge age(X, \text{“}20\ldots29\text{”}) \Rightarrow buys(X, \text{“}computer\text{”}) \quad [1.4\%, 70\%]. \tag{4.2}$$

The former rule states that customers in their thirties, with an annual income of between 40K and 49K, are likely (with 60% confidence) to purchase a VCR, and such cases represent about 2.2% of the total number of transactions. The latter rule states that customers who are students and in their twenties are likely (with 70% confidence) to purchase a computer, and such cases represent about 1.4% of the total number of transactions. ■

Meta-Rule Guided Mining

- Meta-rule can be in the rule form with partially instantiated predicates and constan

  $P_1(X, Y) \wedge P_2(X, W) \Rightarrow buys(X, \text{“iPad”})$

- The resulting rule derived can be

  age(X, “15-25”) ^ profession(X, “student”) => buys(X, “iPad”)

- In general, it can be in the form of

  $P_1 \wedge P_2 \wedge \ldots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \ldots \wedge Q_r$

- Method to find meta-rules
  - Find frequent (l+r) predicates (based on min-support threshold)
  - Push constants deeply when possible into the mining process (see the remaining discussions on constraint-push techniques)
  - Use confidence, correlation, and other measures when possible

Summary

- Roadmap: Many aspects & extensions on pattern mining
- Mining patterns in multi-level, multi dimensional space
- Mining Quantitative Association Rules
- Constraint-based pattern mining