# Association Analysis (4)
# (Data Engineering)

# Type of attributes in assoc. analysis

- Association rule mining assumes the input data consists of binary attributes called items.
  - The presence of an item in a transaction is also assumed to be more important than its absence.
  - As a result, an item is treated as an asymmetric binary attribute.
- Now we extend the formulation to data sets with symmetric binary, categorical, and continuous attributes.

Internet survey data with categorical attributes.

| Gender | Level of Education | State | Computer at Home | Chat Online | Shop Online | Privacy Concerns |
|---|---|---|---|---|---|---|
| Female | Graduate | Illinois | Yes | Yes | Yes | Yes |
| Male | College | California | No | No | No | No |
| Male | Graduate | Michigan | Yes | Yes | Yes | Yes |
| Female | College | Virginia | No | No | Yes | Yes |
| Female | Graduate | California | Yes | No | No | Yes |
| Male | College | Minnesota | Yes | Yes | Yes | Yes |
| Male | College | Alaska | Yes | Yes | Yes | No |
| Male | High School | Oregon | Yes | No | No | No |
| Female | Graduate | Texas | No | Yes | No | No |
| ... | ... | ... | ... | ... | ... | ... |

# Type of attributes

- **Symmetric binary attributes**
  - Gender
  - Computer at Home
  - Chat Online
  - Shop Online
  - Privacy Concerns
- **Nominal attributes**
  - Level of Education
  - State
- **Example of rules**:

  {Shop Online= Yes} → {Privacy Concerns = Yes}.

  This rule suggests that most Internet users who shop online are concerned about their personal privacy.

# Transforming attributes into Asymmetric Binary Attributes

- Create a new item for each distinct attribute-value pair.

- E.g., the nominal attribute Level of Education can be replaced by three binary items:
  - Education = College
  - Education = Graduate
  - Education = High School

- Binary attributes such as Gender are converted into a pair of binary items
  - Gender =Male
  - Gender = Female

# Data after binarizing attributes into "items"

| Male | Female | Education = Graduate | Education = College | ... | Privacy = Yes | Privacy = No |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | ... | 1 | 0 |
| 1 | 0 | 0 | 1 | ... | 0 | 1 |
| 1 | 0 | 1 | 0 | ... | 1 | 0 |
| 0 | 1 | 0 | 1 | ... | 1 | 0 |
| 0 | 1 | 1 | 0 | ... | 1 | 0 |
| 1 | 0 | 0 | 1 | ... | 1 | 0 |
| 1 | 0 | 0 | 1 | ... | 0 | 1 |
| 1 | 0 | 0 | 0 | ... | 0 | 1 |
| 0 | 1 | 1 | 0 | ... | 0 | 1 |
| ... | ... | ... | ... | ... | ... | ... |

# Handling Continuous Attributes

- **Solution**: Discretize

- Example of rule:

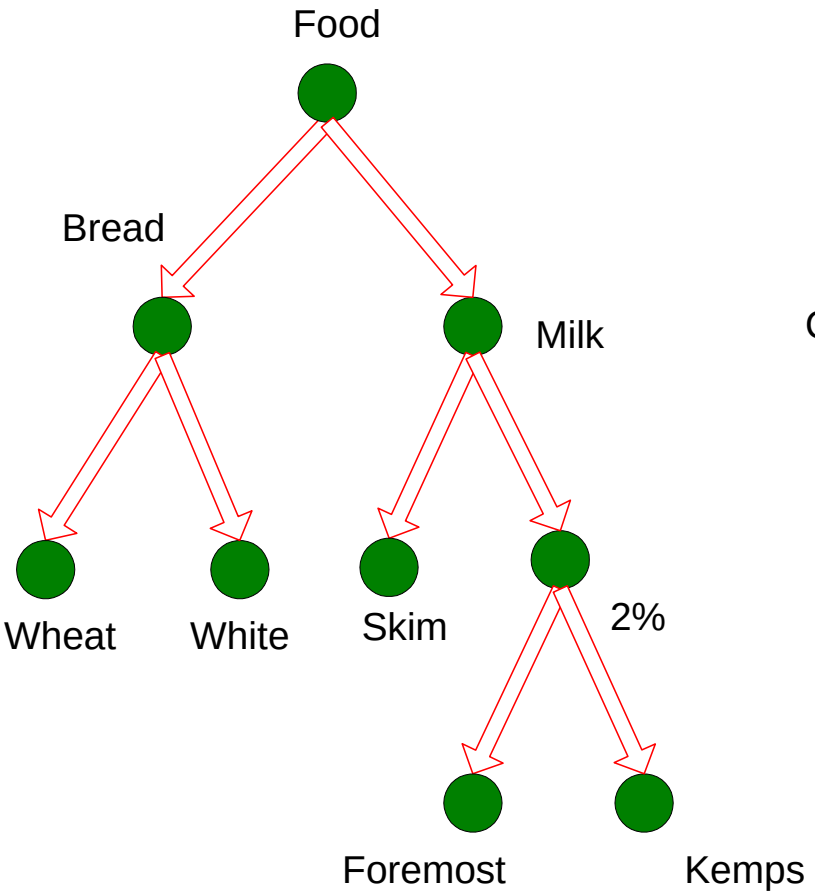
  $\text{Age} \in [21,35) \wedge \text{Salary} \in [70k,120k) \rightarrow \text{Buy}$

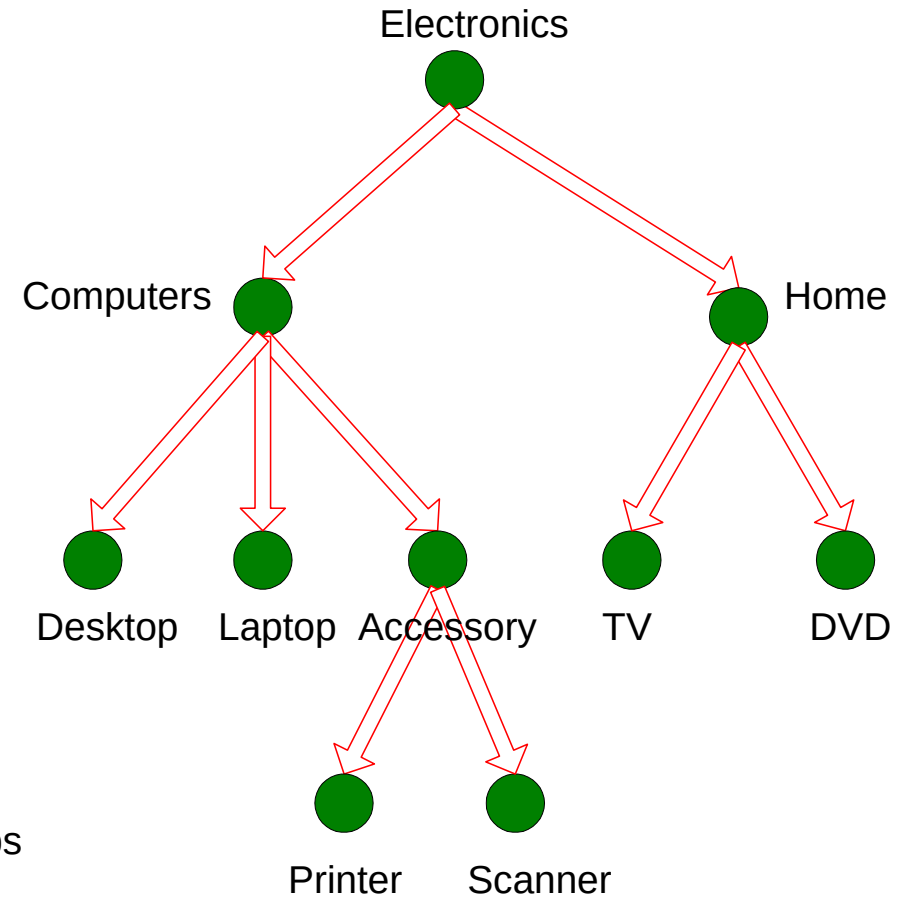
- Of course discretization isn't always easy.
  - If intervals too large, rule may not have enough confidence

    Age $\in$ [12,36) → Chat Online = Yes (s = 30%, c = 57.7%) (minconf=60%)
  - If intervals too small, rule may not have enough support

    Age $\in$ [16,20) → Chat Online = Yes (s = 4.4%, c = 84.6%) (minsup=15%)

# Concept Hierarchies



# Multi-level Association Rules

- Why should we incorporate a concept hierarchy?
  - Rules at lower levels may not have enough support to appear in any frequent itemsets
  - Rules at lower levels of the hierarchy are overly specific e.g.,

    skim milk $\rightarrow$ white bread,

    2% milk $\rightarrow$ wheat bread,

    skim milk $\rightarrow$ wheat bread, etc.

    are all indicative of association between milk and bread

# Multi-level Association Rules

- How do support and confidence vary as we traverse the concept hierarchy?

  - If $\sigma(X1 \cup Y1) \geq minsup$,
    and $X$ is parent of $X1$, $Y$ is parent of $Y1$
    then $\sigma(X \cup Y1) \geq minsup$

    $\sigma(X1 \cup Y) \geq minsup$

    $\sigma(X \cup Y) \geq minsup$

  - If $conf(X1 \Rightarrow Y1) \geq minconf$,
    then $conf(X1 \Rightarrow Y) \geq minconf$

# Multi-level Association Rules

**Approach 1**

- Extend current association rule formulation by augmenting each transaction with higher level items

  Original Transaction: {skim milk, wheat bread}

  Augmented Transaction:
  {skim milk, wheat bread, milk, bread, food}

- Issue:
  - Items that reside at higher levels have much higher support counts

    if support threshold is low, we get too many frequent patterns involving items from the higher levels

# Multi-level Association Rules

**Approach 2**

- Generate frequent patterns at highest level first.

- Then, generate frequent patterns at the next highest level, and so on.

- Issues:
  - May miss some potentially interesting **cross-level** association patterns. E.g.

    skim milk $\rightarrow$ white bread,

    2% milk $\rightarrow$ white bread,

    skim milk $\rightarrow$ white bread

    might not survive because of low support, but

    milk $\rightarrow$ white bread

    could.

    However, we don't generate a cross-level itemset such as

    {milk, white bread}

# Mining word associations (in Web)

**Document-term matrix:**

Frequency of words in a document

"**Itemset**" here is a collection of words

"**Transactions**" are the documents.

**Example:**

W1 and W2 tend to appear together in the same documents.

**Potential solution** for mining frequent itemsets:

Convert into 0/1 matrix and then apply existing algorithms

–Ok, but looses word frequency information

| TID | W1 | W2 | W3 | W4 | W5 |
|---|---|---|---|---|---|
| D1 | 2 | 2 | 0 | 0 | 1 |
| D2 | 0 | 0 | 1 | 2 | 2 |
| D3 | 2 | 3 | 0 | 0 | 0 |
| D4 | 0 | 0 | 1 | 0 | 1 |
| D5 | 1 | 1 | 1 | 0 | 2 |

# Normalize First

- How to determine the support of a word?
- First, normalize the word vectors
  - Each word has a support, which equals to 1.0
- Reason for normalization
  - Ensure that the data is on the same scale so that sets of words that vary in the same way have similar support values.

| TID | W1 | W2 | W3 | W4 | W5 |
|---|---|---|---|---|---|
| D1 | 2 | 20 | 0 | 0 | 1 |
| D2 | 0 | 0 | 1 | 2 | 2 |
| D3 | 2 | 30 | 0 | 0 | 0 |
| D4 | 0 | 0 | 1 | 0 | 1 |
| D5 | 1 | 10 | 1 | 0 | 2 |

Normalize

| TID | W1 | W2 | W3 | W4 | W5 |
|---|---|---|---|---|---|
| D1 | 0.40 | 0.33 | 0.00 | 0.00 | 0.17 |
| D2 | 0.00 | 0.00 | 0.33 | 1.00 | 0.33 |
| D3 | 0.40 | 0.50 | 0.00 | 0.00 | 0.00 |
| D4 | 0.00 | 0.00 | 0.33 | 0.00 | 0.17 |
| D5 | 0.20 | 0.17 | 0.33 | 0.00 | 0.33 |

# Association between words

- **E.g.** How to compute a "meaningful" normalized support for {W1, W2}?
- One might think to sum-up the average normalized supports for W1 and W2.

  s({W1,W2})

  = (0.4+0.33)/2 + (0.4+0.5)/2 + (0.2+0.17)/2

  = 1

| TID | W1 | W2 | W3 | W4 | W5 |
|---|---|---|---|---|---|
| D1 | 0.40 | 0.33 | 0.00 | 0.00 | 0.17 |
| D2 | 0.00 | 0.00 | 0.33 | 1.00 | 0.33 |
| D3 | 0.40 | 0.50 | 0.00 | 0.00 | 0.00 |
| D4 | 0.00 | 0.00 | 0.33 | 0.00 | 0.17 |
| D5 | 0.20 | 0.17 | 0.33 | 0.00 | 0.33 |

- This result is by no means an accident. Why?
- Averaging is useless here.

# Min-APRIORI

- Use instead the **min** value of normalized support (frequencies).

| TID | W1 | W2 | W3 | W4 | W5 |
|---|---|---|---|---|---|
| D1 | 0.40 | 0.33 | 0.00 | 0.00 | 0.17 |
| D2 | 0.00 | 0.00 | 0.33 | 1.00 | 0.33 |
| D3 | 0.40 | 0.50 | 0.00 | 0.00 | 0.00 |
| D4 | 0.00 | 0.00 | 0.33 | 0.00 | 0.17 |
| D5 | 0.20 | 0.17 | 0.33 | 0.00 | 0.33 |

**Example:**

s({W1,W2})

= min{0.4, 0.33} + min{0.4, 0.5}

+ min{0.2, 0.17}

= 0.9

s({W1,W2,W3})

= 0 + 0 + 0 + 0 + 0.17

= 0.17

# Anti-monotone property of Support

| TID | W1 | W2 | W3 | W4 | W5 |
|---|---|---|---|---|---|
| D1 | 0.40 | 0.33 | 0.00 | 0.00 | 0.17 |
| D2 | 0.00 | 0.00 | 0.33 | 1.00 | 0.33 |
| D3 | 0.40 | 0.50 | 0.00 | 0.00 | 0.00 |
| D4 | 0.00 | 0.00 | 0.33 | 0.00 | 0.17 |
| D5 | 0.20 | 0.17 | 0.33 | 0.00 | 0.33 |

**Example:**

**s({W1}) = 0.4 + 0 + 0.4 + 0 + 0.2 = 1**

**s({W1, W2}) = 0.33 + 0 + 0.4 + 0 + 0.17 = 0.9**

**s({W1, W2, W3}) = 0 + 0 + 0 + 0 + 0.17 = 0.17**

So, standard APRIORI algorithm can be applied.