Foundations of Knowledge Management:

Association Rules

Markus Strohmaier

(with slides based on slides by Mark Kröll)

Today's Outline

- Association Rules
  - Motivating Example
  - Definitions
  - The Apriori Algorithm
  - Limitations / Improvements

- Acknowledgements / slides based on:
  - Lecture „Introduction to Machine Learning“ by Albert Orriols i Puig (Illinois Genetic Algorithms Lab)
  - Lecture „Data Management and Exploration“ by Thomas Seidl (RWTH Aachen)
  - Lecture “Association Rules” by Berlin Chen
  - Lecture “PG 402 Wissensmanagment” by Z. Jerroudi
  - Lecture “LS 8 Informatik Computergestützte Statistik“ by Morik and Weihs
  - Association Rules by Prof. Tom Fomby

Today we learn

- Why Association Rules are useful?
  - history + motivation
- What Association Rules are?
  - definitions
- How we can mine them?
  - the Apriori algorithm
  - Illustrating example
- Which challenges they face?
  - + means to address them

Process of Knowledge Discovery

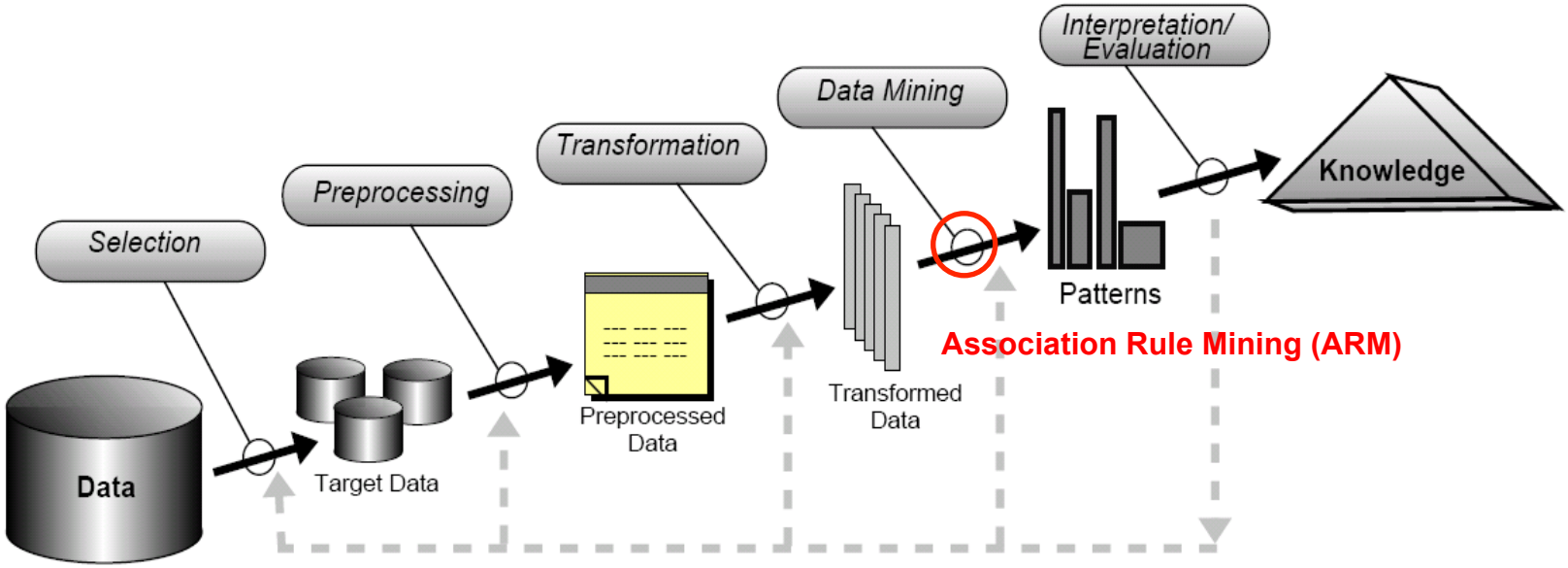
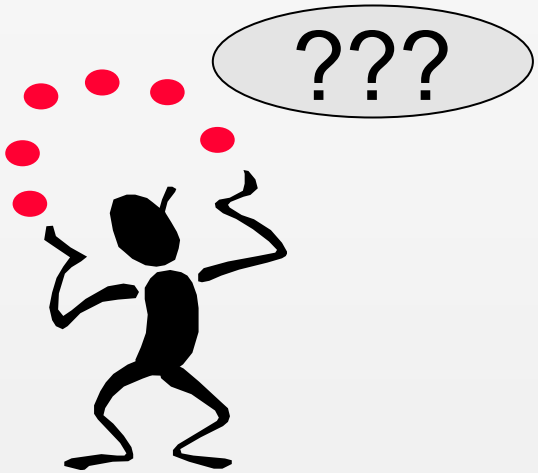


Figure 1: An overview of the steps comprising the KDD process.

Knowledge Discovery and Data Mining: Towards a Unifying Framework (1996)
Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth Knowledge Discovery and Data Mining

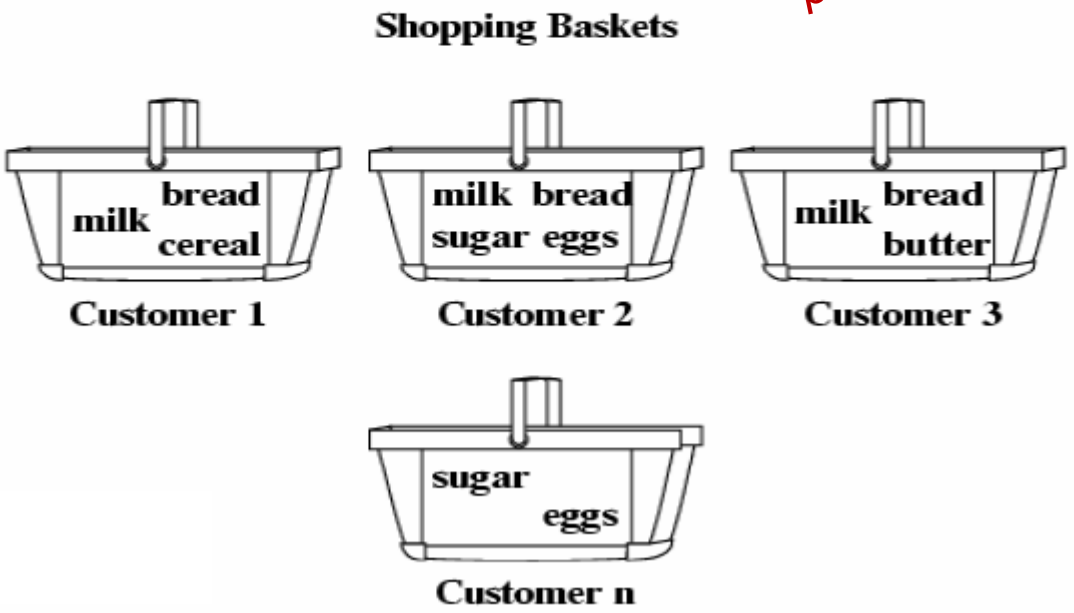
- ARM operates on already structured data (e.g. being in a database)
- ARM represents an unsupervised learning method

Why do we need association rule mining at all?



Motivation for Association Rules(1)

Association Rule Mining can help to better understand purchase behavior!!



For instance, **{beer} => {chips}**

Market Basket Analysis (MBA)(1)

- In retailing, *most purchases are bought on impulse*. Market basket analysis gives clues as to what a customer might have bought *if the idea had occurred to them*.
  - → decide the location and promotion of goods inside a store.

Observation: Purchasers of Barbie dolls are more likely to buy candy.

{barbie doll} => {candy}

→ place high-margin candy near to the Barbie doll display.

Create Temptation: Customers who would have bought candy with their Barbie dolls *had they thought of it* will now be suitably tempted.

Market Basket Analysis (MBA)(2)

- Further possibilities:
  - comparing results between different stores, between customers in different demographic groups, between different days of the week, different seasons of the year, etc.

personalization

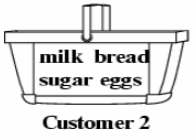
  - If we observe that a rule holds in one store, but not in any other then we know that there is something interesting about that store.
    - different clientele
    - different organization of its displays (in a more lucrative way …)

→ investigating such differences may yield useful insights which will improve company sales.

ReCap: Let's go shopping

- Objective of Association Rule Mining:
  - find associations and correlations between different items (products) that customers place in their shopping basket.
  - to better predict, e.g., :
    (i) what my customers buy? (→ spectrum of products)
    (ii) when they buy it? (→ advertizing)
    (ii) which products are bought together? (→ placement)

Introduction into AR



- Formalizing the problem a little bit
  - Transaction Database T: a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$
  - Each transaction contains a set of items (item set)
  - An item set is a collection of items $I = \{i_1, i_2, \ldots, i_m\}$

- General Aim:
  - Find frequent/interesting patterns, associations, correlations, or causal structures among sets of items or elements in databases or other information repositories.
  - Put this relationships in terms of association rules

$$X \Rightarrow Y$$

    - where X, Y represent two itemsets

Examples of AR

TID	Items
T1	bread, jelly, peanut-butter
T2	bread, peanut-butter
T3	bread, milk, peanut-butter
T4	beer, bread
T5	beer, milk

Examples: Quality?

- bread ⇒ peanut-butter
- beer ⇒ bread

Reads as:
If you buy bread, then you will peanut-butter as well.

- Frequent Item Sets:
  - Items that appear frequently together
  - I = {bread, peanut-butter}
  - I = {beer, bread}

What is an interesting rule?

- Support Count (σ)
  - Frequency of occurrence of an itemset

    σ ({bread, peanut-butter}) = 3

    σ ({beer, bread}) = 1

TID	Items
T1	bread, jelly, peanut-butter
T2	bread, peanut-butter
T3	bread, milk, peanut-butter
T4	beer, bread
T5	beer, milk

- Support (s)
  - Fraction of transactions that contain an itemset

    s({bread, peanut-butter}) = 3/5 (0.6)

    s ({beer, bread}) = 1/5 (0.2)

- Frequent Itemset
  - = an itemset whose support is greater than or equal to a minimum support threshold (minsup)

What is an interesting rule?

- An association rule is an implication of two itemsets

$$X \Rightarrow Y$$

TID	Items
T1	bread, jelly, peanut-butter
T2	bread, peanut-butter
T3	bread, milk, peanut-butter
T4	beer, bread
T5	beer, milk

- Most common measures:
  - Support (s)
    - The occurring frequency of the rule, i.e., the number of transactions that contain both X and Y
  - Confidence (c)
    - The strength of the association, i.e., measures the number of how often items in Y appear in transactions that contain X vs. the number of how often items in X occur in general

$$s = \frac{\sigma(X \cup Y)}{\#\text{ of trans.}}$$

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

Interestingness of Rules

- Let's have a look at some associations + the corresponding measures

TID	s	c
bread ⇒ peanut-butter	0.60	0.75
peanut-butter ⇒ bread		
beer ⇒ bread		
peanut-butter ⇒ jelly		
jelly ⇒ peanut-butter		
jelly ⇒ milk		

TID	Items
T1	bread, jelly, peanut-butter
T2	bread, peanut-butter
T3	bread, milk, peanut-butter
T4	beer, bread
T5	beer, milk

$$s = \frac{\sigma(X \cup Y)}{\#\text{ of trans.}}$$

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

- Support is symmetric / Confidence is asymmetric
- Confidence does not take frequency into account

Confidence vs. Conditional Probability

- Recap Confidence (c)
  - the strength of the association

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

= (number of transactions containing all of the items in X and Y) / (number of transactions containing the items in X)

= (support of X **and** Y)/ (support of X)

= conditional probability Pr(Y | X) = Pr(X **and** Y) / Pr(X)

TID	s	c
jelly ⇒ peanut-butter	0.20	1.00

"If X is bought then Y will be bought with a given probability"

→ "If jelly is bought then peanut-butter will be bought with a probability of 100%

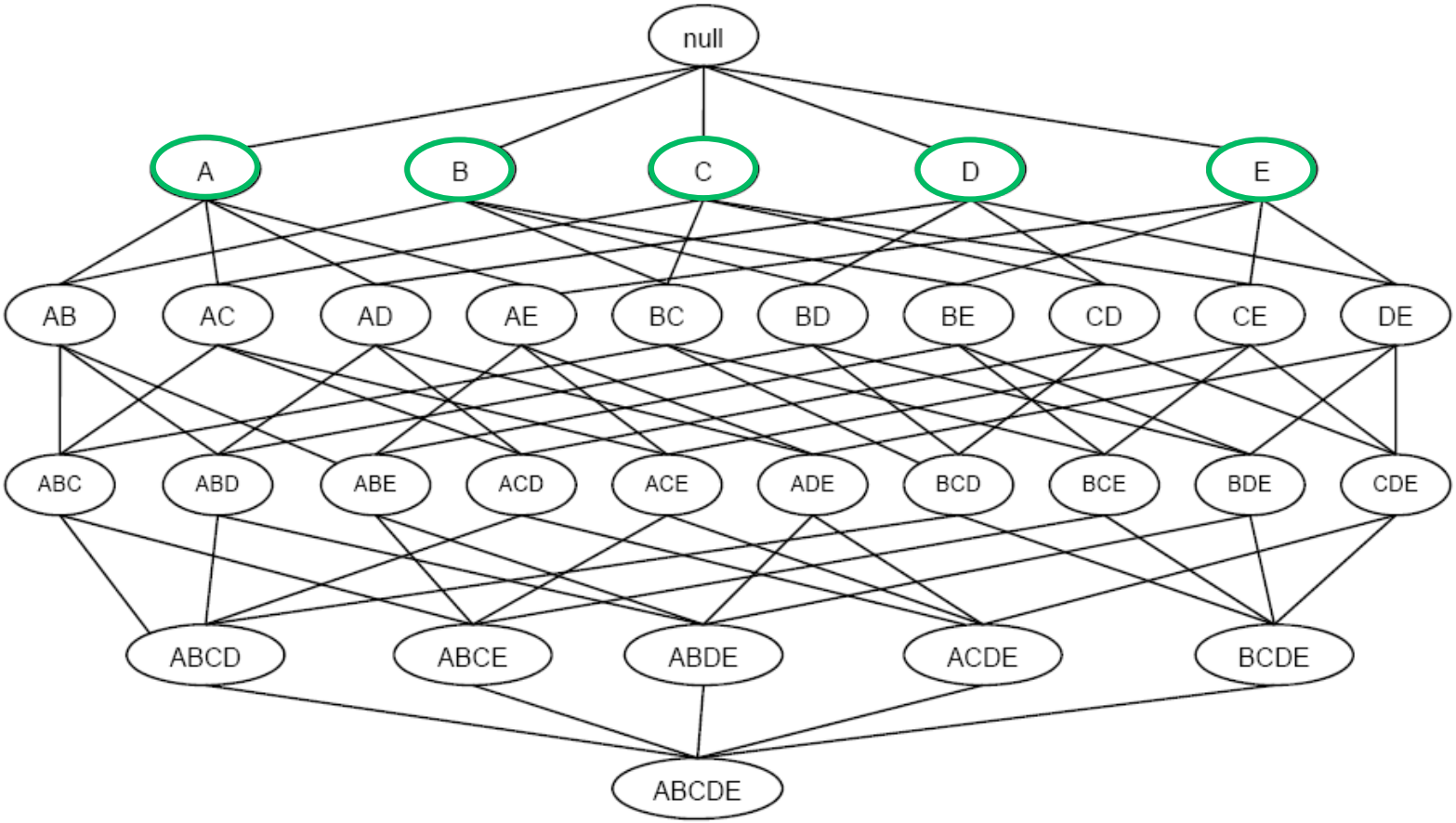
Apriori

- Is the most influential AR miner
  - [Rakesh Agrawal, Tomasz Imieliński, Arun Swami: *Mining Association Rules between Sets of Items in Large Databases.* In: *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data.* 1993.]
- It consists of two steps

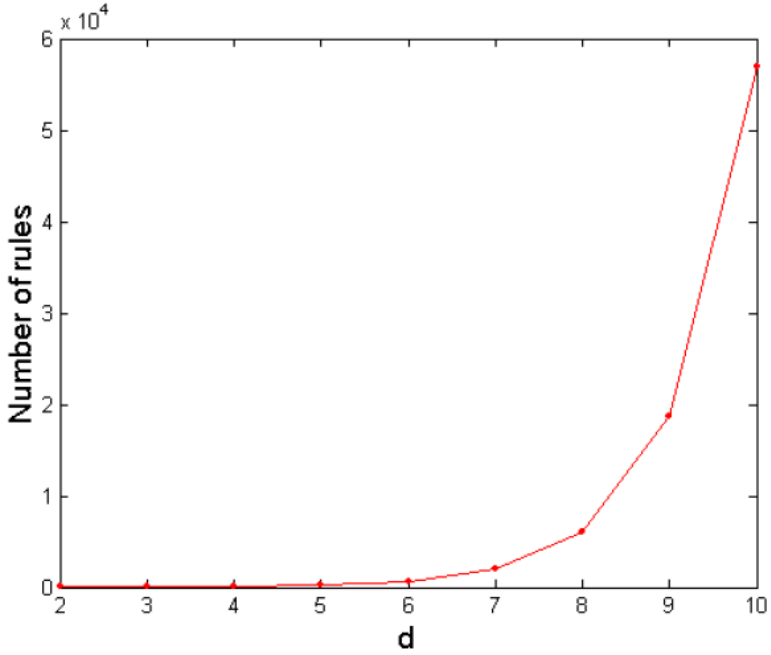
  (1) Generate all frequent itemsets whose support >= minsup

  (2) Use frequent itemsets to craft association rules
- Lets have a look at step one first: Generating Itemsets

Candidate Sets with 5 Items



Computational Complexity



- Given d unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules = $3^d - 2^{d+1} + 1$

→ for d = 5, there are 32 candidate item sets

→ for d = 5, there are 180 rules

d = 25 → $3.4 * 10^7$

d= 25 → $8.5 * 10^{11}$

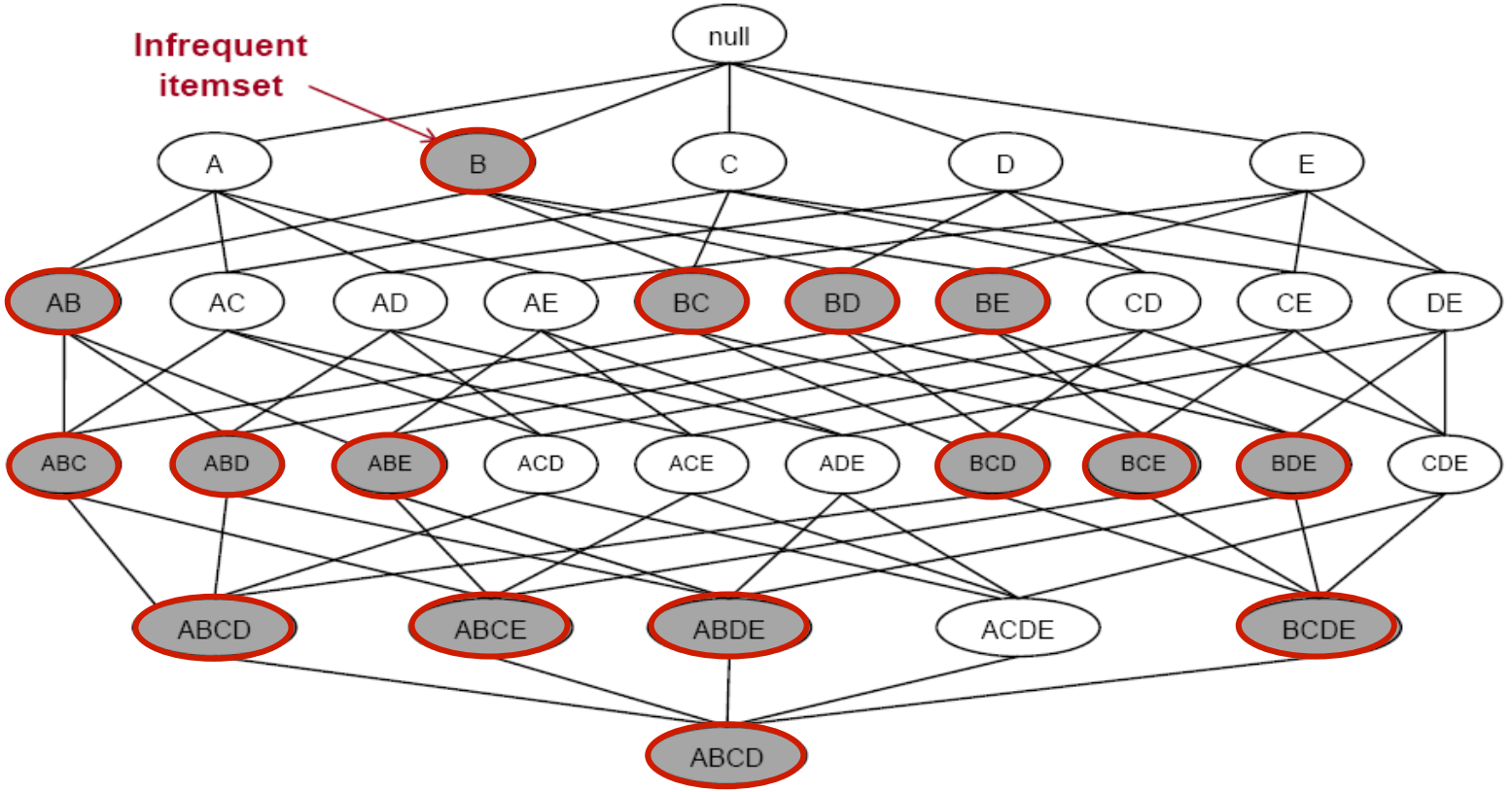
@Generating Itemsets …

- Brute force approach is computationally expensive
  - = take all possible combinations of items
  - → let's select candidates in a smarter way
- Key idea: Downward closure property
  - any subset of a frequent itemset are also frequent itemsets

→ The algorithm iteratively does:

- Create itemsets
- yet, continue exploring only those whose support >= minsup

Example Itemset Generation



- discard infrequent itemsets
  - At the first level B does not meet the required support >= minsup criterion

  → All potential itemsets that contain B can be disregarded (32 → 16)

Let's have a *Frequent Itemset* Example:

Minimum support count = 3

TID	Items
T1	bread, jelly, peanut-butter
T2	bread, peanut-butter
T3	bread, milk, peanut-butter
T4	beer, bread
T5	beer, milk

1-itemsets

Item	count
bread	4
peanut-b	3
jelly	1
milk	1
beer	1

2-itemsets

Item	count
bread, peanut-b	3

Frequent Item Sets for min. support count = 3:

{bread}, {peanut-b} and {bread, peanut-b}

Mining Association Rules

- given the itemset **{bread, peanut-b}** (see last slide)
- corresponding Association Rules:
  - bread → peanut-b. [support = 0.6, confidence = 0.75]
  - peanut-b. → bread [support = 0.6, confidence = 1.0]

- The above rules are binary partitions of the same itemset

- Observation: Rules originating from the same itemset have identical support but can have different confidence

- Support and confidence are decoupled:
  - Support used during candidate generation
  - Confidence used during rule generation

The Apriori – Algorithm(1)

- Let k = 1, set min_support
- Generate frequent itemsets of size 1
- Repeat until no new frequent itemsets are found
  - generate candidate itemsets of size k+1 from size k frequent itemsets
  - prune candidate itemsets containing subsets of size k that are infrequent
  - compute the support of each candidate by scanning the transaction DB
  - eliminate candidates that are infrequent, leaving only those that are frequent

The Apriori – Algorithm(2)

C_k: Candidate itemset of size k
L_k : frequent itemset of size k

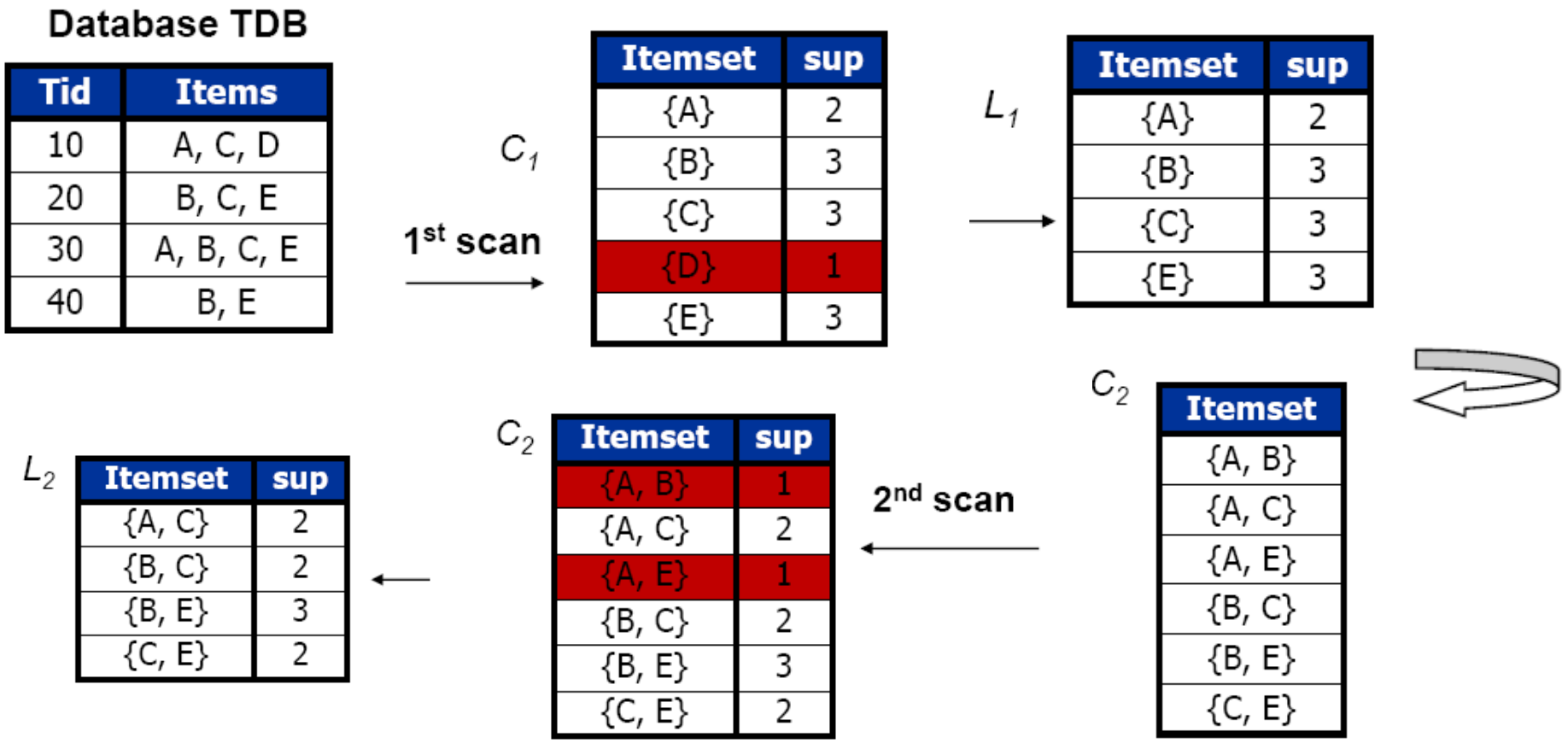
L_1 = {frequent items};
for (k = 1; L_k !=∅; k++) do begin
    C_{k+1} = candidates generated from L_k;
    for each transaction t in database do
        increment the count of all candidates in C_{k+1}
        that are contained in t
    L_{k+1} = candidates in C_{k+1} with min_support
    end
return $\cup_k L_k$;

The Apriori – Algorithm(3)

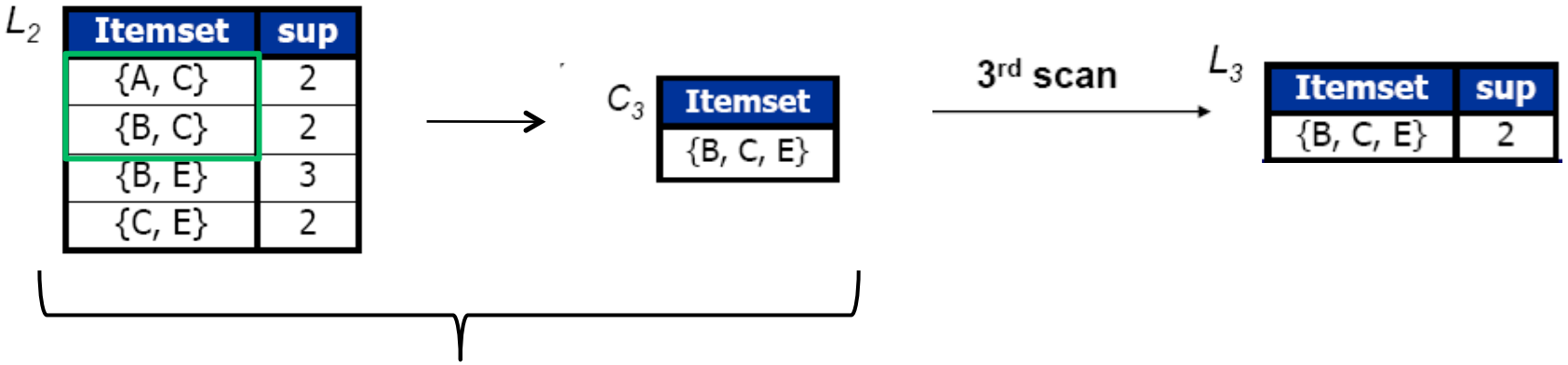
- Join Step
  - C_k is generated by joining L_{k-1} with itself
- Prune Step
  - Any (k-1) itemset that is not frequent cannot be a subset of a frequent k-itemset

Example of Apriori Run (1)

Minimum support count = 2



Example of Apriori Run(2)



Why not e.g. {A,B,C}?

→ only {A,C} and {B,C} are frequent 2-item sets
{A,B} is not
→ decrease database scans

Apriori – The Second Step

- At this stage, we have all frequent itemsets

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

+

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

+

Itemset	sup
{B, C, E}	2

→ now we use these itemsets to generate association rules

Rule Generation in Apriori

- given a frequent itemset L
  - Find all non-empty subsets F in L, such that the association rule F => {L-F} satisfies the minimum confidence
  - Create rule F => {L-F}

- if L = {A, B, C}
  - The candidate itemsets are:

    AB=>C BC = > A AC=> B

    C=>AB A => BC B=>AC

  - In general, there are 2^k-2 candidate solutions,
    - where k is the size of itemset L

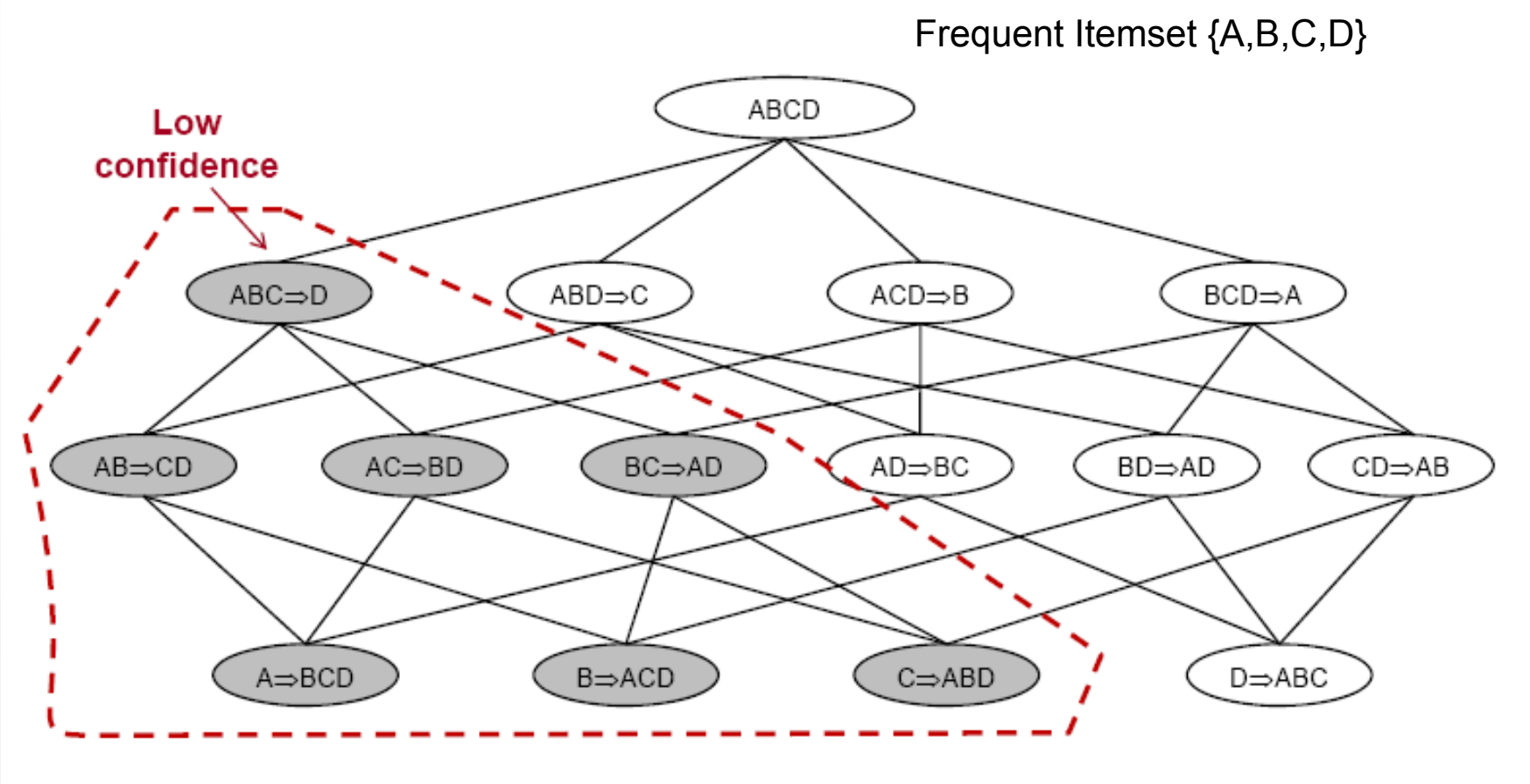
Can we be more efficient?

- can we apply the same heuristic used with itemset generation?
  - Confidence does not have anti-monotone property
  - That is, c(AB=>D) > c(A=>D)?
    - We don't know …
- but confidence of rules generated from the same itemset does have the anti monotone property
  - L = {A, B, C, D}

  → c(ABC =>D) >= c(AB=>CD) >= c(A=>BCD)

  - We can use this property to inform the rule generation

Example of Efficient Rule Generation



@Quality of Generated Rules

- Apriori algorithm produces a lot of rules
  - many of them redundant
  - many of them uninteresting
  - many of them uninterpretable
- Strong Rules can be misleading
  - strong = high support and/or high confidence
  - yet, not all strong association rules are interesting enough to be presented and used (see next slide for an example)

→ If a rule is not interpretable or intuitive in the face of domain-specific knowledge, it need not be adopted and used for decision-making purposes.

Strong Rules Are Not Necessarily Interesting (1)

- Example from [Aggarwal & Yu, PODS98]
- among 5000 students
- 3000 play basketball (=60%), 3750 eat cereal (=75%), 2000 both play basket ball and eat cereal (=40%)
- minsup (40%) and minconf (60%)

- Rule play basketball ⇒ eat cereal [s= 40%, c = 66.7%] is misleading because the overall percentage of students eating cereal is 75% which is higher than 66.7%

  P(eat cereal) > P(eat cereal | play basketball)

  0.75 0.66

  → negative association (“playing basketball” decreases “eat cereal”)

Strong Rules Are Not Necessarily Interesting (2)

- statistical (linear) independence test (e.g. correlation)
  - Heuristics to measure association

    A ⇒ B is interesting if

    - [support(A, B) / support(A)] - support(B) > d
    - or, support(A, B) – [support(A)·support(B)] > k

- example: the association rule in the previous example
  - support(play basketball, eat cereal) - [support(play basketball) ·support(eat cereal)]
  - = 0.4 – [0.6·0.75]
  - = -0.05 < 0 (negative associated !)

Limitations of Apriori

- Bottlenecks:
  - Apriori scans the transaction DB several times
  - usually, there is a large number of candidates
  - calculation of candidate's *support count* can be time-consuming
- Improvements:
  - reduce the number of DB scans
  - shrink the number of candidates
  - more efficient support counting for candidates

Revisiting Candidate Generation

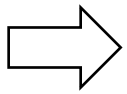
- Remember
  - Use the previous frequent itemsets (k-1) to generate the k-itemsets
  - Count itemsets support by scanning the database

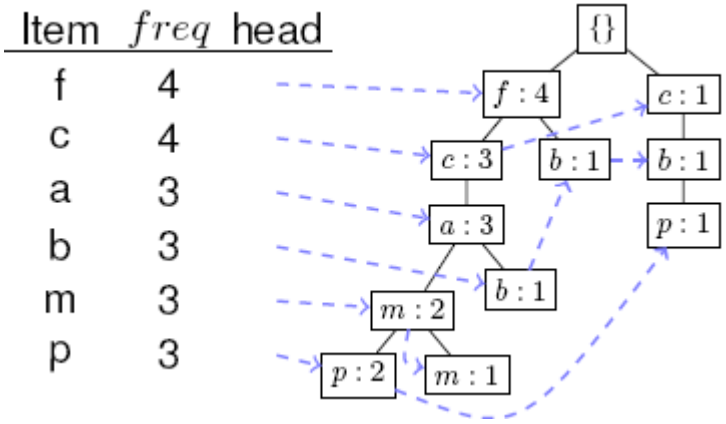
- Bottleneck in the process: Candidate Generation
  - Suppose 100 items
  - First level of the tree → 100 nodes
  - Second level of the tree → $\binom{100}{2}$
  - In general, number of k-itemsets: $\binom{100}{k}$

Avoid Candidate Generation

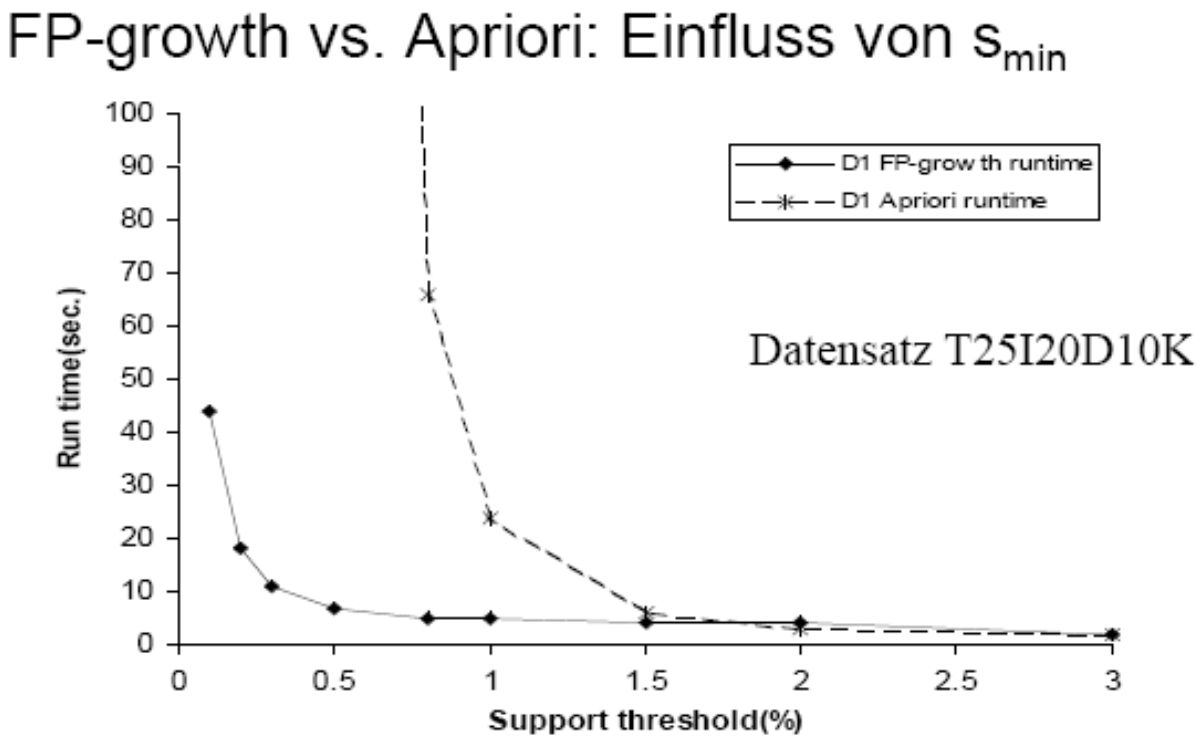
- build auxiliary structure (Frequent Pattern Tree)
  - to get statistics about the itemsets to avoid candidate generation
  - avoid multiple scans of the data

TID	Items bought	(ordered) frequent items
100	$\{f, a, c, d, g, i, m, p\}$	$\{f, c, a, m, p\}$
200	$\{a, b, c, f, l, m, o\}$	$\{f, c, a, b, m\}$
300	$\{b, f, h, j, o\}$	$\{f, b\}$
400	$\{b, c, k, s, p\}$	$\{c, b, p\}$
500	$\{a, f, c, e, l, p, m, n\}$	$\{f, c, a, m, p\}$





- → quick access to nodes of the tree
- for further information see:
  - (Jiawei Han, Jian Pei, Yiwen Yin: Mining Frequent Patterns without Candidate Generation In Proceedings of the 2000 ACM SIGMOD international Conference on Management of Data.)



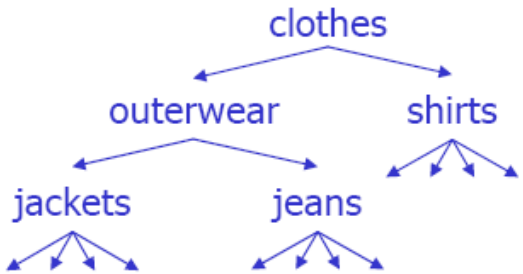
- FP-growth is about a magnitude faster than Apriori because of
  - no candidate generation and testing
  - more compact data structure
  - no iterative database scans

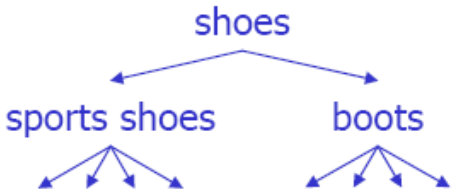
Hierarchical Association Rules

(Srikant & Agrawal , Mining generalized association rules. INnProc. of the 21st Int. Conf. on VLDB, 1995.)

- Problem with plain itemsets (parameter setting):
  - High minsup: apriori finds only few rules
  - Low minsup: apriori finds unmanagably many rules

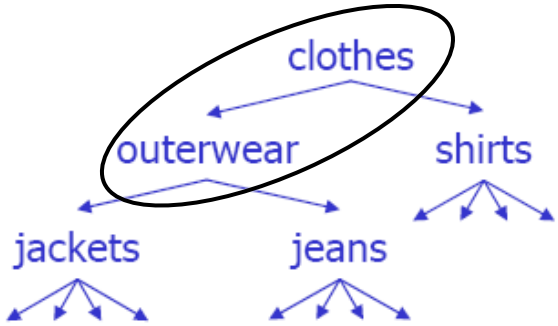
→ exploit item taxonomies (generalizations, is-a hierarchies) which exist in many applications





- Objective: find association rules between generalized items
  - support for sets of item types (e.g., product groups) is higher than support for sets of individual items

Motivation



- Examples:
  - jeans => boots
  - jackets => boots

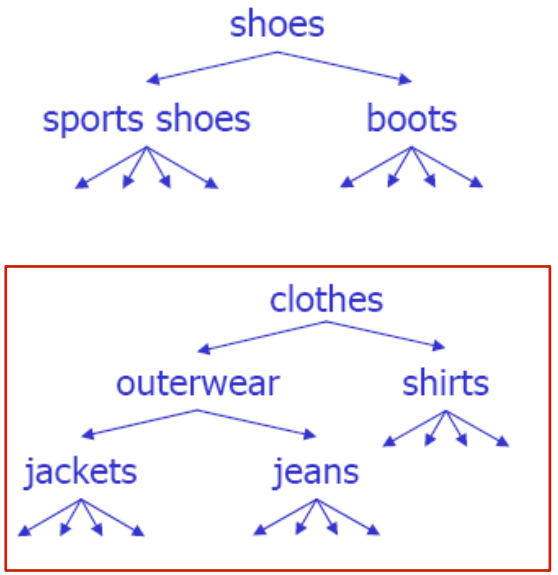
  } support < minsup
  - outerwear => boots

  support > minsup

- Characteristics:
  - support (“outerwear => boots”) is not necessarily equal to the sum support(“jeans => boots”) + support (“jackets => boots”)
  - If the support of rule “outerwear => boots” exceeds minsup, then the support of rule “clothes => boots” does too

Example

transaction id	items
1	shirt
2	jacket, boots
3	jeans, boots
4	sports shoes
5	sports shoes
6	jacket



- Support of {clothes}: 4 of 6 = 67%
- Support of {clothes, boots}: 2 of 6 = 33%
- "shoes => clothes": support 33%, confidence 50%
- "boots => clothes": support 33%, confidence 100%

- Procedure:
  - replace items by items located higher in the hierarchy
  - apply Apriori

Types of Association Rules

- Binary Association Rules
  - Bread => Peanut Butter
- Quantitative Association Rules
  - numeric attributes
  - Weight in [70kg – 90kg] => height in [170cm – 190cm]
- Fuzzy Association Rules
  - allow different degrees of membership (several categories)
  - to overcome the sharp boundary problem
- In this lecture, we focused on Binary Association Rules

Other Application Areas of AR

- Analysis of credit card purchases.
  - identify the most influential factors common to non-profitable customers, e.g. credit card limit, etc.
- Identification of fraudulent medical insurance claims.
  - analyse claim forms submitted by patients to a medical insurance company
  - find relationships among medical procedures that are often performed together
  - might be indicative for fraudulent behavior, when common rules are broken
- Recommendation Systems
  - E.g. **Amazon's** "Customers who bought this item also bought"
  - … is based on association rules

Available Toolkits

- WEKA
  - freely available library implemented in Java
  - provides variants of the Apriori Algorithm

- R
  - http://www.r-project.org/
  - http://rss.acs.unt.edu/Rdoc/library/arules/html/apriori.html

- DBMiner System
  - [Han et al. 1996]

Summary of Today's Lecture (1)

- Association Rules represent an unsupervised learning method
  - that attempts to capture associations between groups of items
- Association Rules are "if-then rules" with two measures
  - which quantify the support and confidence of the rule
  - = if items in group X ap $X \Rightarrow Y$: basket what is the probability that items in group Y will also be purchased?
- Association rule mining is also known as
  - frequent item set mining
  - market basket analysis
  - affinity analysis

Summary of Today's Lecture (2)

- Apriori is most influential rule miner
  - Consisting of two steps:
  - 1) Generating Frequent Itemsets
  - 2) Generating Association Rules from these sets
- Challenges/ Improvements
  - exponential runtime / efficient data structures (FP – tree)
  - rule quality / metrics: interestingness of rules
- Further Directions:
  - Application to sequences in order to look for patterns that evolve over time

Thank you!