

Association Analysis

IE500 Data Mining



Outline

- What is Association Analysis?
- Frequent Itemset Generation
- Rule Generation
- Interestingness Measures
- Handling Continuous and Categorical Attributes
- Subgroup Discovery

Association Analysis

- First algorithms developed in the early 90s at IBM by Agrawal & Srikant
- Motivation
 - Availability of barcode cash registers



Association Analysis

- Initially used for Market Basket Analysis
 - To find how items purchased by customers are related
- Later extended to more complex data structures
 - Sequential patterns
 - Subgraph patterns
- And other application domains
 - Life science
 - Social science
 - Web usage mining

Simple Approaches

- To find out if two items x and y are bought together, we can compute their correlation
- e.g., Pearson's correlation coefficient:

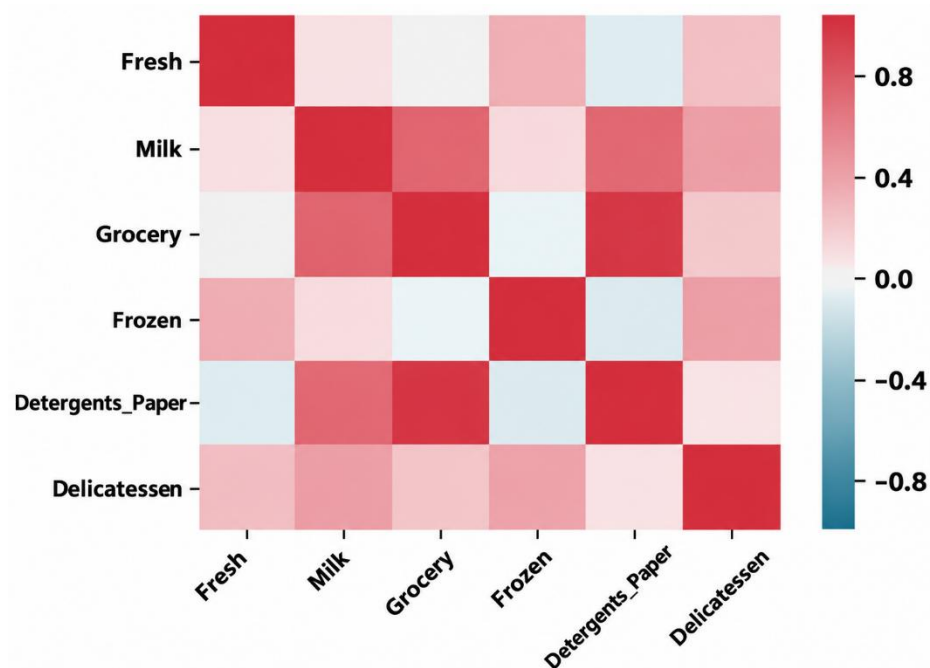
$$\text{PCC} = \frac{\sum_{i=1}^n (x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} * \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

- Numerical coding:
 - 1: item was bought
 - 0: item was not bought
- $\bar{x}$ mean of x : Fraction of transactions containing the item

Correlation Analysis in Python

- e.g., using Pandas:

```
import seaborn as sns  
  
corr = dataframe.corr()  
sns.heatmap(corr)
```



Association Analysis

- Given a set of transactions, **find rules** that will predict the occurrence of an item (set) based on the occurrences of other item(s) (sets) in the transaction.

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Shopping Transactions

- Examples of Association Rules
 - $\{\text{Diaper}\} \rightarrow \{\text{Beer}\}$
 - $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\}$
 - $\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\}$

Implication denotes
co-occurrence

Definition: Frequent Itemset

- Itemset
 - Collection of one or more items
 - Example: {Milk, Bread, Diaper}
 - k-itemset: An itemset that contains k items
- Support count (σ)
 - Frequency of occurrence of an itemset
 - e.g., $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$
- Support (s)
 - Fraction of transactions that contain an itemset
 - e.g., $s(\{\text{Milk, Bread, Diaper}\}) = 2/5 = 0.4$
- Frequent Itemset
 - An itemset whose support is greater than or equal to a minimal support (minsup) threshold specified by the user

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Shopping Transactions

Definition: Association Rule

- Association Rule
 - An implication of the form $X \rightarrow Y$, where X and Y are nonempty and disjoint itemsets
 - Interpretation: when X occurs, Y occurs with a certain probability

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Shopping Transactions

- More formally, it's a *conditional probability*
 - $P(Y|X)$ given X , what is the probability of Y ?

Definition: Association Rule

- Association Rule

- Example:

$\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$
 Condition Consequent

- Rule Evaluation Metrics

- Support s :

Fraction of total transactions which contain both X and Y

$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{|T|}$$

Shopping Transactions

$$\begin{aligned}
 s(\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}) &= \frac{\sigma(\{\text{Milk, Diaper, Beer}\})}{|T|} \\
 &= \frac{2}{5} = 0.4
 \end{aligned}$$

- Confidence c :

Measures how often items in Y appear in transactions that contain X

$$c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

$$\begin{aligned}
 c(\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}) &= \frac{\sigma(\{\text{Milk, Diaper, Beer}\})}{\sigma(\{\text{Milk, Diaper}\})} \\
 &= \frac{2}{3} = 0.67
 \end{aligned}$$

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

The Association Rule Mining Task

- Given a set of transactions T , the goal of association rule mining is to **find all rules** having
 - support $\geq \textit{minsup}$ threshold
 - confidence $\geq \textit{minconf}$ threshold
 - *minsup* and *minconf* are provided by the user
 - Brute Force Approach:
 - List all possible association rules
 - Compute the support and confidence for each rule
 - Remove rules that fail the *minsup* and *minconf* thresholds
- $\Rightarrow$ Computationally prohibitive due to large number of candidates!**

Mining Association Rules

- Example rules:

$\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\} (s=0.4, c=0.67)$

$\{\text{Milk, Beer}\} \rightarrow \{\text{Diaper}\} (s=0.4, c=1.0)$

$\{\text{Diaper, Beer}\} \rightarrow \{\text{Milk}\} (s=0.4, c=0.67)$

$\{\text{Beer}\} \rightarrow \{\text{Milk, Diaper}\} (s=0.4, c=0.67)$

$\{\text{Diaper}\} \rightarrow \{\text{Milk, Beer}\} (s=0.4, c=0.5)$

$\{\text{Milk}\} \rightarrow \{\text{Diaper, Beer}\} (s=0.4, c=0.5)$

- Observations:

- All the above rules are binary partitions of the same itemset:

$\{\text{Milk, Diaper, Beer}\}$

- Rules originating from the same itemset have identical support s

- but can have different confidence

- Thus, we may decouple the support and confidence requirements

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer , Eggs
3	Milk, Diaper, Beer , Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper , Coke

Shopping Transactions

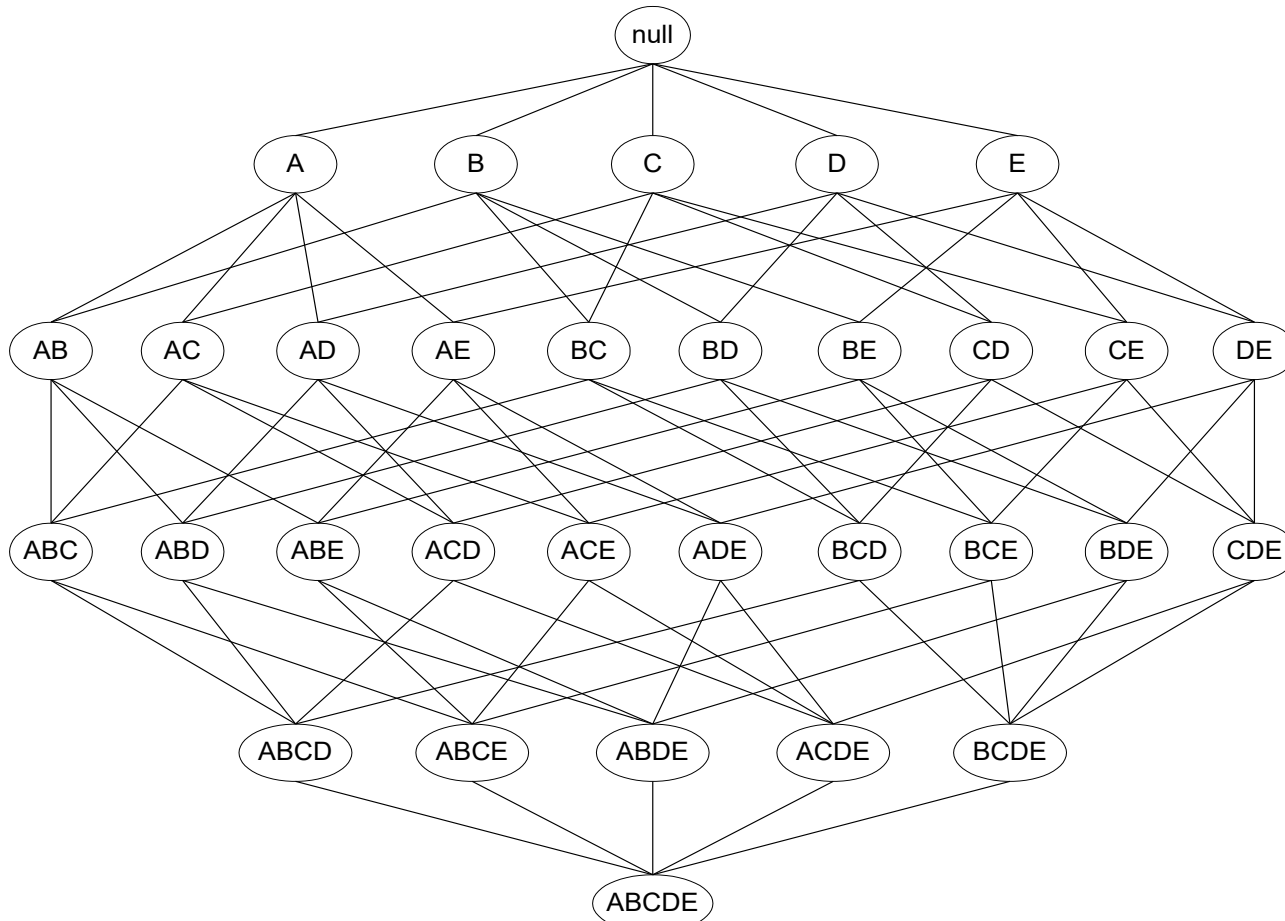
$$s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{|T|}$$

Apriori Algorithm: Basic Idea

- Two-step approach:
 1. Frequent Itemset Generation
 - Generate all itemsets whose support $\geq$ minsup
 2. Rule Generation
 - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset
- Frequent itemset generation is still computationally expensive

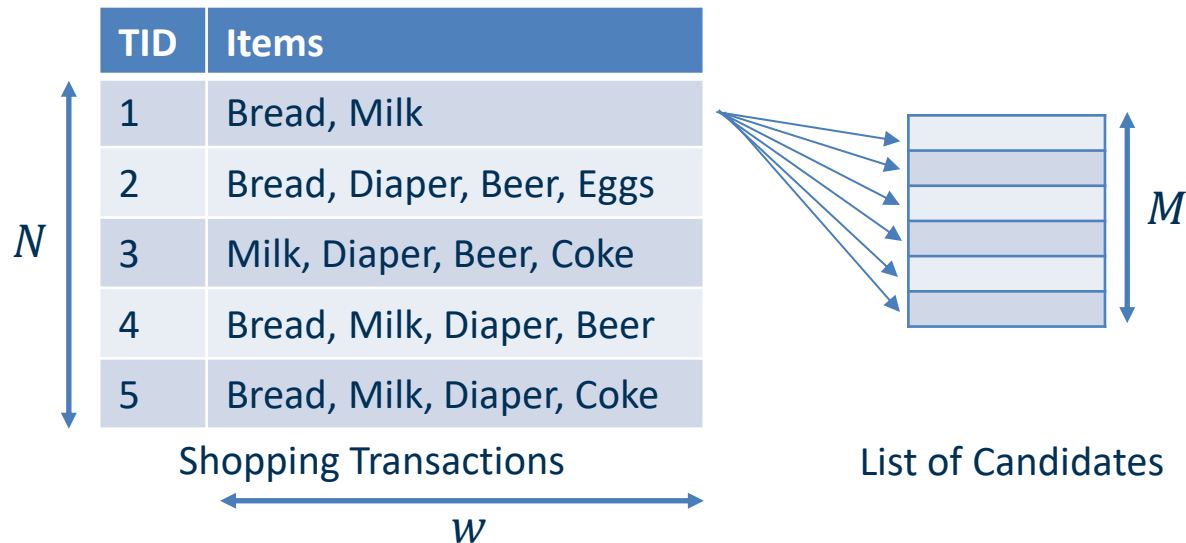
Frequent Itemset Generation

- Given d items, there are $2^d - 1$ candidate itemsets!



Brute-force Approach

- Each itemset in the lattice is a **candidate** frequent itemset
- Count the support of each candidate by scanning the database
- Match each transaction against every candidate

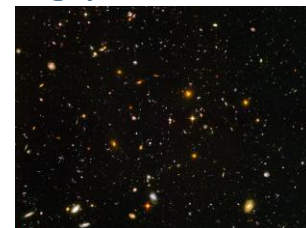


- Complexity $\sim O(NMw) \rightarrow$ Expensive since $M = 2^d - 1$

Brute-force Approach

- Amazon sells 600M different products (as of 2026)
 - That is $2^{600.000.000}$ – 1 possible itemsets
 - That's an 181M digit number
 - Today's supercomputers: 1,200 Petaflops, i.e., 1.2×10^{18} floating point operations per second
 - Even if an itemset could be checked with one single floating point operation this would take $\sim 6.6 \times 10^{180,617,971}$ years (age of universe: 1.4×10^{10} years)
- However:
 - Most itemsets will not be frequent at all, e.g., books on Chinese calligraphy, Inuit cooking, and data mining bought together
 - Thus, smarter algorithms should be possible
 - Intuition for the algorithm:
All itemsets containing Inuit cooking are likely infrequent

amazon



Anti-Monotonicity of Support

- What happens when an itemset gets larger?

- $s(\{\text{Milk}\}) = 0.8$
- $s(\{\text{Milk}, \text{Diaper}\}) = 0.6$
- $s(\{\text{Milk}, \text{Diaper}, \text{Beer}\}) = 0.4$

- $s(\{\text{Bread}\}) = 0.8$
- $s(\{\text{Bread}, \text{Milk}\}) = 0.6$
- $s(\{\text{Bread}, \text{Milk}, \text{Diaper}\}) = 0.4$

- There is a pattern here!

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Reducing the Number of Candidates

- There is a pattern here!
 - It is called anti-monotonicity of support
- If X is a subset of Y
 - $s(Y)$ is at most as large as $s(X)$

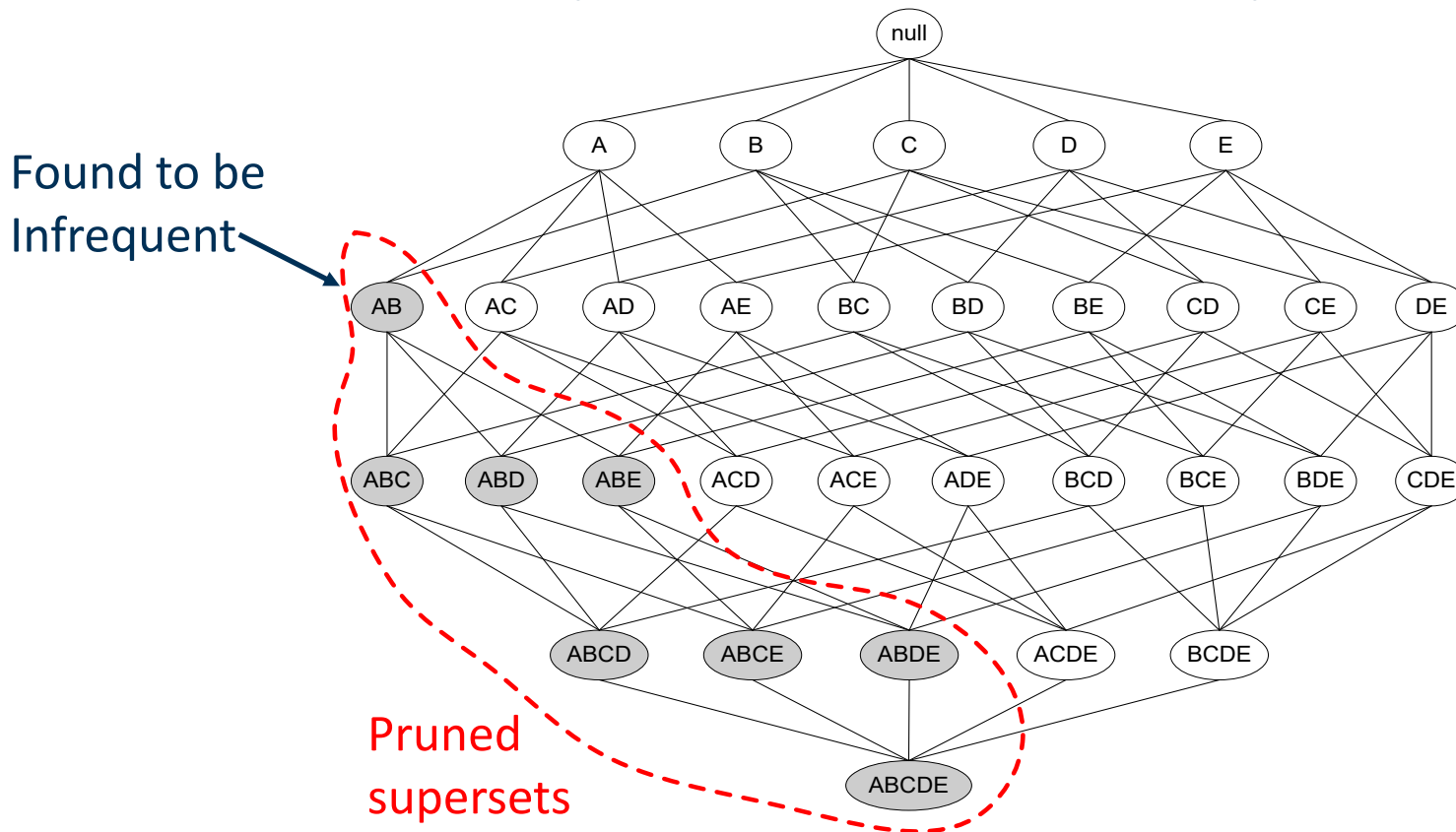
TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

$$\forall X, Y: (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- Consequence for frequent itemset search (aka Apriori principle):
 - If Y is frequent, X also has to be frequent
 - i.e.: **all subsets of frequent itemsets are frequent**

Using the Apriori Principle for Pruning

- If an itemset is infrequent, then all of its supersets must also be infrequent



The Apriori Algorithm

- Let $k=1$
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
 - **Generate** length $(k+1)$ candidate itemsets from length k frequent itemsets
 - **Prune** candidate itemsets that cannot be frequent because they contain subsets of length k that are infrequent (Apriori Principle)
 - **Count** the support of each candidate by scanning the DB
 - **Eliminate** candidates that are infrequent, leaving only those that are frequent
 - Set $k=k+1$

Illustrating the Apriori Principle

Minimum Support Count = 3

Items (1-itemsets)

Item	Count
Bread	4
Coke	2
Milk	5
Beer	3
Diaper	5
Eggs	2

No need to generate
candidates involving
Coke or Eggs

Pairs (2-itemsets)

Itemset	Count
{Bread, Milk}	4
{Bread, Beer}	1
{Bread, Diaper}	3
{Milk, Beer}	2
{Milk, Diaper}	4
{Beer, Diaper}	3

TID	Items
1	Milk, Beer, Diaper
2	Bread, Eggs, Milk, Coke
3	Milk, Diaper, Bread, Eggs
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke
6	Diaper, Beer

Triplets (3-itemsets)

Itemset	Count
{Bread, Milk, Diaper}	3

Illustrating the Apriori Principle

- In the example, we had six items, and examined
 - Six 1-itemsets
 - Six 2-itemsets
 - One 3-itemset
 - i.e., 13 in total
- vs. possible itemsets: $2^6 - 1 = 63$

TID	Items
1	Milk, Beer, Diaper
2	Bread, Eggs, Milk, Coke
3	Milk, Diaper, Bread, Eggs
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke
6	Diaper, Beer

Item	Count
Bread	4
Coke	2
Milk	5
Beer	3
Diaper	5
Eggs	2

From Frequent Itemsets to Rules

- Given a frequent itemset L , find all non-empty subsets $f \subset L$ such that $f \rightarrow L \setminus f$ satisfies the **minimum confidence** requirement

- Example Frequent Itemset L :

– {Milk, Diaper, Bread}

- Example Rule:

– $\underbrace{\{\text{Milk, Diaper}\}}_f \rightarrow \{\text{Bread}\}$

$$c = \frac{\sigma(\text{Milk, Diaper, Bread})}{\sigma(\text{Milk, Diaper})} = \frac{3}{4}$$

TID	Items
1	Milk, Beer, Diaper
2	Bread, Eggs, Milk, Coke
3	Milk, Diaper, Bread, Eggs
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke
6	Diaper, Beer

Challenge: Combinatorial Explosion

- Given a 4-itemset $\{A,B,C,D\}$, we can generate
 - $\{A\} \rightarrow \{B,C,D\}, \quad \{A,B\} \rightarrow \{C,D\}, \quad \{B,C\} \rightarrow \{A,D\}, \quad \{C,D\} \rightarrow \{A,B\}, \quad \{A,B,C\} \rightarrow \{D\}$
 - $\{B\} \rightarrow \{A,C,D\}, \quad \{A,C\} \rightarrow \{B,D\}, \quad \{B,D\} \rightarrow \{A,C\}, \quad \{A,B,D\} \rightarrow \{C\}$
 - $\{C\} \rightarrow \{A,B,D\}, \quad \{A,D\} \rightarrow \{B,C\}, \quad \{A,C,D\} \rightarrow \{B\}$
 - $\{D\} \rightarrow \{A,B,C\}, \quad \{B,C,D\} \rightarrow \{A\}$
 - i.e., a total of 14 rules for just one itemset!
- General number for a k-itemset: $2^k - 2$
 - It's not 2^k since we ignore $\emptyset \rightarrow \{\dots\}$ and $\{\dots\} \rightarrow \emptyset$

Challenge: Combinatorial Explosion

- Wanted:
another pruning trick like Apriori

- However

- $c(\{\text{Milk}, \text{Diaper}\} \rightarrow \{\text{Bread}\}) = 0.75$
- $c(\{\text{Milk}\} \rightarrow \{\text{Bread}\}) = 0.8$
- $c(\{\text{Diaper}\} \rightarrow \{\text{Bread}\}) = 0.6$
- $c(AB \rightarrow C)$ can be larger or smaller than $c(A \rightarrow C)$
 - In general, confidence does not have an anti-monotone property

TID	Items
1	Milk, Beer, Diaper
2	Bread, Eggs, Milk, Coke
3	Milk, Diaper, Bread, Eggs
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke
6	Diaper, Beer

Challenge: Combinatorial Explosion

- But:
confidence of rules generated
from the **same itemset**
has an anti-monotone property

- e.g., $L = \{\text{Milk}, \text{Diaper}, \text{Bread}\}$
 - $\{\text{Milk}, \text{Diaper}\} \rightarrow \{\text{Bread}\} \text{ c}=0.75$
 - $\{\text{Milk}\} \rightarrow \{\text{Diaper}, \text{Bread}\} \text{ c}=0.6$
 - $\{\text{Diaper}\} \rightarrow \{\text{Milk}, \text{Bread}\} \text{ c}=0.6$
 - $\{\text{Milk}, \text{Bread}\} \rightarrow \{\text{Diaper}\} \text{ c}=0.75$
 - $\{\text{Milk}\} \rightarrow \{\text{Diaper}, \text{Bread}\} \text{ c}=0.6$
 - $\{\text{Bread}\} \rightarrow \{\text{Milk}, \text{Diaper}\} \text{ c}=0.75$

TID	Items
1	Milk, Beer, Diaper
2	Bread, Eggs, Milk, Coke
3	Milk, Diaper, Bread, Eggs
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke
6	Diaper, Beer

Observation: moving elements from antecedent to consequent (“left to right”) in the rule never increases confidence!

- e.g., $L = \{A, B, C, D\}$:

$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

Explanation

- For rules generated from the same itemset, moving items from the antecedent to the consequent cannot increase confidence. Reason:

$$c(AB \rightarrow C) := \frac{s(ABC)}{s(AB)} \qquad c(A \rightarrow BC) := \frac{s(ABC)}{s(A)}$$

- Due to anti-monotone property of support, we know

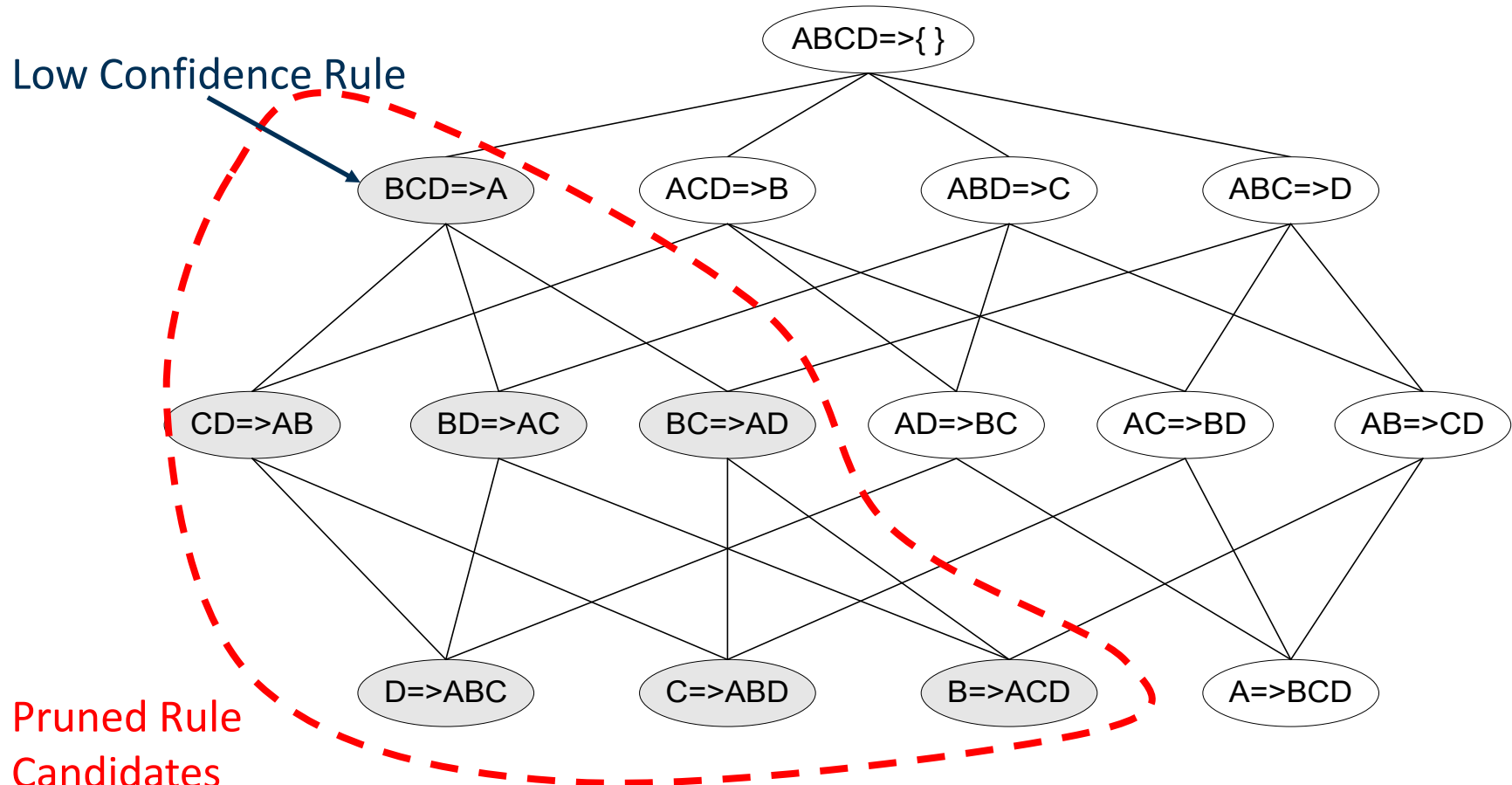
$$s(AB) \leq s(A)$$

- Hence:

$$c(AB \rightarrow C) \geq c(A \rightarrow BC)$$

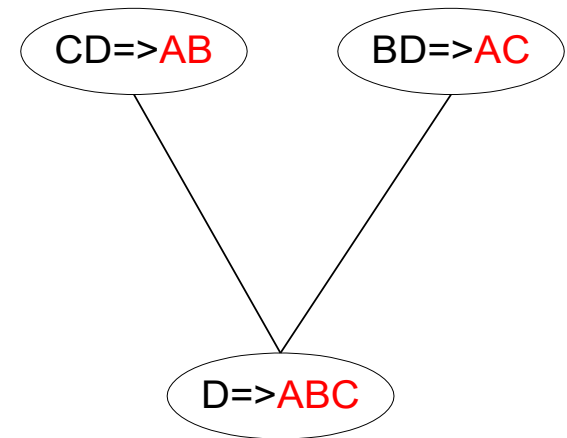
Candidate Rule Pruning

- Moving elements from left to right cannot increase confidence



Rule Generation for Apriori Algorithm

- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent
 - join($CD \rightarrow AB$, $BD \rightarrow AC$)
would produce the candidate rule $D \rightarrow ABC$
 - Prune rule $D \rightarrow ABC$ if one of its parent rules does not have high confidence (e.g., $AD \rightarrow BC$)
- All the required information for confidence computation has already been recorded during itemset generation
 - Thus, there is no need to see the data any more



$$c(X \rightarrow Y) = \frac{s(X \cup Y)}{s(X)}$$

Association Analysis in Python

Various packages exist.

In the exercise, we'll use the mlxtend package

```
from mlxtend.frequent_patterns import apriori, association_rules

# Generate frequent itemsets (min support = 0.1)
frequent_itemsets = apriori(df.astype(bool), min_support=0.1, use_colnames=True)

# Generate association rules (min confidence = 0.5)
rules = association_rules(frequent_itemsets,
                          metric="confidence",
                          min_threshold=0.5)
```

Interestingness Measures

- Association rule algorithms tend to produce too many rules
 - Many of them are uninteresting or redundant
 - Redundant if $\{A,B,C\} \rightarrow \{D\}$ and $\{A,B\} \rightarrow \{D\}$ have the same support & confidence
- Interestingness measures can be used to prune or rank the derived rules
- In the original formulation of association rules, support & confidence were the only interestingness measures used
- Later, various other measures have been proposed
 - We will have a look at one: Lift
 - See Tan/Steinbach/Kumar, Chapter 6.7

Drawback of Confidence

Contingency table

	Coffee	$\overline{\text{Coffee}}$	
Tea	3	1	4
$\overline{\text{Tea}}$	15	1	16
	18	2	20



TID	Items
1	Coffee
2	Coffee
3	Coffee
4	Coffee
5	Coffee
6	Coffee
7	Coffee
8	Coffee
9	Coffee
10	Coffee
11	Coffee
12	Coffee
13	Coffee
14	Coffee
15	Coffee
16	Tea, Coffee
17	Tea, Coffee
18	Tea, Coffee
19	Tea
20	Bread

- Association Rule:
Tea $\rightarrow$ Coffee
- $\text{confidence}(\text{Tea} \rightarrow \text{Coffee}) = \frac{3}{4} = 0.75$
- **but** $\text{support}(\text{Coffee}) = \frac{18}{20} = 0.9$
- Although confidence is high, rule is misleading as the fraction of coffee drinkers is higher than the confidence of the rule
 - We want $\text{confidence}(X \rightarrow Y) > \text{support}(Y)$
 - If $\text{confidence} < \text{support}(Y)$, X reduces the probability of Y. Equality indicates independence.

- We discover a high confidence rule for tea $\rightarrow$ coffee
 - 75% of all people who drink tea also drink coffee
 - Hypothesis: people who drink tea are likely to drink coffee
 - Implicitly: **more likely than all people**
- Test: Compare the confidence of the two rules
 - Rule: Tea $\rightarrow$ coffee $c(\text{tea} \rightarrow \text{coffee}) = \frac{s(\{\text{tea}\} \cup \{\text{coffee}\})}{s(\{\text{tea}\})}$
 - Default rule: all $\rightarrow$ coffee $c(\text{all} \rightarrow \text{coffee}) = s(\{\text{coffee}\})$
- We accept a rule if its confidence is higher than the default rule
$$\frac{c(\text{tea} \rightarrow \text{coffee})}{c(\text{all} \rightarrow \text{coffee})} = \frac{c(\text{tea} \rightarrow \text{coffee})}{s(\{\text{coffee}\})} > 1$$

- The lift of an association rule $X \rightarrow Y$ is defined as:

$$c(X \rightarrow Y) = \frac{s(X \cup Y)}{s(X)}$$

$$Lift = \frac{c(X \rightarrow Y)}{s(Y)} = \frac{s(X \cup Y)}{s(X) * s(Y)}$$

Support, not support count!!!!!!

- Confidence normalized by support of consequent
- Interpretation
 - if **lift** > 1, then X and Y are positively correlated
 - if **lift** = 1, then X and Y are independent
 - if **lift** < 1, then X and Y are negatively correlated

Lift (Example)

- Association Rule:
Tea $\rightarrow$ Coffee

- confidence(Tea $\rightarrow$ Coffee) = $\frac{3}{4} = 0.75$

- but** support(Coffee) = $\frac{18}{20} = 0.9$

$$\begin{aligned} \text{Lift}(\text{Tea} \rightarrow \text{Coffee}) &= \frac{c(\text{tea} \rightarrow \text{coffee})}{c(\text{all} \rightarrow \text{coffee})} = \frac{c(\text{tea} \rightarrow \text{coffee})}{s(\{\text{coffee}\})} \\ &= \frac{0.75}{0.9} = 0.833 < 1 \end{aligned}$$

- Lift < 1, so Tea and Coffee are negatively associated;
discard this rule when seeking positive associations.

Contingency table

	Coffee	$\overline{\text{Coffee}}$	
Tea	3	1	4
$\overline{\text{Tea}}$	15	1	16
	18	2	20



TID	Items
1	Coffee
2	Coffee
3	Coffee
4	Coffee
5	Coffee
6	Coffee
7	Coffee
8	Coffee
9	Coffee
10	Coffee
11	Coffee
12	Coffee
13	Coffee
14	Coffee
15	Coffee
16	Tea, Coffee
17	Tea, Coffee
18	Tea, Coffee
19	Tea
20	Bread

Interestingness Measures

- There are lots of measures proposed in the literature
- Some measures are good for certain applications, but not for others
- Details: see literature (e.g., Tan et al.)

#	Measure	Formula
1	ϕ -coefficient	$\frac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j, B_k) + \sum_k \max_j P(A_j, B_k) - \max_j P(A_j) - \max_k P(B_k)}{2 - \max_j P(A_j) - \max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\bar{A},\bar{B})}{P(A,\bar{B})P(\bar{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\bar{A}\bar{B}) - P(A,\bar{B})P(\bar{A},B)}{P(A,B)P(\bar{A}\bar{B}) + P(A,\bar{B})P(\bar{A},B)} = \frac{\alpha - 1}{\alpha + 1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\bar{A}\bar{B})} - \sqrt{P(A,\bar{B})P(\bar{A},B)}}{\sqrt{P(A,B)P(\bar{A}\bar{B})} + \sqrt{P(A,\bar{B})P(\bar{A},B)}} = \frac{\sqrt{\alpha} - 1}{\sqrt{\alpha} + 1}$
6	Kappa (κ)	$\frac{P(A,B) + P(\bar{A},\bar{B}) - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i, B_j) \log \frac{P(A_i, B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i) \log P(A_i), -\sum_j P(B_j) \log P(B_j))}$
8	J-Measure (J)	$\max \left(P(A, B) \log \left(\frac{P(B A)}{P(B)} \right) + P(\bar{A}\bar{B}) \log \left(\frac{P(\bar{B} \bar{A})}{P(\bar{B})} \right), \right. \\ \left. P(A, B) \log \left(\frac{P(A B)}{P(A)} \right) + P(\bar{A}\bar{B}) \log \left(\frac{P(\bar{A} \bar{B})}{P(\bar{A})} \right) \right)$
9	Gini index (G)	$\max \left(P(A)[P(B A)^2 + P(\bar{B} A)^2] + P(\bar{A})[P(B \bar{A})^2 + P(\bar{B} \bar{A})^2] \right. \\ \left. - P(B)^2 - P(\bar{B})^2, \right. \\ \left. P(B)[P(A B)^2 + P(\bar{A} B)^2] + P(\bar{B})[P(A \bar{B})^2 + P(\bar{A} \bar{B})^2] \right. \\ \left. - P(A)^2 - P(\bar{A})^2 \right)$
10	Support (s)	$P(A, B)$
11	Confidence (c)	$\max(P(B A), P(A B))$
12	Laplace (L)	$\max \left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2} \right)$
13	Conviction (V)	$\max \left(\frac{P(A)P(\bar{B})}{P(\bar{A}\bar{B})}, \frac{P(B)P(\bar{A})}{P(\bar{B}\bar{A})} \right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A, B) - P(A)P(B)$
17	Certainty factor (F)	$\max \left(\frac{P(B A) - P(B)}{1 - P(B)}, \frac{P(A B) - P(A)}{1 - P(A)} \right)$
18	Added Value (AV)	$\max(P(B A) - P(B), P(A B) - P(A))$
19	Collective strength (S)	$\frac{P(A,B) + P(\bar{A}\bar{B})}{P(A)P(B) + P(\bar{A})P(\bar{B})} \times \frac{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A,B) - P(\bar{A}\bar{B})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A) + P(B) - P(A,B)}$
21	Klogsen (K)	$\sqrt{P(A,B)} \max(P(B A) - P(B), P(A B) - P(A))$

Handling Continuous and Categorical Attributes

- How to apply association analysis to attributes that are not asymmetric binary variables?

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

- Example Rule:
 $\{\text{Number of Pages} \in [5,10) \wedge (\text{Browser}=\text{Firefox})\} \rightarrow \{\text{Buy} = \text{No}\}$

Handling Categorical Attributes

- Transform a categorical attribute into asymmetric binary variables
- Introduce a new “item” for each distinct attribute-value pair
 - e.g. replace “Browser Type” attribute with
 - attribute: “Browser Type = Chrome”
 - attribute: “Browser Type = Firefox”
 -
- Issues
 - What if the attribute has many possible values?
 - Many of the attribute values may have very low support
 - Potential solution: aggregate low-support attribute values
 - What if the distribution of attribute values is highly skewed?
 - Example: 95% of the visitors have Buy = No
 - Most of the items will be associated with the (Buy=No) item
 - Potential solution: drop the highly frequent item

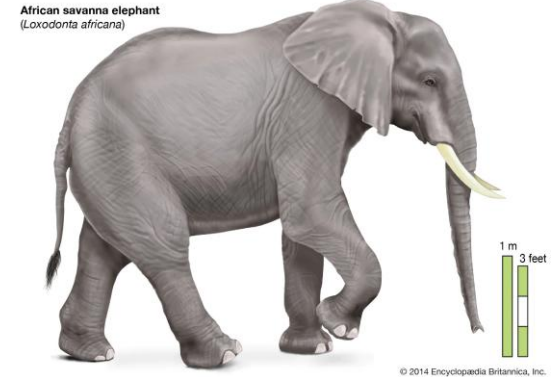
Handling Continuous Attributes

- Transform a continuous attribute into binary variables using discretization
 - equal-width binning
 - equal-frequency binning
- Issue: Size of the discretization intervals affects support & confidence
 - {Refund=No, (Income=\$51,251)} → {Cheat=No}
 - {Refund=No, (60K ≤ Income ≤ 80K)} → {Cheat=No}
 - {Refund=No, (0K ≤ Income ≤ 1B)} → {Cheat=No}
 - If intervals are too small
 - Itemsets may not have enough support
 - If intervals are too large
 - Rules may not have enough confidence
 - e.g. combination of different age groups compared to a specific age group

Subgroup Discovery

- Association Rule Mining:
 - Find all patterns in the data
- Classification:
 - Identify the best patterns that can predict a target variable
 - These need not include all predictive patterns.
- Subgroup Discovery:
 - Find **all patterns** that can explain a target variable

Subgroup Discovery vs. Classification



- Example: learn to classify animals
 - Two possible models
 - has Trunk
→ Elephant (acc. 98%)
 - has Trunk AND weight>3000kg AND color=grey AND height>2m
→ Elephant (acc 99%)
 - Which one do you prefer?
 - Occam's Razor:
if you have two theories that explain a phenomenon equally well,
choose the simpler one (has Trunk → Elephant)
 - What is our goal?
 - Classify animals with high accuracy
 - Learn as much about elephants (more generally: the data) as possible

Subgroup Discovery – Algorithms

- Early algorithms (e.g., EXPLORA, MIDOS, 1990s)
 - Learn an unpruned decision tree
 - Extract rules
 - Compute measures for rules, rate and rank
- Newer algorithms
 - Based on association rule mining (APRIORI-SD and others, 2000s)
 - Based on evolutionary algorithms (2000s)

Subgroup Discovery – Metrics

- One of the most common metrics in Subgroup Discovery is **WRAcc (Weighted Relative Accuracy)**, using the probabilities of subgroup membership (S) and the target class (T)
 - WRAcc = $P(ST) - P(S) * P(T)$

	Elephant	¬Elephant	
has Trunk AND weight>3000kg AND color=grey AND height>2m	1894	0	
¬(has Trunk AND weight>3000kg AND color=grey AND height>2m)	32	54874	



Subgroup Discovery – Metrics

- One of the most common metrics in Subgroup Discovery is **WRAcc (Weighted Relative Accuracy)**, using the probabilities of subgroup membership (S) and the target class (T)
 - WRAcc = $P(ST) - P(S) * P(T) = 0.033 - 0.033 * 0.034 = 0.032$

	Elephant	¬Elephant	
has Trunk AND weight>3000kg AND color=grey AND height>2m	0.033	0.0	0.033
¬(has Trunk AND weight>3000kg AND color=grey AND height>2m)	0.0006	0.966	0.967
	0.034	0.966	

Subgroup Discovery – Metrics

$$\text{WRAcc} = P(\text{ST}) - P(\text{S}) * P(\text{T})$$

Observations:

- If subgroup and target are independent: $P(\text{ST}) = P(\text{S}) * P(\text{T}) \rightarrow \text{WRAcc} = 0.0$
- Best case for a subgroup:
 - The subgroup contains only target examples $\rightarrow P(\text{ST}) = P(\text{S})$
 - And covers as many target examples as possible $\rightarrow P(\text{S}) = P(\text{T})$
 - $\rightarrow$ Maximum possible WRAcc for target class: $P(\text{T}) - P(\text{T})^2$
- Important consequence:
 - WRAcc depends on the base frequency of the target class $P(\text{T})$
 - WRAcc_max is largest when $P(\text{T}) = 0.5$ (balanced classes)
 - $\rightarrow$ WRAcc values are only comparable for subgroups referring to the same target class.
- Our elephant rule: $P(\text{ST}) - P(\text{S}) * P(\text{T}) \approx 0.033 - 0.033 * 0.034 = 0.032$
 - Maximum WRAcc: $P(\text{T}) - P(\text{T})^2 \approx 0.034 - 0.034^2 = 0.032844$
 - i.e., our rule is pretty good!
- **Bottom line: WRAcc represents both coverage and accuracy**

Questions?



Literature for this Slideset

- Pang-Ning Tan, Michael Steinbach, Karpatne, Vipin Kumar:
Introduction to Data Mining.
2nd Edition. Pearson.
- Chapter 4: Association Analysis:
Basic Concepts and
Algorithms
- Chapter 7: Association Analysis:
Advanced Concepts

