

Association Rules Mining

Extending Association Analysis

Mining Massive Datasets

Materials provided by Prof. Carlos Castillo — <https://chato.cl/teach>

Instructor: Dr. Teodora Sandra Buda — <https://tbuda.github.io/>

Sources

- Data Mining, The Textbook (2015) by Charu Aggarwal (Chapters 4, 5)
 - See [slides by Lijun Zhang](#)
- Introduction to Data Mining 2nd edition (2019) by Tan et al. (Chapter 6)
 - See [slides ch6](#)

Alternative to minsup itemsets
and minconf rules:
interesting itemsets and rules

Advantages of frequent itemsets

- Simple and intuitive
 - Support computed from frequency
 - Confidence computed from conditional probabilities
- Efficient
 - Downward closure property enables efficient algorithms

Disadvantages of frequent itemsets

- Patterns **are not always significant** from an application-specific perspective
- {Milk} can be appended to any itemset and does not change its frequency
- For any itemset X , the rule $X \Rightarrow \{\text{Milk}\}$ has 100% confidence

tid	Set of items
1	$\{Bread, Butter, Milk\}$
2	$\{Eggs, Milk, Yogurt\}$
3	$\{Bread, Cheese, Eggs, Milk\}$
4	$\{Eggs, Milk, Yogurt\}$
5	$\{Cheese, Milk, Yogurt\}$

Disadvantages of frequent itemsets

(cont.)

- Cannot adjust to the **skew** in the individual item support values
 - Support of {Milk, Butter} is different from $\{\neg\text{Milk}, \neg\text{Butter}\}$
 - We would like a method that treats both presence and absence similarly

tid	Set of items
1	$\{Bread, Butter, Milk\}$
2	$\{Eggs, Milk, Yogurt\}$
3	$\{Bread, Cheese, Eggs, Milk\}$
4	$\{Eggs, Milk, Yogurt\}$
5	$\{Cheese, Milk, Yogurt\}$

Statistical coefficient of correlation

- Pearson coefficient

$$\rho = \frac{E[X \cdot Y] - E[X] \cdot E[Y]}{\sigma(X)\sigma(Y)}$$

- Estimated correlation

$$\rho_{ij} = \frac{\text{sup}(\{i, j\}) - \text{sup}(i) \cdot \text{sup}(j)}{\sqrt{\text{sup}(i) \cdot \text{sup}(j) \cdot (1 - \text{sup}(i)) \cdot (1 - \text{sup}(j))}}$$

- Properties

- Lines in $[-1, 1]$; is symmetric wrt presence/absence
- It is hard to understand intuitively

χ^2 measure

- Given a k-itemset X, there are 2^k possible states
 - 2-itemset:
{Bread, Butter}
 - 2^2 states:
{Bread, Butter}, {¬Bread, Butter},
{Bread, ¬Butter}, {¬Bread, ¬Butter}

χ^2 measure (cont.)

- The χ^2 measure for an itemset: $\chi^2(X) = \sum_{i=1}^{2^{|X|}} \frac{(O_i - E_i)^2}{E_i}$
 - O_i observed support, E_i expected support
- Properties
 - Larger values \Rightarrow greater dependence
 - Treats presence and absence similarly
 - High computational complexity
 - Does not satisfy the downward closure property

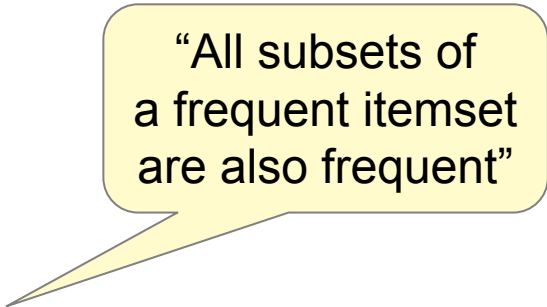
“All subsets of a frequent itemset are also frequent”

Interest ratio

- Definition
$$I(\{i_1, \dots, i_k\}) = \frac{\sup(\{i_1, \dots, i_k\})}{\prod_{j=1}^k \sup(i_j)}$$
- Properties
 - When items are statistically independent $\Rightarrow 1$
 - Value >1 indicates positive correlations
 - Misleading if some items are extremely rare
 - Does not satisfy the downward closure property

Symmetric confidence

- In general $conf(X \Rightarrow Y) \neq conf(Y \Rightarrow X)$
- We can symmetrize:
 - Minimum, maximum, average of both
 - Geometric mean: cosine measure
- Can be generalized to k-itemsets
- Does not satisfy the downward closure property



“All subsets of a frequent itemset are also frequent”

Cosine measure (cosine coefficient) on columns

$$\text{cosine}(i, j) = \frac{\text{sup}(\{i, j\})}{\sqrt{\text{sup}(i)} \cdot \sqrt{\text{sup}(j)}}$$

- Interpretation: cosine between two columns of the data matrix, which we interpret as vectors
- It is symmetric

Prove this is the
geometric mean
of $\text{conf}(i \Rightarrow j)$ and $\text{conf}(j \Rightarrow i)$

$$\left(\prod_{i=1}^n x_i \right)^{\frac{1}{n}}$$

Cosine as geometric mean (answer)

$$\text{cosine}(i, j) = \frac{\text{sup}(\{i, j\})}{\sqrt{\text{sup}(i)} \cdot \sqrt{\text{sup}(j)}}$$

- Note this is $\text{sqrt}(\text{conf}(i \Rightarrow j) \times \text{conf}(j \Rightarrow i))$, i.e., the geometric mean

Jaccard Coefficient

- Jaccard coefficient

$$J(S_1, \dots, S_k) = \frac{|\cap S_i|}{|\cup S_i|}$$

- Properties

- Satisfies downward closure property

$$J(S_1, \dots, S_k) \geq J(S_1, \dots, S_k, S_{k+1})$$

Why?

Why this satisfies downward closure? (Answer)

- Jaccard coefficient (generalization)

$$J(S_1, \dots, S_k) = \frac{|\cap S_i|}{|\cup S_i|}$$

← Cannot increase with k
← Cannot decrease with k

- Properties

- **Satisfies downward closure property**

$$J(S_1, \dots, S_k) \geq J(S_1, \dots, S_k, S_{k+1})$$

Jaccard Coefficient

- Jaccard coefficient (generalization)

$$J(S_1, \dots, S_k) = \frac{|\cap S_i|}{|\cup S_i|}$$

- Properties

- Satisfies downward closure property

$$J(S_1, \dots, S_k) \geq J(S_1, \dots, S_k, S_{k+1})$$

- Speed up by min-hash trick

Collective strength

- An itemset is in violation in a transaction if **some, but not all** of the items in I are present in the transaction
- Violation rate $v(I)$
 - Fraction of violations of itemset I over all transactions

Collective strength (cont.)

$$C(I) = \frac{1 - v(I)}{1 - E[v(I)]} \cdot \frac{E[V(I)]}{v(I)}$$

- The expected value of $v(I)$ is calculated assuming statistical independence of the individual items

$$E[v(I)] = 1 - \prod_{i \in I} p_i - \prod_{i \in I} (1 - p_i)$$

- 0 indicates perfect negative correlation
- ∞ indicates perfect positive correlation

Collective strength (cont.)

- Simple interpretation of collective strength

$$C(I) = \frac{1 - v(I)}{1 - E[v(I)]} \cdot \frac{E[V(I)]}{v(I)} = \frac{\text{Good Events}}{E[\text{Good Events}]} \cdot \frac{E[\text{Bad Events}]}{\text{Bad Events}}$$

- An itemset I is said to be **strongly collective** at level s , if it satisfies the following properties:
 - The collective strength $C(I)$ of the itemset I is at least s .
 - **Closure property**: The collective strength $C(J)$ of every subset J of I is at least s .

Useful techniques/tricks
(meta algorithms)

Sampling

1. **Sample** a subset of the transactions
2. Apply mining to sample data
3. Profit!

However, there can be:

false positives (frequent in sample but not overall) or
false negatives (frequent overall but not in the sample)

Data partitioned ensembles

1. **Partition** the DB into k disjoint segments
2. Apply the mining algorithm **independently** on each segment
3. Post-process to remove false positives

There are never false negatives

Extension to categorical attributes

Other types of association analysis

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 the kind of rule we want to mine:
- $\{\text{Gender}=\text{Male}, \text{Age} \in [21, 30)\} \Rightarrow \{\text{Hours online} \geq 10\}$

Other types of association (cont.)

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
...

- Example of the kind of rule we want to mine:
- {Level of Education=Graduate, Online Banking=Yes}
⇒ {Privacy Concerns = Yes}

Solution: a *dummy* binary variable for each attribute=value pair

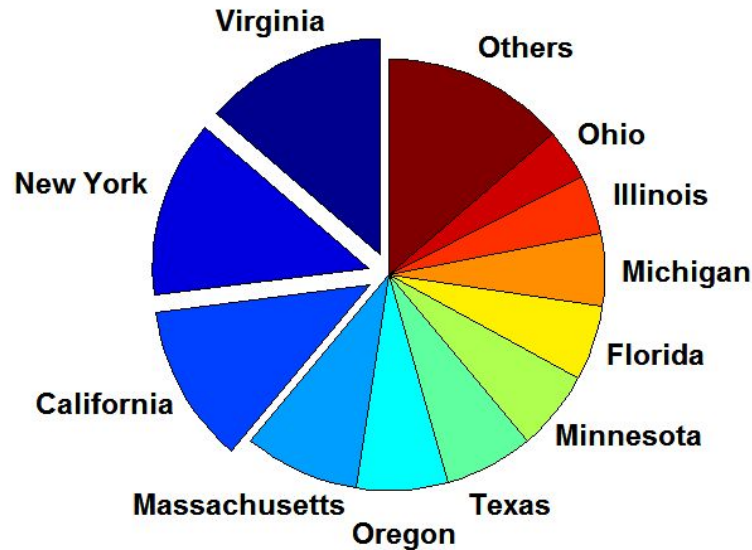
Gender	Level of Education	State	Computer at Home	Online Auction	Chat Online	Online Banking	Privacy Concerns
Female	Graduate	Illinois	Yes	Yes	Daily	Yes	Yes
Male	College	California	No	No	Never	No	No
Male	Graduate	Michigan	Yes	Yes	Monthly	Yes	Yes
Female	College						
Female	Graduate						
Male	College	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

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
...

Problem:

too infrequent attribute values

It is not necessary to add an extra column for attribute values that rarely appear. Bundle them.



Problem:

too frequent attribute values

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
- Computation becomes expensive; many frequent itemsets involving this
- Potential solution: **discard the highly frequent items**

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

Extension to continuous attributes

Handling continuous attributes

- Different methods:
 - Discretization-based
 - Statistics-based
 - Non-discretization based (e.g., minApriori)
- Different kinds of rules can be produced:
- $\{\text{Age} \in [21, 30), \text{No of hours online} \in [10, 20)\}$
 $\Rightarrow \{\text{Chat Online} = \text{Yes}\}$
- $\{\text{Age} \in [21, 30), \text{Chat Online} = \text{Yes}\}$
 - $\Rightarrow \text{No of hours online: } \mu=14, \sigma=4$

Discretization-based methods

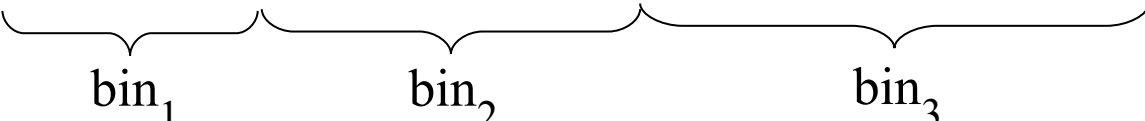
Gender	...	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern					
Female	...	26	90K	20	4	Yes					
Male	...	51	135K	10	2	No					
Male	...	29	80K	10	3	Yes					
Female	...	45	120K	15	3	Yes					
Female	...	31	95K	20	5	Yes					
Male	...	25	55K	25	5	Yes					
Male	...	37	100K	10	1	No					
Male	...	41									
Female	...	26									
...									
			Male	Female	...	Age < 13	Age ∈ [13, 21)	Age ∈ [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
		

Types of discretization

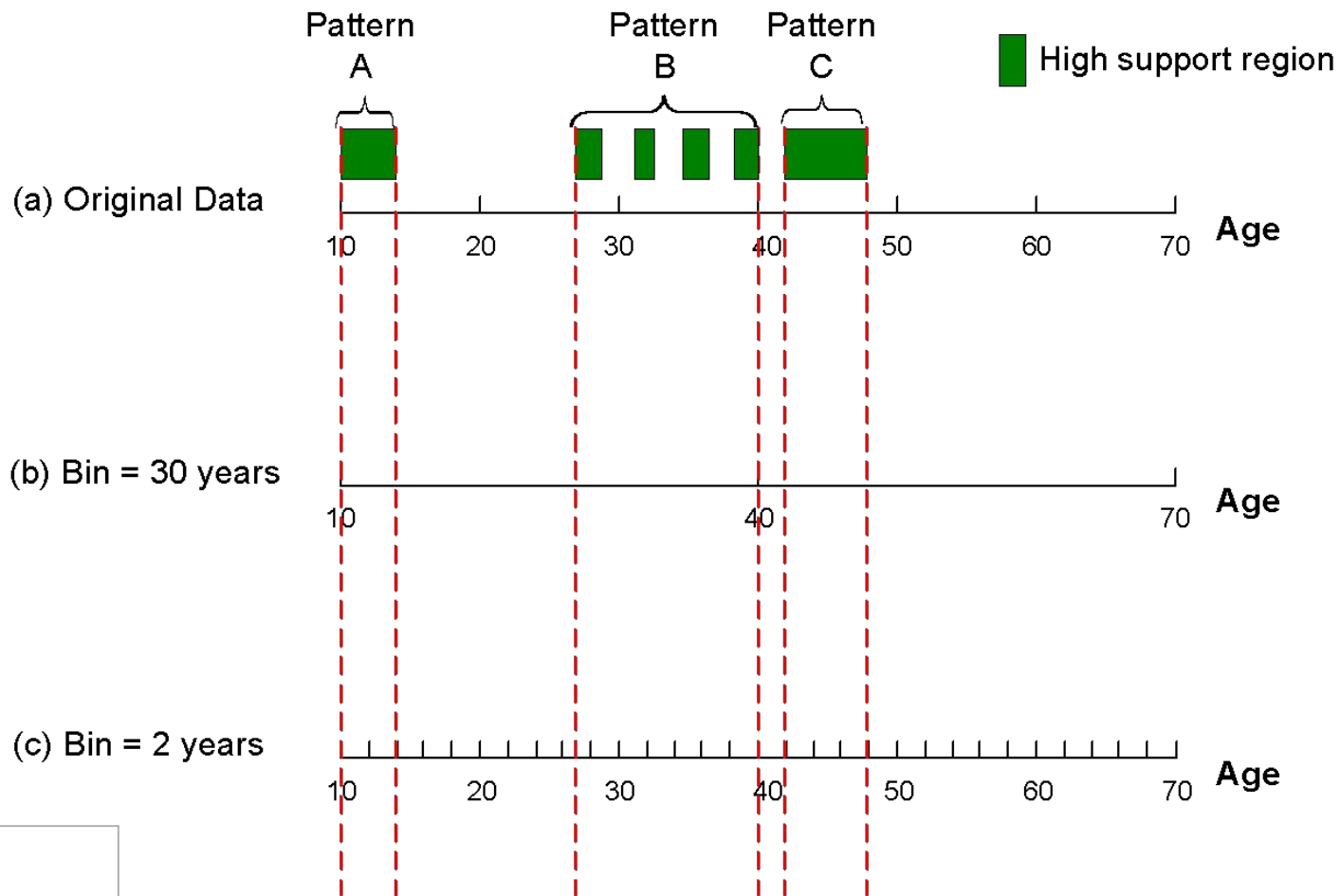
- Unsupervised discretization
 - Equal-width (max-min in each bin is constant)
 - Equal-depth (number of elements per bin is constant)
 - Clustering-based
- Supervised discretization

Continuous attribute, v

	1	2	3	4	5	6	7	8	9
Chat Online = Yes	0	0	20	10	20	0	0	0	0
Chat Online = No	150	100	0	0	0	100	100	150	100



Interval width issues



Example patterns:



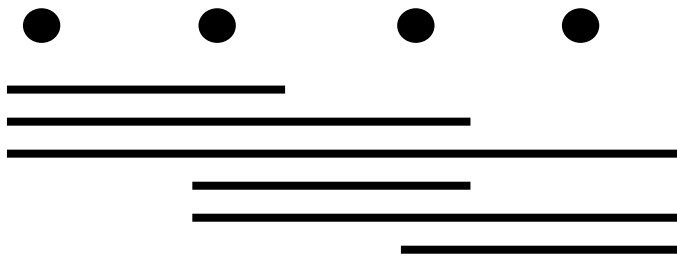
Interval width issues (cont.)

- **Interval too wide** (e.g., Bin size= 30 in the example of previous slide)
 - 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 in the example of previous slide)
 - 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 intervals may not meet support threshold

Discretization: all possible intervals

Number of intervals = 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 $\{\text{Age} \in [21,25), \text{Chat Online}=\text{Yes}\}$ is frequent,
then $\{\text{Age} \in [10,50), \text{Chat Online}=\text{Yes}\}$ is also frequent
- Improve efficiency:
 - Use maximum support to avoid intervals that are too wide

Handling redundant rules: keep the most general one

R1: $\{\text{Age} \in [18, 20), \text{Age} \in [10, 12)\} \Rightarrow \{\text{Chat Online} = \text{Yes}\}$

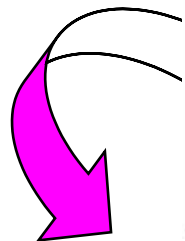
R2: $\{\text{Age} \in [18, 23), \text{Age} \in [10, 20)\} \Rightarrow \{\text{Chat Online} = \text{Yes}\}$

- If both rules have the same support and confidence,
 - prune (remove) the most specific rule (R1)
 - keep the most general rule (R2)

Statistics-based methods

- Example:
- $\{\text{Income} > 100\text{K}, \text{Online Banking}=\text{Yes}\} \Rightarrow \text{Age: } \mu=34$
- Approach
 - Remove target attribute (e.g., “Age”)
 - Extract frequent itemsets
 - For each frequent itemset X , create a rule **$X \Rightarrow \text{Age stats}$**
 - Apply statistical test to determine interestingness

Statistics-based methods (example)



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}

....

Candidate association rules:

{Male, Income > 100K} → Age: $\mu = 30$

{Income < 40K, No hours ∈ [10,15]} → Age: $\mu = 24$

**{Income > 100K, Online Banking = Yes}
→ Age: $\mu = 34$**

....

Statistics-based methods: what is an **interesting** candidate?

- Compare the statistics for segment covered by the rule vs rest of the population:

$A \Rightarrow B: \mu$ versus $A \Rightarrow B: \mu'$

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

- Statistical hypothesis testing:
 - Null hypothesis: $H_0: \mu' = \mu + \Delta$
 - Alternative hypothesis: $H_1: \mu' > \mu + \Delta$
 - Z has zero mean and variance 1 under null hypothesis

Finding an **interesting** candidate

- Example: Buy=Yes \Rightarrow Age $\mu=23$
- Suppose rule is interesting if difference between μ and μ' is more than 5 years ($\Delta = 5$)
 - For r , suppose $n_1 = 50$, $\sigma_1 = 3.5$, $\mu=23$
 - For r' (complement): $n_2 = 250$, $\sigma_2 = 6.5$, $\mu'=30$

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} = \frac{30 - 23 - 5}{\sqrt{\frac{3.5^2}{50} + \frac{6.5^2}{250}}} = 3.11$$

Finding an **interesting** candidate (cont.)

- Example: Buy=Yes \Rightarrow Age $\mu=23$

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} = \frac{30 - 23 - 5}{\sqrt{\frac{3.5^2}{60} + \frac{6.5^2}{250}}} = 3.11$$

- For 1-sided test at 95% confidence level, critical Z-value for rejecting null hypothesis is 1.64.
- Since $Z=3.11$ is greater than 1.64, r is an interesting rule

Min-Apriori

- Document-term matrix showing that W1 and W2 tend to appear together

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

Min-Apriori (cont.)

- What is the support of a word? Can't be frequency; instead, we normalize word vectors (columns add up to one)

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

Normalize



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

Min-Apriori (cont.)

- New definition of support

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

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

Min-Apriori (cont.)

- 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

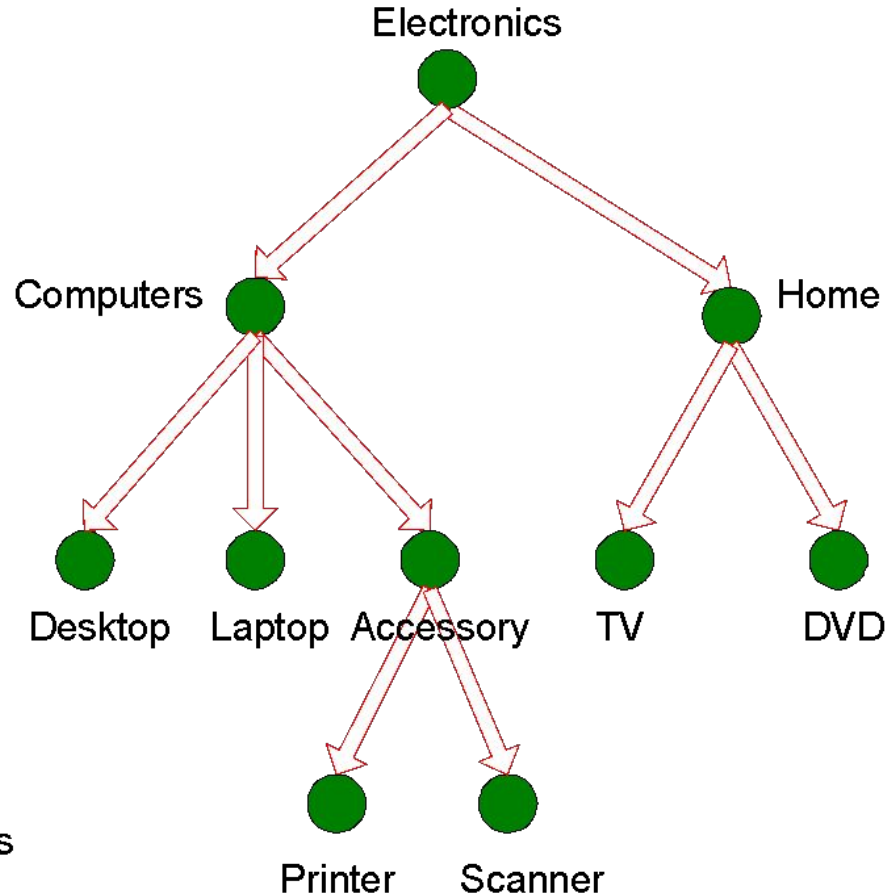
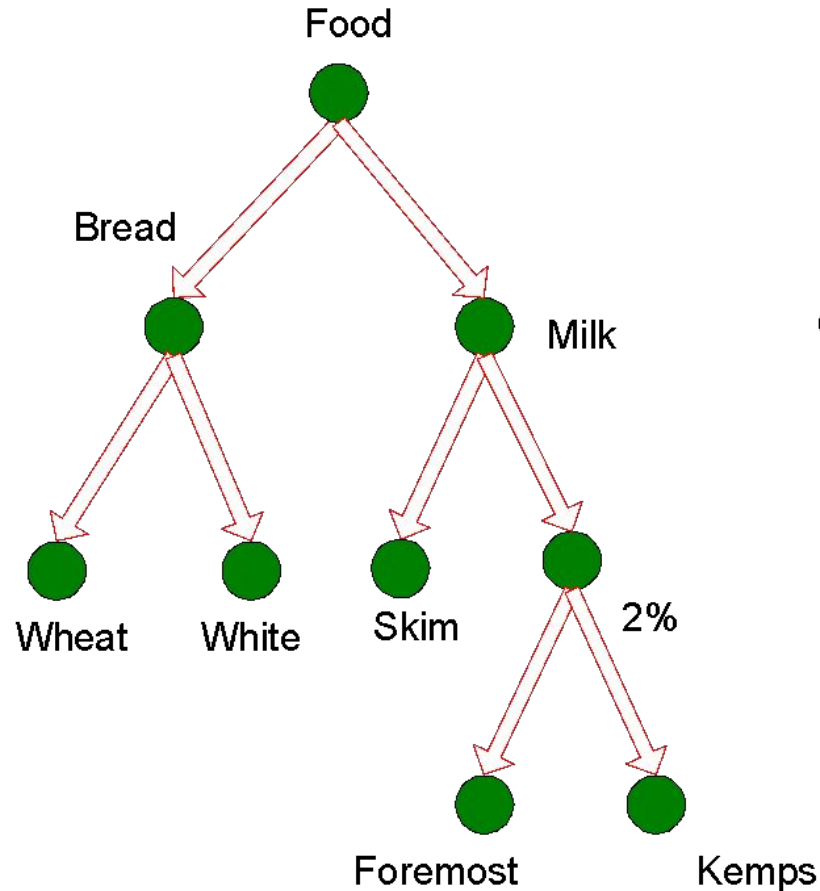
Example:

$$\text{Sup}(W1) = 0.4 + 0 + 0.4 + 0 + 0.2 = 1$$

$$\text{Sup}(W1, W2) = 0.33 + 0 + 0.4 + 0 + 0.17 = 0.9$$

$$\text{Sup}(W1, W2, W3) = 0 + 0 + 0 + 0 + 0.17 = 0.17$$

Categories in a hierarchy



Why should we incorporate 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., soy milk → tofu, almond milk → smoked tofu, soy milk → smoked tofu, etc.
are indicative of a **more general association** between soy milk and tofu
- But, rules at higher level may be too generic

Support and confidence in a hierarchy

If X is the parent item for both X_1 and X_2 ,
then $\text{sup}(X) \leq \text{sup}(X_1) + \text{sup}(X_2)$

- **If** $\text{sup}(X_1 \cup Y_1) \geq \text{minsup}$,
and X is parent of X_1 , Y is parent of Y_1
then $\text{sup}(X \cup Y_1) \geq \text{minsup}$, $\text{sup}(X_1 \cup Y) \geq \text{minsup}$
 $\text{sup}(X \cup Y) \geq \text{minsup}$
- **If** $\text{conf}(X_1 \Rightarrow Y_1) \geq \text{minconf}$,
then $\text{conf}(X_1 \Rightarrow Y) \geq \text{minconf}$

Mining 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

Mining 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

Summary

Things to remember

- Interestingness measures
- Categorical attributes
- Continuous attributes
- Min-Apriori
- Mining multi-level rules on a hierarchy

Exercises on this topic

- Introduction to Data Mining 2nd edition (2019) by Tan et al.
 - Exercises 6.8 → 2-4, 6-9
- Data Mining, The Textbook (2015) by Charu Aggarwal
 - Exercises 4.9 → 20-21