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:

predicted value p_i
actual value a_i

$$\mathrm{PCC} = \frac{\sum_{i=1}^{n}(p_i - \bar{p}) * (a_i - \bar{a})}{\sqrt{\sum_{i=1}^{n}(p_i - \bar{p})^2} * \sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2}}$$

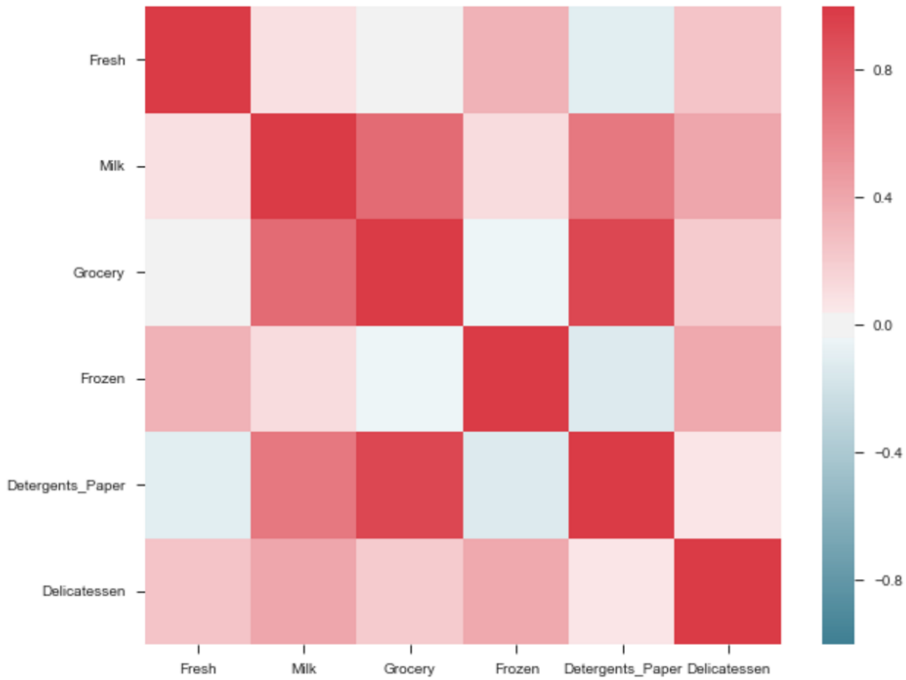
- Numerical coding:
  - 1: item was bought
  - 0: item was not bought
- $\bar{p}$ average of p (i.e., how often x was bought)

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 based on the occurrences of other items in the transaction.

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

Shopping Transactions

- Examples of Association Rules
  - {Diaper} → {Beer}
    {Beer, Bread} → {Milk}
    {Milk, Bread} → {Eggs, Coke}

Implication denotes co-occurence, not causality!

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. σ({Milk, Bread, Diaper}) = 2
- Support (s)
  - Fraction of transactions that contain an itemset
  - e.g. s({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	Break, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Shopping Transactions

Definition: Association Rule

- Association Rule
  - An implication of the form X → Y, where X and Y are 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	Break, 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:
    {Milk, Diaper} → {Beer}
    Condition Consequent
- Rule Evaluation Metrics
  - Support **s**:
    Fraction of total transactions which contain both X and Y
  - Confidence **c**:
    Measures how often items in Y appear in transactions that contain X

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

Shopping Transactions

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

$$\text{s}(\{\text{Milk, Diaper}\} \rightarrow \{Beer\}) = \frac{\sigma(\{Milk, Diaper, Beer\})}{|T|}$$

$$= \frac{2}{5} = 0.4$$

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

$$c(\{\text{Milk, Diaper}\} \rightarrow \{Beer\})) = \frac{\sigma(\{Milk, Diaper, Beer\})}{\sigma(\{Milk, Diaper\})}$$

$$= \frac{2}{3} = 0.67$$

The Association Rule Mining Task

- Given a set of transactions T,
  the goal of association rule mining is to **find all rules** having
  - support ≥ *minsup* threshold
  - confidence ≥ *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

⇒ **Computationally prohibitive due to large number of candidates!**

Mining Association Rules

- Example rules:
  {Milk, Diaper} → {Beer} (s=0.4, c=0.67)
  {Milk, Beer} → {Diaper} (s=0.4, c=1.0)
  {Diaper, Beer} → {Milk} (s=0.4, c=0.67)
  {Beer} → {Milk, Diaper} (s=0.4, c=0.67)
  {Diaper} → {Milk, Beer} (s=0.4, c=0.5)
  {Milk} → {Diaper, Beer} (s=0.4, c=0.5)

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

Shopping Transactions

- Observations:
  - All the above rules are binary partitions of the same itemset: {Milk, Diaper, Beer}
  - Rules originating from the same itemset have identical support s
    - but can have different confidence

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

  - Thus, we may decouple the support and confidence requirements

Apriori Algorithm: Basic Idea

- Two-step approach:

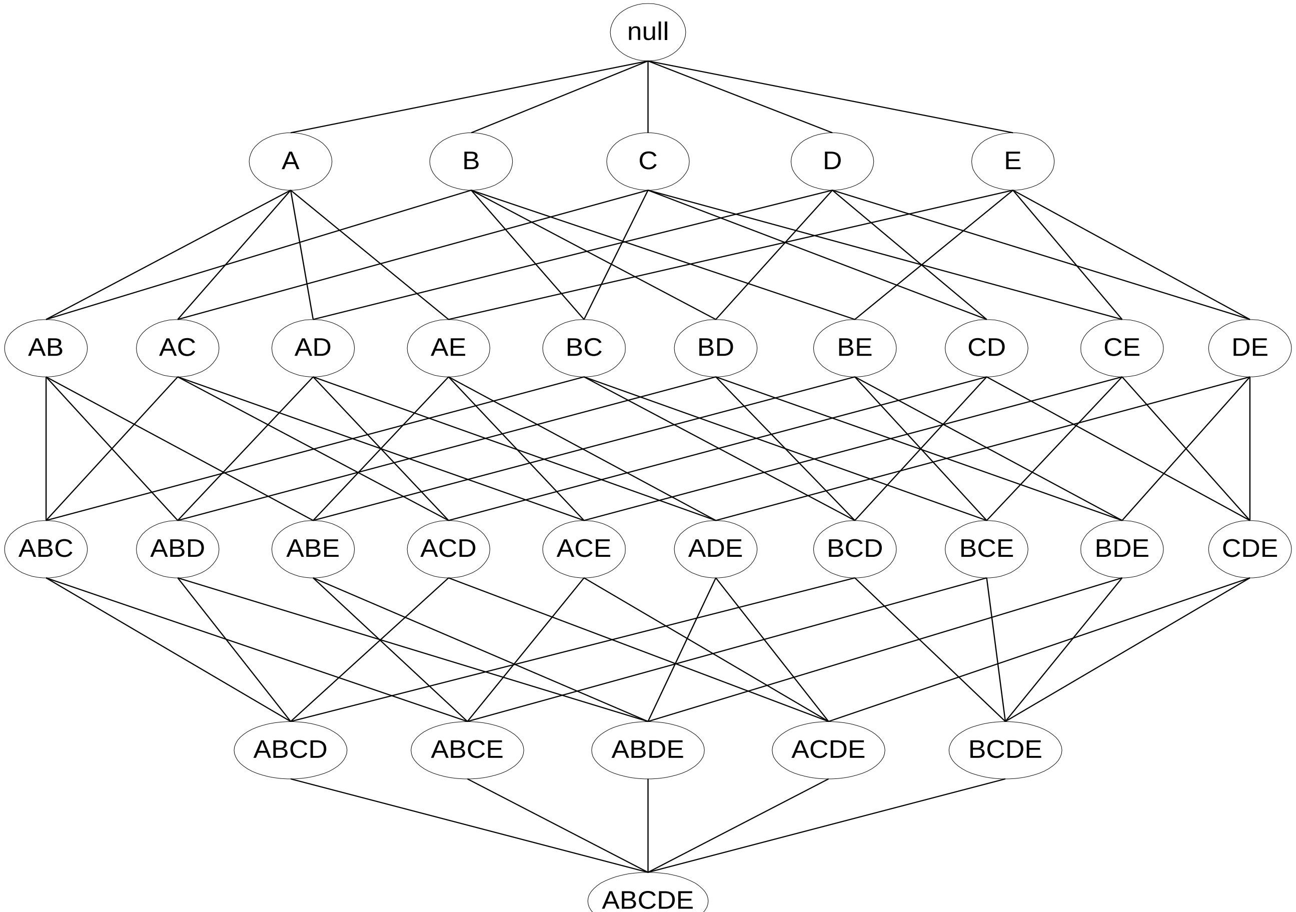
  1. Frequent Itemset Generation
     - Generate all itemsets whose support $\geq$ minsup

  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 candidate itemsets!



Brute-force Approach

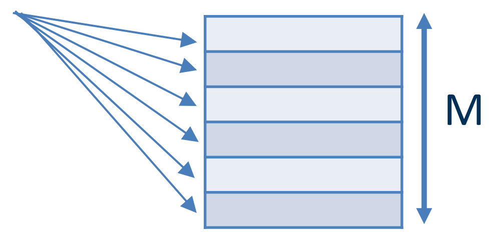
- Each itemset in the lattice is a **candidate** frequent itemset
- Count the support of each candidate by scanning the database
- Match each transaction against every candidate

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

N

Shopping Transactions

w



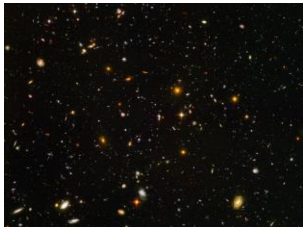
List of Candidates

- Complexity ~ O(NMw) ➔ Expensive since $M = 2^d$

Brute-force Approach

- Amazon sells 12M different products (as of 2023)
  - That is $2^{12.000.000}$ possible itemsets
    - That's a 3.6M digit number
  - Today's supercomputers: 1,200 Petaflops, i.e., $1.2x\ 10^{18}$ floating point operations per second
  - Even if an itemset could be checked with one single floating point operation this would take ~ $10^{3,612,334}$ years

    (age of universe: $1.4x10^{10}$ years)



- However:
  - Most itemsets will not be important 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

Anti-Monotonicity of Support

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- What happens when an itemset gets larger?
  - s({Milk}) = 0.8
  - s({Milk,Diaper}) = 0.6
  - s({Milk,Diaper,Beer}) = 0.4

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

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

- There is a pattern here!

Reducing the Number of Candidates

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

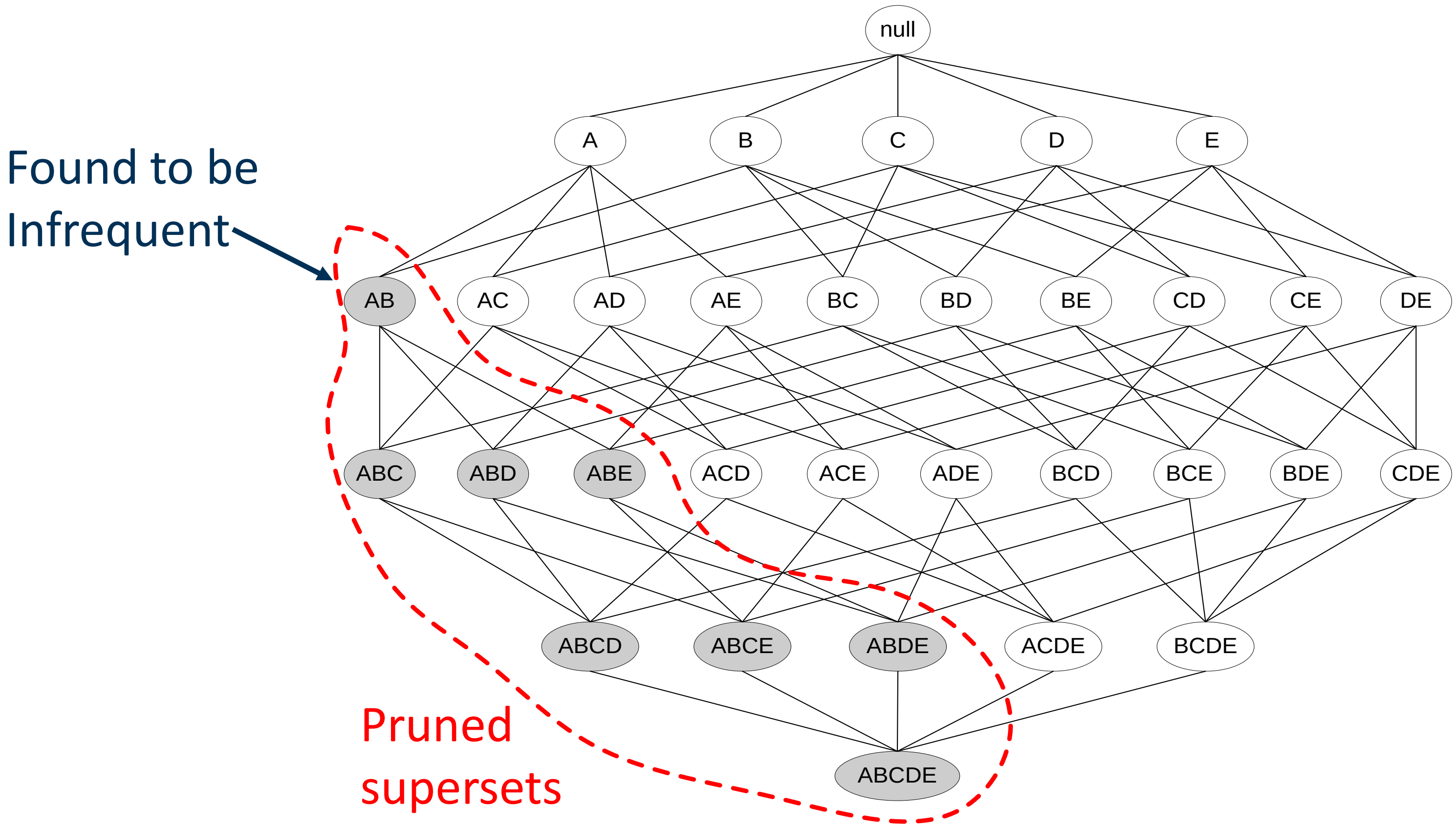
TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Break, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

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

- 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 can not 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

Illustrating the Apriori Principle

Minimum Support Count = 3

Items (1-itemsets)

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

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

No need to generate candidates involving Coke or Eggs

Pairs (2-itemsets)

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

No need to generate candidate {Milk, Diaper, Beer} as count {Milk, Beer} = 2

Triplets (3-itemsets)

Itemset	Count
{Bread, Milk, Diaper}	3

Illustrating the Apriori Principle

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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 = 64

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

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

From Frequent Itemsets to Rules

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

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

- Example Frequent Itemset L:
  - {Milk,Diaper,Beer}
- Example Rule:
  - $\underbrace{\{\text{Milk,Diaper}\}}_{f} \rightarrow \{\text{Beer}\}$

$$c = \frac{\sigma(Milk, Diaper, Beer)}{\sigma(Milk, Diaper)} = \frac{2}{3}$$

Challenge: Combinatorial Explosion

- Given a 4-itemset {A,B,C,D}, we can generate

  {A} → {B,C,D}, {A,B} → {C,D}, {B,C} → {A,D}, {C,D} → {A,B}, {A,B,C} → {D}
  {B} → {A,C,D}, {A,C} → {B,D}, {B,D} → {A,C}, {A,B,D} → {C}
  {C} → {A,B,D}, {A,D} → {B,C}, {A,C,D} → {B}
  {D} → {A,B,C}, {B,C,D} → {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 ∅ → {…} and {…} → ∅

Challenge: Combinatorial Explosion

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

- Wanted:
  another pruning trick like Apriori

- However
  - c({Milk,Diaper} → {Beer}) =0.67
  - c({Milk} → {Beer}) =0.5
  - c({Diaper} → {Beer}) =0.8

  - c(ABC → D) can be larger or smaller than c(AB → D)
    - In general, confidence does not have an anti-monotone property

Challenge: Combinatorial Explosion

- But:
  confidence of rules generated from the **same itemset** has an anti-monotone property
  - E.g. L = {Milk,Diaper,Beer}
    - {Milk,Diaper,Beer} → Ø c=1.0
      - {Milk,Diaper} → {Beer} c=0.67
      - {Milk} → {Diaper,Beer} c=0.5
      - {Diaper} → {Milk,Beer} c=0.5
    - {Milk,Beer} → {Diaper} c=1.0
      - {Milk} → {Diaper,Beer} c=0.5
      - {Beer} → {Milk,Diaper} c=0.67
  - e.g., L = {A,B,C,D}:
    $c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$

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

Observation: moving elements from antecedent to consequence ("left to right") in the rule never increases confidence!

Explanation

- Confidence is anti-monotone with respect to the number of items on the right-hand side (RHS) of the rule
  - i.e., “moving elements from left to right” 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

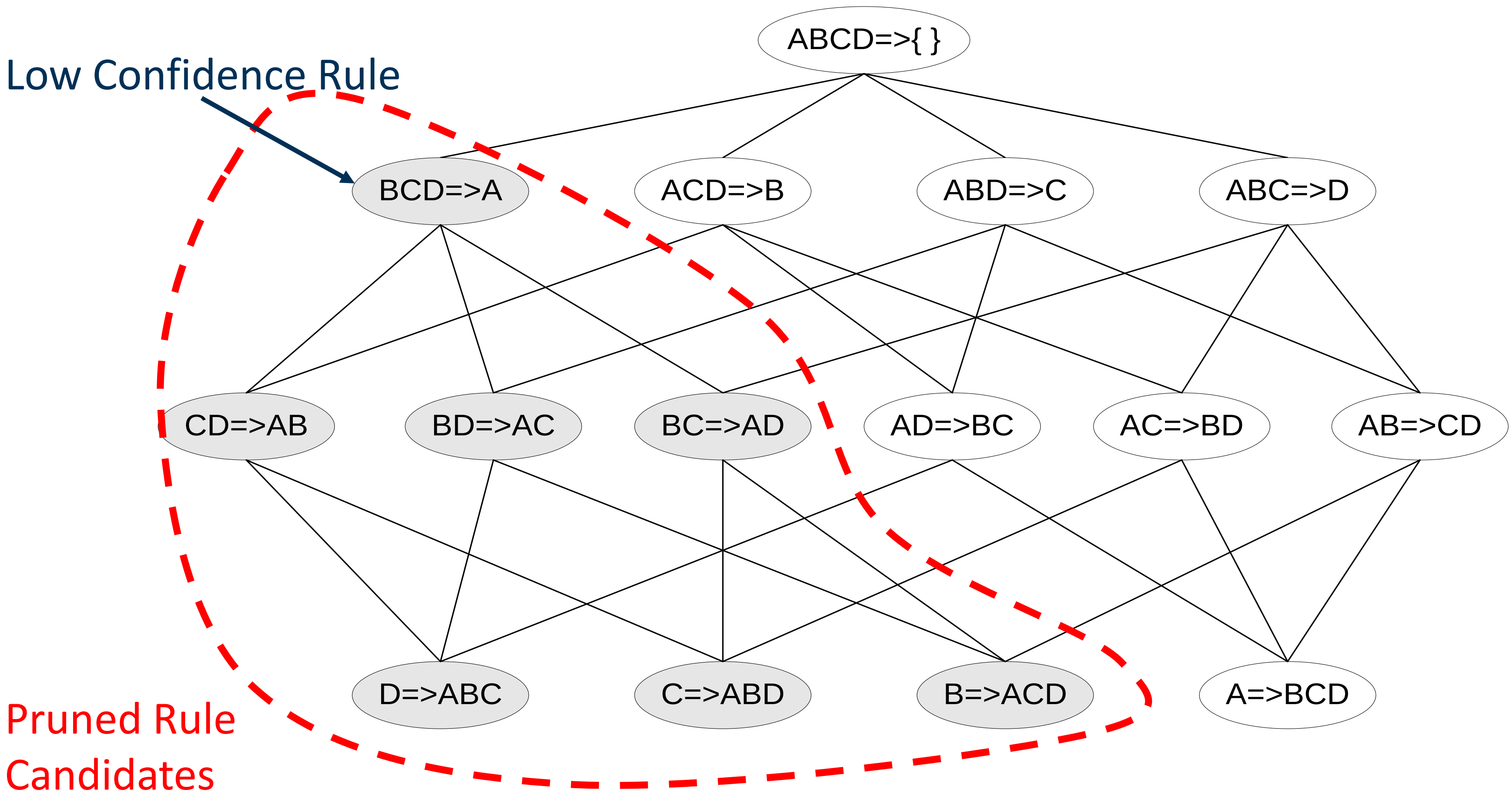
$$\mathrm{s(AB)} \leq \mathrm{s(A)}$$

  - Hence

$$c(A \rightarrow BC) \geq \mathrm{c(A \rightarrow BC)}$$

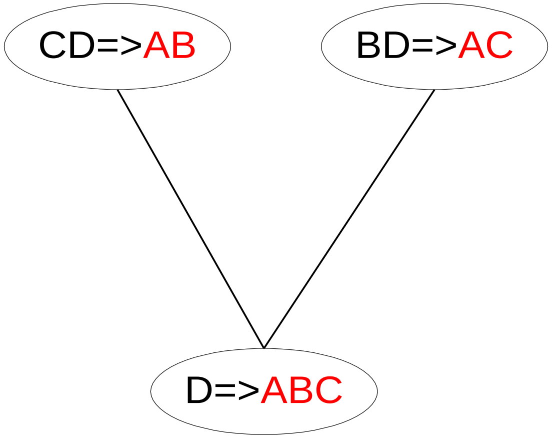
Candidate Rule Pruning

- 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 → AB, BD → AC) would produce the candidate rule D → ABC
  - Prune rule D → ABC if one of its parent rules does not have high confidence (e.g. AD → 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 Orange3 package
  - Frequent Itemset Generation

```python
from orangecontrib.associate.fpgrowth import frequent_itemsets

# Calculate frequent itemsets
itemsets = dict(frequent_itemsets(dataset.values, 0.20))
```

Min support

  - Creating Association Rules

```python
from orangecontrib.associate.fpgrowth import association_rules

# Calculate association rules from itemsets
rules = association_rules(itemsets, 0.70)
```

Min confidence

Interestingness Measures

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Association rule algorithms tend to produce too many rules
  - Many of them are uninteresting or redundant
  - Redundant if {A,B,C} → {D} and {A,B} → {D} have 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 → Coffee
- confidence(Tea → Coffee) = $\frac{3}{4} = 0.75$
- **but** support(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 confidence(X → Y) > support(Y)
  - otherwise rule is misleading as X reduces probability of Y

Lift

- We discover a high confidence rule for tea → 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 → coffee $c(tea \rightarrow coffee) = \frac{s(\{tea\} \cup \{coffee\})}{s(\{tea\})}$
  - Default rule: all → coffee $c(all \rightarrow coffee) = \frac{s(\{all\} \cup \{coffee\})}{s(\{all\})} = \frac{s(\{coffee\})}{1}$ $= s(\{coffee\})$
- We accept a rule iff its confidence is higher than the default rule

$$\frac{c(tea \rightarrow coffee)}{c(all \rightarrow coffee)} = \frac{c(tea \rightarrow coffee)}{s(\{coffee\})} > 1$$

Lift

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

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

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

  – 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)

University of Mannheim
Data and Web Science Group

- Association Rule:
  Tea → Coffee
- confidence(Tea → Coffee) = $\frac{3}{4} = 0.75$
- **but** support(Coffee) = $\frac{18}{20} = 0.9$

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

$$Lift(Tea \rightarrow Coffee) = \frac{c(tea \rightarrow coffee)}{c(all \rightarrow coffee)} = \frac{c(tea \rightarrow coffee)}{s(\{coffee\})}$$

$$= \frac{0.75}{0.9} = 0.833 \; < 1$$

  – lift < 1, therefore is negatively correlated and removed

Interestingness Measures

University of Mannheim
Data and Web Science Group

- 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)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{A}\overline{B})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A}\overline{B})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{A}\overline{B})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A}\overline{B})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
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)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]$ $-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]$ $-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A),P(A\|B))$
12	Laplace (L)	$\max\Big(\frac{NP(A,B)+1}{NP(A)+2},\frac{NP(A,B)+1}{NP(B)+2}\Big)$
13	Conviction (V)	$\max\Big(\frac{P(A)P(\overline{B})}{P(A\overline{B})},\frac{P(B)P(\overline{A})}{P(B\overline{A})}\Big)$
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)	$\max\Big(\frac{P(B\|A)-P(B)}{1-P(B)},\frac{P(A\|B)-P(A)}{1-P(A)}\Big)$
18	Added Value (AV)	$\max(P(B\|A)-P(B),P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{A}\overline{B})}{P(A)P(B)+P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{A}\overline{B})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (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:

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

Handling Categorical Attributes

- Transform 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 attribute has many possible values?
    - Many of the attribute values may have very low support
    - Potential solution: aggregate low-support attribute values
  - What if distribution of attribute values is highly skewed?
    - Example: 95% of the visitors have Buy = No
    - Most of the items will be associated with (Buy=No) item
    - Potential solution: drop the highly frequent item

Handling Continuous Attributes

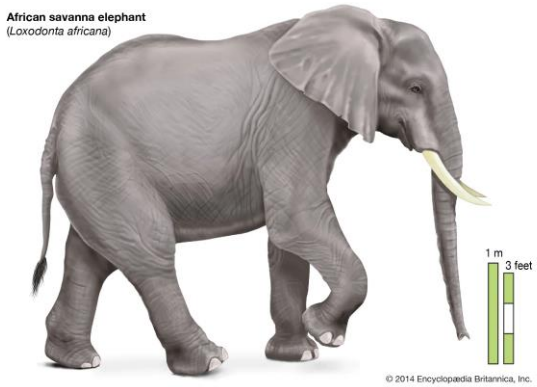
- Transform continuous attribute into binary variables using discretization
  - equal-width binning
  - equal-frequency binning
- Issue: Size of the discretization intervals affects support & confidence
  - {Refund=No, (Income=$51,251)} → {Cheat=No}
  - {Refund=No, (60K<= Income <=80K)} → {Cheat=No}
  - {Refund=No, (0K<= Income <=1B)} → {Cheat=No}

  - If intervals are too small
    - Itemsets may not have enough support
  - If intervals 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
    - Those need not to be all
- 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 at high accuracy
    - Learn as much about elephants (more general: the data) as possible

Subgroup Discovery – Algorithms

- Early algorithms (e.g., EXPLORA, MIDOS, 1999s)
  - Learn 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 probability of subgroup (S) and target (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 probability of subgroup (S) and target (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

WRAcc = P(ST) – P(S)*P(T)

- Observations:
  - The higher P(ST), the more examples are covered
    - i.e., higher WRAcc means high coverage (like support)
  - The lower P(S) – P(ST), the more accurate the subgroup
    - i.e., the higher P(ST)-P(S), the more accurate the subgroup
    - P(T) is a constant factor anyways, given a dataset
    - i.e., higher WRacc means higher accuracy

- Bottom line: WRacc represents both coverage and accuracy

Subgroup Discovery – Metrics

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

- Observations:
  - If P(S) and P(T) are independent, P(ST) = P(S)*P(T), i.e., WRAcc = 0.0
    - Subgroup and target do not interact, this is not interesting
  - Best case:
    - P(ST) = P(S), i.e., no non-target examples covered by subgroup
    - P(ST) = P(T), i.e., no target examples not covered by subgroup
    - i.e., optimimum is $P(T) - P^2(T)$
  - Our elephant rule: P(ST) – P(S)*P(T) = 0.033 – 0.033*0.034 = 0.032
    - Maximum WRacc: $P(T) - P(T)^2 = 0.034 - 0.034^2 = 0.032844$
    - i.e., our rule is pretty good!

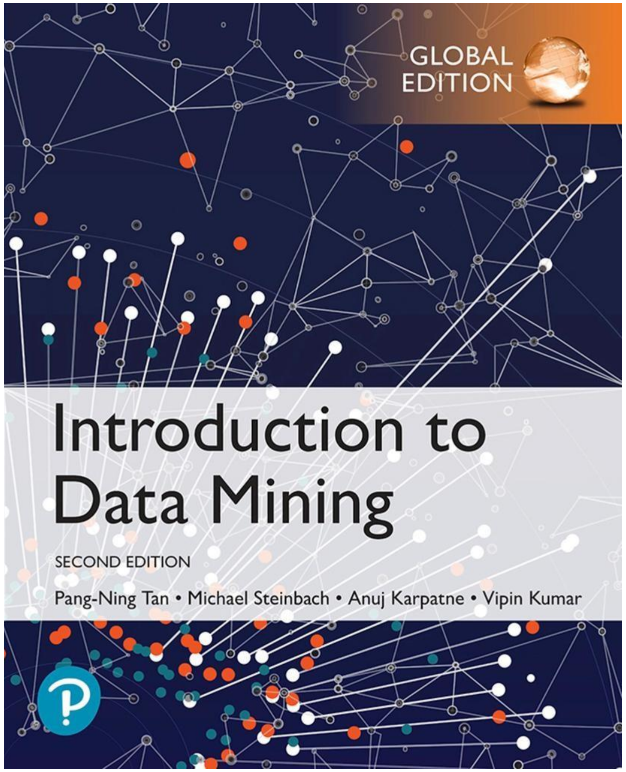
What’s Next?

- Prof. Gemulla
  - HWS: Large-Scale Data Management, Machine Learning
  - FSS: Deep Learning
- Prof. Bizer
  - HWS: Web Data Integration, Large Language Models and Agents
  - FSS: Web Mining
- Prof. Stuckenschmidt
  - HWS: Decision Support
- Prof. Ponzetto
  - HWS: Information Retrieval and Web Search
  - FSS: Advanced Methods in Text Analytics
- Prof. Keuper
  - HWS: Higher Level Computer Vision, Image Processing
  - FSS:Generative Computer Vision Models
- Prof. Rehse
  - FSS: Advanced Process Mining

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