HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

582364 Data mining, 4 cu
Lecture 6:
Quantitative association rules
Multi-level association rules

Spring 2010

Lecturer: Juho Rousu

Teaching assistant: Taru Itäpelto

Generalizing frequent pattern discovery

- So far we have discussed methods that discover frequent patterns from specific type of data
  - Asymmetric attributes: 0/1 data with lots of 0's
  - Binary data: item present/not present in the transaction
  - Transactions/Itemsets are unstructured ('flat'): baskets of items with arbitrary order

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Handling Continuous and Multi-valued Nominal Attributes

- In practice, we encounter a much more diverse set of attributes
  - Multi-valued nominal attributes
  - Ordered value ranges: ordinal, interval, ratio scale; real and integer numbers
  - Relational structure: temporal, spatial relationships, concept hierarchies

Session Id	Country	Session Length (sec)	Number of Web Pages viewed	Gender	Browser Type	Buy
1	USA	982	8	Male	IE	No
2	China	811	10	Female	Netscape	No
3	USA	2125	45	Female	Mozilla	Yes
4	Germany	596	4	Male	IE	Yes
5	England	123	9	Male	Mozilla	No
…	…	…	…	…	…	…

Handling Multi-Valued Nominal Attributes

- Transform categorical attribute into asymmetric binary variables
- Introduce a new “item” for each distinct attribute-value pair
  - Example: replace Browser Type attribute with
    - Browser Type = Internet Explorer
    - Browser Type = Mozilla
    - Browser Type = Mozilla

Handling Multi-Valued Nominal Attributes

- What if attribute has many possible values
  - Example: attribute country has more than 200 possible values
  - Many of the attribute values may have very low support
  - Potential solution: Aggregate the low-support attribute values:
    - Group by frequency alone: “Other” group
    - Group by some semantic connection: “Scandinavian countries”
    - Use of concept hierarchy and multi-level assocation rules
- What if distribution of attribute values is highly skewed
    - Example: assume 95% of the web site visitors are from USA
    - Most of the items will be associated with (Country=USA) item
      - Simple solution: drop the highly frequent items
    - Use of multiple minimum support & all-confidence measures (c.f Lecture 5)

Quantitative association rules

- Association rules that contain real or integer-valued attributes
- We look at two basic types of methods
  - Discretization-based methods for generating association rules
    - $Age \in [21,35) \wedge Salary \in [70k,120k) \rightarrow Buy$
  - Statistics-based methods for characterizing the sub-population coverered by the rule
    - $Salary \in [70k,120k) \wedge Buy \rightarrow Age: \mu=28, \sigma=4$

Discretization-based approach

- Split the range of the attribute into intervals using some discretization method
  - equal-width, equal frequency, clustering
- Generate one asymmetric binary attribute per interval
- Main problem is to choose the number and the boundaries of the intervals
  - Too wide intervals lead to loss of confidence in the association rules
  - Too narrow intervals lead to loss of support

Discretization example

- Consider thresholds
  - minsup = 5%
  - minconf = 65%
- The example data has two strong association rules embedded:
  - Age in [16,24) → Chat=Yes (s 8.8%, c 81.5%
  - Age in [44,60] →Chat=No (s 16.8%, c 70%)
- Discovering these rules requires getting the discretization of the age groups exactly right

Age group	Chat online = Yes	Chat online = No
[12,16)	12	13
[16,20)	11	2
[20,24)	11	3
[24,28)	12	13
[28,32)	14	12
[32,36)	15	12
[36,40)	16	14
[40,44)	16	14
[44,48)	4	10
[48,52)	5	11
[52,56)	5	10
[56,60)	4	11

Discretization example

- Too wide intervals lead to dropped confidence (<65%):
  - Age in [12,36) → Chat=Yes (s 30%, 57.7%
  - Age in [36,60] →Chat=No (s 28%, c 58.3%)
- Too narrow intervals lead to dropped support (<5%):
  - Age in [16,20) → Chat=Yes (s 4.4%, c 84.6%
  - Age in [20,24] →Chat=No (s 4.4%, c 78.6%)

Age group	Chat online = Yes	Chat online = No
[12,16)	12	13
[16,20)	11	2
[20,24)	11	3
[24,28)	12	13
[28,32)	14	12
[32,36)	15	12
[36,40)	16	14
[40,44)	16	14
[44,48)	4	10
[48,52)	5	11
[52,56)	5	10
[56,60)	4	11

Discretization example

- Intermediate sized intervals recover some of the embedded rules:
  - Age in [44,52) → Chat=No (s 8.4%, c 70%
  - Age in [52,60] →Chat=No (s 8.4%, c 70%)
  - Age in [12,20) → Chat=Yes (s 9.2%, c 60.5%
  - Age in [20,28] →Chat=Yes (s 9.2%, c 60%)
- By changing the interval lengths alone, recovering all patterns does not seem possible

Age group	Chat online = Yes	Chat online = No
[12,16)	12	13
[16,20)	11	2
[20,24)	11	3
[24,28)	12	13
[28,32)	14	12
[32,36)	15	12
[36,40)	16	14
[40,44)	16	14
[44,48)	4	10
[48,52)	5	11
[52,56)	5	10
[56,60)	4	11

Discretization example

- One way to circumvent this problem is to use all groupings of attribute values into intervals
  - [12,16],[12,20),[12,24),...[52,60),[56,60)
- This would recover our two strong rules:
  - Age in [16,24) → Chat=Yes (s 8.8%, c 81.5%
  - Age in [44,60] →Chat=No (s 16.8%, c 70%)
- However, a lot more candidates to examine!

Age group	Chat online = Yes	Chat online = No
[12,16)	12	13
[16,20)	11	2
[20,24)	11	3
[24,28)	12	13
[28,32)	14	12
[32,36)	15	12
[36,40)	16	14
[40,44)	16	14
[44,48)	4	10
[48,52)	5	11
[52,56)	5	10
[56,60)	4	11

Discretization Issues

- Execution time
  - If the attribute has *v* values existing in the database, there are $O(v^2)$ different intervals that can be created
  - Significant expansion of the data
- Potential to create redundant rules
  - If an interval I is frequent, all intervals J that contain I must be frequent as well

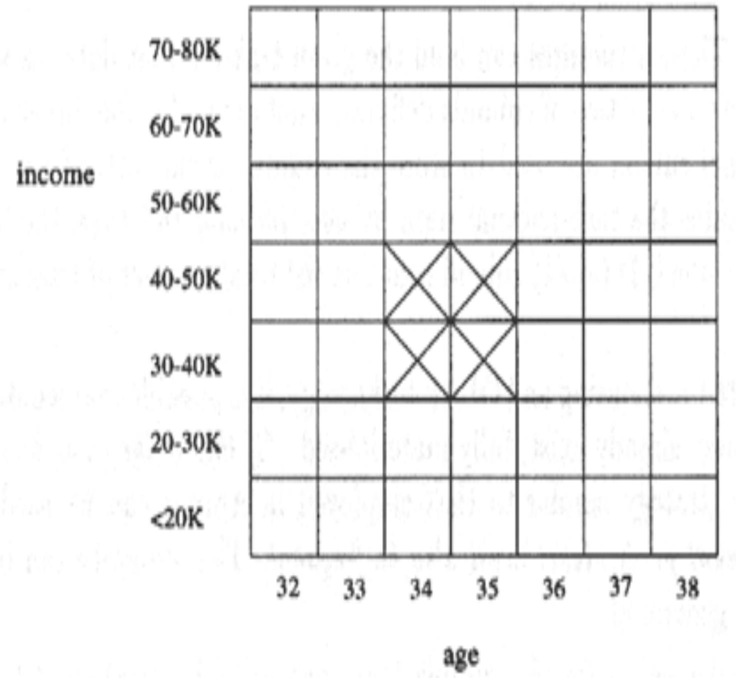
    {Refund = No, (Income = \$51,250)} → {Cheat = No}

    {Refund = No, (51K ≤ Income ≤ 52K)} → {Cheat = No}

    {Refund = No, (50K ≤ Income ≤ 60K)} → {Cheat = No}
- Methods that generate dynamically a smaller set of intervals exist, however they are out of the scope of this course

2D Discretization

- If the numerical attributes are correlated, discretizing two attributes at once may be beneficial
  - e.g Age and Income
- One approach is to use equi-width discretization to create a grid
- From the grid dense rectangles are extracted to form the left hand side of the rule
- The intervals extracted can change dynamically during the frequent pattern mining



age in [30-34) $\wedge$ income in [24K – 48K))
$\Rightarrow$ big screen TV

Statistics-based Methods

- Quantitative association rules can be used to infer statistical properties of a population
- Example:
  - Browser=Mozilla $\wedge$ Buy=Yes $\rightarrow$ Age: μ=23
  - Income > \$100K $\wedge$ Shop Online =Yes $\rightarrow$ Age: μ=38
- Rule right-hand side consists of a continuous variable, characterized by their statistics
  - mean, median, standard deviation, etc.
- Key issue in statistics-based methods is interestingness
  - Are the statistics of the sub-population covered by the rule significantly different from the rest of the population

Statistics-based Methods

- Example:

  Browser=Mozilla $\wedge$ Buy=Yes $\rightarrow$ Age: μ=23

- Approach:
  - Withhold the target variable (e.g. Age) from the rest of the data
  - Apply existing frequent itemset generation on the rest of the data
  - For each frequent itemset, compute the descriptive statistics for the corresponding target variable
    - Frequent itemset becomes a rule by introducing the target variable as rule right-hand side
  - Apply statistical test to determine interestingness of the rule

Statistics-based Methods: interestingness

- Compare the statistics for segment of population covered by the rule vs segment of population not covered by the rule:

  A $\Rightarrow$ B: μ versus not A $\Rightarrow$ B: μ'

- Statistical hypothesis testing:
  - s_1 and s_2 : standard deviations of the two populations
  - Δ is user-specified threshold for interesting difference
  - Null hypothesis: H0: $\mu' = \mu + \Delta$
  - Alternative hypothesis: H1: $\mu' > \mu + \Delta$
  - Z has zero mean and variance 1 under null hypothesis

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Statistics-based Methods

- Example:

  r: Browser=Mozilla $\wedge$ Buy=Yes $\rightarrow$ Age: μ=23

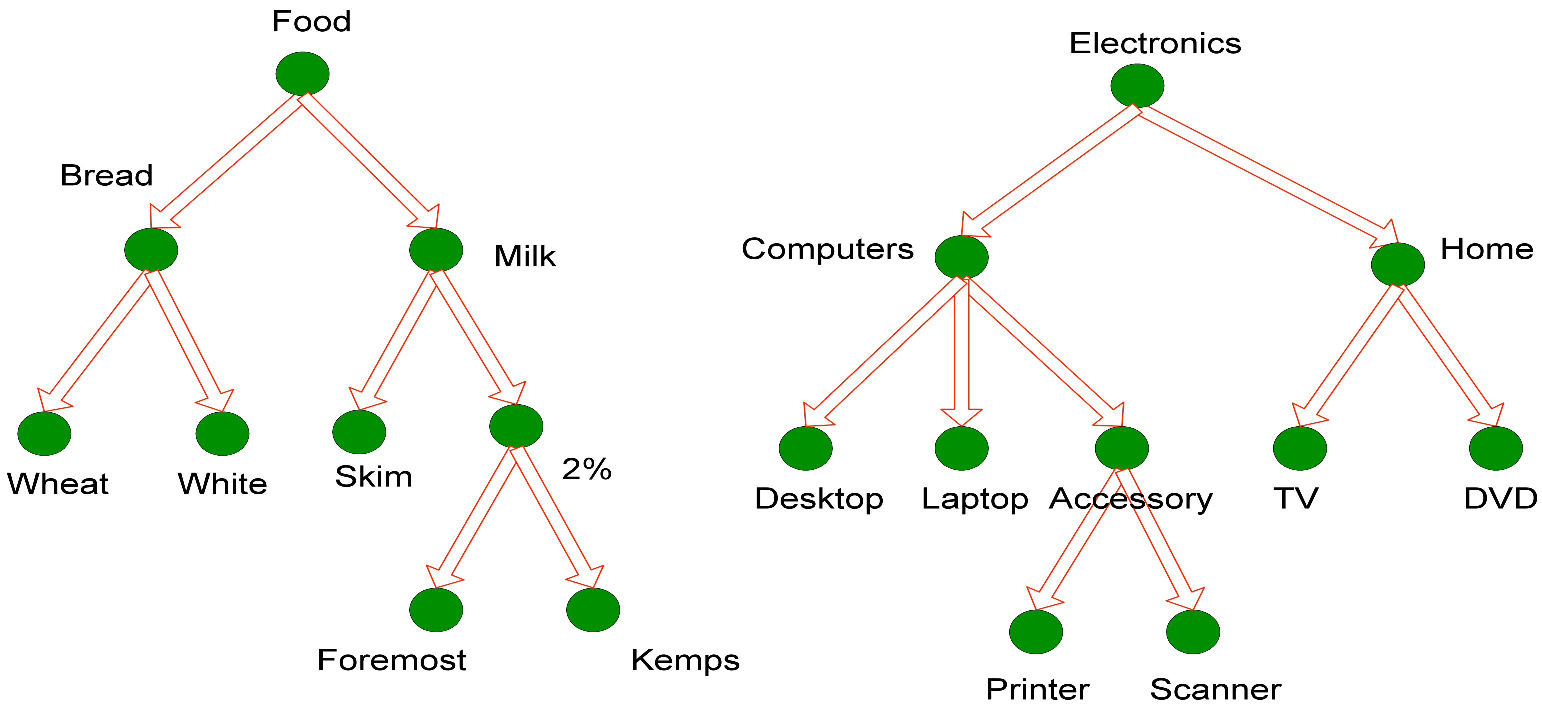
  - Rule is interesting if difference between μ and μ' is greater than 5 years (i.e., $\Delta = 5$)
  - For r, suppose $n_1 = 50$, $s_1 = 3.5$
  - For r' (complement): $n_2 = 250$, $s_2 = 6.5$

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = \frac{30 - 23 - 5}{\sqrt{\frac{3.5^2}{50} + \frac{6.5^2}{250}}} = 3.11$$

  - For 1-sided test at 95% confidence level (5% p-value), critical Z-value for rejecting null hypothesis is 1.64.
  - Since Z is greater than 1.64, r is an interesting rule

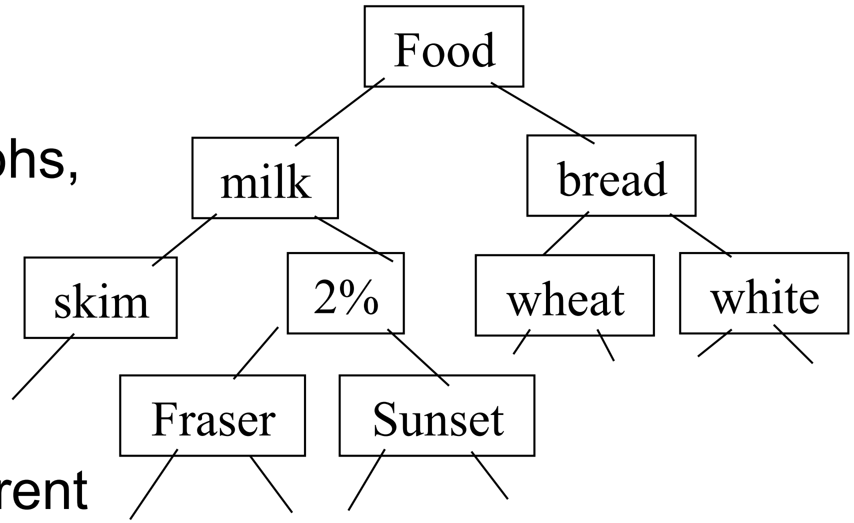
Handling concept hierarchies

- Organization of items in taxonomies is often encountered
- Typically the concept hierarchy is defined by domain knowledge
- Interesting associations may be contained in different levels
  - e.g. Milk → Bread, Skim Milk → Wheat Bread



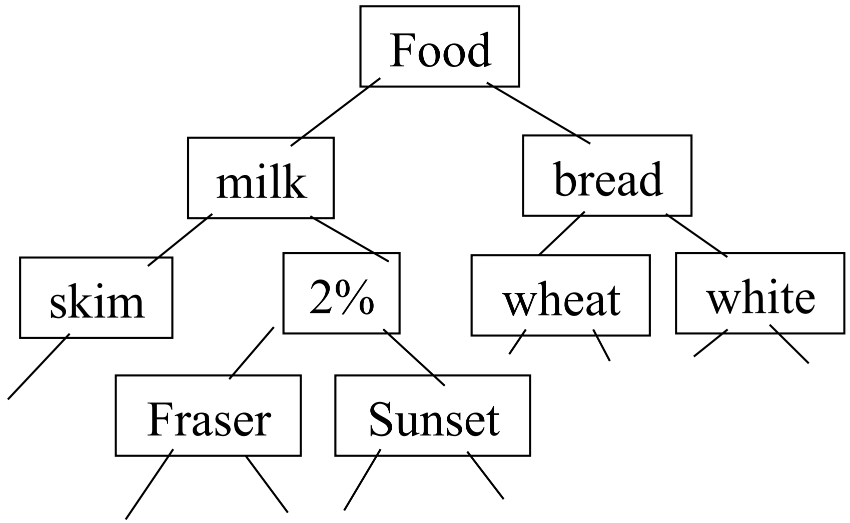
Handling concept hierarchies

- We look at concept hierarchies represented as directed acyclic graphs, where the edges represent an *is-a* relationship
  - e.g. 'Milk is-a Food'
- Given a edge (*p*,*q*), we call *p* the parent and *q* the child
- A node s is called an ancestor of node t if there is a directed path from s to t; t is called the descendant of s
  - e.g. 'Skim Milk' is a descendant of 'Food'



Transactions and concept hierarchies

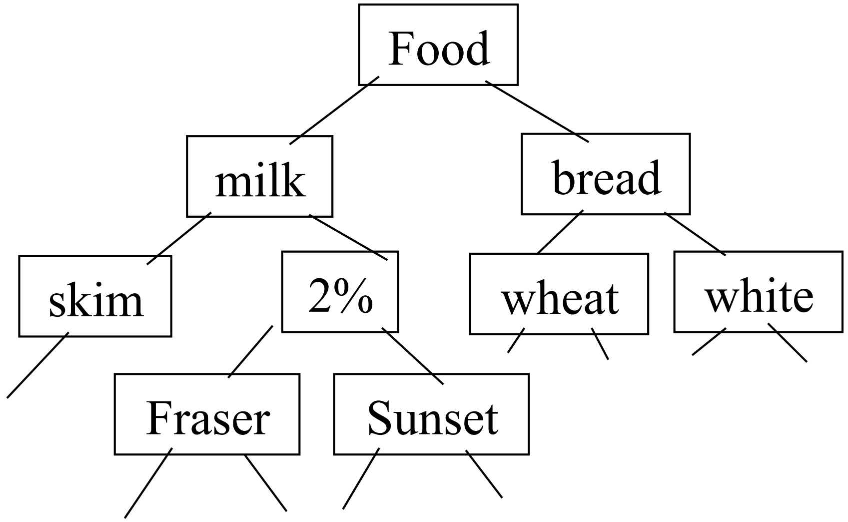
- Given a concept hierarchy, transactions become structured:
  - each item corresponds to a path from root to a leaf
    - E.g. (Food,Milk,Skim Milk), (Food,Bread,Wheat Bread)
- Representation options
  - Encode the higher levels as extra items
  - Encode the database in terms of the paths in the hierarchy



TID	Items
T1	{111, 121, 211, 221}
T2	{111, 211, 222, 323}
T3	{112, 122, 221, 411}
T4	{111, 121}
T5	{111, 122, 211, 221, 413}

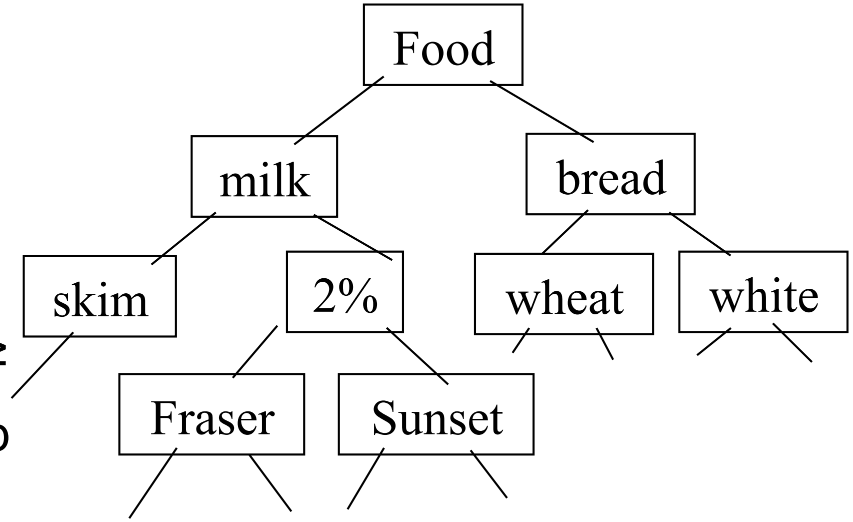
Support in concept hierarchies

- Support goes down monotonically as we travel a path from root to a leaf:
  - If X1 is the child of X, then $\sigma(X) \geq \sigma(X1)$
  - $\sigma(\text{Milk}) \geq \sigma(\text{Skim Milk})$
- If all items correspond to leaves, the support of a parent is the sum of children supports
  - If X has two children X1 and X2 then $\sigma(X) = \sigma(X1) + \sigma(X2)$
  - $\sigma(\text{Milk}) = \sigma(\text{Skim Milk}) + \sigma(\text{2\% Milk})$



Support in concept hierarchies

- In an itemset containing multiple items, moving in the same direction in all paths causes monotonic change in support
  - e.g. σ(Skim Milk, Wheat Bread) ≥ minsup then σ(Milk,Wheat Bread) ≥ minsup and σ(Milk,Bread) ≥ minsup
- Moves in opposite directions does not behave monotonically
  - σ(Milk, Wheat Bread) vs. σ(Skim Milk,Bread) can be ranked in any order by changing the underlying database

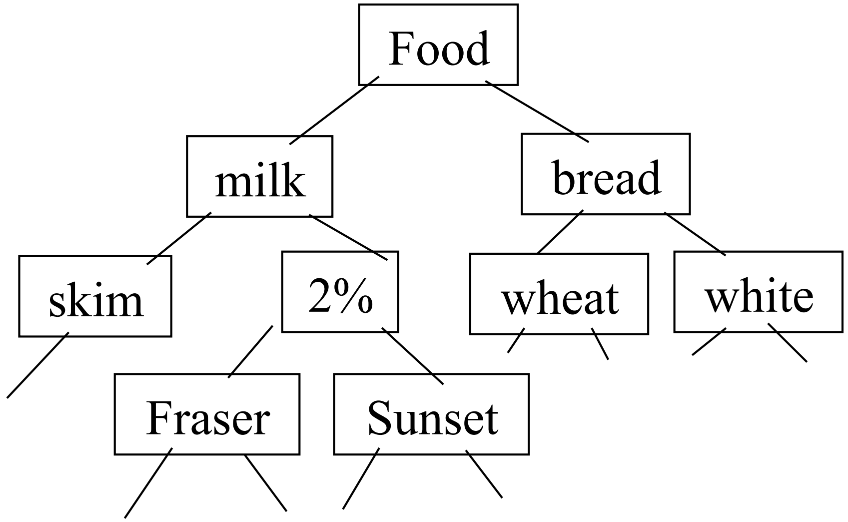


Confidence in concept hierarchies

- Confidence

  $c(X \rightarrow Y) = \sigma(X \cup Y)/\sigma(X)$

  goes monotonically up as we go up the hierarchy of the right-hand side itemset Y and keep left-hand side itemset X fixed
- e.g. if conf(Skim Milk → Wheat Bread) ≥ minconf then conf(Skim Milk →Bread) ≥ minconf



Properties of concept hierarchies

- Rules at lower levels may not have enough support to appear in any frequent itemsets
  - e.g. power adapter of particular mobile phone type
- Rules at lower levels of the hierarchy are overly specific
  - e.g., skim milk → white bread, 2% milk → wheat bread,

    skim milk → wheat bread, etc.

    are (probably) only indicative of association between milk and bread
- Rules at higher levels may become too general
  - e.g. electronics → food is probably not useful even though it satisfied the support and confidence thresholds
- Need a flexible approach to use the concept hierarchy

Mining multi-level association rules

- Assocation rules that contain the higher levels in the concept hierarchy are called multi-level association rules
- Simple approach: Augment each transaction with higher level items

  Original Transaction: {skim milk, wheat bread}

  Augmented Transaction:

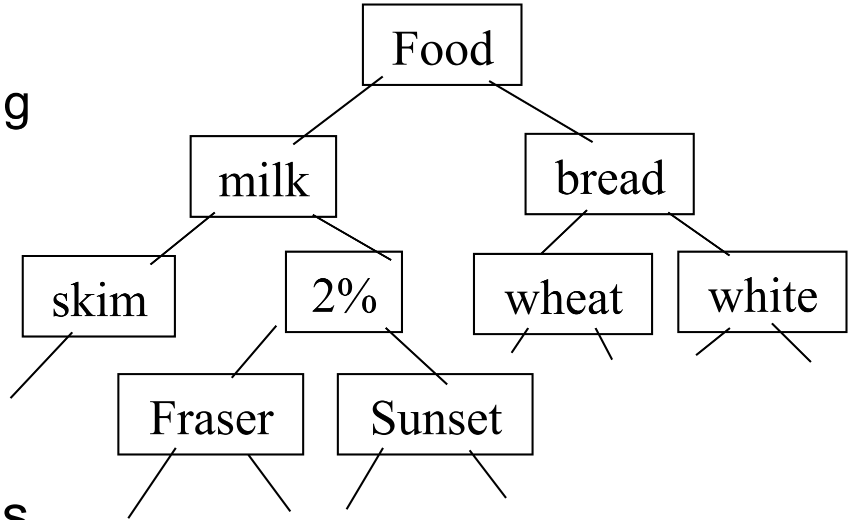
  {skim milk, wheat bread, milk, bread, food}
- Issues:
  - Items that reside at higher levels have much higher support counts
    - if support threshold is low, too many frequent patterns involving items from the higher levels
  - Increased dimensionality of the data

Mining Multi-level Association Rules

- Second approach uses a top-down exploration of the concept hierarchy
- Generate frequent patterns at highest level first:
  - e.g. milk → bread [20%, 60%].
- Then, generate frequent patterns at the next highest level
  - e.g 2% milk → wheat bread [6%, 50%]
- Continue deeper into the hierarchy until support goes below the *minsup* threshold
- Issues:
  - I/O requirements will increase dramatically because we need to perform more passes over the data
  - May miss some potentially interesting cross-level association patterns

Uniform Support vs. Reduced Support

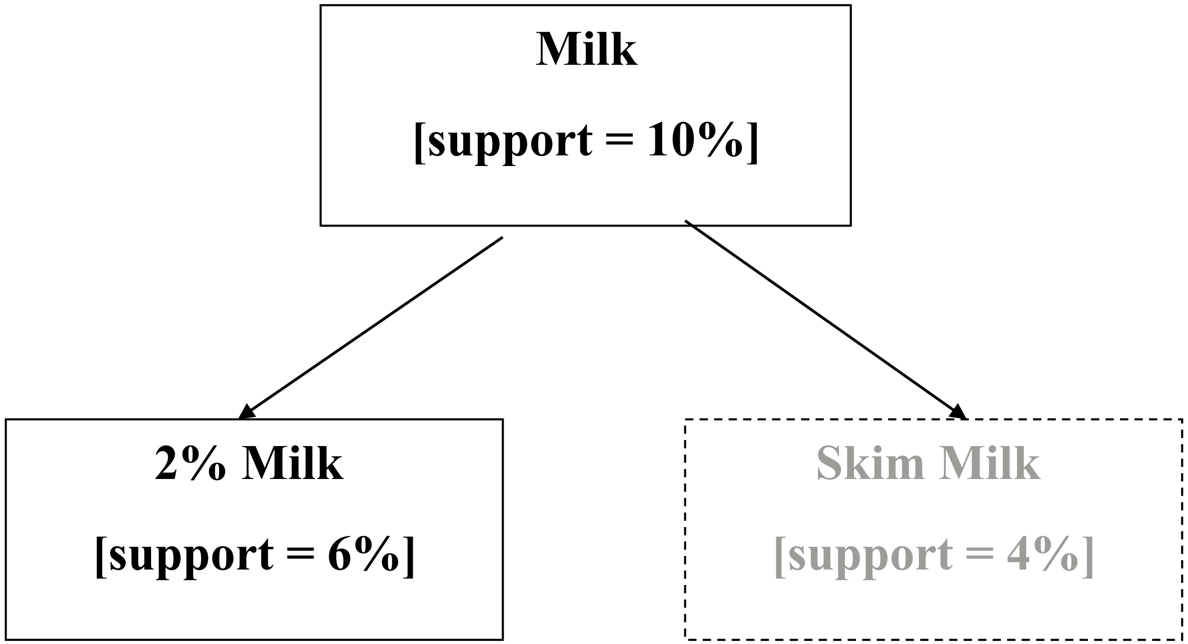
- The approach outlined uses a *uniform support threshold* for all levels
  - No need to examine itemsets containing any item whose ancestors do not have minimum support.
- A potential problem: Lower level items do not occur as frequently.
  - If support threshold too high ⇒ miss low level associations
  - too low ⇒ generate too many high level associations
- Alternative is to use a *reduced* minimum support at lower levels



Uniform Support: example

Level 1
min_sup = 5%

Level 2
min_sup = 5%

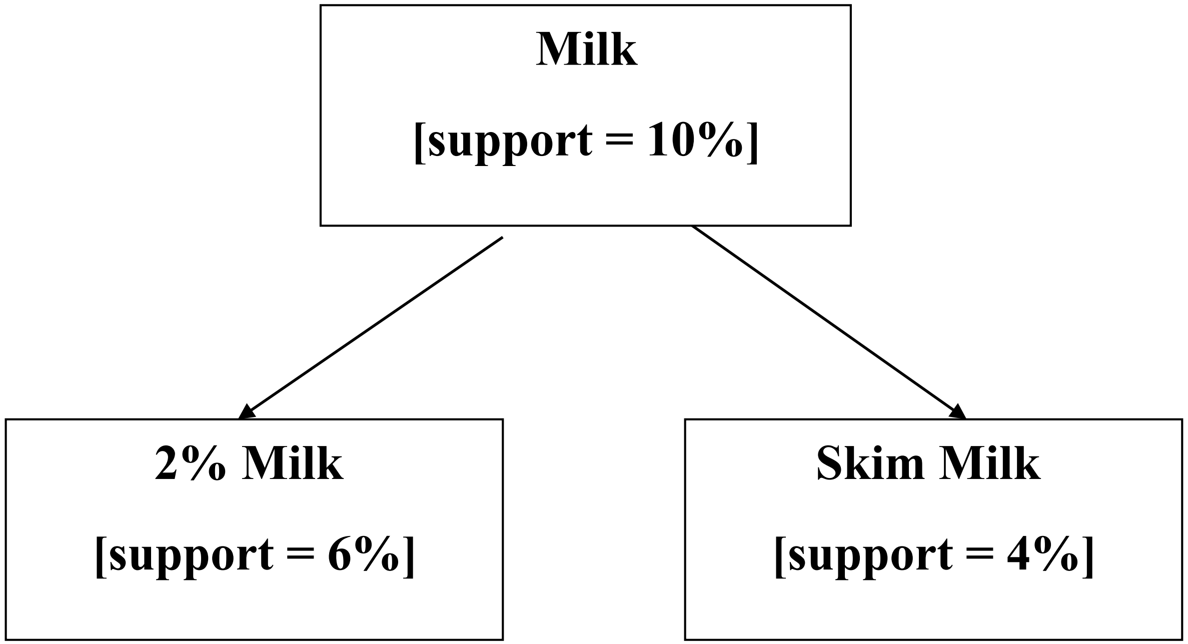


Back

Reduced Support: example

Level 1
min_sup = 5%

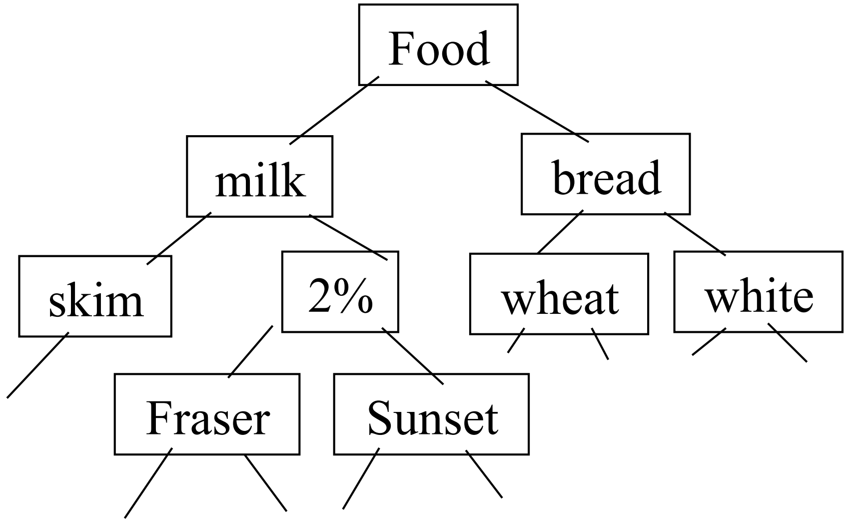
Level 2
min_sup = 3%



Back

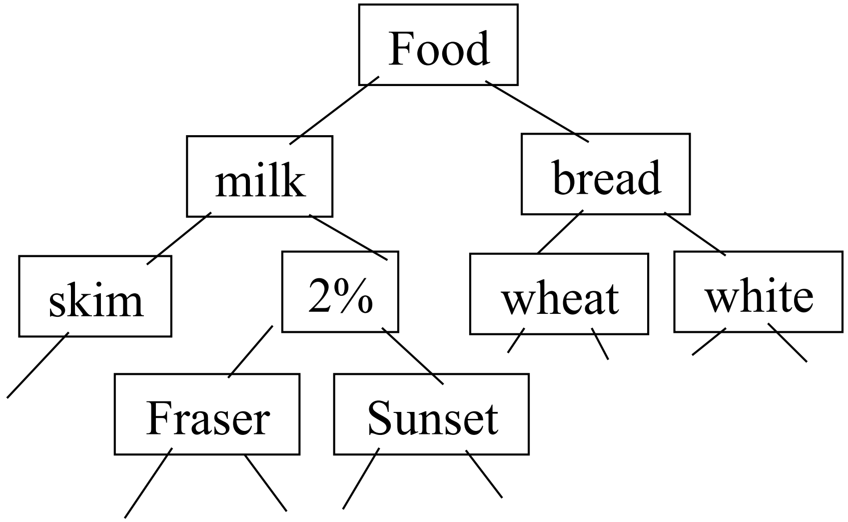
Reduced support: search strategies

- First stategy: Level-by-level independent
- Full breadth first search, children are examined regardless if parent was frequent
- e.g. itemsets containing Skim Milk would be searched even if itemsets containing Milk are all infrequent
- Rationale: since the minsup threshold is lower for Skim Milk it can still be a part of a frequent itemset
- However, causes a lot of exploration of lower levels of the hierarchy



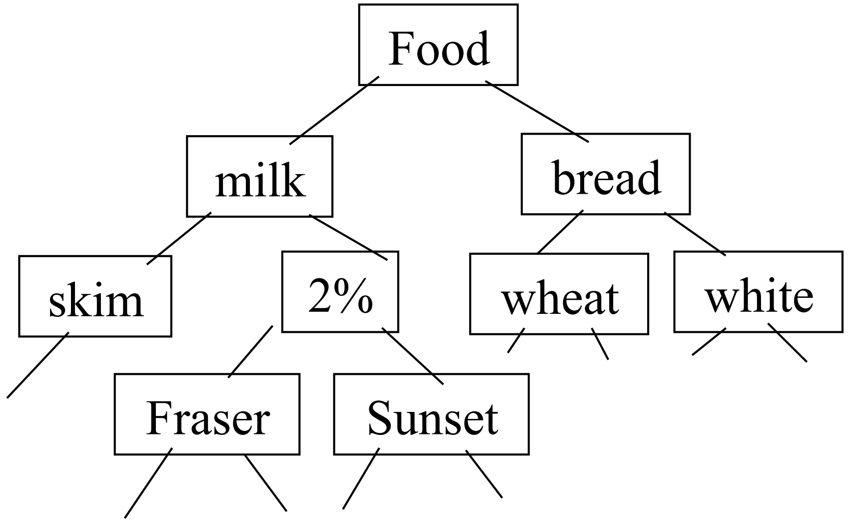
Reduced support: search strategies

- Second strategy: Level-cross filtering by single item
- Examine itemsets containing child (e.g. Skim Milk) if parent (Milk) is frequent, otherwise prune the subtrees below from search
- Prunes the search space more effectively than the level-by-level independent
- May miss some associations, where the reduced minimum support requirement makes the lower level item frequent



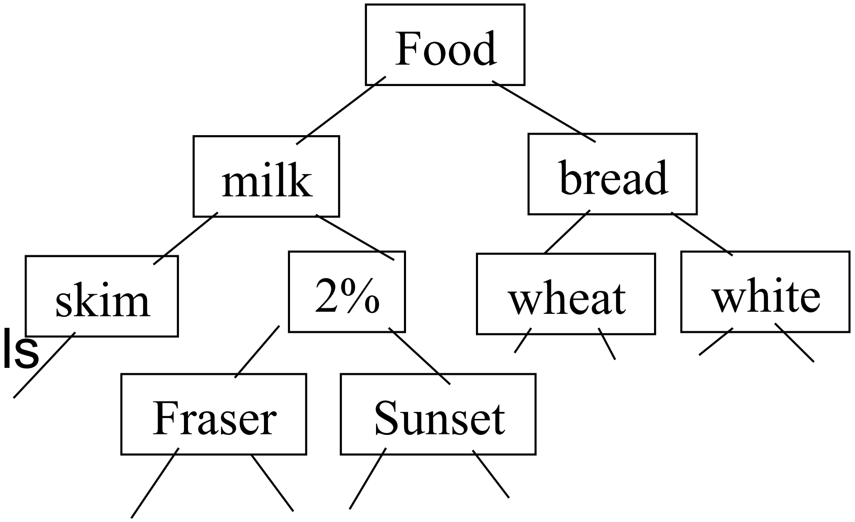
Reduced support: search strategies

- Third strategy: Level-cross filtering by k-itemset
- Examine a k-itemset on level i if the corresponding itemsets on level i-1 is frequent, otherwise prune the subtrees below from search
  - e.g. Examine {Skim Milk, Wheat Bread} only if {Milk, Bread} is frequent
- Heaviest pruning of the search, thus most efficient, but also misses more itemsets



Mining Cross-level Association Rules

- The approaches above mine for rules that lie on a single level of the hierachy
  - {Milk, Bread},{Skim Milk, Wheat Bread}
- In cross-level association rules levels can mix
  - {Skim Milk, Bread}, {Milk, Wheat bread}
- Given a itemset with items on different levels, take the minsup threshold of the deepest level as the thereshold to be used in pruning



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.
  - If 2% Milk accounts for 25% of sales of Milk, then the second rule does not carry new information