Association Rules Mining

K. Gibert[1,2]

(1) Knowledge Engineering and Machine Learning group at
Intelligent Data Science and Artificial Intelligence research Center
Universitat Politècnica de Catalunya, Barcelona

[2] Vicedean on Ethics and Equity
Official Professional College on Informatics Engineering from Catalonia

Universitat Politècnica de Catalunya

Association Rules

- Main Goal:
  Find associations between packs of items

  Items occurring often together can be associated to each other

- Origin: Market basket analysis
  packs of items purchased by customers

Five important algorithms (Yilmaz et al., 2003):

- AIS algorithm 1993
- SETM algorithm 1995
- Apriori (Agrawaal 1994), AprioriTid and AprioriHybrid 1994

Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487-499).

Applications: Association Rule Mining

- $* \Rightarrow$ Maintenance Agreement
  - What the store should do to boost Maintenance Agreement sales
- Home Electronics $\Rightarrow *$
  - What other products should the store stocks up?
- Attached mailing in direct marketing
- Detecting "ping-ponging" of patients
- Marketing and Sales Promotion
- Supermarket shelf management

Itemset

- **Itemset**
  - A collection of one or more items occurring toghether
    - 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

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

Itemset

- **Itemset**
  - A 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**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5

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**

Frequent Itemset

- **Itemset**
  - A 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**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

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 Number of Candidates

- Apriori principle:
  - If an itemset is frequent, then all of its subsets must also be frequent

- Property of monotonicity of the support measure:

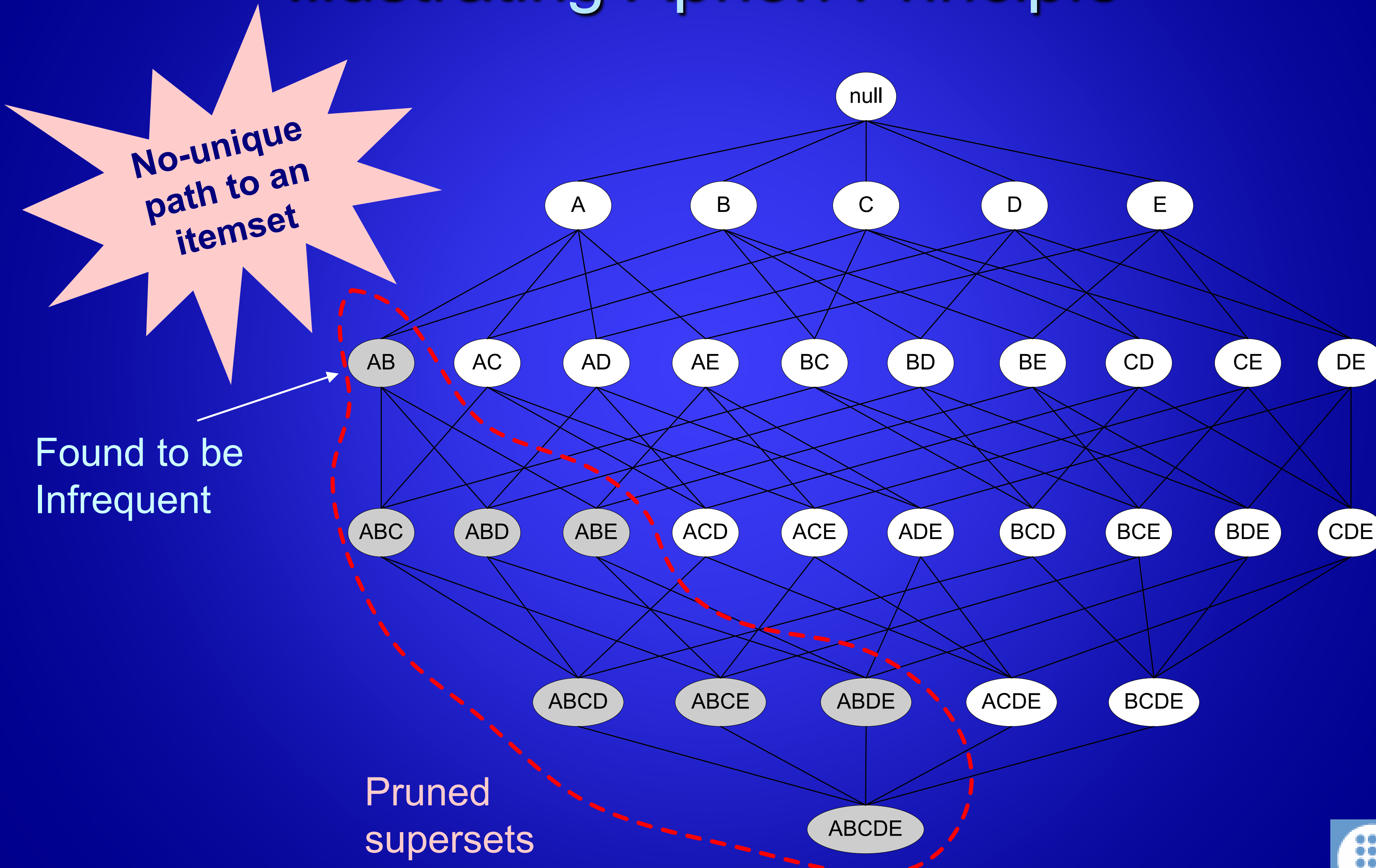
$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

Guarantees
A priori principle

X non-frequent -> Y non-frequent

  - ***anti-monotone*** property of support:
    Support of an itemset never exceeds the support of its subsets
    If X is non-frequent none of its supersets is frequent

Illustrating Apriori Principle



Illustrating Apriori Principle

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

Items (1-itemsets)

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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

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

Minimum Support = 3

Triplets (3-itemsets)

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

If every subset is considered,

$^6C_1 + {}^6C_2 + {}^6C_3 = 41$

With support-based pruning,

$6 + 6 + 1 = 13$

ECLAT algorithm

Finding frequent itemsets

- Method:

    1. Let k=1, $S_o <> \emptyset$
    2. S_1=Generate frequent itemsets of length 1 *(support>minsup)*
    3. While $S_k \backslash S_{k-1} <> \emptyset$
        - Generate length (k+1) candidate itemsets (adding one frequent item to frequent k-itemsets)
        - Count the support of each candidate by scanning the DB
        - Prune candidate itemsets with support $<$ minsup
        - $S_{k+1}=S_k$ U surviving candidates
    4. Return S_{k+1}

Use subset tree or prefix tree to avoid repeated solutions

Finding frequent itemsets

- Method:

1. Let $k=1$, $S_o <> \varnothing$
2. S_1=Generate frequent itemsets of length 1 *(support>minsup)*
3. While $S_k \backslash S_{k-1} <> \varnothing$
   - Generate length (k+1) candidate itemsets (adding one frequent item to frequent k-itemsets)
   - Count the support of each candidate by scanning the DB
   - Prune candidate itemsets with support < minsup
   - $S_{k+1}=S_k$ U surviving candidates
4. Return S_{k+1}

Use subset tree or prefix tree to avoid repeated solutions

r scans of the DB (r maximum length of frequent itemsets)

Association Rules

- Association Rule: An implication expression of the form X → Y, where X and Y are itemsets

r1: {Diaper} → {Beer}
r2: {Milk, Bread} → {Eggs,Coke}
r3: {Beer, Bread} → {Milk}

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**

Only categorical data

ments involved in the rule

items
2: 4 items
r3: 3 items

Dimension of ass. rule: number of items involved

r1: dimemsion 2
r2: dimension 4
r3: dimension 3

Y 1-itemset classification

Association Rules

- Given a frequent itemset

  (i1, i2, i3, i4…….i ℓ)

The number of association rules derived is

$$\sum_{l=1}^{\ell} C_{\ell,l} = \sum_{l=1}^{\ell} \frac{\ell! l!}{(\ell-l)!}$$

Example

(Beer, Milk, Deaper)

Beer -> Milk, Deaper
Milk -> Beer, Deaper
Deaper -> Beer, Milk
Beer, Milk -> Deaper
Beer, Deaper -> Milk
Milk, Deaper -> Beer

Quality of rules

- Interestingness problem (Liu et al., 1999):
  - some generated rules can be self-evident
  - some marginal events can dominate
  - interesting events can be rarely occuring
- Need to estimate how interesting the rules are
- Subjective and objective measures

Subjective measures

- Often based on earlier user experiences/beliefs
- Unexpectedness: rules are interesting if they are unknown or contradict the existing knowledge (or expectations).
- Actionability: rules are interesting if users can get advantage by using them
- Weak and strong beliefs

Objective measures

- Based on threshold values controlled by the user
- Some typical measures (Han and Kamber, 2001):
  - Simplicity (short, small items considered)
  - support (utility)
  - confidence (certainty)

  - **Exact rule:** confidence =100 %
  - Usefulness requires also on high support
  - **strong rule:** high confidence and support
  - Some competing alternative approaches (other that Apriori) can generate useful rules even with low support values

Metrics for Association Rules

- **Association Rule**
  - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
  - Example:
    $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

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**

- **Rule Evaluation Metrics**
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

  \- Lift (c)
    - LIFT

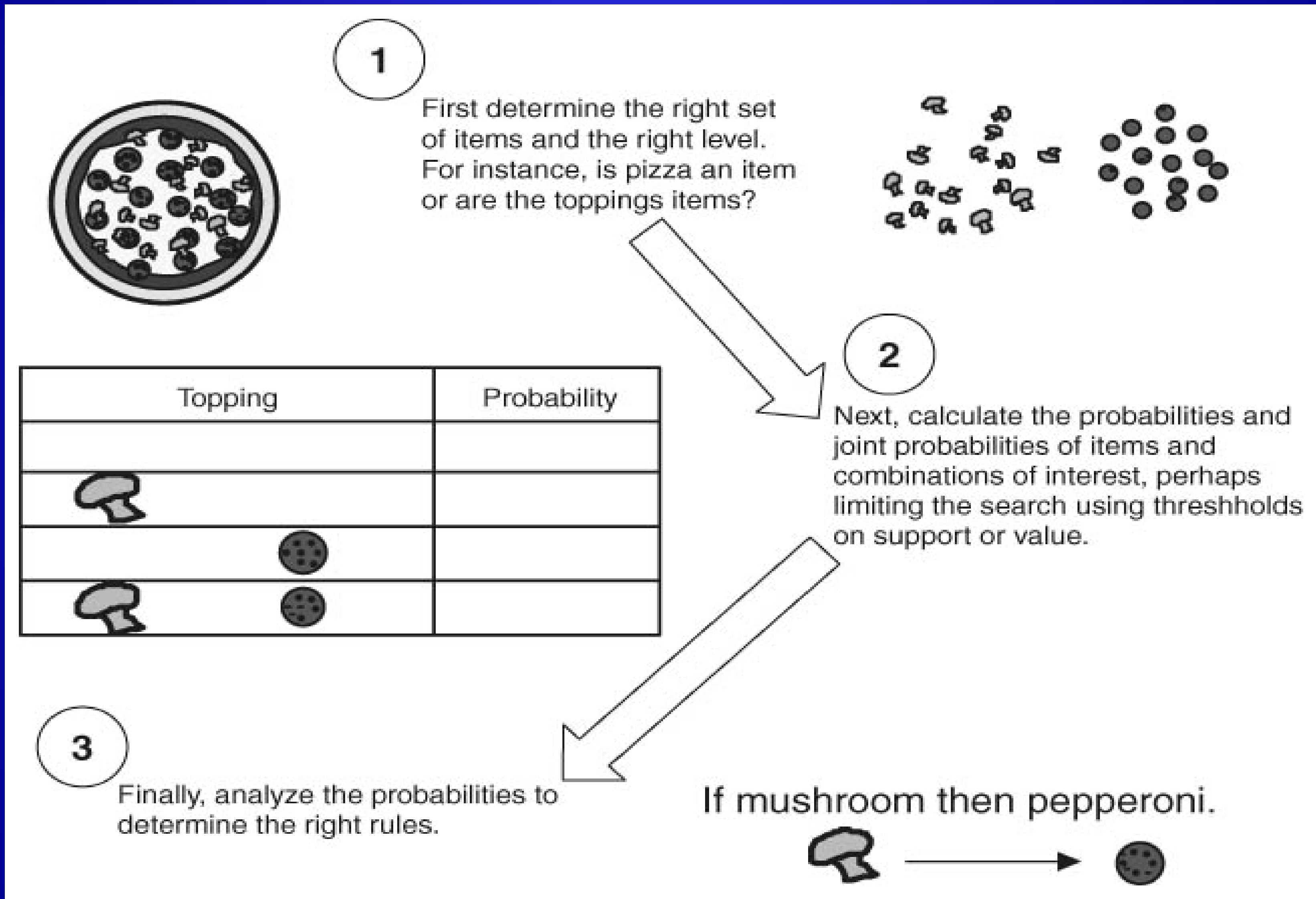
Example:

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

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

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

Creating Association Rules



Generating association rules

- Usually consists of two subproblems (Han and Kamber, 2001):
  1) Find frequent itemsets whith enough occurences
     (predefined minimum support threshold)
  1) Derive association rules from those frequent itemsets
     (*predefined minimum confidence threshold)*

- Solve 1) and 2) iteratively til new rules no more emerge

- Most of the research focus is on the first subproblem

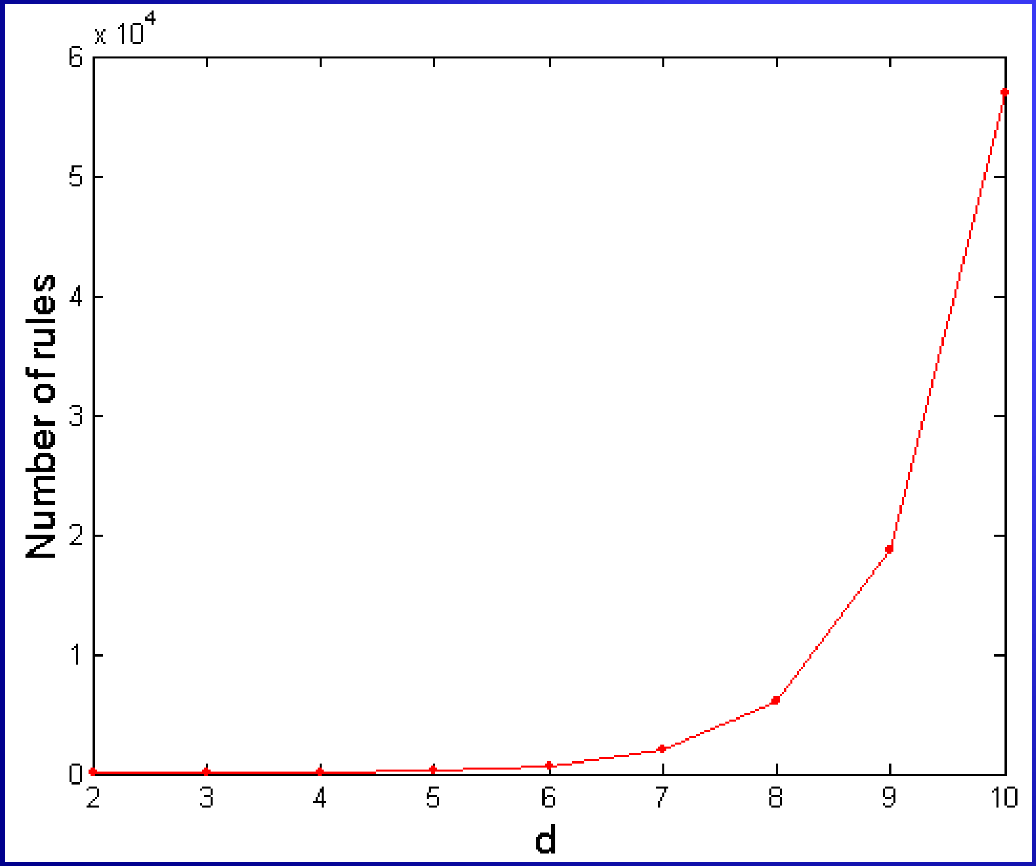
Brute-force approach:

List all possible association rules
Compute the support and confidence for each rule
Prune rules that fail the *minsup* and *minconf* threshold

Computationall prohibitive

Computational Complexity

- Given d unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules:



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

Apriori algorithm

- Developed by Agrawal and Srikant 1994
- Find association rules on large scale
- Allow implication outcomes of several items
- Based on minimum support threshold

Illustrating Apriori From FIS to Ass Rules

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

Items (1-itemsets)

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

Pairs (2-itemsets)

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

Bread -> milk
Milk -> Bread
:
Bread -> Diaper
Diaper -> Bread
:

Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	**3**

Bread -> Milk, Diaper
Milk -> Bread, Diaper
Diaper -> Bread, Milk
Bread, Milk -> Diaper
Bread, Diaper -> Milk
Milk, Diaper -> Bread

Illustrating Apriori From FIS to Ass Rules

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

Confidences might change

Bread -> milk	s: σ(Bread, Milk)/n= 3/5	c: σ(Bread, Milk)/σ(Bread) = ¾
Milk -> Bread	s: σ(Bread, Milk)/n= 3/5	c: σ(Bread, Milk)/σ(Milk) = ¾
Bread -> Beer	s: σ(Bread, Beer)/n= 2/5	c: σ(Bread, Beer)/σ(Bread) = 2/4
Beer -> Bread	s: σ(Bread, Beer)/n= 2/5	c: σ(Bread, Beer)/σ(Beer) = 2/3
Bread -> Diaper	s: σ(Bread, Diaper)/n= 3/5	c: σ(Bread, Diaper)/σ(Bread) = ¾
Diaper -> Bread	s: σ(Bread, Diaper)/n= 3/5	c: σ(Bread, Diaper)/σ(Diaper) = ¾
:	:	
:		

s: σ(Bread, Milk, Diaper)/n= 3/5

Bread -> Milk, Diaper	c: σ(Bread, Milk, Diaper)/σ(Bread) = ¾
Milk -> Bread, Diaper	c: σ(Bread, Milk, Diaper)/σ(Milk) = ¾
Diaper -> Bread, Milk	c: σ(Bread, Milk, Diaper)/σ(Diaper) = ¾
Bread, Milk -> Diaper	c: σ(Bread, Milk, Diaper)/σ(Bread, Milk) = 3/3
Bread, Diaper -> Milk	c: σ(Bread, Milk, Diaper)/σ(Bread, Diaper) = 3/3
Milk, Diaper -> Bread	c: σ(Bread, Milk, Diaper)/σ(Milk, Diaper) = 3/3

.

Apriori Algorithm

- Method:

1. Let k=1, $S_o <> \varnothing$
2. S_1=Generate frequent itemsets of length 1 *(support>minsup)*
3. While $S_k \backslash S_{k-1} <> \varnothing$
    - Generate length (k+1) candidate itemsets (adding one frequent item to frequent k-itemsets)
    - Count the support of each candidate by scanning the DB
    - Prune candidate itemsets with support < minsup
    - $S_{k+1}=S_k$ U surviving candidates
4. Return S_{k+1}

Apriori algorithm

- Developed by Agrawal and Srikant 1994
- Find association rules on large scale
- Allow implication outcomes of several items
- Based on minimum support threshold

- Three versions:
  – Apriori (basic version) faster in first iterations
  – AprioriTid faster in later iteratons
  – AprioriHybrid can change from Apriori to AprioriTid after first iterations

Limitations of Apriori algorithm

- Needs several iterations of the data
- Combinatorial treatment from itemset to rules
- Uses a **uniform** minimum support threshold
- Difficulties to find rarely occuring events
- Alternative using a **non-uniform** min support thresold
- Some competing alternative approaches focus on **partition** and **sampling**