BITS Pilani
Hyderabad Campus

BITS Pilani

Dr. Manik Gupta
Assistant Professor
Department of CSIS

innovate achieve lead

BITS Pilani
Hyderabad Campus

Data Mining (CS F415)
Lecture 13 – Advanced Concepts in Association Analysis

Tuesday, 11th February 2020

Today's Agenda

- Handling categorical attributes
- Handling continuous attributes
- Multi level associations

Categorical and Continuous Attributes

How to apply association analysis to non-asymmetric binary variables?

Gender	...	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	...	26	90K	20	4	Yes
Male	...	51	135K	10	2	No
Male	...	29	80K	10	3	Yes
Female	...	45	120K	15	3	Yes
Female	...	31	95K	20	5	Yes
Male	...	25	55K	25	5	Yes
Male	...	37	100K	10	1	No
Male	...	41	65K	8	2	No
Female	...	26	85K	12	1	No
...	...	...	...	...	...	...

Example of Association Rule:

$$\{\text{Gender=Male, Age} \in [21,30)\} \rightarrow \{\text{No of hours online} \geq 10\}$$

Handling Categorical Attributes

- Example: Internet Usage Data
- {Level of Education=Graduate, Online Banking=Yes} → {Privacy Concerns = Yes}

Gender	Level of Education	State	Computer at Home	Online Auction	Chat Online	Online Banking	Privacy Concerns
Female	Graduate	Illinois	Yes	Yes	Daily	Yes	Yes
Male	College	California	No	No	Never	No	No
Male	Graduate	Michigan	Yes	Yes	Monthly	Yes	Yes
Female	College	Virginia	No	Yes	Never	Yes	Yes
Female	Graduate	California	Yes	No	Never	No	Yes
Male	College	Minnesota	Yes	Yes	Weekly	Yes	Yes
Male	College	Alaska	Yes	Yes	Daily	Yes	No
Male	High School	Oregon	Yes	No	Never	No	No
Female	Graduate	Texas	No	No	Monthly	No	No
...	...	...	...	...	...	...	...

What are the symmetric binary and nominal attributes here??

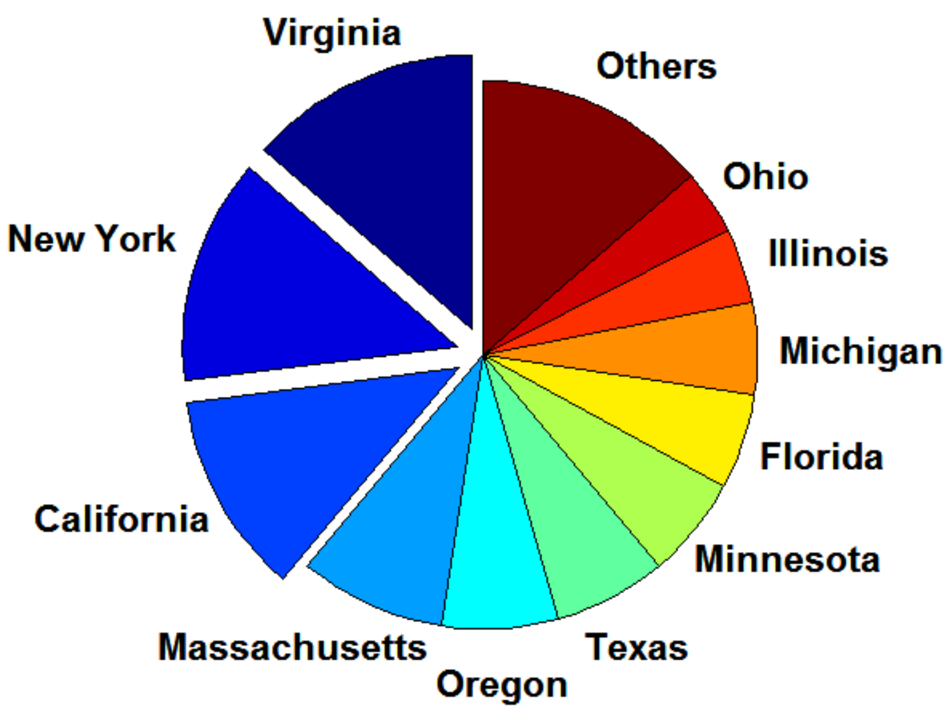
Handling Categorical Attributes

- Introduce a new "item" for each distinct attribute-value pair

Male	Female	Education = Graduate	Education = College	Education = High School	...	Privacy = Yes	Privacy = No
0	1	1	0	0	...	1	0
1	0	0	1	0	...	0	1
1	0	1	0	0	...	1	0
0	1	0	1	0	...	1	0
0	1	1	0	0	...	1	0
1	0	0	1	0	...	1	0
1	0	0	0	0	...	0	1
1	0	0	0	1	...	0	1
0	1	1	0	0	...	0	1
...	...	...	...	...	...	...	...

Handling Categorical Attributes

- Some attributes can have many possible values
  - Many of their attribute values have very low support
  - Potential solution: Aggregate the low-support attribute values



Handling Categorical Attributes

- Distribution of attribute values can be highly skewed
  - Example: 85% of survey participants own a computer at home
    - Most records have Computer at home = Yes
    - Leads to generation of redundant rules
    - Computation becomes expensive; many frequent itemsets involving the binary item (Computer at home = Yes)
- Computational Complexity
  - Binarizing the data increases the number of items
  - But the width of the "transactions" remain the same as the number of original (non-binarized) attributes
  - Produce more frequent itemsets, but maximum size of frequent itemset is limited to the number of original attributes

Handling Continuous Attributes

- Association Rules that contain continuous attributes are called **quantitative association rules.**
  - Used to infer statistical properties of a population
- Different methods:
  – Discretization-based
  – Statistics-based
  – Non-discretization based
    - minApriori
- Different kinds of rules can be produced:
  – {Age$\in$[21,30), No of hours online$\in$[10,20)} $\rightarrow$ {Chat Online =Yes}
  – {Age$\in$[21,30), Chat Online = Yes} $\rightarrow$ No of hours online: μ=14, σ=4

Discretization-based Methods

- Groups adjacent values into a finite number of intervals
- The discrete intervals can be mapped into asymmetric binary attributes so that existing association analysis algorithms can be applied.

Gender	...	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	...	26	90K	20	4	Yes
Male	...	51	135K	10	2	No
Male	...	29	80K	10	3	Yes
Female	...	45	120K	15	3	Yes
Female	...	31	95K	20	5	Yes
Male	...	25	55K	25	5	Yes
Male	...	37	100K	10	1	No
Male	...	41	65K	8	2	No
Female	...	26	85K	12	1	No
...	...	...	...	...	...	...

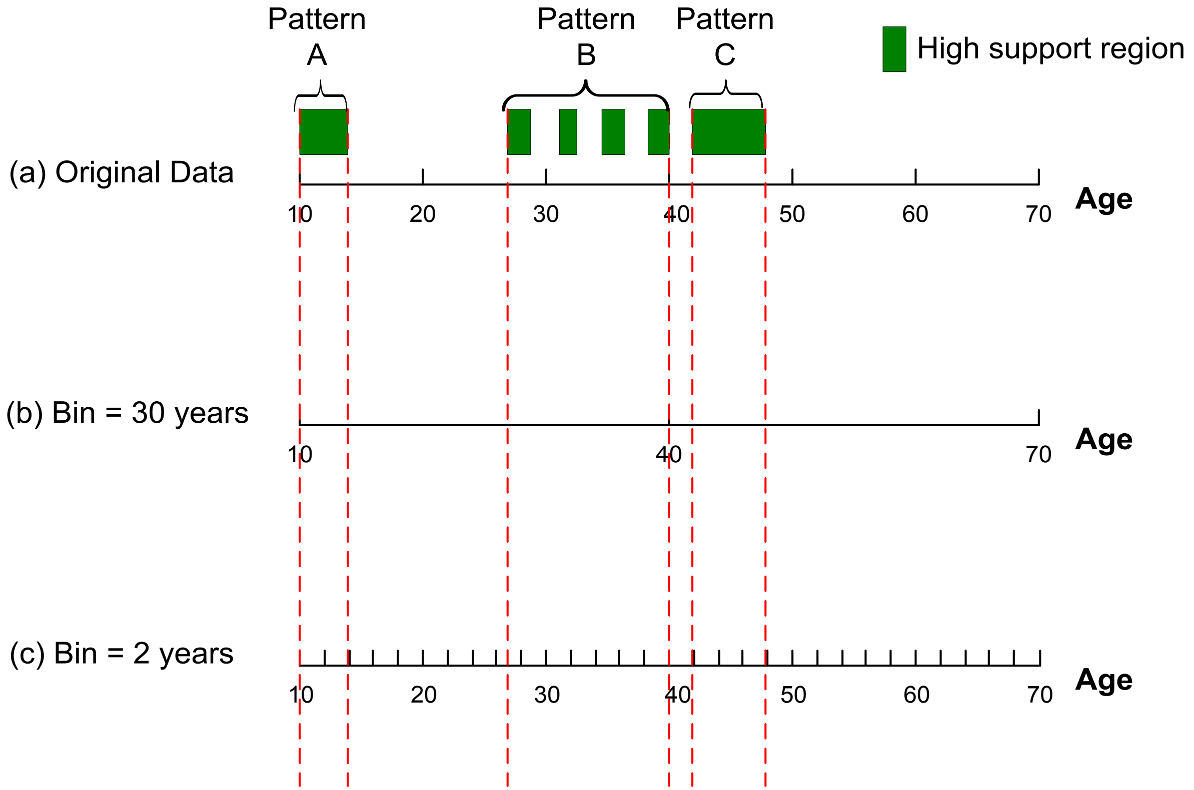
Male	Female	... ...	Age < 13	Age $\in [13, 21)$	Age $\in [21, 30)$	... ...	Privacy $=$ Yes	Privacy $=$ No
0	1	...	0	0	1	...	1	0
1	0	...	0	0	0	...	0	1
1	0	...	0	0	1	...	1	0
0	1	...	0	0	0	...	1	0
0	1	...	0	0	0	...	1	0
1	0	...	0	0	1	...	1	0
1	0	...	0	0	0	...	0	1
1	0	...	0	0	0	...	0	1
0	1	...	0	0	1	...	0	1
...	...	...	...	...	...	...	...	...

Discretization-based Methods

- Unsupervised discretization
  - Equal Interval binning
  - Equal frequency binning
- Supervised discretization

Discretization Issues

Key parameter is number of intervals to partition each attribute



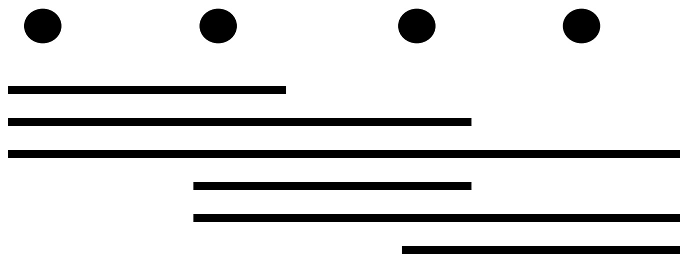
$$\text{Pattern A:} \quad \text{Age} \in [10, 15) \longrightarrow \text{Chat Online} = \text{Never}$$
$$\text{Pattern B:} \quad \text{Age} \in [26, 41) \longrightarrow \text{Chat Online} = \text{Never}$$
$$\text{Pattern C:} \quad \text{Age} \in [42, 48) \longrightarrow \text{Online Banking} = \text{Yes}$$

Discretization Issues

- Interval too wide (e.g., Bin size= 30)
  - May merge several disparate patterns
    - Patterns A and B are merged together
  - May lose some of the interesting patterns
    - Pattern C may not have enough confidence
- Interval too narrow (e.g., Bin size = 2)
  - Pattern A is broken up into two smaller patterns
    - Can recover the pattern by merging adjacent subpatterns
  - Pattern B is broken up into smaller patterns
    - Cannot recover the pattern by merging adjacent subpatterns
  - Some windows may not meet support threshold

Discretization: all possible grouping of adjacent intervals

Number of intervals within a range = k
Total number of adjacent intervals = k(k-1)/2



Execution time

– If the range is partitioned into k intervals, there are $O(k^2)$ new items
– If an interval [a,b) is frequent, then all intervals that subsume [a,b) must also be frequent
  - E.g.: if {Age $\in$ [21,25), Chat Online=Yes} is frequent,
    then {Age $\in$ [10,50), Chat Online=Yes} is also frequent
– Generate too many candidate and frequent itemsets

Discretization Issues

- Redundant rules are generated
  - R1: {Age $\in$[18,20), Age $\in$[10,12)} $\rightarrow$ {Chat Online=Yes}
  - R2: {Age $\in$[18,23), Age $\in$[10,20)} $\rightarrow$ {Chat Online=Yes}
  - If both rules have the same support and confidence, prune the more specific rule (R1) and retain the generalized rule (R2 has a wider interval for age attribute).

Statistics-based Methods

- Quantitative association rules can be used to infer the statistical properties of a population
- Rule consequent consists of a continuous variable and can be characterized by their statistics
  – mean, median, standard deviation, etc.
- Example:
  – {Income > 100K, Online Banking=Yes} → Age: μ=34

Statistics-based Methods

- Approach
  - **Rule Generation** - Specify the target attribute used to characterize interesting segments of the population
    - Withhold the target attribute from the rest of the data
  - Extract frequent itemsets from the rest of the attributes
    - Binarize the continuous attributes (except for the target attribute)
  - For each frequent itemset that identifies an interesting segment of the population, compute the corresponding descriptive statistics of the target attribute
    - Frequent itemset becomes a rule by introducing the target variable as rule consequent
  - **Rule Validation** - Apply statistical test to determine interestingness of the rule

Statistics-based Methods

Gender	...	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	...	26	90K	20	4	Yes
Male	...	51	135K	10	2	No
Male	...	29	80K	10	3	Yes
Female	...	45	120K	15	3	Yes
Female	...	31	95K	20	5	Yes
Male	...	25	55K	25	5	Yes
Male	...	37	100K	10	1	No
Male	...	41	65K	8	2	No
Female	...	26	85K	12	1	No
...	...	...	...	...	...	...

Frequent Itemsets:

{Male, Income > 100K}

{Income < 30K, No hours ∈[10,15)}

{Income > 100K, Online Banking = Yes}

....

Association Rules:

{Male, Income > 100K} → Age: μ = 30

{Income < 40K, No hours ∈[10,15)} → Age: μ = 24

{Income > 100K,Online Banking = Yes} → Age: μ = 34

....

Statistics-based Methods

- How to determine whether an association rule interesting?
  - Rule is interesting only if statistics computed from transactions covered by rule are different than those computed from transactions not covered by the rule
  - Statistical hypothesis testing should be applied to determine whether difference is statistically different or no

Statistics-based Methods

- 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 $\overline{A} \Rightarrow B$: μ'
- Goal is to test whether the difference between μ and μ' is greater than a user specified threshold Δ
- Statistical hypothesis testing:
  - Null hypothesis: H0: $\mu' = \mu + \Delta$
  - Alternative hypothesis: H1: $\mu' > \mu + \Delta$
  - To determine which hypothesis should be accepted, Z-statistic is computed as follows:

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

Statistics-based Methods

- Example:
  - r: Browser=Mozilla $\wedge$ Buy=Yes $\rightarrow$ Age: μ=23
  - Assume that the rule is interesting if difference between μ and μ' is more than 5 years (i.e., Δ = 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, critical Z-value for rejecting null hypothesis is 1.64.
  - Since Z is greater than 1.64, the null hypothesis can be rejected and hence, r is an interesting rule

Non discretization methods - Min-Apriori

- Finding associations among continuous attributes
- Finding word associations in text documents using a document-term matrix

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

Example:

W1 and W2 tends to appear together in the same document

Min-Apriori

- Data contains only continuous attributes of the same "type"
  - e.g., frequency of words in document

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

- Potential solution:
  - Convert into 0/1 matrix and then apply existing algorithms
    - lose word frequency information
  - Discretization does not apply as users want association among words (e.g. data and mining) not ranges of words frequencies (e.g. data $\in$ [1,4] and mining $\in$ [2,3])

Min-Apriori

- How to determine the support of a word?
  - If we simply sum up its frequency, support count will be greater than total number of documents!
    - Normalize the word vectors – e.g., divide frequency of each word by sum of word frequencies across all documents
    - Each word has a support equals to 1.0

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 →

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

Min-Apriori

- In min-apriori, the association among words in a given document is obtained by taking minimum value of their normalized frequencies.
- Support of an itemset is computed by aggregating its association across all the documents
- New definition of support:

$$\sup(C) = \sum_{i \in T} \min_{j \in C} D(i, j)$$

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:

Sup(W1,W2,W3)

= 0 + 0 + 0 + 0 + 0.17

= 0.17

Anti-monotone property of Support

Support has properties that makes it suitable for finding word associations in documents.

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:

– Sup(W1) = 0.4 + 0 + 0.4 + 0 + 0.2 = 1

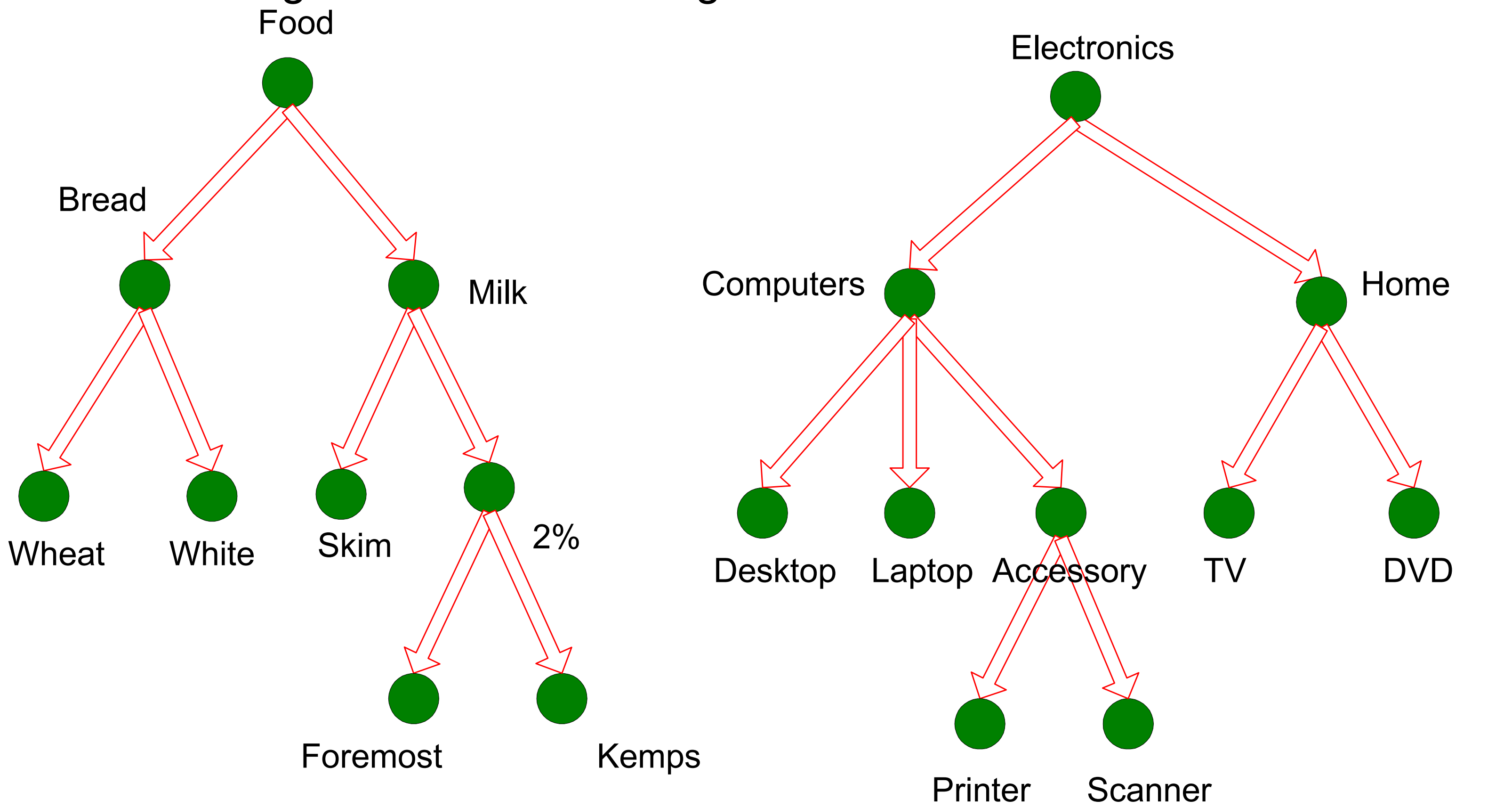
– Sup(W1, W2) = 0.33 + 0 + 0.4 + 0 + 0.17 = 0.9

– Sup(W1, W2, W3) = 0 + 0 + 0 + 0 + 0.17 = 0.17

Support decreases monotonically as the number of words in an itemset increases

Concept Hierarchies

- Multilevel organization of the various entities or concepts defined in a particular domain.
- Defined according to domain knowledge or a standard classification scheme



Multi-level Association Rules

Why should we incorporate concept hierarchy?

- Rules at lower levels may not have enough support to appear in any frequent itemsets and hence there is a **chance to miss interesting patterns.**
- 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 indicative of association between milk and bread
- Rules at higher level of hierarchy may be too **generic**
  - If milk and batteries are the only items sold together frequently, the pattern {food, electronics} may overgeneralize the situation

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}

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 and hence increased computation time
- Redundant rules may be produced involving items from lower level of hierarchy

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:

- 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

Thanks!

Next Lecture:

- Subsequence mining

Readings:

  - Chapter 7 - Tan & Kumar