Interestingness Measurements

- *Objective* measures
  - Two popular measurements:
  - support and
  - confidence

- *Subjective* measures [Silberschatz & Tuzhilin, KDD95]
  - A rule (pattern) is interesting if it is
  - unexpected (surprising to the user) and/or
  - actionable (the user can do something with it)

Criticism to Support and Confidence

Example 1 [Aggarwal & Yu, PODS98]

- Among 5000 students
  - 3000 play basketball (=60%)
  - 3750 eat cereal (=75%)
  - 2000 both play basket ball and eat cereal (=40%)
- Rule *play basketball* $\Rightarrow$ *eat cereal* [40%, 66.7%] is misleading because the overall percentage of students eating cereal is 75% which is higher than 66.7%
- Rule *play basketball* $\Rightarrow$ *not eat cereal* [20%, 33.3%] is far more accurate, although with lower support and confidence
- Observation: *play basketball* and *eat cereal* are *negatively correlated*

➢ Not all strong association rules are interesting and some can be misleading.
→ augment the support and confidence values with interestingness measures such as the correlation $A \Rightarrow B\ [supp, conf, corr]$

Other Interestingness Measures: Correlation

- ***Lift*** is a simple correlation measure between two items *A* and *B:*

$$corr_{A,B} = \frac{P(A \cup B)}{P(A)P(B)} = \frac{P(B|A)}{P(B)} = \frac{conf(A \Rightarrow B)}{supp(B)}$$

! The two rules $A \Rightarrow B$ and $B \Rightarrow A$ have the same correlation coefficient.

- take both P(A) and P(B) in consideration

- $corr_{A,B} > 1$ the two items A and B are positively correlated
- $corr_{A,B} = 1$ there is no correlation between the two items A and B
- $corr_{A,B} < 1$ the two items A and B are negatively correlated

Other Interestingness Measures: Correlation

- Example 2:

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

- X and Y: positively correlated
- X and Z: negatively related
- support and confidence of X=>Z dominates
- but items X and Z are negatively correlated
- Items X and Y are positively correlated

rule	support	confidence	correlation
$X \Rightarrow Y$	25%	50%	2
$X \Rightarrow Z$	37.5%	75%	0.86
$Y \Rightarrow Z$	12.5%	50%	0.57

Chapter 3: Frequent Itemset Mining

1) Introduction
   – Transaction databases, market basket data analysis
2) Mining Frequent Itemsets
   – Apriori algorithm, hash trees, FP-tree
3) Simple Association Rules
   – Basic notions, rule generation, interestingness measures
4) Further Topics
   – Hierarchical Association Rules
      • Motivation, notions, algorithms, interestingness
   – Quantitative Association Rules
      • Motivation, basic idea, partitioning numerical attributes, adaptation of apriori algorithm, interestingness
5) Extensions and Summary

Hierarchical Association Rules: Motivation

- Problem of association rules in plain itemsets
  - *High minsup:* apriori finds only few rules
  - *Low minsup:* apriori finds unmanagably many rules
- Exploit item taxonomies (generalizations, *is-a* hierarchies) which exist in many applications

- New task: find all generalized association rules between generalized items → Body and Head of a rule may have items of any level of the hierarchy
- *Generalized association rule*: $X \Rightarrow Y$
  with $X, Y \subset I, X \cap Y = \emptyset$ and no item in Y is an ancestor of any item in X
  e.g. $jackets \Rightarrow clothes$ is essentially trivial

Hierarchical Association Rules: Motivating Example

- Examples

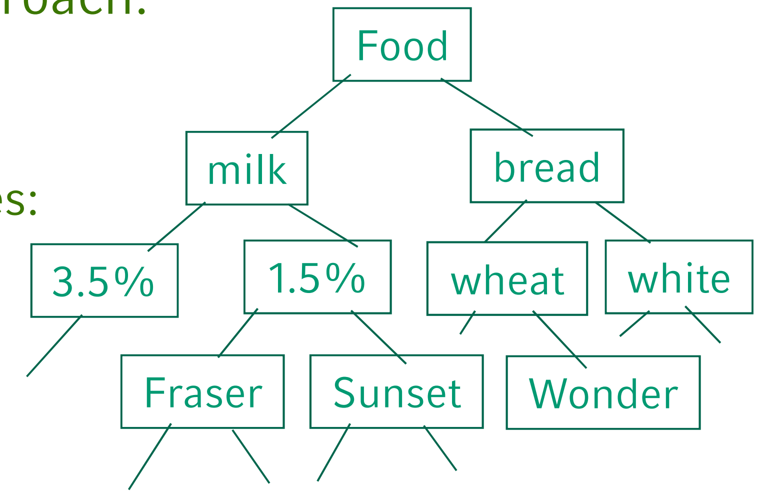
  Jeans $\Rightarrow$ boots
  jackets $\Rightarrow$ boots
  } Support < minSup

  Outerwear $\Rightarrow$ boots   Support > minsup

- Characteristics
  - Support("outerwear $\Rightarrow$ boots") is not necessarily equal to the sum support("jackets $\Rightarrow$ boots") + support("jeans $\Rightarrow$ boots")
    e.g. if a transaction with jackets, jeans and boots exists
  - Support for sets of generalizations (e.g., product groups) is higher than support for sets of individual items
    If the support of rule "outerwear $\Rightarrow$ boots" exceeds minsup, then the support of rule "clothes $\Rightarrow$ boots" does, too

Mining Multi-Level Associations

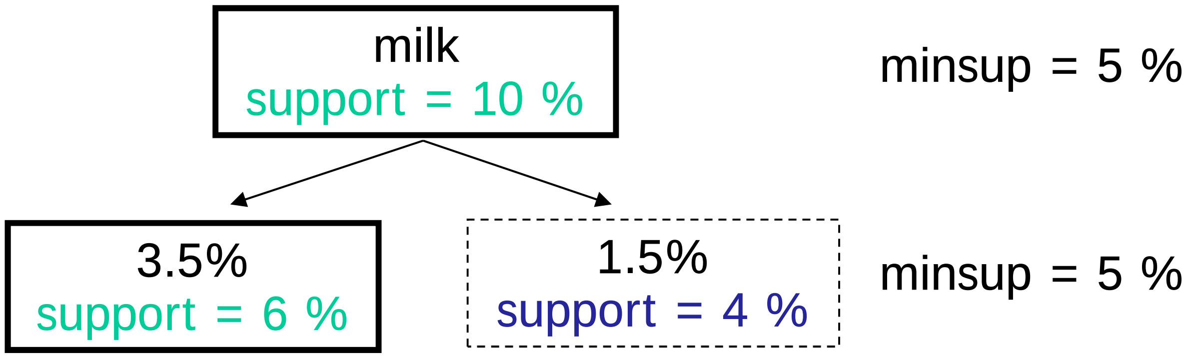
- A *top_down, progressive deepening* approach:
  - First find high-level strong rules:
    - *milk* ⇒ *bread* [20%, 60%].
  - Then find their lower-level "weaker" rules:
    - 1.5% *milk* ⇒ *wheat bread* [6%, 50%].



- Different min_support threshold across multi-levels lead to different algorithms:
  - adopting the same min_support across multi-levels
  - adopting reduced min_support at lower levels

Minimum Support for Multiple Levels

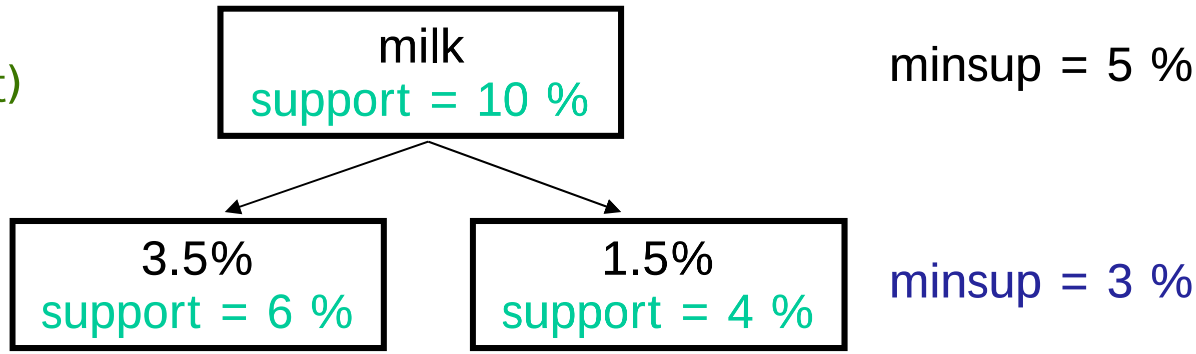
- Uniform Support



+ the search procedure is simplified (monotonicity)

+ the user is required to specify only one support threshold

- Reduced Support (Variable Support)

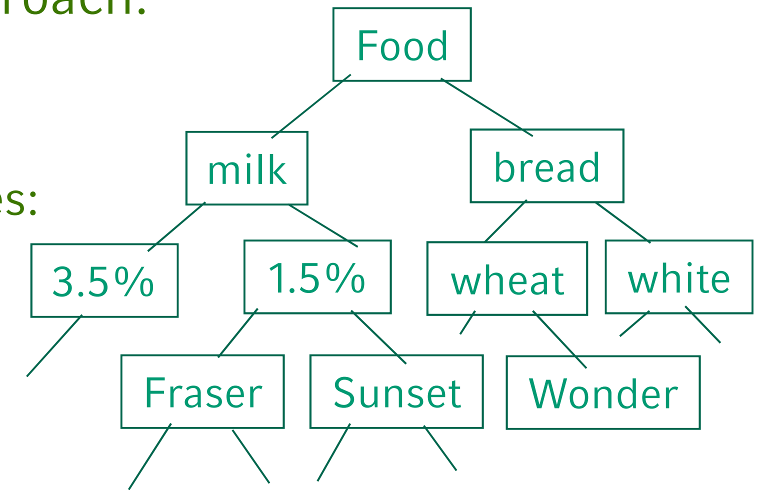


+ takes the lower frequency of items in lower levels into consideration

Multilevel Association Mining using Reduced Support

- A *top_down, progressive deepening* approach:
  - First find high-level strong rules:
    - *milk* ⇒ *bread* [20%, 60%].
  - Then find their lower-level "weaker" rules:
    - 1.5% *milk* ⇒ *wheat bread* [6%, 50%].

level-wise processing (breadth first)



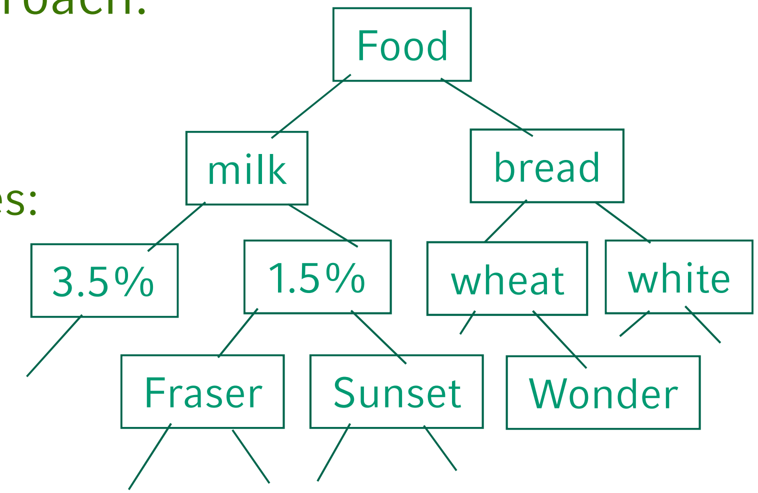
3 approaches using reduced Support:

- *Level-by-level independent method*:
  - Examine each node in the hierarchy, regardless of whether or not its parent node is found to be frequent
- *Level-cross-filtering by single item*:
  - Examine a node only if its parent node at the preceding level is frequent
- *Level-cross- filtering by k-itemset*:
  - Examine a k-itemset at a given level only if its parent k-itemset at the preceding level is frequent

Multilevel Associations: Variants

- A *top_down, progressive deepening* approach:
  - First find high-level strong rules:
    - *milk* ⇒ *bread* [20%, 60%].
  - Then find their lower-level "weaker" rules:
    - 1.5% *milk* ⇒ *wheat bread* [6%, 50%].

level-wise processing (breadth first)



- Variations at mining multiple-level association rules.
  - Level-crossed association rules:
    - 1.5 % *milk* ⇒ *Wonder wheat bread*
  - Association rules with multiple, alternative hierarchies:
    - 1.5 % *milk* ⇒ *Wonder bread*

Multi-level Association: Redundancy Filtering

- Some rules may be redundant due to "ancestor" relationships between items.
- Example
  - R_1: milk $\Rightarrow$ wheat bread [support = 8%, confidence = 70%]
  - R_2: 1.5% milk $\Rightarrow$ wheat bread [support = 2%, confidence = 72%]
- We say that rule 1 is an ancestor of rule 2.
- *Redundancy:*
  A rule is redundant if its support is close to the "expected" value, based on the rule's ancestor

  (see [SA'95] R. Srikant, R. Agrawal: *Mining Generalized Association Rules*. In VLDB, 1995.)

Expected Support and Expected Confidence

- How to compute the expected support?
  Given the rule for X ⇒ Y and its ancestor rule X´ ⇒ Y´ the expected support of X ⇒ Y is defined as:

$$E_{Z'}[\mathrm{P}(Z)] = \frac{\mathrm{P}(z_1)}{\mathrm{P}(z'_1)} \times \cdots \times \frac{\mathrm{P}(z_j)}{\mathrm{P}(z'_j)} \times \mathrm{P}(Z')$$

  where $Z = X \cup Y = \{z_1, \ldots, z_n\}$, $Z' = X' \cup Y' = \{z'_1, \ldots, z'_j, z_{j+1}, \ldots, z_n\}$ and each $z'_i \in Z'$ is an ancestor of $z_i \in Z$

[SA'95] R. Srikant, R. Agrawal: *Mining Generalized Association Rules*. In VLDB, 1995.

Expected Support and Expected Confidence

- How to compute the expected confidence?
  Given the rule for X ⇒ Y and its ancestor rule X´ ⇒ Y´, then the expected confidence of X ⇒ Y is defined as:

$$E_{X' \Rightarrow Y'}[\mathrm{P}(Y|X)] = \frac{\mathrm{P}(y_1)}{\mathrm{P}(y_1')} \times \cdots \times \frac{\mathrm{P}(y_j)}{\mathrm{P}(y_j')} \times \mathrm{P}(Y'|X')$$

  where $Y = \{y_1, \ldots, y_n\}$ and $Y' = \{y_1', \ldots, y_j', y_{j+1}, \ldots, y_n\}$ and each $y_i' \in Y'$ is an ancestor of $y_i \in Y$

[SA'95] R. Srikant, R. Agrawal: *Mining Generalized Association Rules*. In VLDB, 1995.

Chapter 3: Frequent Itemset Mining

1) Introduction
   – Transaction databases, market basket data analysis
2) Simple Association Rules
   – Basic notions, rule generation, interestingness measures
3) Mining Frequent Itemsets
   – Apriori algorithm, hash trees, FP-tree
4) Further Topics
   – Hierarchical Association Rules
      • Motivation, notions, algorithms, interestingness
   – Multidimensional and Quantitative Association Rules
      • Motivation, basic idea, partitioning numerical attributes, adaptation of apriori algorithm, interestingness
5) Summary

Multi-Dimensional Association: Concepts

- Single-dimensional rules:
  - buys milk $\Rightarrow$ buys bread

- Multi-dimensional rules: $\geq$ 2 dimensions
  - Inter-dimension association rules (*no repeated dimensions*)
    - **age** between 19-25 $\wedge$ **status** is student $\Rightarrow$ **buys** coke
  - hybrid-dimension association rules (*repeated dimensions*)
    - **age** between 19-25 $\wedge$ **buys** popcorn $\Rightarrow$ **buys** coke

Techniques for Mining Multi-Dimensional Associations

- Search for frequent *k*-predicate set:
  - Example: {age, occupation, buys} is a 3-predicate set.
  - Techniques can be categorized by how age is treated.

1. Using static discretization of quantitative attributes
   - Quantitative attributes are statically discretized by using predefined concept hierarchies.
2. Quantitative association rules
   - Quantitative attributes are dynamically discretized into "bins"based on the distribution of the data.
3. Distance-based association rules
   - This is a dynamic discretization process that considers the distance between data points.

Quantitative Association Rules

- Up to now: associations of *boolean* attributes only
- Now: *numerical* attributes, too
- Example:
  - Original database

ID	age	marital status	# cars
1	23	single	0
2	38	married	2

  - Boolean database

ID	age: 20..29	age: 30..39	m-status: single	m-status: married	. . .
1	1	0	1	0	. . .
2	0	1	0	1	. . .

Quantitative Association Rules: Ideas

- Static discretization
  - Discretization of all attributes *before* mining the association rules
  - E.g. by using a generalization hierarchy for each attribute
  - Substitute numerical attribute values by ranges or intervals

- Dynamic discretization
  - Discretization of the attributes *during* association rule mining
  - Goal (e.g.): maximization of confidence
  - Unification of neighboring association rules to a generalized rule

Partitioning of Numerical Attributes

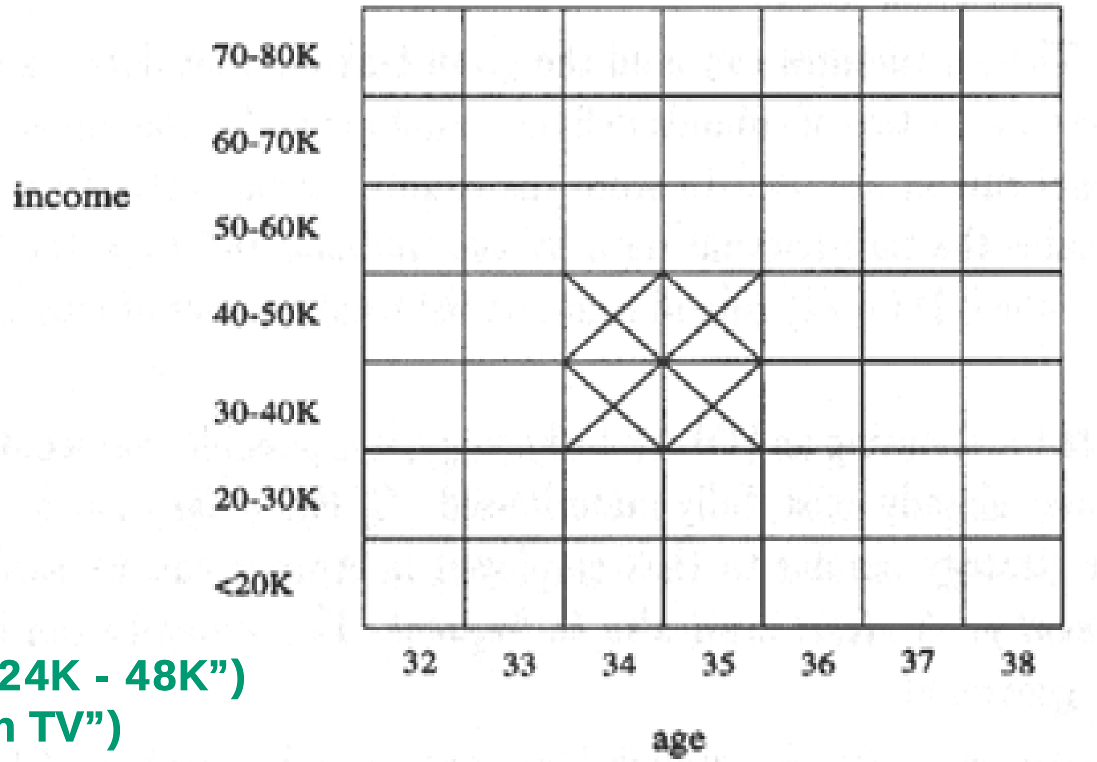
- Problem: Minimum support
  - Too many intervals → too small support for each individual interval
  - Too few intervals → too small confidence of the rules

- Solution
  - First, partition the domain into many intervals
  - Afterwards, create new intervals by merging adjacent interval

- Numeric attributes are *dynamically* discretized such that the confidence or compactness of the rules mined is maximized.

Quantitative Association Rules

- 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,"30-34") $\wedge$ income(X,"24K - 48K")
$\Rightarrow$ buys(X,"high resolution TV")

Chapter 3: Frequent Itemset Mining

1) Introduction
   - Transaction databases, market basket data analysis
2) Mining Frequent Itemsets
   - Apriori algorithm, hash trees, FP-tree
3) Simple Association Rules
   - Basic notions, rule generation, interestingness measures
4) Further Topics
   - Hierarchical Association Rules
     - Motivation, notions, algorithms, interestingness
   - Quantitative Association Rules
     - Motivation, basic idea, partitioning numerical attributes, adaptation of apriori algorithm, interestingness
5) Summary

Chapter 3: Summary

- Mining frequent itemsets
  - Apriori algorithm, hash trees, FP-tree
- Simple association rules
  - support, confidence, rule generation, interestingness measures (correlation), …
- Further topics
  - Hierarchical association rules: algorithms (top-down progressive deepening), multilevel support thresholds, redundancy and R-interestingness
  - Quantitative association rules: partitioning numerical attributes, adaptation of apriori algorithm, interestingness
- Extensions: multi-dimensional association rule mining

Further Applications

- Customer analysis
- Facilitator for other data mining techniques
- Indexing and retrieval: provide a concise data representation
- Web mining tasks: sequential pattern mining for traversal patterns which help in designing and organizing web sites
- Temporal applications, e.g. event detection
- Spatial and spatiotemporal analysis: association rules can characterize useful relationships between spatial and non-spatial properties
- Image and multimedia data mining: frequent image features help in several mining tasks for image data
- Chemical and biological applications: often important motifs correspond to frequent patterns in graphs and structured data (toxicological analysis, chemical compound prediction, RNA analysis …)

Sequential pattern mining

Outlook to KDD 2

- Task 1: find all subsets of items that occur with a specific sequence in many transactions.
  - E.g.: 97% of transactions contain the *sequence* {jogging → high ECG → sweating}
- Task 2: find all rules that correlate the order of one set of items after that of another set of items in the transaction database.
  - E.g.: 72% of users who perform a web search *then* make a long eye gaze over the ads *follow that* by a successful add-click
- The order of the items matters, thus all possible permutations of items must be considered when checking possible frequent sequences, not only the combinations of items
- Applications: data with temporal order (streams), e.g.: bioinformatics, Web mining, text mining, sensor data mining, process mining etc.

Sequential Pattern Mining vs. Frequent Itemset Mining

- Both can be applied on similar dataset
  - Each customer has a customer id and aligned with transactions.
  - Each transaction has a transaction id and belongs to one customer.
  - Based on the transaction id, each customer also aligned to a transaction sequence.

Cid	Tid	Item
1	1	{butter}
	2	{milk}
	3	{sugar}
2	4	{butter, sugar}
	5	{milk, sugar}
	6	{butter, milk, sugar}
	7	{eggs}
3	8	{sugar}
	9	{butter, milk}
	10	{eggs}
	11	{milk}

Cid	Item
1	{butter} ,{milk}, {sugar}
2	{butter, sugar}, {milk, sugar}, {butter, milk, sugar}, {eggs}
3	{sugar}, {butter, milk}, {eggs}, {milk}

Frequent itemset mining

- No temporal importance in the order of items happening together

items	frequency
{butter}	4
{milk}	5
{butter, milk}	2
…	

Sequential pattern mining

- The order of items matters

sequences	frequency
{butter}	4
{butter, milk}	2
{butter},{milk}	4
{milk},{butter}	1
{butter},{butter,milk}	1
…	

Sequential Pattern Mining Algorithm

- Breadth-first search based
  - GSP (*Generalized Sequential Pattern*) algorithm[1]
  - SPADE[2]
  - …
- Depth-first search based
  - PrefixSpan[3]
  - SPAM[4]
  - …

[1]Sirkant & Aggarwal: *Mining sequential patterns: Generalizations and performance improvements.* EDBT 1996
[2]Zaki M J. *SPADE: An efficient algorithm for mining frequent sequences[J].* Machine learning, 2001, 42(1-2): 31-60.
[3]Pei at. al.: *Mining sequential patterns by pattern-growth: PrefixSpan approach.* TKDE 2004
[4]Ayres, Jay, et al: *Sequential pattern mining using a bitmap representation.* SIGKDD 2002.