Association Analysis

IE500 Data Mining

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Outline

UNIVERSITY OF MANNHEIM
Data and Web Science Group

1. What is Association Analysis?
2. Frequent Itemset Generation
3. Rule Generation
4. Handling Continuous and Categorical Attributes
5. Interestingness Measures
6. Sequential Pattern Mining

Correlation Analysis

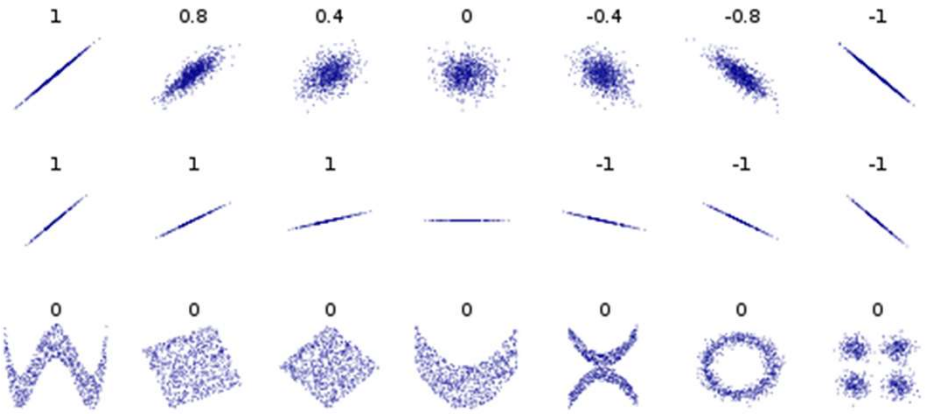
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Correlation analysis measures the degree of dependency between **two variables**
  - Continuous variables: Pearson's correlation coefficient (PCC)
  - Binary variables: Phi coefficient

$$PCC(x,y) = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}}$$

$$Phi(x,y) = \frac{f_{11} f_{00} - f_{01} f_{10}}{\sqrt{f_{1+} f_{+1} f_{0+} f_{+0}}}$$

- Value range [-1,1]
  - 1 : positive correlation
  - 0 : variables independent
  - -1 : negative correlation



Example: Correlations between Products in Shopping Baskets

UNIVERSITY OF MANNHEIM
Data and Web Science Group

	P1	P2	P3	P4	P5
Basket 1	1	1	0	1	1
Basket 2	1	0	0	1	1
Basket 3	1	0	0	0	1

1 : always bought together
0 : sometimes bought together
-1 : never bought together

Correlation Matrix (Correlation Matrix)

Table View | Pairwise Table | Plot View | Annotations

Attributes	ThinkPad X...	Asus EeePC	HP Laserjet...	2 GB DDR3...	8 GB DDR3...	Lenovo Tab...	Netbook-Sc...	HP CE50 T...	LT Laser M...	LT Minimaus
ThinkPad X2	1	-1	0.356	-0.816	0.612	0.583	-0.667	0.356	0.167	-0.408
Asus EeePC	-1	1	-0.356	0.816	-0.612	-0.583	0.667	-0.356	-0.167	0.408
HP Laserjet	0.356	-0.356	1	-0.218	-0.327	0.356	-0.535	1	-0.089	-0.655
2 GB DDR3	-0.816	0.816	-0.218	1	-0.500	-0.816	0.816	-0.218	0	0.200
8 GB DDR3	0.612	-0.612	-0.327	-0.500	1	0.102	-0.408	-0.327	0.102	0
Lenovo Tabl	0.583	-0.583	0.356	-0.816	0.102	1	-0.667	0.356	-0.250	0
Netbook-Sch	-0.667	0.667	-0.535	0.816	-0.408	-0.667	1	-0.535	0.167	0.408
HP CE50 To	0.356	-0.356	1	-0.218	-0.327	0.356	-0.535	1	-0.089	-0.655
LT Laser Ma	0.167	-0.167	-0.089	0	0.102	-0.250	0.167	-0.089	1	-0.408
LT Minimaus	-0.408	0.408	-0.655	0.200	0	0	0.408	-0.655	-0.408	1

Shortcoming: Measures correlation only between two items but not between multiple items, e.g. {ThinkPad, Cover} → {Minimaus}

1. Association Analysis

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Association analysis can find **multiple item co-occurrence relationships** (descriptive method)
- first algorithms developed in the early 90s at IBM by Agrawal & Srikant
- initially used for **shopping basket analysis** to find out how items purchased are related
- later extended to more complex data structures
  - sequential patterns
  - subgraph patterns
- and other application domains
  - web usage mining, social science, life science

Association Analysis

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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.

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

Implication denotes co-occurence, not causality!

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: Frequent Itemset

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- **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	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Shopping Transactions

Definition: Association Rule

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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	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

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Association Rule
  - Example:
    {Milk, Diaper} → {Beer}
    Condition **Consequent**

- Rule Evaluation Metrics
  - **Support s:**
    Fraction of total transactions which contain both X and Y

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

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

  - **Confidence c:**
    Measures how often items in Y appear in transactions that contain X

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

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

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

Main Challenges concerning Association Analysis

UNIVERSITY OF MANNHEIM
Data and Web Science Group

1. Mining association rules from large amounts of data can be **computationally expensive**
   - algorithms need to apply smart pruning strategies
2. Algorithms often discover a **large number of associations**
   - many of them are uninteresting or redundant
   - the user needs to select the subset of the associations that is relevant given her task at hand

The Association Rule Mining Task

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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:
  1. List all possible association rules
  2. Compute the support and confidence for each rule
  3. Remove rules that fail the *minsup* and *minconf* thresholds

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

Mining Association Rules

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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
  - Thus, we may decouple the support and confidence requirements

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

Apriori Algorithm: Basic Idea

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Two-step approach:

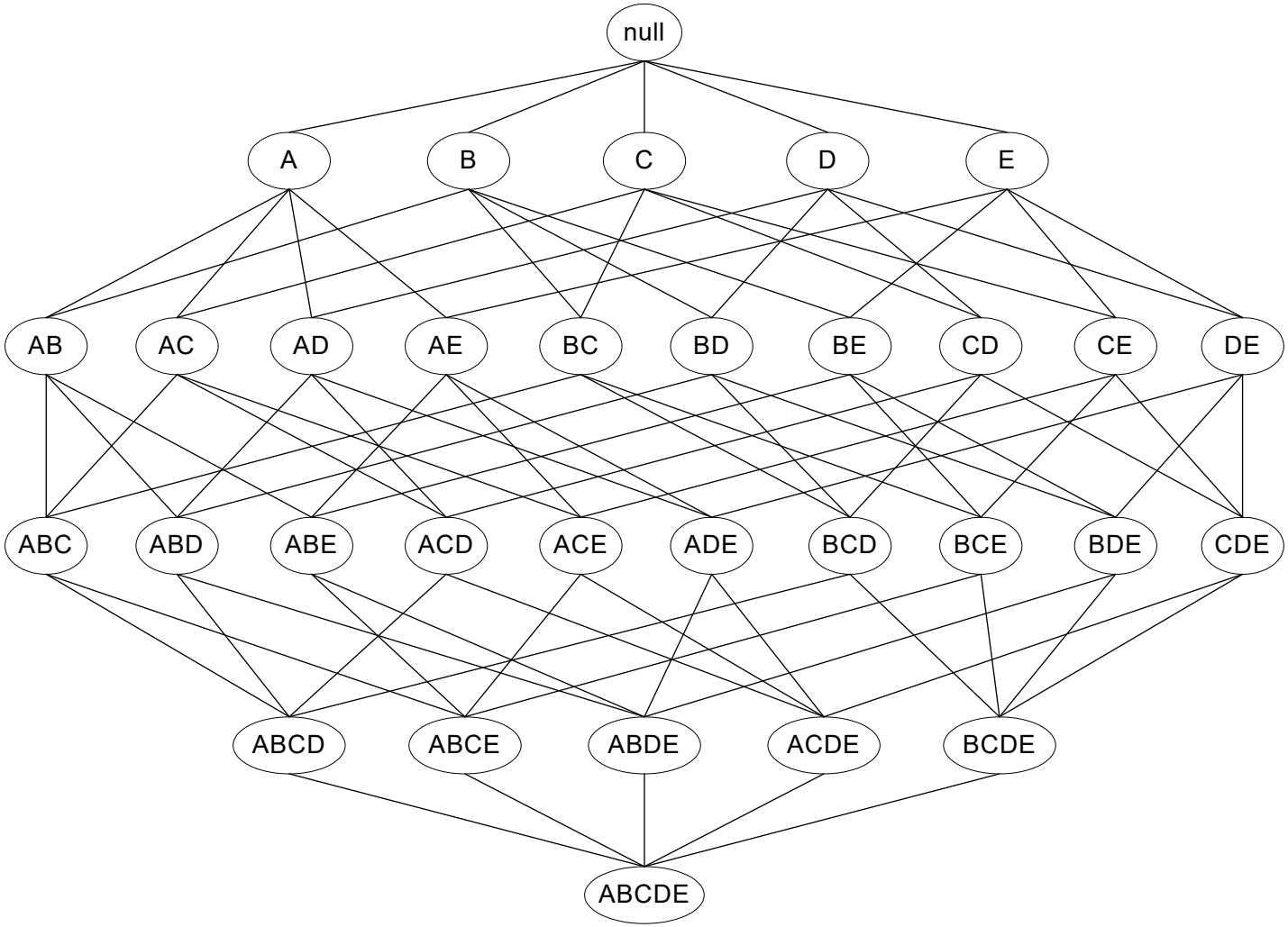
1. Frequent Itemset Generation
   - Generate all itemsets whose support $\geq$ minsup

2. Rule Generation
   - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

2. Frequent Itemset Generation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

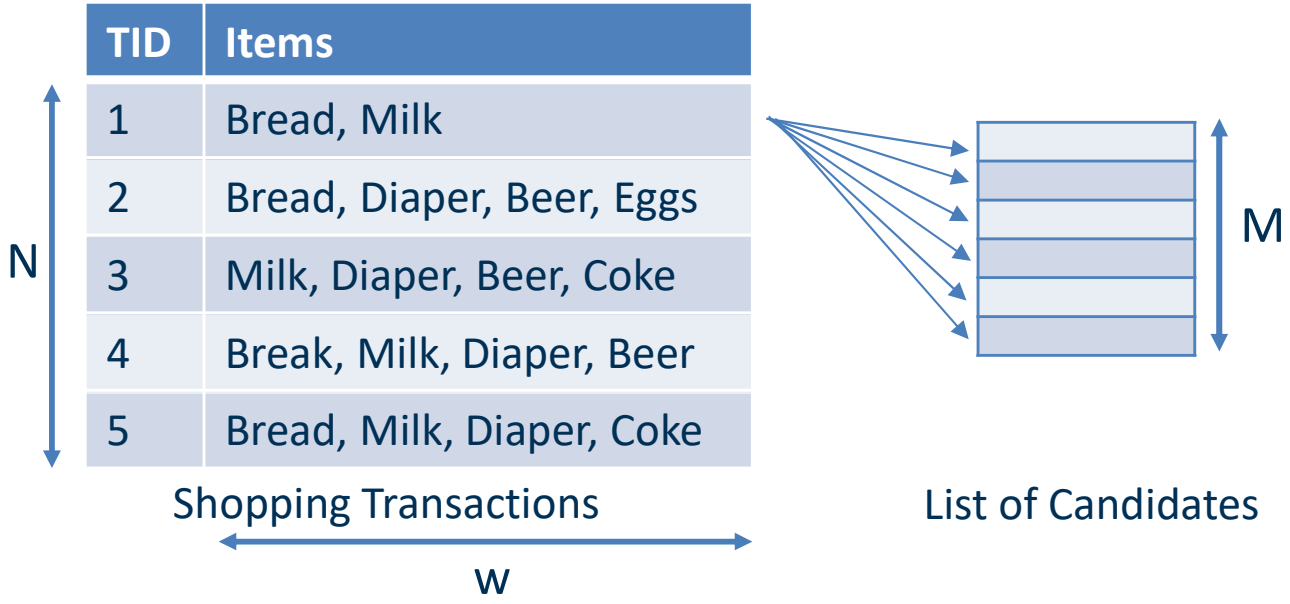
- Given d items, there are 2^d candidate itemsets!



Brute-force Approach

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

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

Brute-force Approach

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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 frequent at all, e.g., books on Chinese calligraphy, Inuit cooking, and data mining bought together
  - Thus, smarter algorithms should be possible
  - Intuition for smarter algorithms: 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

- There is a pattern here!

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

Reducing the Number of Candidates

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

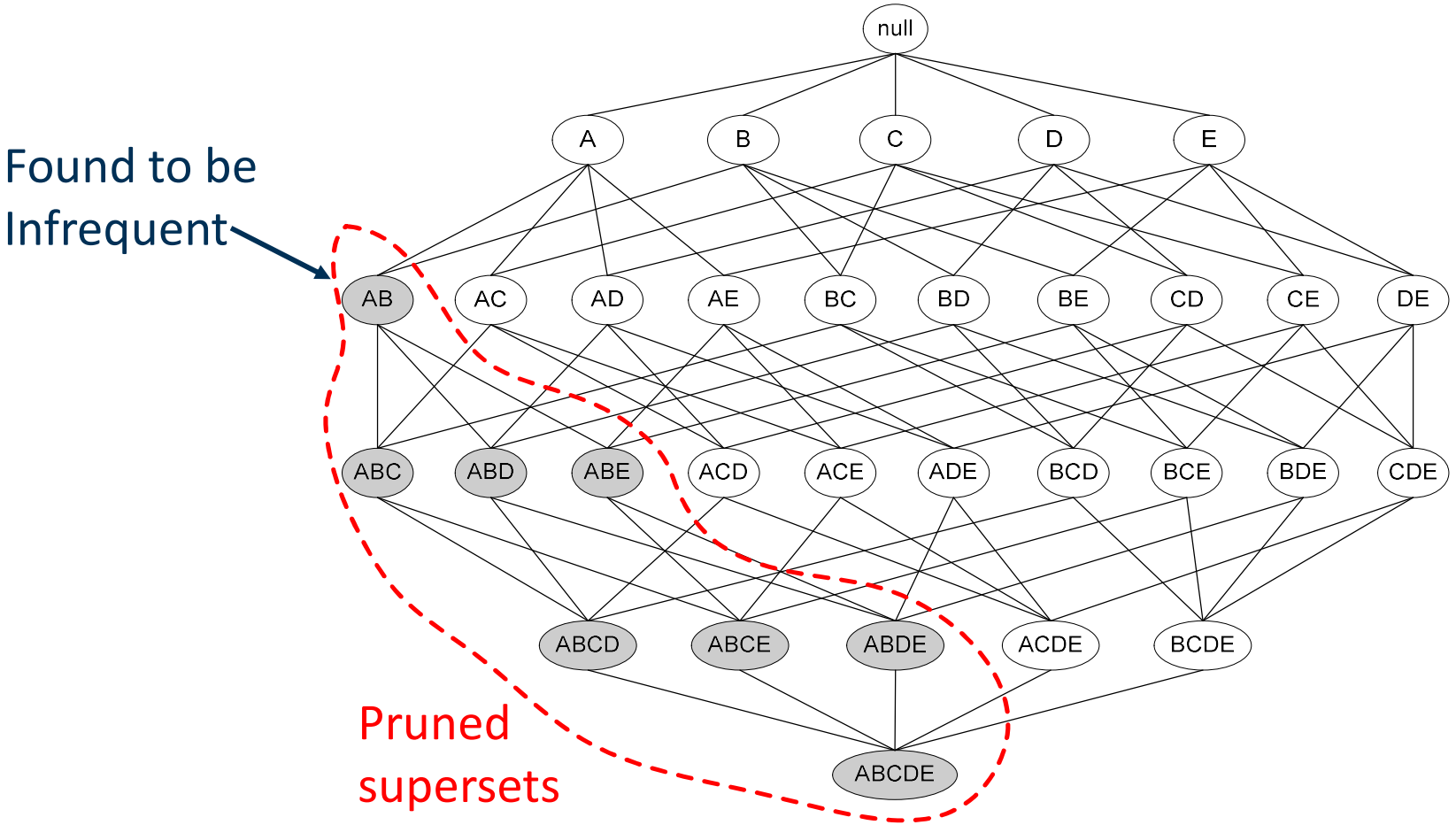
$$\forall X, Y: (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

- 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

UNIVERSITY OF MANNHEIM
Data and Web Science Group

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



The Apriori Algorithm

UNIVERSITY OF MANNHEIM
Data and Web Science Group

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

Illustrating the Apriori Principle

Minimum Support Count = 3

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Items (1-itemsets)

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

No need to generate candidates involving Coke or Eggs

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

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 in total**

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

Pruning strongly reduced the number of itemsets for which we need to scan the DB

Example of Frequent Itemsets

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Result History | FrequentItemSets

Data

Annotations

No. of Sets: 83

Total Max. Size: 4

Min. Size: 1

Max. Size: 4

Contains Item:

Update View

Size	Support ↓	Item 1	Item 2	Item 3
1	0.600	Asus EeePC		
1	0.500	LT Minimaus		
1	0.500	2 GB DDR3 RAM		
2	0.500	Asus EeePC	2 GB DDR3 RAM	
1	0.400	ThinkPad X220		
1	0.400	Netbook-Schutzhülle		
1	0.400	Lenovo Tablet Sleeve		
1	0.400	LT Laser Maus		
2	0.400	Asus EeePC	LT Minimaus	
2	0.400	Asus EeePC	Netbook-Schutzhülle	
2	0.400	2 GB DDR3 RAM	Netbook-Schutzhülle	
3	0.400	Asus EeePC	2 GB DDR3 RAM	Netbook-Schutzhülle
1	0.300	HP Laserjet P2055		
1	0.300	HP CE50 Toner		
2	0.300	LT Minimaus	2 GB DDR3 RAM	
2	0.300	LT Minimaus	Netbook-Schutzhülle	
2	0.300	ThinkPad X220	Lenovo Tablet Sleeve	
2	0.300	HP Laserjet P2055	HP CE50 Toner	
3	0.300	Asus EeePC	LT Minimaus	2 GB DDR3 RAM

Application of Frequent Itemsets

UNIVERSITY OF MANNHEIM
Data and Web Science Group

1. Take top-k frequent itemsets of size 2 containing item A
2. Rank second item according to
   - profit made by selling item
   - whether you want to reduce number of items B in stock
   - knowledge about customer preferences
3. Offer special price for combination with top-ranked second item

amazon

3. 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 \ 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:
  - $\underbrace{\{Milk,Diaper\}}_{f} \rightarrow \{Beer\}$

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

Challenge: Combinatorial Explosion

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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 Ø → {…} and {…} → Ø

Challenge: Combinatorial Explosion

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

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

Challenge: Combinatorial Explosion

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

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

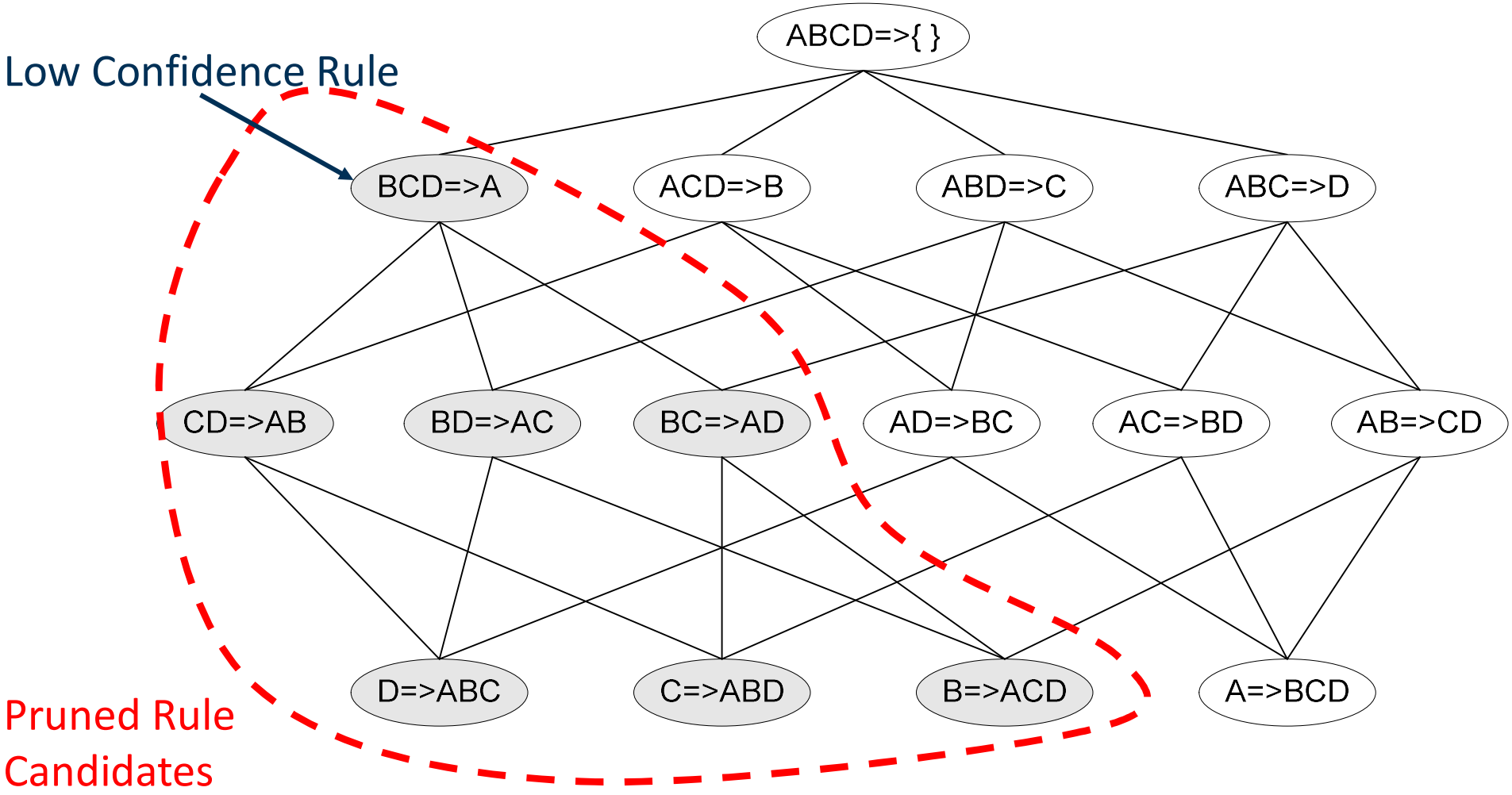
  - Hence

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

Candidate Rule Pruning

UNIVERSITY OF MANNHEIM
Data and Web Science Group

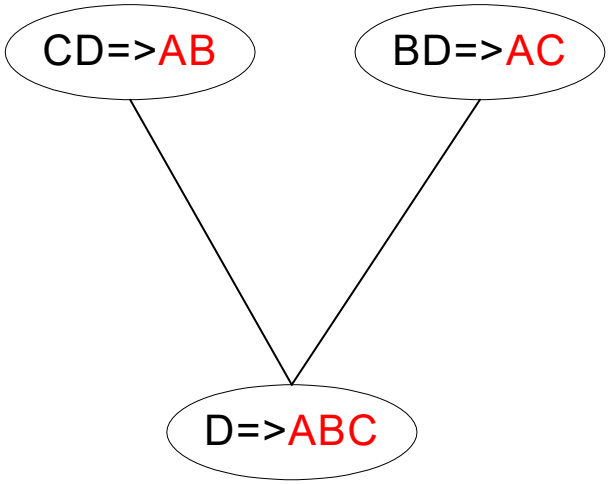
- Moving elements from left to right cannot increase confidence



Rule Generation for Apriori Algorithm

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

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

4. Handling Continuous and Categorical Attributes

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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
…	…	…	…	…	…	…

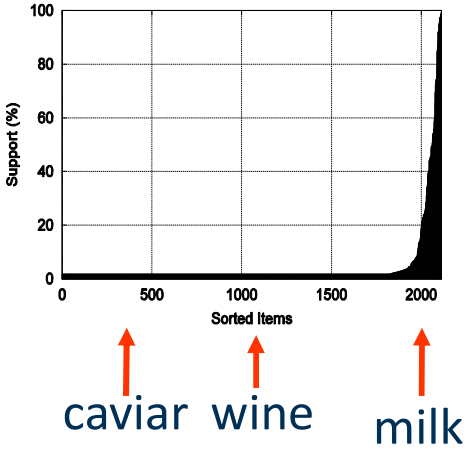
- Example Rule:

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

Handling Categorical Attributes

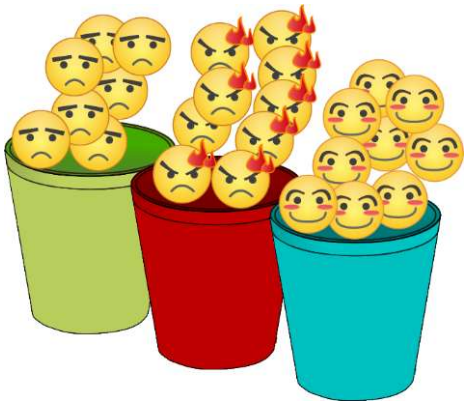
UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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”
    - …..
  - Python: OneHotEncoder
- 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

UNIVERSITY OF MANNHEIM
Data and Web Science Group



- Transform continuous attribute into binary variables using discretization
  - Values of the attribute, e.g., age of a person:
    - 0, 4, 12, 16, 16, 18, 24, 26, 30
  - **Equal-interval binning** – for bin width of e.g., 10:
    - Bin 1: 0, 4 [-∞,10) bin
    - Bin 2: 12, 16, 16, 18 [10,20) bin
    - Bin 3: 24, 26, 30 [20,+∞) bin
  - **Equal-frequency binning** – for bin density of e.g., 3:
    - Bin 1: 0, 4, 12 [-, 14) bin
    - Bin 2: 16, 16, 18 [14, 21) bin
    - Bin 3: 24, 26, 30 [21,+] bin

Handling Continuous Attributes

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

Association Analysis in Python

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Various packages exist
  - In the exercise, we'll use the Orange3 package
  - Frequent Itemset Generation

Python

```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

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

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

Min confidence

5. 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

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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

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) = $\frac{18}{20} = 0.9$

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

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

  - lift < 1, therefore is negatively correlated and removed

Conclusion

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- The algorithm does the counting for you and finds patterns in the data

- You need to do the interpretation based on your knowledge about the application domain.
  - Which patterns are meaningful?
  - Which patterns are surprising?

6. Sequential Pattern Mining

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Association rule mining does not consider **the order of transactions**
- In many applications such orderings are relevant

Sequence Database	Sequence	Element (Transaction)	Event (Item)
Customer Data	Purchase history of a given customer	A set of items bought by a customer at time t	Books, diary products, CDs, etc
Web Server Logs	Browsing activity of a particular visitor	A collection of files viewed by a Web visitor after a single mouse click	Home page, index page, contact info, etc
Sensor Data	History of events generated by a given sensor	Events triggered by a sensor at time t	Types of alarms generated by sensors

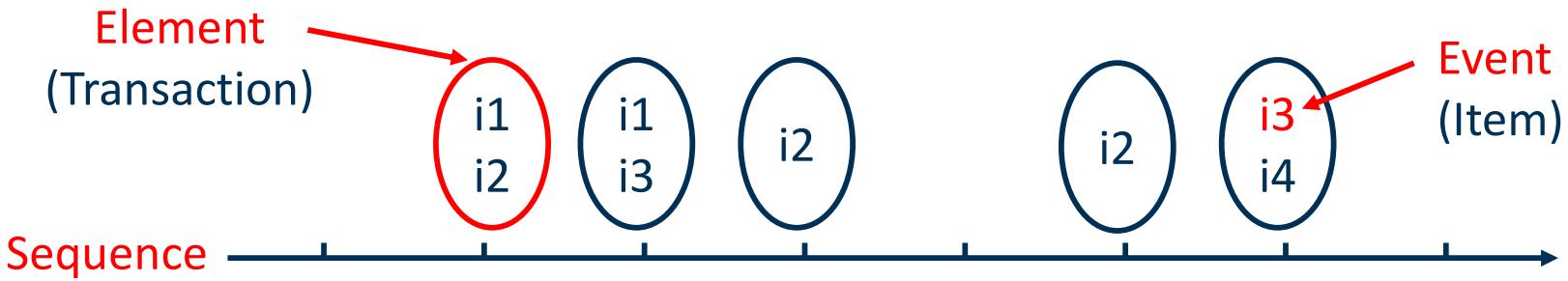
Definition of a Sequence

- A **sequence** is an ordered list of elements (transactions)

$$s = < e_1\ e_2\ e_3\ \ldots >$$

- Each **element** contains a collection of **events** (items)

$$e_i = \{i_1, i_2, \ldots, i_k\}$$

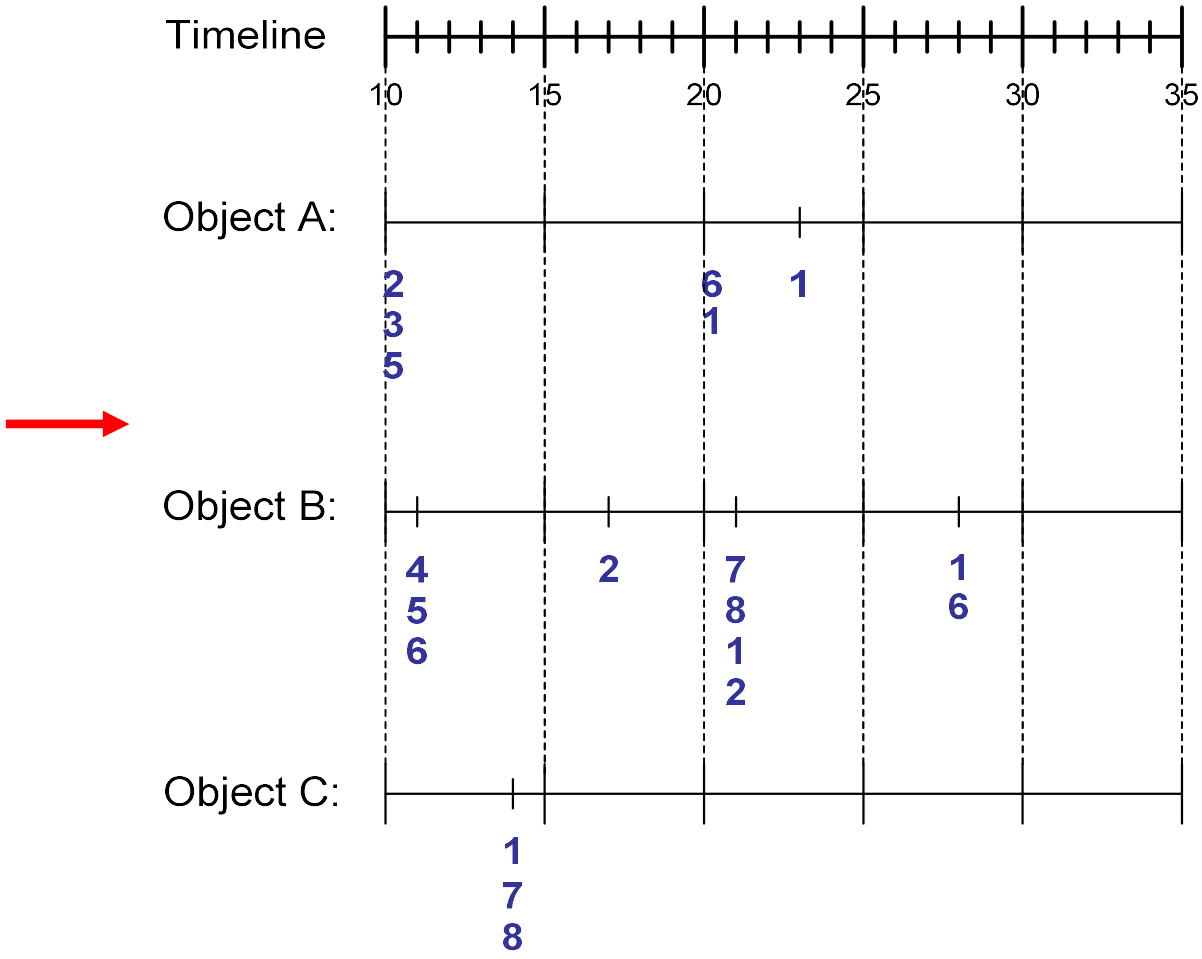


Sequence Data

UNIVERSITY OF MANNHEIM
Data and Web Science Group

Sequence Database:

Object	Timestamp	Events
A	10	2, 3, 5
A	20	6, 1
A	23	1
B	11	4, 5, 6
B	17	2
B	21	7, 8, 1, 2
B	28	1, 6
C	14	1, 8, 7



Sequential Pattern Mining

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Formal Definition of a Subsequence
  - A sequence $\langle a_1\ a_2 \ldots a_n \rangle$ is contained in another sequence $\langle b_1\ b_2 \ldots b_m \rangle$ ($m \geq n$) if there exist integers $i_1 < i_2 < \ldots < i_n$ such that $a_1 \subseteq b_{i_1}$, $a_2 \subseteq b_{i_2}$, …, $a_n \subseteq b_{i_n}$

Data sequence <b>	Subsequence <a>	Contain?
< {2,4} {3,5,6} {8} >	< {2} {3,5} >	Yes
< {1,2} {3,4} >	< {1} {2} >	No
< {2,4} {2,4} {2,5} >	< {2} {4} >	Yes

- The **support** of a subsequence w is defined as the fraction of data sequences that contain w
- A **sequential pattern** is a frequent subsequence (i.e., a subsequence whose support is ≥ minsup)

Sequential Pattern Mining

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Given:
  - a database of sequences
  - a user-specified minimum support threshold, *minsup*
- Task:
  - Find all subsequences with support ≥ *minsup*
- Challenge:
  - Very large number of candidate subsequences that need to be checked against the sequence database.
  - By applying the Apriori principle, the number of candidates can be pruned significantly.

Determining the Candidate Subsequences

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Given n events: $i_1, i_2, i_3, \ldots, i_n$
- Candidate 1-subsequences:

  $\langle\{i_1\}\rangle, \langle\{i_2\}\rangle, \langle\{i_3\}\rangle, \ldots, \langle\{i_n\}\rangle$
- Candidate 2-subsequences:

  $\langle\{i_1, i_2\}\rangle, \langle\{i_1, i_3\}\rangle, \ldots, \langle\{i_1\}\ \{i_1\}\rangle, \langle\{i_1\}\ \{i_2\}\rangle, \ldots, \langle\{i_{n-1}\}\ \{i_n\}\rangle$
- Candidate 3-subsequences:

  $\langle\{i_1, i_2, i_3\}\rangle, \langle\{i_1, i_2, i_4\}\rangle, \ldots, \langle\{i_1, i_2\}\ \{i_1\}\rangle, \langle\{i_1, i_2\}\ \{i_2\}\rangle, \ldots,$

  $\langle\{i_1\}\ \{i_1, i_2\}\rangle, \langle\{i_1\}\ \{i_1, i_3\}\rangle, \ldots, \langle\{i_1\}\ \{i_1\}\ \{i_1\}\rangle, \langle\{i_1\}\ \{i_1\}\ \{i_2\}\rangle, \ldots$

Generalized Sequential Pattern Algorithm (GSP)

UNIVERSITY OF MANNHEIM
Data and Web Science Group

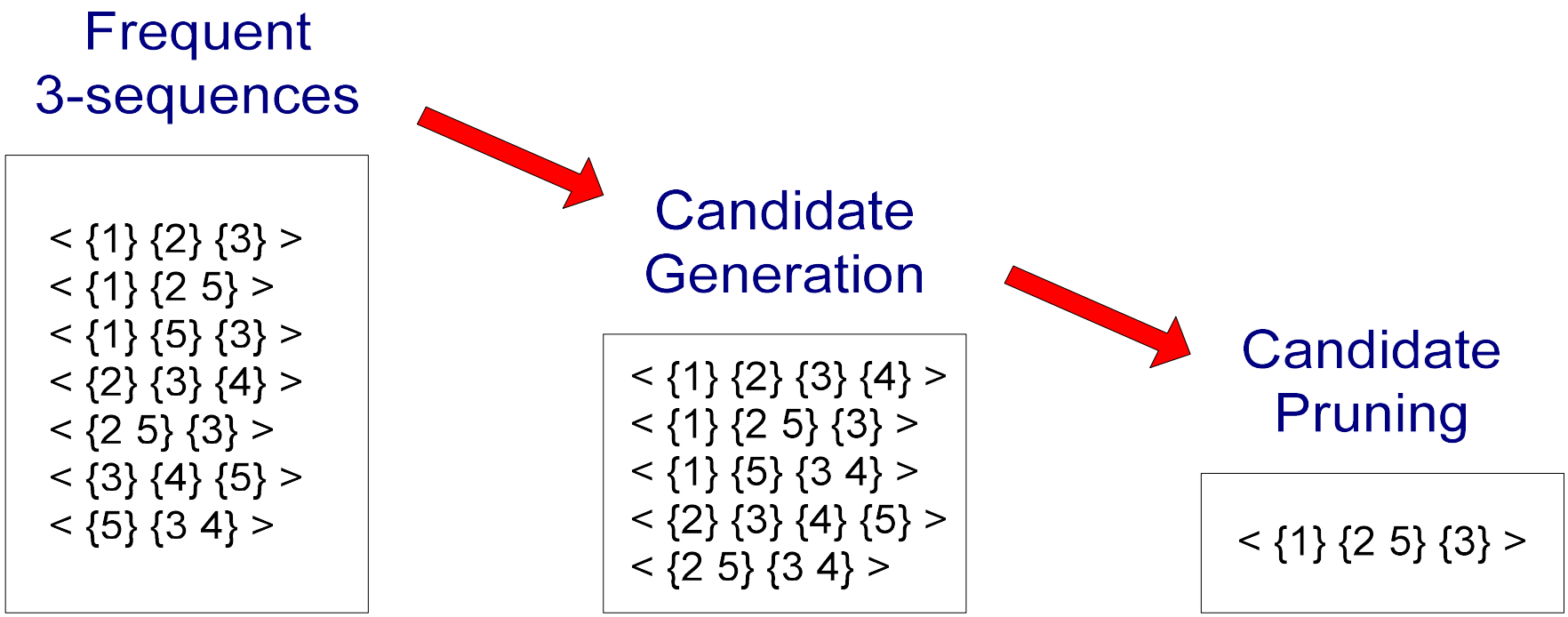
- Make the first pass over the sequence database D to yield all the 1-element frequent subsequences
- Repeat until no new frequent subsequences are found
  1. Candidate **Generation**:
     Merge pairs of frequent subsequences found in the (k-1)*th* pass to generate candidate sequences that contain k items
  2. Candidate **Pruning**:
     Prune candidate *k*-sequences that contain infrequent (*k-1)*-subsequences (**Apriori principle**)
  3. Support **Counting**:
     Make a new pass over the sequence database D to find the support for these candidate sequences
  4. Candidate **Elimination**:
     Eliminate candidate *k*-sequences whose actual support is less than *minsup*

Candidate Generation

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Base case (k=2):
  - Merging two frequent 1-sequences $<\{i_1\}>$ and $<\{i_2\}>$ will produce two candidate 2-sequences: $<\{i_1\}\ \{i_2\}>$ and $<\{i_1\ i_2\}>$
- General case (k>2):
  - A frequent (*k-1*)-sequence w_1 is merged with another frequent (*k-1*)-sequence w_2 to produce a candidate *k*-sequence if the subsequence obtained by **removing the first event in w_1** is the same as the subsequence obtained by **removing the last event** in w_2
  - The resulting candidate after merging is given by the sequence w_1 extended with the last event of w_2.
    - If the last two events in w_2 belong to the same element, then the last event in w_2 becomes part of the last element in w_1
    - Otherwise, the last event in w_2 becomes a separate element appended to the end of w_1

GSP Example



- Only one 4-sequence survives the candidate pruning step
- All other 4-sequences are removed because they contain subsequences that are not part of the set of frequent 3-sequences.

Timing Constraints

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Timing constraints allow us to pose **additional restrictions** on whether a sequence is counted to support a pattern or not.
- Motivating Example:

  < {Statistics} {Database Systems} {Data Mining} >

  < {Database Systems} {Statistics} {Data Mining} >
- We are interested in students that support the pattern

  < {Database Systems, Statistics} {Data Mining} >
- We don’t care about the order of Database Systems and Statistics.
- We care about that the gap between these courses and Data Mining is not too long.

Window Size

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Spezifies a time window in the data sequence in which all events will be considered to belong to the same element.
- Given a candidate pattern: <{a, c}>
- Any data sequences that contain

  <… {a c} … >,
  <… {a} … {c}…> (where time({c}) – time({a}) $\leq$ ws)
  <…{c} … {a} …> (where time({a}) – time({c}) $\leq$ ws)

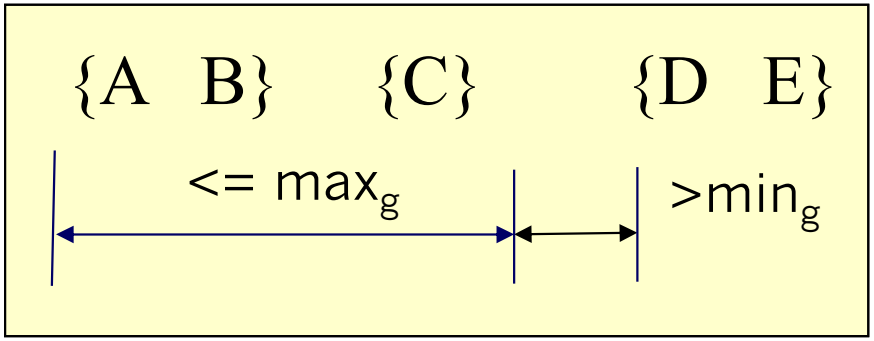
  will contribute to the support count of the candidate pattern
- max_g = 2, min_g = 0, ws = 1

Data sequence	Subsequence	Contain?
< {2,4} {3,5,6} {4,7} {4,6} {8} >	< {3} {5} >	No
< {1} {2} {3} {4} {5}>	< {1,2} {3} >	Yes
< {1,2} {2,3} {3,4} {4,5}>	< {1,2} {3,4} >	Yes

Max-Gap, Min-Gap

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- Max-Gap: Sequence is counted if gap between consecutive elements is smaller than max_g
- Min-Gap: Sequence is counted if gap between consecutive elements is bigger than min_g
- $max_g = 2$, $min_g = 0$



Data sequence	Subsequence	Contain?
< {2,4} {3,5,6} {4,7} {4,5} {8} >	< {6} {5} >	Yes
< {1} {2} {3} {4} {5}>	< {1} {4} >	No
< {1} {2,3} {3,4} {4,5}>	< {2} {3} {5} >	Yes
< {1,2} {3} {2,3} {3,4} {2,4} {4,5}>	< {1,2} {5} >	No

Literature for this Slideset

UNIVERSITY OF MANNHEIM
Data and Web Science Group

- 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