

The background of the slide is a complex, abstract composition. It features a network of thin, reddish-brown lines forming a web-like structure. Scattered throughout are numerous small, colored dots in shades of green, blue, and orange. On the left side, there is a vertical strip with a grid of small, light-colored squares. In the center, a large, white, angular shape resembling a stylized 'V' or a folded piece of paper is superimposed over the other elements. The overall aesthetic is technical and data-driven.

Mining Multiple-Level Associations

Mining Multiple-Level Frequent Patterns

- Items often form hierarchies

- Ex.: Dairyland 2% milk;
Wonder wheat bread

- How to set min-support thresholds?

Uniform support

Level 1
min_sup = 5%

Level 2
min_sup = 5%

Milk
[support = 10%]

2% Milk
[support = 6%]

Skim Milk
[support = 2%]

Reduced support

Level 1
min_sup = 5%

Level 2
min_sup = 1%

- Uniform min-support across multiple levels (reasonable?)
 - Level-reduced min-support: Items at the lower level are expected to have lower support
- Efficient mining: *Shared* multi-level mining
 - Use the lowest min-support to pass down the set of candidates

Redundancy Filtering at Mining Multi-Level Associations

- Multi-level association mining may generate many redundant rules

- Redundancy filtering: Some rules may be redundant due to “ancestor” relationships between items

(Suppose the 2% milk sold is about $\frac{1}{4}$ of milk sold in gallons)

- $\text{milk} \Rightarrow \text{wheat bread}$ [support = 8%, confidence = 70%] (1)

- $2\% \text{ milk} \Rightarrow \text{wheat bread}$ [support = 2%, confidence = 72%] (2)

- A rule is *redundant* if its support is close to the “expected” value, according to its “ancestor” rule, and it has a similar confidence as its “ancestor”

- Rule (1) is an ancestor of rule (2), which one to prune?

(2)可以由(1)派生。
∴ (2)是冗余。

Customized Min-Supports for Different Kinds of Items

- We have used the same min-support threshold for all the items or item sets to be mined in each association mining
- In reality, some items (e.g., diamond, watch, ...) are valuable but less frequent
- It is necessary to have customized min-support settings for different kinds of items
- One Method: Use group-based “individualized” min-support 根据分组来分配 ms.
 - E.g., {diamond, watch}: 0.05%; {bread, milk}: 5%; ...
 - How to mine such rules efficiently?
 - Existing scalable mining algorithms can be easily extended to cover such cases