Interestingness Measurements

- **Objective** measures
  - Two popular measurements:
    - support
    - 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 basketball and eat cereal (=40%)
  - Rule **play basketball ⇒ 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 ⇒ 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
Interestingness of Association Rules

- Goal: Delete misleading association rules
- Condition for a rule $A \Rightarrow B$
  $$\frac{P(A \cup B)}{P(A)} > P(B) + d \quad \text{for a suitable threshold } d > 0$$

- Measure for the interestingness of a rule
  $$\frac{P(A \cup B)}{P(A)} - P(B)$$
  - The larger the value, the more interesting the relation between $A$ and $B$, expressed by the rule.
  - Other measures: correlation between $A$ and $B \Rightarrow \frac{P(A \cup B)}{P(A)P(B)}$

Criticism to Support and Confidence: Correlation of Itemsets

- Example 2

<table>
<thead>
<tr>
<th>Rule</th>
<th>Support</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X \Rightarrow Y$</td>
<td>25%</td>
<td>50%</td>
</tr>
<tr>
<td>$X \Rightarrow Z$</td>
<td>37.50%</td>
<td>75%</td>
</tr>
</tbody>
</table>

- $X$ and $Y$: positively correlated
- $X$ and $Z$: negatively related
- support and confidence of $X \Rightarrow Z$ dominates
- We need a measure of dependent or correlated events
  $$corr_{A,B} = \frac{P(A \cup B)}{P(A)P(B)}$$
  - $P(B|A)/P(B)$ is also called the lift of rule $A \Rightarrow B$
Other Interestingness Measures: Interest

- **Interest (correlation, lift):** \( \frac{P(A \cup B)}{P(A)P(B)} \)

- taking both \( P(A) \) and \( P(B) \) in consideration
- Correlation equals 1, i.e. \( P(A \cup B) = P(B) \cdot P(A) \), if \( A \) and \( B \) are independent events
- \( A \) and \( B \) negatively correlated, if the value is less than 1; otherwise \( A \) and \( B \) positively correlated

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
<th>Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>X,Y</td>
<td>25%</td>
<td>2</td>
</tr>
<tr>
<td>X,Z</td>
<td>37.50%</td>
<td>0.9</td>
</tr>
<tr>
<td>Y,Z</td>
<td>12.50%</td>
<td>0.57</td>
</tr>
</tbody>
</table>

Chapter 8: Mining Association Rules

- **Introduction**
  - Transaction databases, market basket data analysis
- **Simple Association Rules**
  - Basic notions, apriori algorithm, hash trees, interestingness
- **Hierarchical Association Rules**
  - Motivation, notions, algorithms, interestingness
- **Quantitative Association Rules**
  - Motivation, basic idea, partitioning numerical attributes, adaptation of apriori algorithm, interestingness
- **Constraint-based Association Mining**
- **Summary**
Hierarchical Association Rules: Motivation

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

![Hierarchical Itemset Diagram]

- Task: find association rules between generalized items
- Support for sets of item types (e.g., product groups) is higher than support for sets of individual items

Hierarchical Association Rules: Motivating Example

- Examples
  - jeans $\Rightarrow$ boots
  - jackets $\Rightarrow$ boots
  - outerwear $\Rightarrow$ boots
  \[
  \text{Support} < \text{minsup}
  \]

- Characteristics
  - Support(“outerwear $\Rightarrow$ boots”) is not necessarily equal to the sum support(“jackets $\Rightarrow$ boots”) + support( “jeans $\Rightarrow$ boots”)
  - If the support of rule “outerwear $\Rightarrow$ boots” exceeds minsup, then the support of rule “clothes $\Rightarrow$ boots” does, too
Mining Multi-Level Associations

- Example generalization hierarchy:

- A top-down, progressive deepening approach:
  - First find high-level strong rules:
    - \( \text{milk} \rightarrow \text{bread} \ [20\%, \ 60\%] \).
  - Then find their lower-level “weaker” rules:
    - \( 1.5\% \text{ milk} \rightarrow \text{wheat bread} \ [6\%, \ 50\%] \).

- Variations at mining multiple-level association rules.
  - Level-crossed association rules:
    - \( 1.5\% \text{ milk} \rightarrow \text{Wonder wheat bread} \)
  - Association rules with multiple, alternative hierarchies:
    - \( 1.5\% \text{ milk} \rightarrow \text{Wonder bread} \)

Hierarchical Association Rules: Basic Notions

- [Srikant & Agrawal 1995]

- Let \( U = \{i_1, \ldots, i_m\} \) be a universe of literals called items
- Let \( h \) be a directed acyclic graph defined as follows:
  - The universe of literals \( U \) forms the set of vertices in \( h \)
  - A pair \((i, j)\) forms an edge in \( h \) if \( i \) is a generalization of \( j \)
    - \( i \) is called parent or direct predecessor of \( j \)
    - \( j \) is called a child or a direct successor of \( i \)
  - \( x' \) is an ancestor of \( x \) and, thus, \( x \) is a descendant of \( x' \) wrt. \( h \), if there is a path from \( x' \) to \( x \) in \( h \)
  - A set of items \( z' \) is called an ancestor of a set of items \( z \) if at least one item in \( z' \) is an ancestor of an item in \( z \).
Hierarchical Association Rules: Basic Notions (2)

- Let \( D \) be a set of transactions \( T \) with \( T \subseteq U \)
  - Typically, transactions \( T \) in \( D \) only contain items from the leaves of graph \( h \)
- A transaction \( T \) supports an item \( i \in U \) if \( i \) or any ancestor of \( i \) is contained in \( T \)
- A transaction \( T \) supports a set \( X \subseteq U \) of items if \( T \) supports each item in \( X \)

- Support of a set \( X \subseteq U \) of items in \( D \):
  - Percentage of transactions in \( D \) that support \( X \)

Hierarchical Association Rules: Basic Notions (3)

- Hierarchical association rule
  - \( X \Rightarrow Y \) with \( X \subseteq U, Y \subseteq U, X \cap Y = \emptyset \)
  - No item in \( Y \) is ancestor of an item in \( X \) wrt. \( h \)
- Support of a hierarchical association rule \( X \Rightarrow Y \) in \( D \):
  - Support of the set \( X \cup Y \) in \( D \)
- Confidence of a hierarchical association rule \( X \Rightarrow Y \) in \( D \):
  - Percentage of transactions that support \( Y \) among the subset of transactions that support \( X \)
Hierarchical Association Rules: Example

Support of \{clothes\}: 4 of 6 = 67%
Support of \{clothes, boots\}: 2 of 6 = 33%

"shoes ⇒ clothes": support 33%, confidence 50%
"boots ⇒ clothes": support 33%, confidence 100%

<table>
<thead>
<tr>
<th>transaction id</th>
<th>items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>shirt</td>
</tr>
<tr>
<td>2</td>
<td>jacket, boots</td>
</tr>
<tr>
<td>3</td>
<td>jeans, boots</td>
</tr>
<tr>
<td>4</td>
<td>sports shoes</td>
</tr>
<tr>
<td>5</td>
<td>sports shoes</td>
</tr>
<tr>
<td>6</td>
<td>jacket</td>
</tr>
</tbody>
</table>

Determination of Frequent Itemsets: Basic Algorithm for Hierarchical Rules

Idea: Extend the transactions in the database by all the ancestors of the items contained

Method:
- For all transactions \( t \) in the database
  - Create an empty new transaction \( t' \)
  - For each item \( i \) in \( t \), insert \( i \) and all its ancestors wrt. \( h \) in \( t' \)
  - Avoid inserting duplicates
- Based on the new transactions \( t' \), find frequent itemsets for simple association rules (e.g., by using the apriori algorithm)
Determination of Frequent Itemsets: Optimization of Basic Algorithm

- Precomputation of ancestors
  - Additional data structure that holds the association of each item to the list of its ancestors: \( \text{item} \rightarrow \text{list of successors} \)
  - supports a more efficient access to the ancestors of an item

- Filtering of new ancestors
  - Add only ancestors to a transaction which occur in an element of the candidate set \( C_k \) of the current iteration

Example
- \( C_k = \{\{\text{clothes, shoes}\}\} \)
- Substitute „jacketABC“ by „clothes“

Algorithm \textit{Cumulate}: Exclude redundant itemsets

- Let \( X \) be a \( k \)-itemset, \( i \) an item and \( i' \) an ancestor of \( i \)
- \( X = \{i, i', \ldots\} \)
- Support of \( X - \{i'\} = \text{support of } X \)
- When generating candidates, \( X \) can be excluded
- \( k \)-itemsets that contain an item \( i \) and an ancestor \( i' \) of \( i \) as well are not counted
Multi-level Association: Redundancy Filtering

- Some rules may be redundant due to “ancestor” relationships between items.
- Example
  - milk $\Rightarrow$ wheat bread  \[\text{support} = 8\%, \text{confidence} = 70\%\]
  - 2\% milk $\Rightarrow$ wheat bread  \[\text{support} = 2\%, \text{confidence} = 72\%\]

- 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.

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 “weaker” frequent itemsets:
    - 1.5\% milk (5\%), wheat bread (4\%)

- Different min_support threshold across multi-levels lead to different algorithms:
  - If adopting the same min_support across multi-levels
    - toss t if any of t’s ancestors is infrequent.
  - If adopting reduced min_support at lower levels
    - then examine only those descendents whose ancestor’s support is frequent/non-negligible.
Progressive Refinement of Data Mining Quality

- Why progressive refinement?
  - Mining operator can be expensive or cheap, fine or rough
- Superset coverage property:
  - Preserve all the positive answers—allow a positive false test but not a false negative test.
- Two- or multi-step mining:
  - First apply rough/cheap operator (superset coverage)
  - Then apply expensive algorithm on a substantially reduced candidate set (Koperski & Han, SSD’95).

Determination of Frequent Itemsets: Stratification

- Alternative to basic algorithm (i.e., to apriori algorithm)
- Stratification: build layers from the sets of itemsets

- Basic observation
  - If itemset \( X' \) does not have minimum support, and \( X' \) is an ancestor of \( X \), then \( X \) does not have minimum support, too.

- Method
  - For a given \( k \), do not count all \( k \)-itemsets simultaneously
  - Instead, count the more general itemsets first, and count the more specialized itemsets only when required
Determination of Frequent Itemsets: Stratification (2)

Example

- \( C_k = \{\{\text{clothes, shoes}\}, \{\text{outerwear, shoes}\}, \{\text{jackets, shoes}\}\} \)
- First, count the support for \{clothes, shoes\}
- Only if support to small, count the support for \{outerwear, shoes\}

Notions

- **Depth** of an itemset
  - For itemsets \( X \) from a candidate set \( C_k \) without direct ancestors in \( C_k \): \( \text{depth}(X) = 0 \)
  - For all other itemsets \( X \) in \( C_k \):
    - \( \text{depth}(X) = 1 + \max \{\text{depth}(X'), X' \in C_k \text{ is a parent of } X\} \)
- \((C_k^n)\): set of itemsets of depth \( n \) from \( C_k \) where \( 0 \leq n \leq \text{maximum depth } t \)

Determination of Frequent Itemsets: Algorithm Stratify

Method

- Count the itemsets from \( C_k^0 \)
- Remove all descendants of elements from \((C_k^0)\) that do not have minimum support
  - Count the remaining elements in \( (C_k^1) \)
  - ...

Trade-off between number of itemsets for which support is counted simultaneously and number of database scans

- If \(|C_k^n|\) is small, then count candidates of depth \((n, n+1, ..., t)\) at once
Determination of Frequent Itemsets: Stratification – Problems

- Problem of algorithm Stratify
  - If many itemsets with small depth share the minimum support, only few itemsets of a higher depth are excluded

- Improvements of algorithm Stratify
  - Estimate the support of all itemsets in $C_k$ by using a sample
  - Let $C_k'$ be the set of all itemsets for which the sample suggests that all or at least all their ancestors in $C_k$ share the minimum support
  - Determine the actual support of the itemsets in $C_k'$ by a single database scan
  - Remove all descendants of elements in $C_k'$ that have a support below the minimum support from the set $C_k'' = C_k - C_k'$
  - Determine the support of the remaining itemsets in $C_k''$ in a second database scan

Determination of Frequent Itemsets: Stratification – Experiments

- Test data
  - Supermarket data
    - 548,000 items; item hierarchy with 4 levels; 1.5M transactions
  - Department store data
    - 228,000 items; item hierarchy with 7 levels; 570,000 transactions

- Results
  - Optimizations of algorithms cumulate and stratify can be combined
  - *cumulate* optimizations yield a strong efficiency improvement
  - *Stratification* yields a small additional benefit only
Progressive Refinement Mining of Spatial Association Rules

- Hierarchy of spatial relationship:
  - “g_close_to”: near_by, touch, intersect, contain, etc.
  - First search for rough relationship and then refine it.

- Two-step mining of spatial association:
  - Step 1: rough spatial computation (as a filter)
    - Using MBR or R-tree for rough estimation.
  - Step2: Detailed spatial algorithm (as refinement)
    - Apply only to those objects which have passed the rough spatial association test (no less than min_support)

Interestingness of Hierarchical Association Rules – Notions

- Rule $X' \Rightarrow Y'$ is an ancestor of rule $X \Rightarrow Y$ if:
  - Itemset $X'$ is an ancestor of itemset $X$ or itemset $Y'$ is an ancestor of itemset $Y$

- Rule $X' \Rightarrow Y'$ is a direct ancestor of rule $X \Rightarrow Y$ in a set of rules if:
  - Rule $X' \Rightarrow Y'$ is an ancestor of rule $X \Rightarrow Y$, and
  - There is no rule $X'' \Rightarrow Y''$ such that $X'' \Rightarrow Y''$ is an ancestor of $X \Rightarrow Y$ and $X' \Rightarrow Y'$ is an ancestor of $X'' \Rightarrow Y''$

- A hierarchical association rule $X \Rightarrow Y$ is called $R$-interesting if:
  - There are no direct ancestors of $X \Rightarrow Y$ or
  - Actual support is larger than $R$ times the expected support or
  - Actual confidence is larger than $R$ times the expected confidence
Interestingness of Hierarchical Association Rules – Example

**Example**
- Let $R = 2$

<table>
<thead>
<tr>
<th>Item</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>clothes</td>
<td>20</td>
</tr>
<tr>
<td>outerwear</td>
<td>10</td>
</tr>
<tr>
<td>jackets</td>
<td>4</td>
</tr>
</tbody>
</table>

- **No.** | **rule** | **support** | **$R$-interesting?**
- 1 | clothes $\Rightarrow$ shoes | 10 | yes: no ancestors
- 2 | outerwear $\Rightarrow$ shoes | 9  | yes: Support $>> R^*$ expected support (wrt. rule 1)
- 3 | jackets $\Rightarrow$ shoes | 4  | no: Support $< R^*$ expected support (wrt. rule 2)

Multi-level Association: Uniform Support vs. Reduced Support

- **Uniform Support:** the same minimum support for all levels
  - Benefit for efficiency: One minimum support threshold
    - No need to examine itemsets containing any item whose ancestors do not have minimum support.
  - Limited effectiveness: Lower level items do not occur as frequently. Look at support threshold minsup, if ...
    - Minsup too high $\Rightarrow$ miss low level associations
    - Minsup too low $\Rightarrow$ generate too many high level associations

- **Reduced Support:** reduced minimum support at lower levels
  - There are four search strategies:
    - Level-by-level independent
    - Level-cross filtering by $k$-itemset
    - Level-cross filtering by single item
    - Controlled level-cross filtering by single item
Hierarchical Association Rules – How to Choose Minimum Support?

- **Uniform Support**
  - outerwear
    - support = 10%
  - jackets
    - support = 6%
  - jeans
    - support = 4%
  - minsup = 5%

- **Reduced Support** (Variable Support)
  - outerwear
    - support = 10%
  - jackets
    - support = 6%
  - jeans
    - support = 4%
  - minsup = 3%

Multi-Dimensional Association: Concepts

- Single-dimensional rules:
  - buys(X, “milk”) ⇒ buys(X, “bread”)

- Multi-dimensional rules: ≥ 2 dimensions or predicates
  - Inter-dimension association rules (**no repeated predicates**)
    - age(X,”19-25”) ∧ occupation(X,”student”) ⇒ buys(X,”coke”)
  - Hybrid-dimension association rules (**repeated predicates**)
    - age(X,”19-25”) ∧ buys(X,”popcorn”) ⇒ buys(X, “coke”)

- Categorical Attributes
  - finite number of possible values, no ordering among values

- Quantitative Attributes
  - numeric, implicit ordering among values
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 are 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.

Chapter 8: Mining Association Rules

- Introduction
  - Transaction databases, market basket data analysis

- Simple Association Rules
  - Basic notions, apriori algorithm, hash trees, interestingness

- Hierarchical Association Rules
  - Motivation, notions, algorithms, interestingness

- Quantitative Association Rules
  - Motivation, basic idea, partitioning numerical attributes, adaptation of apriori algorithm, interestingness

- Constraint-based Association Mining

- Summary