Data Mining

Fosca Giannotti and Mirco Nanni

Pisa KDD Lab, ISTI-CNR & Univ. Pisa

http://www-kdd.isti.cnr.it/

DIPARTIMENTO DI INFORMATICA - Università di Pisa
anno accademico 20011/2012

Association rules and market basket analysis

Association rules - module outline

1. **What are association rules (AR) and what are they used for:**
    1. The paradigmatic application: Market Basket Analysis
    2. The single dimensional AR (intra-attribute)
3. **How to compute AR**
    1. Basic Apriori Algorithm and its optimizations
    2. Multi-Dimension AR (inter-attribute)
    3. Quantitative AR
    4. Constrained AR
5. **How to reason on AR and how to evaluate their quality**
    1. Multiple-level AR
    2. Interestingness
    3. Correlation vs. Association

Market Basket Analysis: the context

Customer buying habits by finding associations and correlations between the different items that customers place in their “shopping basket”

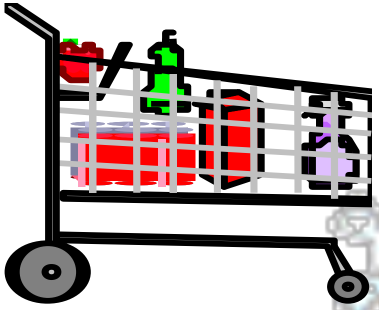
Milk, eggs, sugar, bread

Customer1

Milk, eggs, cereal, bread

Customer2

Eggs, sugar

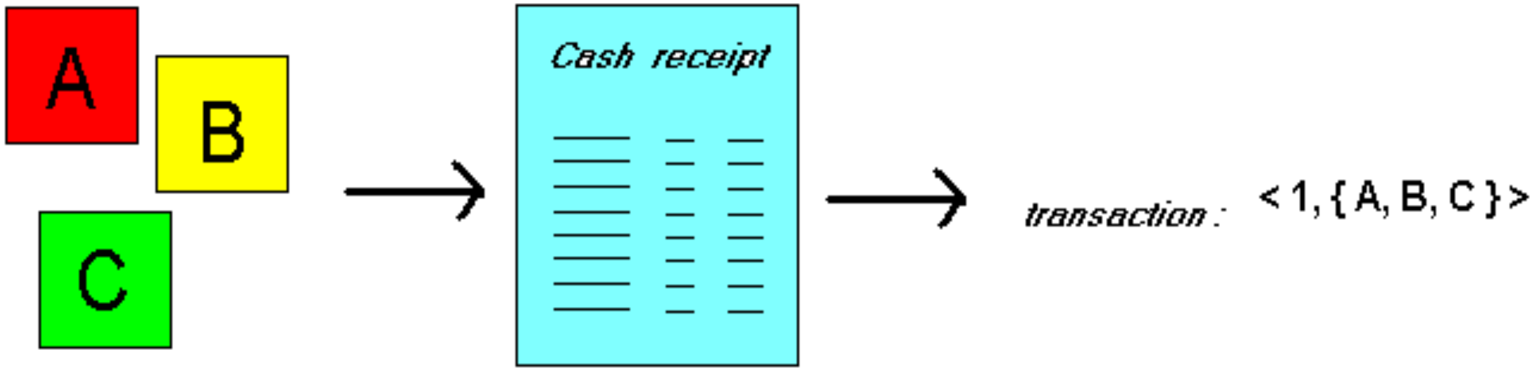


Customer3

Market Basket Analysis: the context

Given: a database of customer transactions, where each transaction is a set of items

- Find groups of items which are frequently purchased together



Goal of MBA

- Extract information on purchasing behavior
- Actionable information: can suggest
  - new store layouts
  - new product assortments
  - which products to put on promotion
- MBA applicable whenever a customer purchases multiple things in proximity
  - credit cards
  - services of telecommunication companies
  - banking services
  - medical treatments

Association Rules

- Express how product/services relate to each other, and tend to group together
- Examples.
  - Rule form: “Body → Head [support, confidence]”.
  - buys(x, “diapers”) → buys(x, “beers”) [0.5%, 60%]
  - major(x, “CS”) ^ takes(x, “DB”) → grade(x, “A”) [1%, 75%]

Useful, trivial, unexplicable

- Useful: “On Thursdays, grocery store consumers often purchase diapers and beer together”.
- Trivial: “Customers who purchase maintenance agreements are very likely to purchase large appliances”.
- Unexplicable: “When a new hardaware store opens, one of the most sold items is toilet rings.”

Association Rules Road Map

- Single dimension vs. multiple dimensional AR
  - E.g., association on items bought vs. linking on different attributes.
  - Intra-Attribute vs. Inter-Attribute
- Qualitative vs. quantitative AR
  - Association on categorical vs. numerical attributes
- Simple vs. constraint-based AR
  - E.g., small sales (sum < 100) trigger big buys (sum > 1,000)?
- Single level vs. multiple-level AR
  - E.g., what brands of beers are associated with what brands of diapers?
- Association vs. correlation analysis.
  - Association does not necessarily imply correlation.

Basic Concepts

Transaction:

Relational format	**Compact format**
<Tid,item>	**<Tid,itemset>**
<1, item1>	<1, {item1,item2}>
<1, item2>	<2, {item3}>
<2, item3>	

Item: single element, **Itemset**: set of items

Support_count of an itemset I: # of transactions containing I

Support of an itemset I: # of transactions containing I/ # Tot. of transactions

Minimum Support MinSup : threshold for support

Frequent Itemset : with support ≥ MinSup.

Frequent Itemsets represents set of items which are positively correlated

Frequent Itemsets

Transaction ID	Items Bought
1	dairy,fruit
2	dairy,fruit, vegetable
3	dairy
4	fruit, cereals

Support({dairy}) = 3/4 (75%)
Support({fruit}) = 3/4 (75%)
Support({dairy, fruit}) = 2/4 (50%)

If σ = 60%, then

{dairy} and {fruit} are frequent while {dairy, fruit} is not.

Definition: Frequent Itemset (repetita juvant)

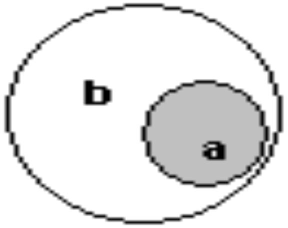
- Itemset
  - A collection of one or more items
    - ✓ Example: {Milk, Bread, Diaper}
  - k-itemset
    - ✓ An itemset that contains k items
- Support count (σ)
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread,Diaper}) = 2
- Support
  - Fraction of transactions that contain an itemset
  - 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

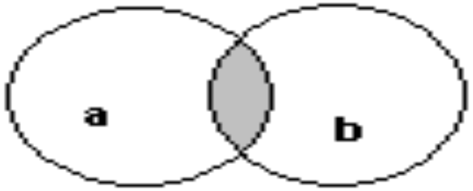
Frequent Itemsets vs. Logic Rules

Frequent itemset I = {a, b} does not distinguish between (1) and (2)

(1)

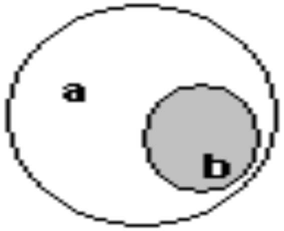


a => b

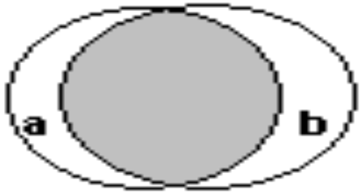


Almost no relation

(2)



b => a



Almost a=>b e b=>a

Logic does: x ⇒ y *iff* when x holds, y holds too

Association Rules: Measures

- Let A and B be a partition of an itemset I :

$$A \Rightarrow B\ [s, c]$$

A and B are itemsets

s = **support of A ⇒ B** = support(A∪B)

c = **confidence of A ⇒ B** = support(A∪B)/support(A)

- Measure for rules:
  - ✓ minimum support σ
  - ✓ minimum confidence γ
- The rules holds if : $s \geq \sigma$ and $c \geq \gamma$

Association Rules: Meaning

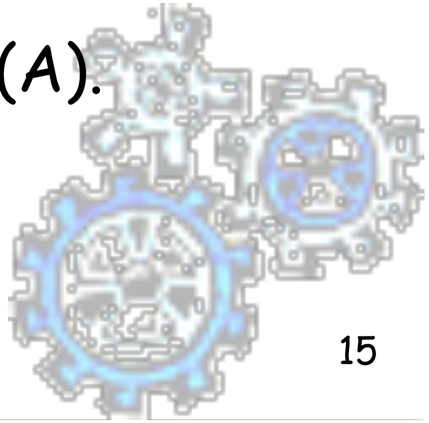
$A \Rightarrow B\ [\ s, c\ ]$

Support: denotes the frequency of the rule within transactions. A high value means that the rule involve a great part of database.

$$\text{support}(A \Rightarrow B) = p(A \cup B)$$

Confidence: denotes the percentage of transactions containing A which contain also B. It is an estimation of conditioned probability .

$$\text{confidence}(A \Rightarrow B) = p(B|A) = p(A \,\&\, B)/p(A).$$



Association Rules - Example

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

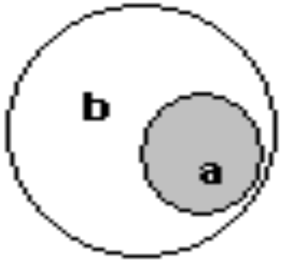
Frequent Itemset	Support
{A}	0,75
{B}	0,50
{C}	0,50
{A,C}	0,50

For rule $A \Rightarrow C$:

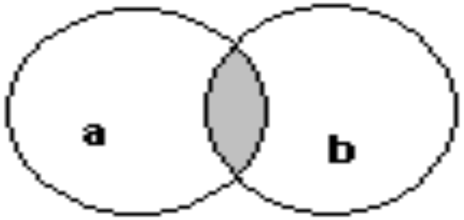
support = support({A, C}) = 50%

confidence = support({A, C})/support({A}) = 66.6%

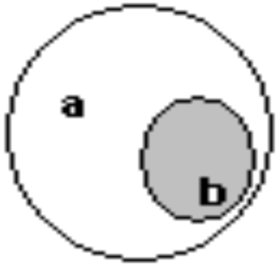
Association Rules – the effect



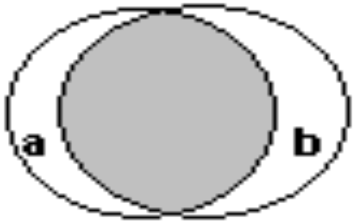
conf(a => b) = 100%
conf(b => a) = ~ 0%



conf(a => b) = ~ 0%
conf(b => a) = ~ 0%



conf(a => b) = ~ 0%
conf(b => a) = 100%



conf(a => b) = ~100%
conf(b => a) = ~100%

Association Rules – the parameters σ and γ

Minimum Support σ :

High ⇒ few frequent itemsets

⇒ few valid rules which occur very often

Low ⇒ many valid rules which occur rarely

Minimum Confidence γ :

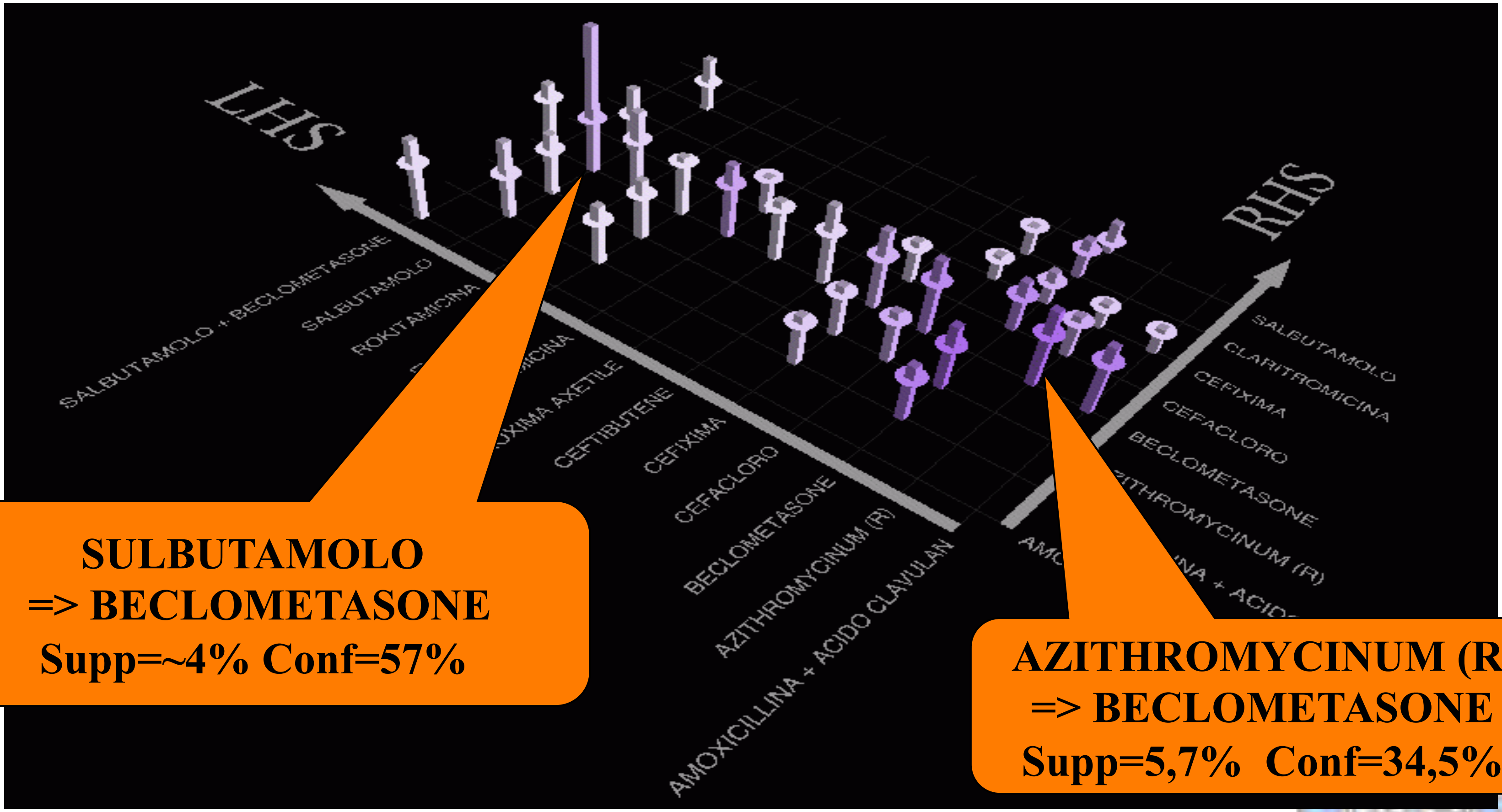
High ⇒ few rules, but all "almost logically true"

Low ⇒ many rules, but many of them very "uncertain"

Typical Values: $\sigma = 2 \div 10$ % $\gamma = 70 \div 90$ %

Association Rules – visualization

(Patients <15 old for USL 19 (a unit of Sanitary service), January-September 1997)



SULBUTAMOLO
=> BECLOMETASONE
Supp=~4% Conf=57%

AZITHROMYCINUM (R)
=> BECLOMETASONE
Supp=5,7% Conf=34,5%

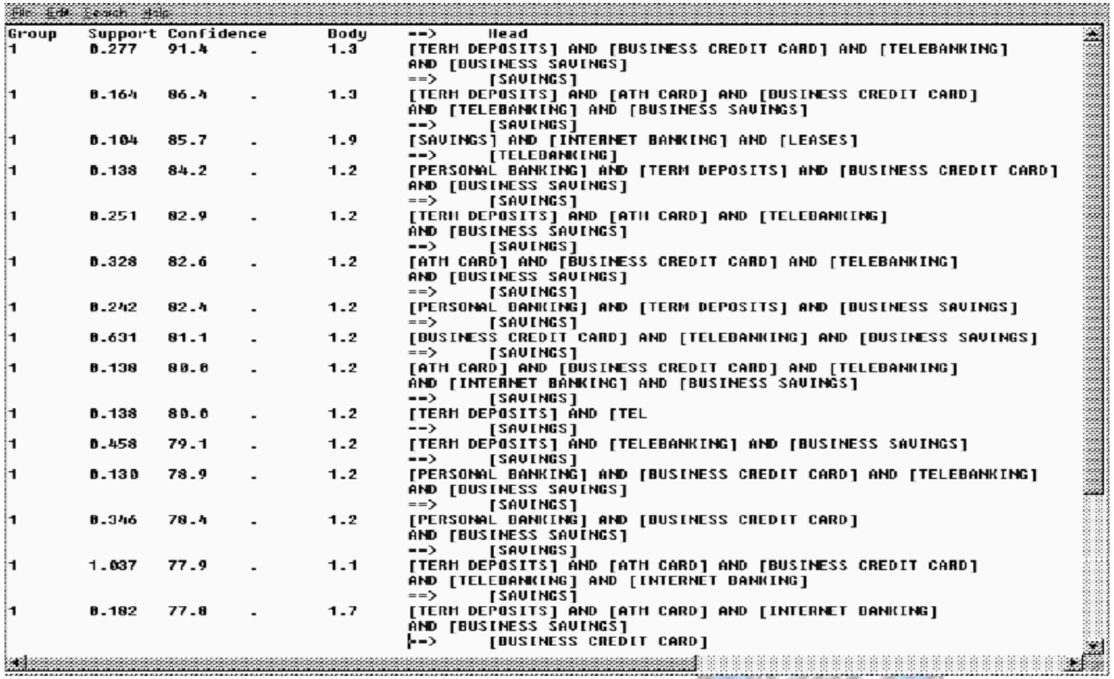
Association Rules – bank transactions

Step 1: Create groups of customers (cluster) on the base of demographical data.

Step 2: Describe customers of each cluster by mining association rules.

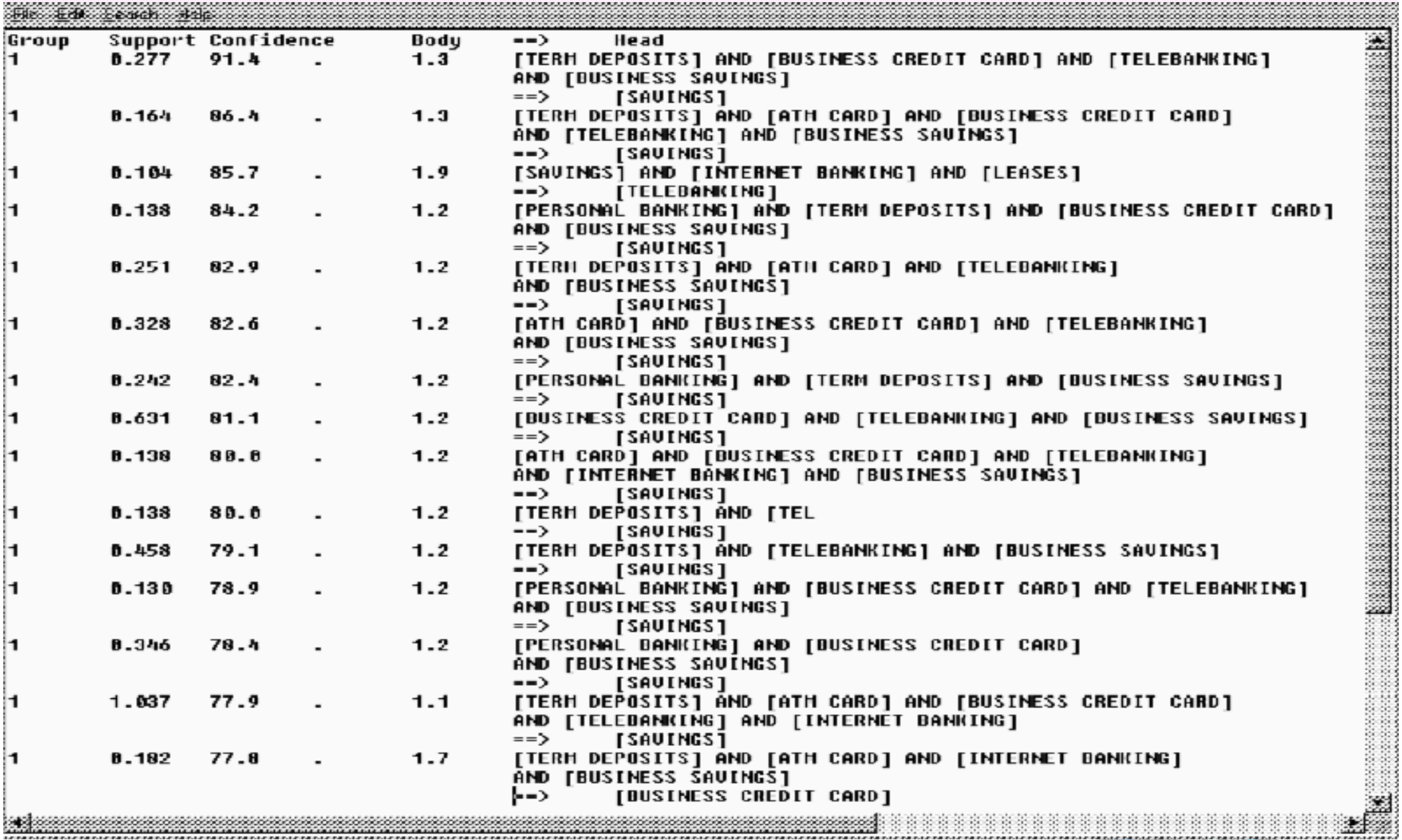
Example:

Rules on cluster 6 (23,7% of dataset):



```
Group    Support Confidence      Body    ==>      Head
1        0.277   91.4    .       1.3     [TERM DEPOSITS] AND [BUSINESS CREDIT CARD] AND [TELEBANKING]
                                         AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.164   86.4    .       1.3     [TERM DEPOSITS] AND [ATM CARD] AND [BUSINESS CREDIT CARD]
                                         AND [TELEBANKING] AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.104   85.7    .       1.9     [SAVINGS] AND [INTERNET BANKING] AND [LEASES]
                                         ==>      [TELEBANKING]
1        0.138   84.2    .       1.2     [PERSONAL BANKING] AND [TERM DEPOSITS] AND [BUSINESS CREDIT CARD]
                                         AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.251   82.9    .       1.2     [TERM DEPOSITS] AND [ATM CARD] AND [TELEBANKING]
                                         AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.328   82.6    .       1.2     [ATM CARD] AND [BUSINESS CREDIT CARD] AND [TELEBANKING]
                                         AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.242   82.4    .       1.2     [PERSONAL BANKING] AND [TERM DEPOSITS] AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.631   81.1    .       1.2     [BUSINESS CREDIT CARD] AND [TELEBANKING] AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.138   80.0    .       1.2     [ATM CARD] AND [BUSINESS CREDIT CARD] AND [TELEBANKING]
                                         AND [INTERNET BANKING] AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.138   80.0    .       1.2     [TERM DEPOSITS] AND [TEL
                                         ==>      [SAVINGS]
1        0.458   79.1    .       1.2     [TERM DEPOSITS] AND [TELEBANKING] AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.130   78.9    .       1.2     [PERSONAL BANKING] AND [BUSINESS CREDIT CARD] AND [TELEBANKING]
                                         AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        0.346   78.4    .       1.2     [PERSONAL BANKING] AND [BUSINESS CREDIT CARD]
                                         AND [BUSINESS SAVINGS]
                                         ==>      [SAVINGS]
1        1.037   77.9    .       1.1     [TERM DEPOSITS] AND [ATM CARD] AND [BUSINESS CREDIT CARD]
                                         AND [TELEBANKING] AND [INTERNET BANKING]
                                         ==>      [SAVINGS]
1        0.182   77.8    .       1.7     [TERM DEPOSITS] AND [ATM CARD] AND [INTERNET BANKING]
                                         AND [BUSINESS SAVINGS]
                                         ==>      [BUSINESS CREDIT CARD]
```

Cluster 6 (23.7% of customers)



```
Group     Support Confidence       Body      -->       Head
1         0.277   91.4      .      1.3       [TERM DEPOSITS] AND [BUSINESS CREDIT CARD] AND [TELEBANKING]
                                             AND [BUSINESS SAVINGS]
                                             ==>       [SAVINGS]
1         0.164   86.4      .      1.3       [TERM DEPOSITS] AND [ATM CARD] AND [BUSINESS CREDIT CARD]
                                             AND [TELEBANKING] AND [BUSINESS SAVINGS]
                                             -->       [SAVINGS]
1         0.104   85.7      .      1.9       [SAVINGS] AND [INTERNET BANKING] AND [LEASES]
                                             -->       [TELEBANKING]
1         0.138   84.2      .      1.2       [PERSONAL BANKING] AND [TERM DEPOSITS] AND [BUSINESS CREDIT CARD]
                                             AND [BUSINESS SAVINGS]
                                             ==>       [SAVINGS]
1         0.251   82.9      .      1.2       [TERM DEPOSITS] AND [ATM CARD] AND [TELEBANKING]
                                             AND [BUSINESS SAVINGS]
                                             -->       [SAVINGS]
1         0.328   82.6      .      1.2       [ATM CARD] AND [BUSINESS CREDIT CARD] AND [TELEBANKING]
                                             AND [BUSINESS SAVINGS]
                                             ==>       [SAVINGS]
1         0.242   82.4      .      1.2       [PERSONAL BANKING] AND [TERM DEPOSITS] AND [BUSINESS SAVINGS]
                                             ==>       [SAVINGS]
1         0.631   81.1      .      1.2       [BUSINESS CREDIT CARD] AND [TELEBANKING] AND [BUSINESS SAVINGS]
                                             ==>       [SAVINGS]
1         0.138   80.0      .      1.2       [ATM CARD] AND [BUSINESS CREDIT CARD] AND [TELEBANKING]
                                             AND [INTERNET BANKING] AND [BUSINESS SAVINGS]
                                             -->       [SAVINGS]
1         0.138   80.0      .      1.2       [TERM DEPOSITS] AND [TEL
                                             -->       [SAVINGS]
1         0.458   79.1      .      1.2       [TERM DEPOSITS] AND [TELEBANKING] AND [BUSINESS SAVINGS]
                                             -->       [SAVINGS]
1         0.130   78.9      .      1.2       [PERSONAL BANKING] AND [BUSINESS CREDIT CARD] AND [TELEBANKING]
                                             AND [BUSINESS SAVINGS]
                                             ==>       [SAVINGS]
1         0.346   78.4      .      1.2       [PERSONAL BANKING] AND [BUSINESS CREDIT CARD]
                                             AND [BUSINESS SAVINGS]
                                             -->       [SAVINGS]
1         1.037   77.9      .      1.1       [TERM DEPOSITS] AND [ATM CARD] AND [BUSINESS CREDIT CARD]
                                             AND [TELEBANKING] AND [INTERNET BANKING]
                                             ==>       [SAVINGS]
1         0.182   77.8      .      1.7       [TERM DEPOSITS] AND [ATM CARD] AND [INTERNET BANKING]
                                             AND [BUSINESS SAVINGS]
                                             -->       [BUSINESS CREDIT CARD]
```

Association rules - module outline

- **What are association rules (AR) and what are they used for:**
  - The paradigmatic application: Market Basket Analysis
  - The single dimensional AR (intra-attribute)
- **How to compute AR**
  - Basic Apriori Algorithm and its optimizations
  - Multi-Dimension AR (inter-attribute)
  - Quantitative AR
  - Constrained AR
- **How to reason on AR and how to evaluate their quality**
  - Multiple-level AR
  - Interestingness
  - Correlation vs. Association

Basic Apriori Algorithm

Problem Decomposition

① Find the *frequent itemsets*: the sets of items that satisfy the support constraint

- A subset of a frequent itemset is also a frequent itemset, i.e., if {*A*,*B*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset
- Iteratively find frequent itemsets with cardinality from 1 to *k* (*k*-itemset)

② Use the frequent itemsets to generate association rules.

Problem Decomposition

Transaction ID	Purchased Items
1	{1, 2, 3}
2	{1, 4}
3	{1, 3}
4	{2, 5, 6}

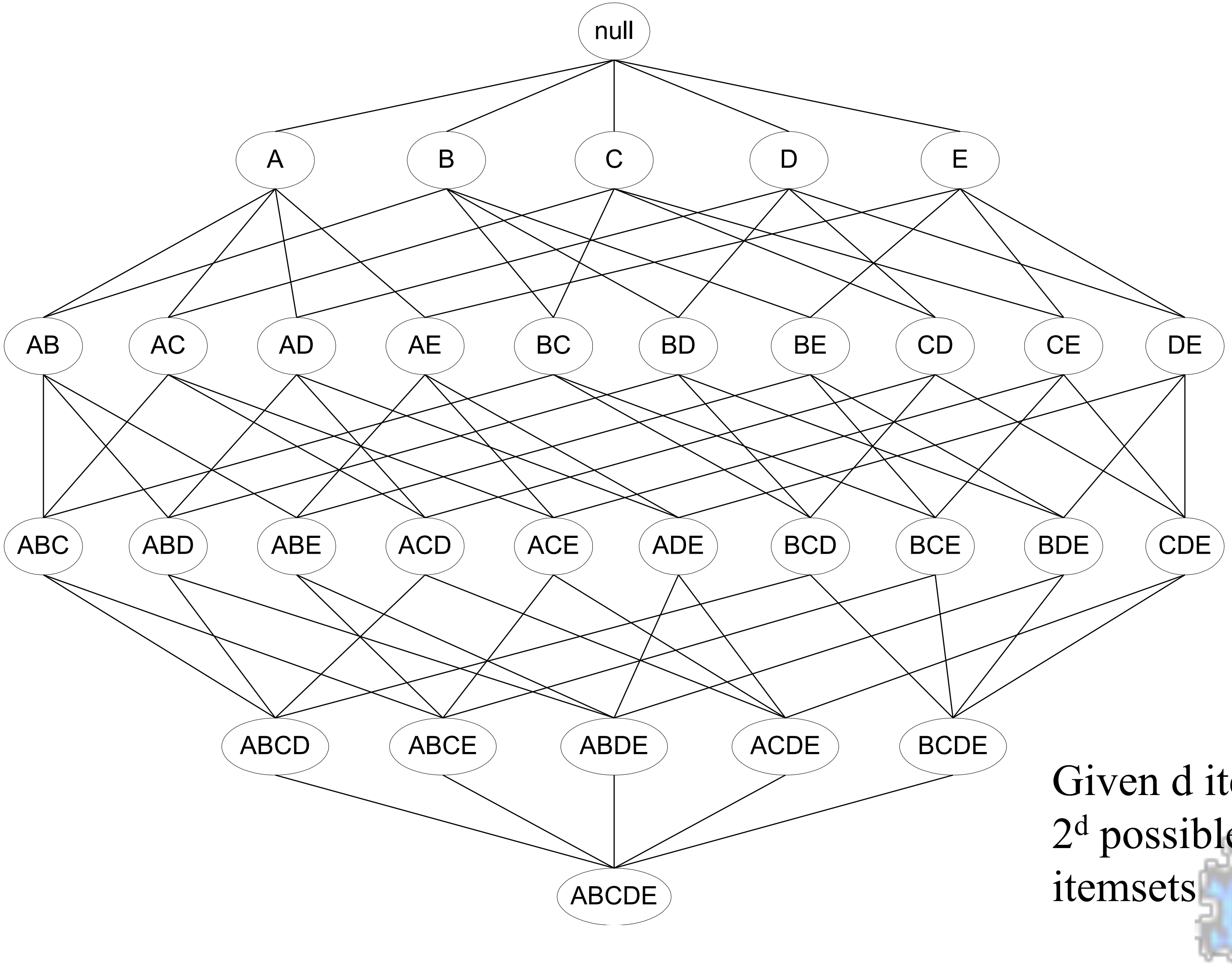
- **For minimum support = 50% = 2 transactions and minimum confidence = 50%**

Frequent Itemsets	Support
{1}	75%
{2}	50%
{3}	50%
{1,3}	50%

For the rule 1 ⇒ 3:

- Support = Support({1, 3}) = 50%
- Confidence = Support({1,3})/Support({1}) = 66%

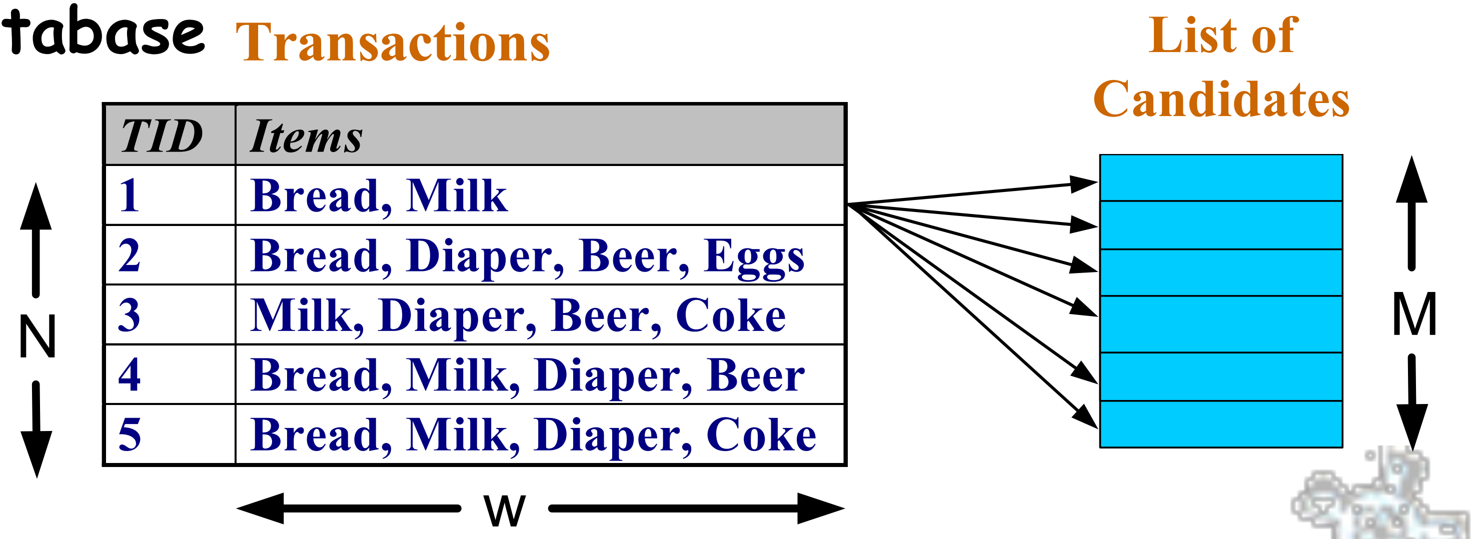
Frequent Itemset Generation



Given d items, there are 2^d possible candidate itemsets

Frequent Itemset Generation

- Brute-force approach:
  - Each itemset in the lattice is a candidate frequent itemset
  - Count the support of each candidate by scanning the database



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

  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => Expensive since $M = 2^d$!!!

Frequent Itemset Generation Strategies

- Reduce the number of candidates (M)
  - Complete search: $M=2^d$
  - Use pruning techniques to reduce M
- Reduce the number of transactions (N)
  - Reduce size of N as the size of itemset increases
  - Used by DHP and vertical-based mining algorithms
- Reduce the number of comparisons (NM)
  - Use efficient data structures to store the candidates or transactions
  - No need to match every candidate against every transaction

The Apriori property

- **If B is frequent and A ⊆ B then A is also frequent**
  - Each transaction which contains B contains also A, which implies supp.(A) ≥ supp.(B))

- **Consequence:** if A is not frequent, then it is not necessary to generate the itemsets which include A.

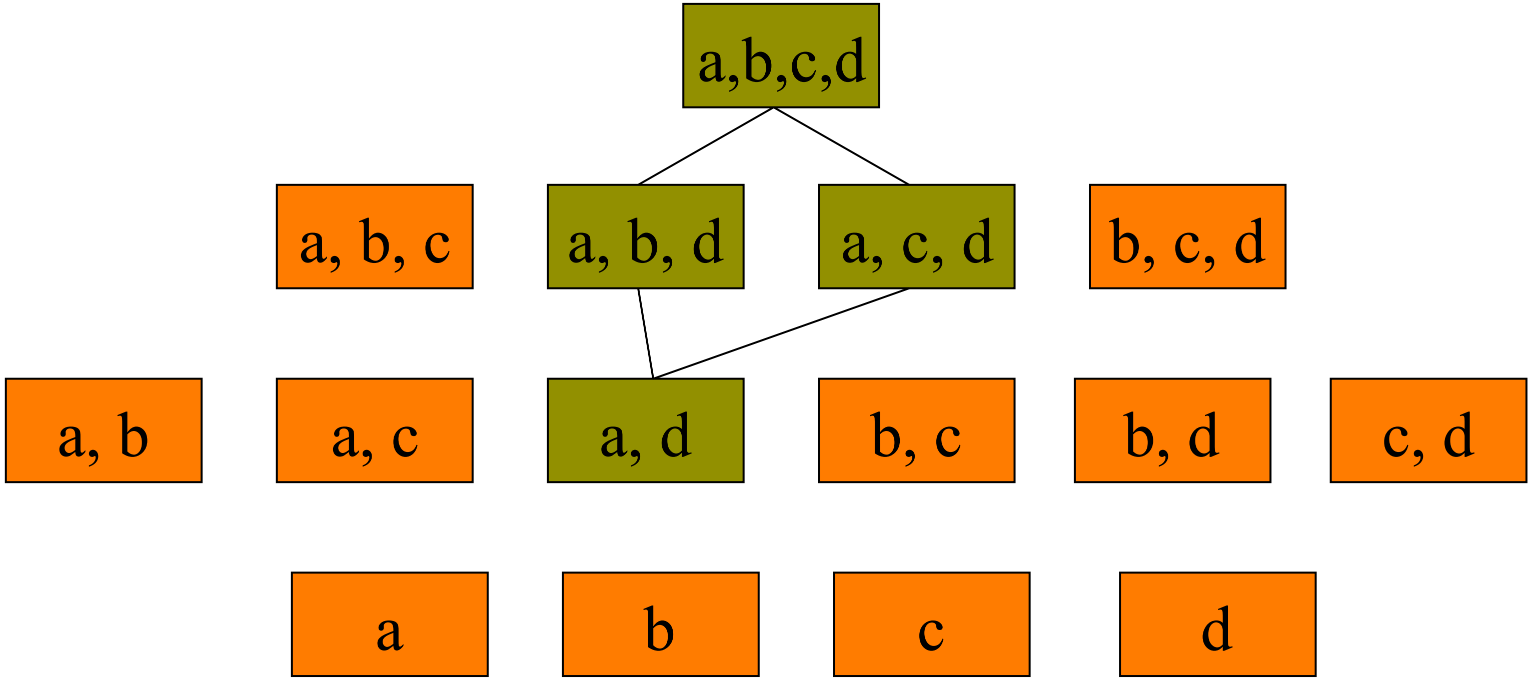
- **Example:**
  - <1, {a, b}>   <2, {a} >
  - <3, {a, b, c}>   <4, {a, b, d}>

  with minimum support = 30%.

  The itemset {c} is not frequent so is not necessary to check for:

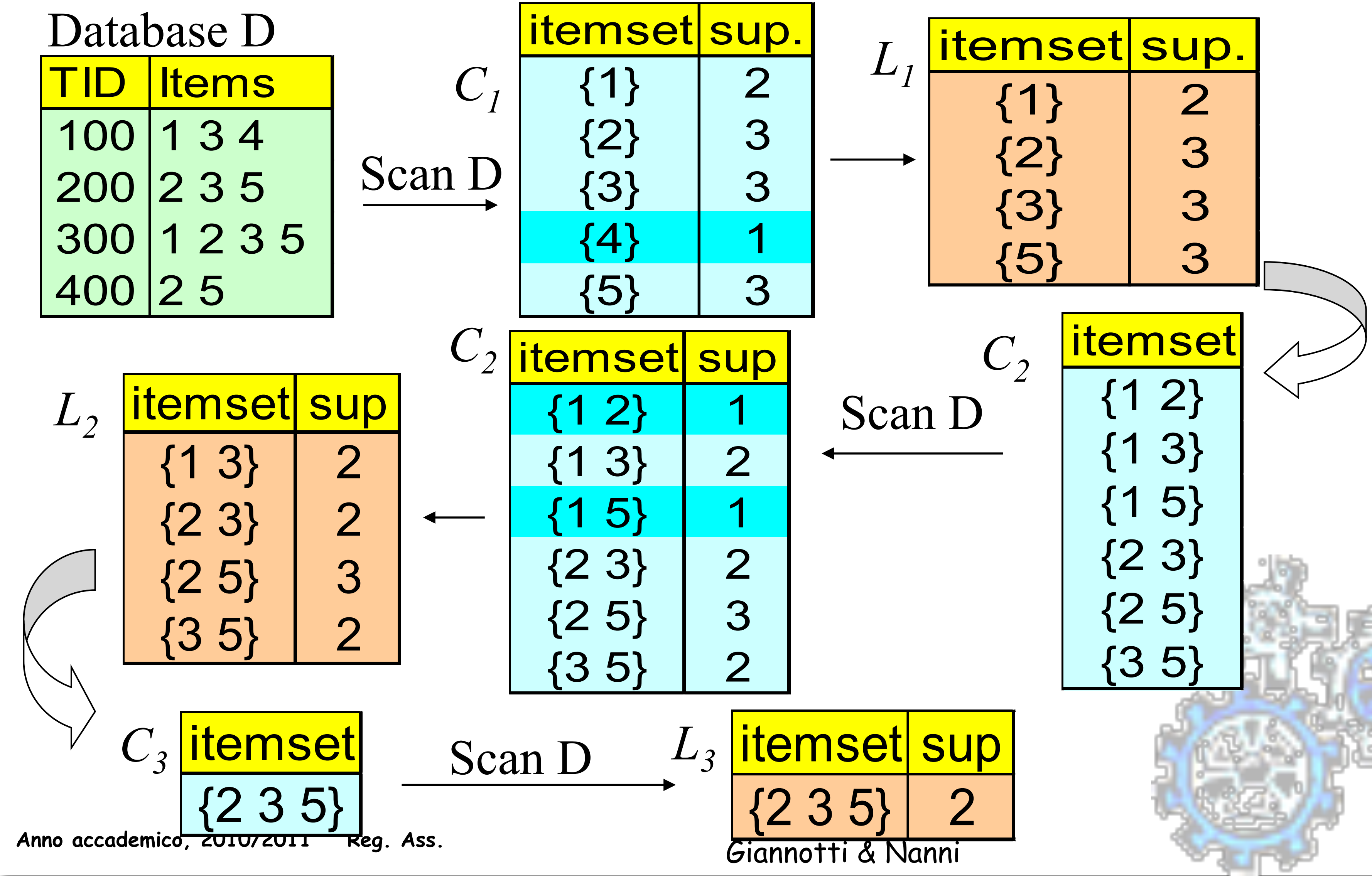
  {c, a}, {c, b}, {c, d}, {c, a, b}, {c, a, d}, {c, b, d}

Apriori - Example



{a,d} is not frequent, so the 3-itemsets {a,b,d}, {a,c,d} and the 4-itemset {a,b,c,d}, are not generated.

The Apriori Algorithm — Example



Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

The Apriori Algorithm

- **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**
- **Pseudo-code:**

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

$L_1$ = {frequent items};
for ($k$ = 1; $L_k$ !=$\emptyset$; $k$++) 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 $\cup_k L_k$;
```

How to Generate Candidates?

- **Suppose the items in L_{k-1} are listed in an order**
- **Step 1: self-joining L_{k-1}**

```
insert into $C_k$
select $p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}$
from $L_{k-1}\ p, L_{k-1}\ q$
where $p.item_1=q.item_1, \ldots, p.item_{k-2}=q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$
```

- **Step 2: pruning**

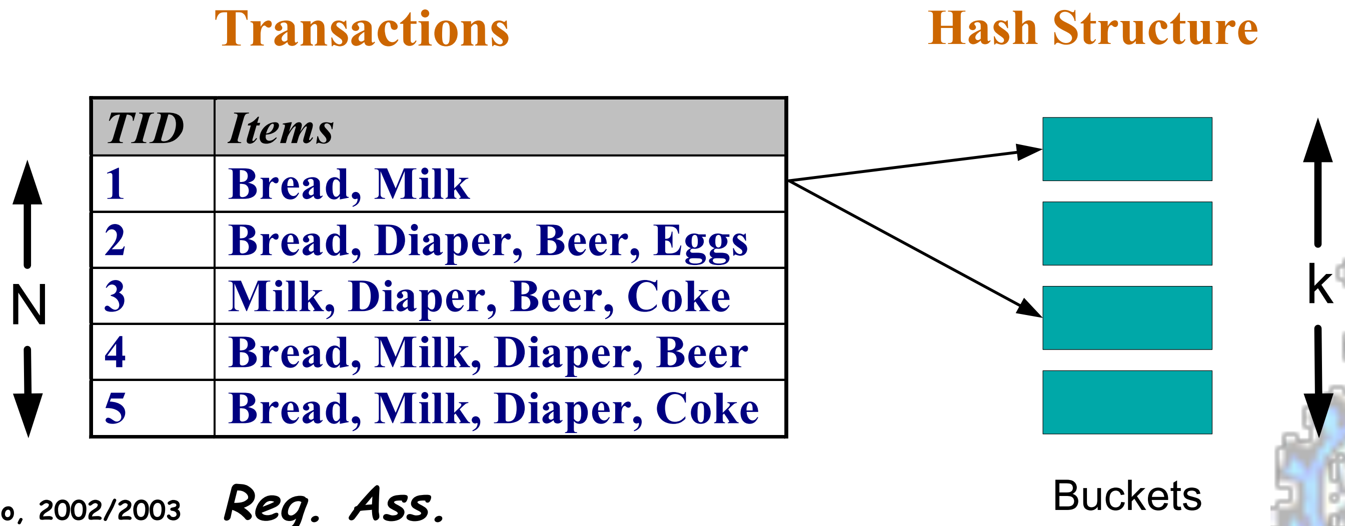
```
forall itemsets $c$ in $C_k$ do
    forall (k-1)-subsets $s$ of $c$ do
        if ($s$ is not in $L_{k-1}$) then delete $c$ from $C_k$
```

Example of Generating Candidates

- L_3={*abc, abd, acd, ace, bcd*}
- Self-joining: L_3*L_3
  - *abcd* from *abc* and *abd*
  - *acde* from *acd* and *ace*
- Pruning:
  - *acde* is removed because *ade* is not in L_3
- C_4={*abcd*}

Reducing Number of Comparisons

- Candidate counting:
  - Scan the database of transactions to determine the support of each candidate itemset
  - To reduce the number of comparisons, store the candidates in a hash structure
    - ✓ Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets



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 Mining Problem (repe.)

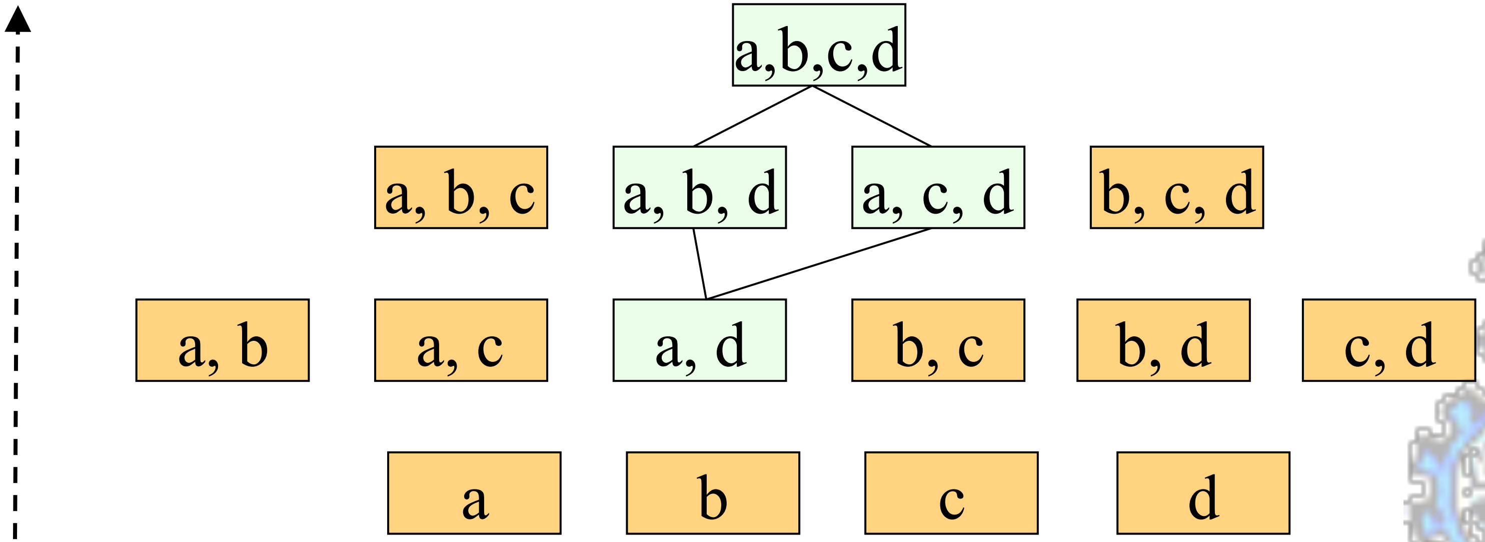
- *$I=\{x_1, \ldots, x_n\}$ set of distinct literals (called items)*
- *$X \subseteq I$, $X \neq \emptyset$, $|X| = k$, X is called k-itemset*
- *A transaction is a couple $\langle tID, X \rangle$ where X is an itemset*
- *A transaction database TDB is a set of transactions*
- *An itemset X is contained in a trans. $\langle tID, Y \rangle$ if $X \subseteq Y$*
- *Given a TDB the subset of transactions of TDB in which X is contained is named TDB[X].*
- *The support of an itemset X , written $supp_{TDB}(X)$ is the cardinality of TDB[X].*
- *Given a user-defined min_sup threshold an itemset X is frequent in TDB if its support is no less than min_sup.*
- *Given a min_sup and a transaction database TDB, the Frequent Itemset Mining Problem requires to compute all frequent itensets in TDB w.r.t min_sup.*

The Apriori Algorithm (rep.)

- *The classical Apriori algorithm [1994] exploits a nice property of frequency in order to prune the exponential search space of the problem:*

"if an itemset is infrequent all its supersets will be infrequent as well"

- *This property is known as "the antimonotonicity of frequency" (aka the "Apriori trick").*
- *This property suggests a breadth-first level-wise computation.*



The Apriori Algorithm

C_k: set of candidate k-itemsets
L_k: set of frequent k-itemsets

```
scan TDB and generate $L_1$;
for ($k = 1$; $L_k \,!= \emptyset$; $k$++) do begin
    $C_{k+1}$ = Apriori-gen($L_k$);
    for each transaction t in TDB do
        for each itemset X in $C_{k+1}$, X in t do X.count++
    $L_{k+1} = \{X \text{ in } C_{k+1} \mid X.count \geq min\_sup\}$;
end;
return $\cup_k L_k$.
```

Candidate generation function (Apriori-gen) is performed in 2 steps:

1. *Join step: candidate k+1-itemsets are generated by joining two frequent k-itemsets which share the same k-1 prefix;*
2. *Prune step: candidate itemsets generated at the previous point are pruned if they have at least one k-subset infrequent.*

Methods to Improve Apriori's Efficiency

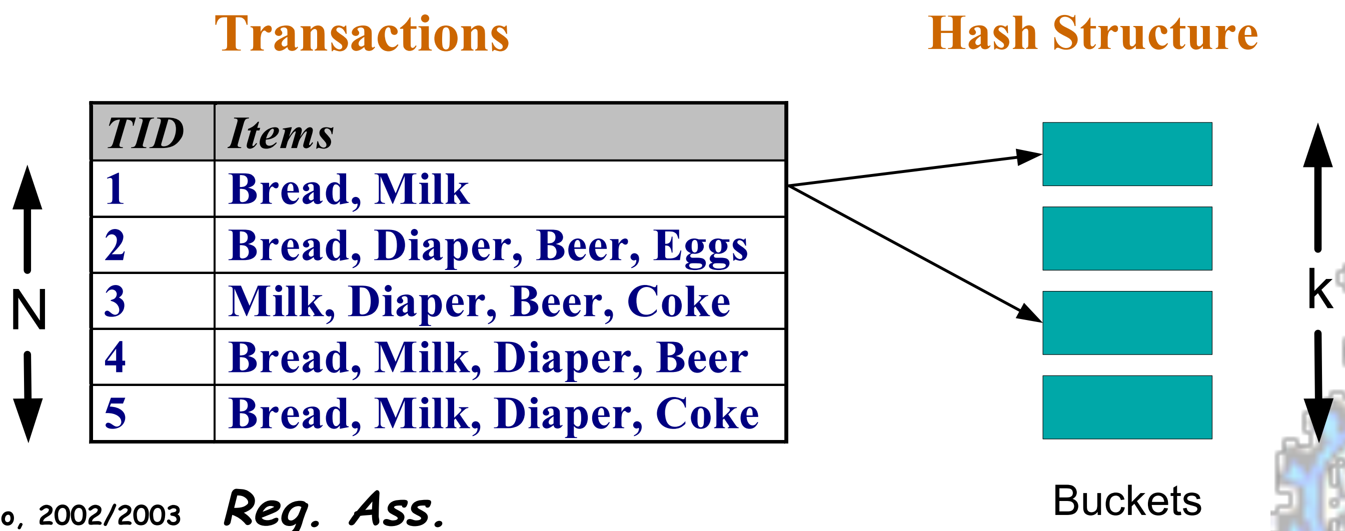
- Hash-based itemset counting: A *k*-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent
- Transaction reduction: A transaction that does not contain any frequent k-itemset is useless in subsequent scans
- Partitioning: Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
- Sampling: mining on a subset of given data, lower support threshold + a method to determine the completeness
- Dynamic itemset counting: add new candidate itemsets only when all of their subsets are estimated to be frequent

How to Count Supports of Candidates?

- Why counting supports of candidates is a problem?
  - The total number of candidates can be very huge
  - One transaction may contain many candidates
- Method:
  - Candidate itemsets are stored in a *hash-tree*
  - *Leaf* node of hash-tree contains a list of itemsets and counts
  - *Interior* node contains a hash table
  - *Subset function*: finds all the candidates contained in a transaction

Reducing Number of Comparisons

- Candidate counting:
  - Scan the database of transactions to determine the support of each candidate itemset
  - To reduce the number of comparisons, store the candidates in a hash structure
    - ✓ Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets



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

Optimizations

- DHP: Direct Hash and Pruning (Park, Chen and Yu, SIGMOD'95).
- Partitioning Algorithm (Savasere, Omiecinski and Navathe, VLDB'95).
- Sampling (Toivonen'96).
- Dynamic Itemset Counting (Brin et. al. SIGMOD'97)

Factors Affecting Complexity

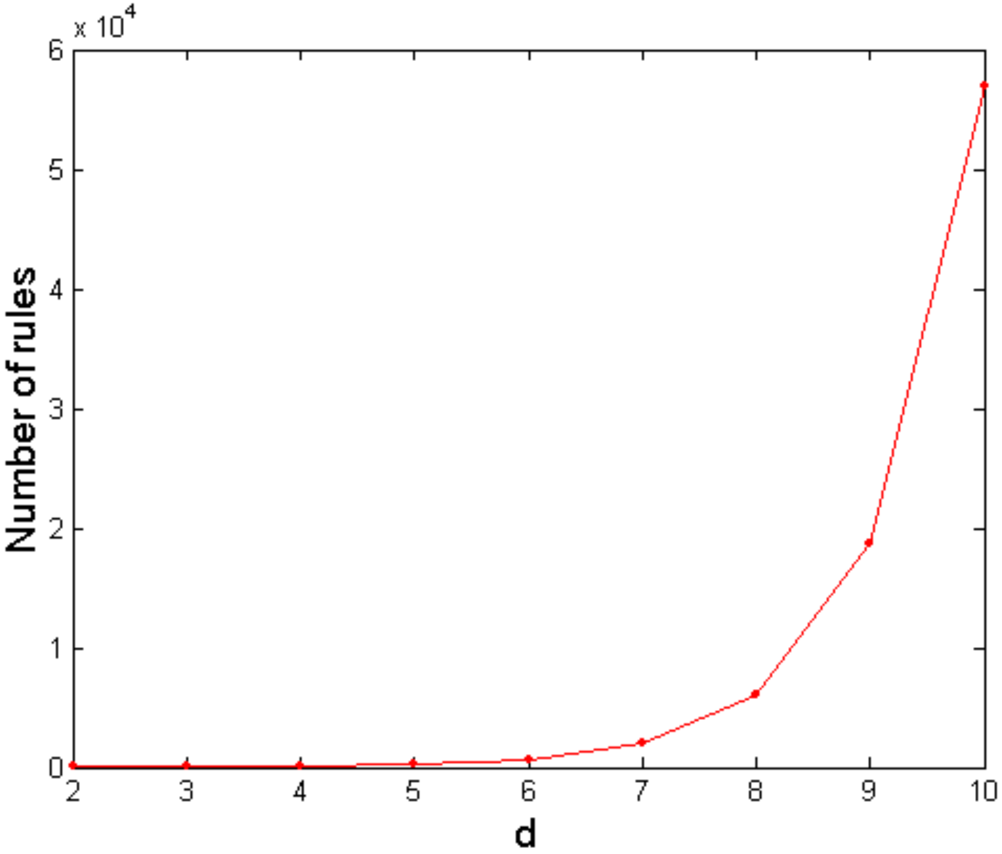
- Choice of minimum support threshold
  - lowering support threshold results in more frequent itemsets
  - this may increase number of candidates and max length of frequent itemsets
- Dimensionality (number of items) of the data set
  - more space is needed to store support count of each item
  - if number of frequent items also increases, both computation and I/O costs may also increase
- Size of database
  - since Apriori makes multiple passes, run time of algorithm may increase with number of transactions
- Average transaction width
  - transaction width increases with denser data sets
  - This may increase max length of frequent itemsets and traversals of hash tree (number of subsets in a transaction increases with its width)

Association rules - module outline

- **What are association rules (AR) and what are they used for:**
  - The paradigmatic application: Market Basket Analysis
  - The single dimensional AR (intra-attribute)
- **How to compute AR**
  - Basic Apriori Algorithm and its optimizations
  - Multi-Dimension AR (inter-attribute)
  - Quantitative AR
  - Constrained AR
- **How to reason on AR and how to evaluate their quality**
  - Multiple-level AR
  - Interestingness
  - Correlation vs. Association

Rule generation Computational Complexity

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



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$
$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

Generating Association Rules from Frequent Itemsets

- Only strong association rules are generated
- Frequent itemsets satisfy minimum support threshold
- Strong rules are those that satisfy minimum confidence threshold

- $confidence(A \implies B) = \Pr(B \mid A) =$

$$\text{support}(A \cup B)/\text{support}(A)$$

Strong Rule generation

```
For each frequent itemset, f, generate all non-empty subsets of f
For every non-empty subset s of f do
    if $\text{support}(f)/\text{support}(s) \geq \text{min\_confidence}$ then
        output rule s ==> (f-s)
end
```

Rule Generation

- If {A,B,C,D} is a frequent itemset, candidate rules:

ABC →D, ABD →C, ACD →B, BCD →A,
A →BCD, B →ACD, C →ABD, D →ABC
AB →CD, AC → BD, AD → BC, BC →AD,
BD →AC, CD →AB,

- If |L| = k, then there are 2^k – 2 candidate association rules (ignoring L → ∅ and ∅ → L)

Rule Generation

- **How to efficiently generate rules from frequent itemsets?**
  - **In general, confidence does not have an anti-monotone property**

    $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$

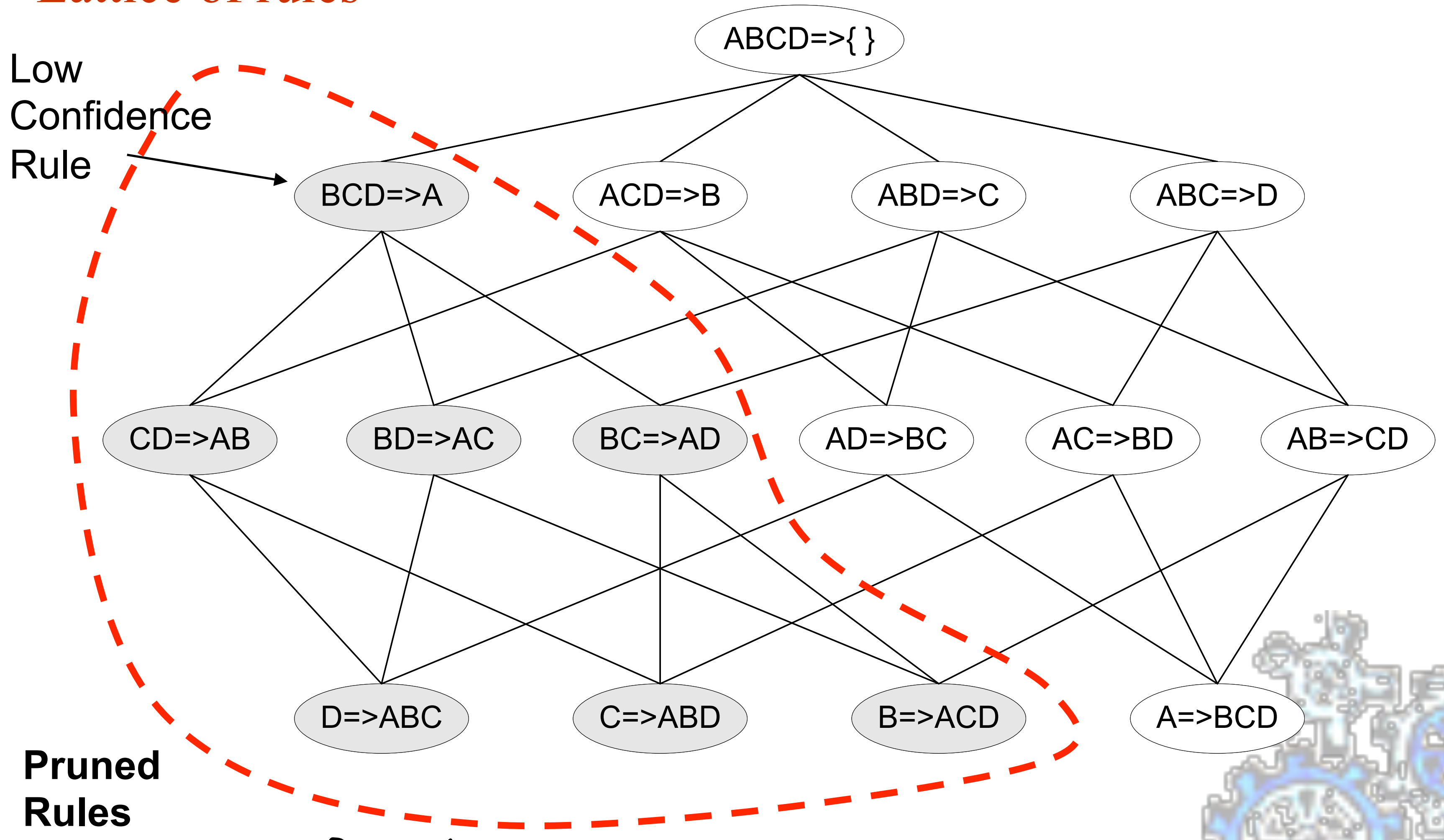
  - **But confidence of rules generated from the same itemset has an anti-monotone property**
  - **e.g., $L = \{A,B,C,D\}$:**

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

✓ Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

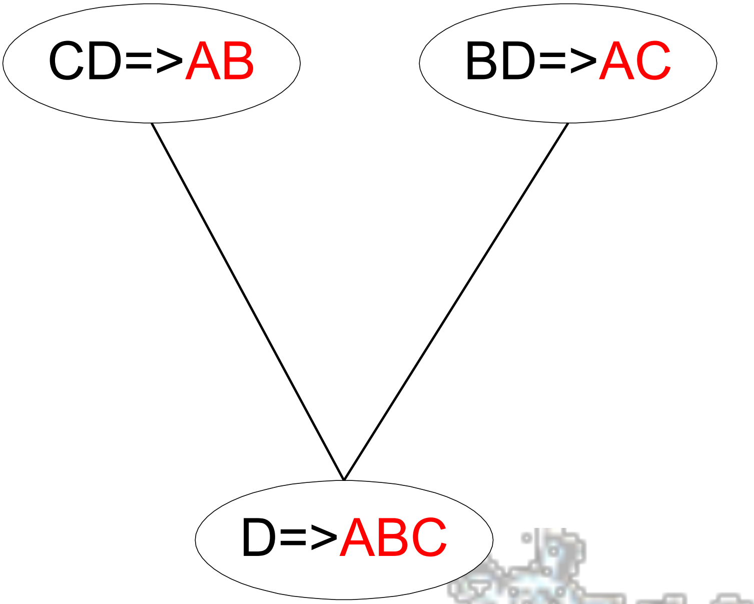
Rule Generation for Apriori Algorithm

Lattice of rules



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 its subset AD=>BC does not have high confidence



Association rules - module outline

- **What are association rules (AR) and what are they used for:**
  - The paradigmatic application: Market Basket Analysis
  - The single dimensional AR (intra-attribute)
- **How to compute AR**
  - Basic Apriori Algorithm and its optimizations
  - **Multi-Dimension AR (inter-attribute)**
  - **Quantitative AR**
  - Constrained AR
- **How to reason on AR and how to evaluate their quality**
  - Multiple-level AR
  - Interestingness
  - Correlation vs. Association

Single-dimensional vs multi-dimensional AR

Single-dimensional (Intra-attribute)

The events are: *items A, B and C belong to the same transaction*

Occurrence of events: *transactions*

Multi-dimensional (Inter-attribute)

The events are : *attribute A assumes value* a, *attribute B assumes value* b *and attribute C assumes value* c.

Occurrence of events: *tuples*

Multidimensional AR

Associations between values of different attributes :

CID	**nationality**	**age**	**income**
1	Italian	50	low
2	French	40	high
3	French	30	high
4	Italian	50	medium
5	Italian	45	high
6	French	35	high

RULES:

nationality = French ⇒ **income** = high [50%, 100%]

income = high ⇒ **nationality** = French [50%, 75%]

age = 50 ⇒ **nationality** = Italian [33%, 100%]

Single-dimensional vs Multi-dimensional AR

Multi-dimensional

<1, Italian, 50, low>

<2, French, 45, high>

Single-dimensional

<1, {nat/Ita, age/50, inc/low}>

<2, {nat/Fre, age/45, inc/high}>

Schema: <ID, a?, b?, c?, d?>

<1, yes, yes, no, no>

<2, yes, no, yes, no>

<1, {a, b}>

<2, {a, c}>

Quantitative Attributes

- Quantitative attributes (e.g. age, income)
- Categorical attributes (e.g. color of car)

CID	**height**	**weight**	**income**
1	168	75,4	30,5
2	175	80,0	20,3
3	174	70,3	25,8
4	170	65,2	27,0

Problem: too many distinct values

Solution: transform quantitative attributes in categorical ones via discretization.

Quantitative Association Rules

CID	Age	Married	NumCars
1	23	No	1
2	25	Yes	1
3	29	No	0
4	34	Yes	2
5	38	Yes	2

[Age: 30..39] and [Married: Yes] ⇒ [NumCars:2]

support = 40%
confidence = 100%

Discretization of quantitative attributes

Solution: each value is replaced by the interval to which it belongs.

height: 0-150cm, 151-170cm, 171-180cm, >180cm

weight: 0-40kg, 41-60kg, 60-80kg, >80kg

income: 0-10ML, 11-20ML, 20-25ML, 25-30ML, >30ML

CID	**height**	**weight**	**income**
1	151-171	60-80	>30
2	171-180	60-80	20-25
3	171-180	60-80	25-30
4	151-170	60-80	25-30

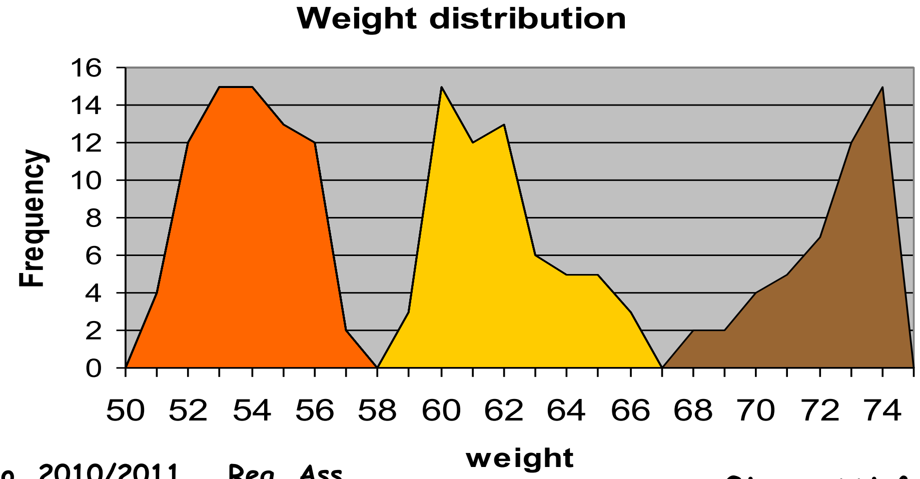
Problem: the discretization may be useless (see **weight**).

How to choose intervals?

1. Interval with a fixed "reasonable" granularity
   Ex. intervals of 10 cm for height.

2. Interval size is defined by some domain dependent criterion
   Ex.: 0-20ML, 21-22ML, 23-24ML, 25-26ML, >26ML

3. Interval size determined by analyzing data, studying the distribution or using clustering



50 - 58 kg

59-67 kg

> 68 kg

Discretization of quantitative attributes

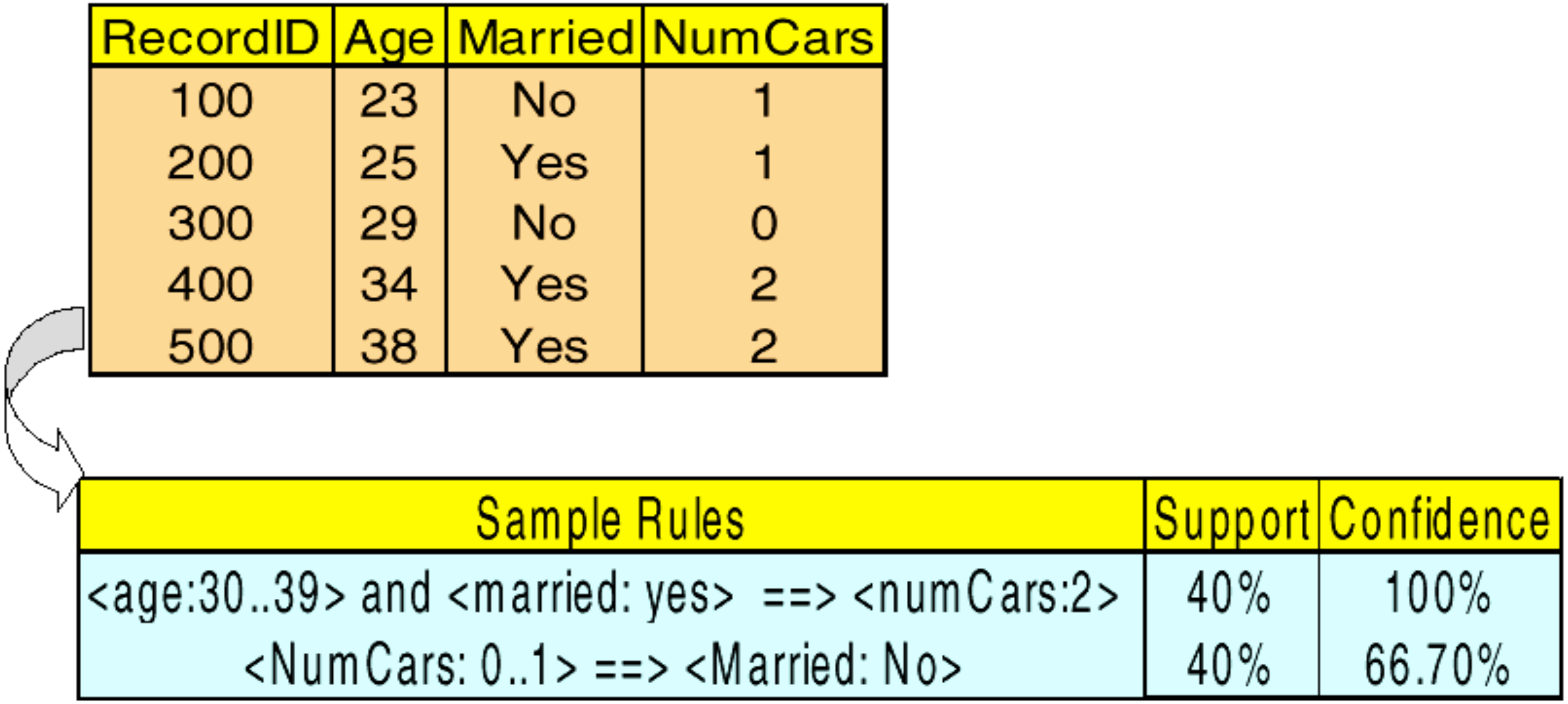
1. Quantitative attributes are statically discretized by using predefined concept hierarchies:
   - elementary use of background knowledge

Loose interaction between Apriori and discretizer

2. Quantitative attributes are dynamically discretized
   - into "bins" based on the distribution of the data.
   - considering the distance between data points.

Tighter interaction between Apriori and discretizer

Quantitative Association Rules



RecordID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	Yes	2

Sample Rules	Support	Confidence
<age:30..39> and <married: yes> ==> <numCars:2>	40%	100%
<NumCars: 0..1> ==> <Married: No>	40%	66.70%

Handling quantitative rules may require mapping of the continuous variables into Boolean

Mapping Quantitative to Boolean

- One possible solution is to map the problem to the Boolean association rules:
  - discretize a non-categorical attribute to intervals, e.g., Age [20,29], [30,39],...
  - categorical attributes: each value becomes one item
  - non-categorical attributes: each interval becomes one item
- Problems with the mapping
  - too few intervals: lost information
  - too low support: too many rules

RecordID	Age	Married	NoCars
100	23	No	1
500	38	Yes	2

RecID	Age: 20..29	Age: 30..39	Married: Yes	Married: No	Cars: 0	Cars: 1	Cars: 2
100	1	0	0	1	0	1	0
500	0	1	1	0	0	0	1

Association rules - module outline

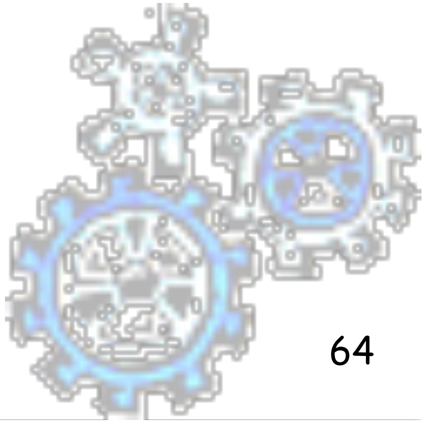
- **What are association rules (AR) and what are they used for:**
  - The paradigmatic application: Market Basket Analysis
  - The single dimensional AR (intra-attribute)
- **How to compute AR**
  - Basic Apriori Algorithm and its optimizations
  - Multi-Dimension AR (inter-attribute)
  - Quantitative AR
  - **Constrained AR**
- **How to reason on AR and how to evaluate their quality**
  - Multiple-level AR
  - Interestingness
  - Correlation vs. Association

Constraints and AR

- Preprocessing: use constraints to focus on a subset of transactions
  - Example: find association rules where the prices of all items are at most 200 Euro
- Optimizations: use constraints to optimize Apriori algorithm
  - Anti-monotonicity: when a set violates the constraint, so does any of its supersets.
  - Apriori algorithm uses this property for pruning
- Push constraints as deep as possible inside the frequent set computation

Constraint-based AR

- **What kinds of constraints can be used in mining?**
  - **Data constraints:**
    - ✓ SQL-like queries
      - Find product pairs sold together in Vancouver in Dec.'98.
    - ✓ OLAP-like queries (Dimension/level)
      - in relevance to region, price, brand, customer category.
  - **Rule constraints:**
    - ✓ specify the form or property of rules to be mined.
    - ✓ Constraint-based AR



Rule Constraints

- Two kind of constraints:
  - Rule form constraints: meta-rule guided mining.
    - ✓ P(x, y) ^ Q(x, w) → takes(x, "database systems").
  - Rule content constraint: constraint-based query optimization (Ng, et al., SIGMOD'98).
    - ✓ sum(LHS) < 100 ^ min(LHS) > 20 ^ sum(RHS) > 1000
- 1-variable vs. 2-variable constraints (Lakshmanan, et al. SIGMOD'99):
  - 1-var: A constraint confining only one side (L/R) of the rule, e.g., as shown above.
  - 2-var: A constraint confining both sides (L and R).
    - ✓ sum(LHS) < min(RHS) ^ max(RHS) < 5* sum(LHS)

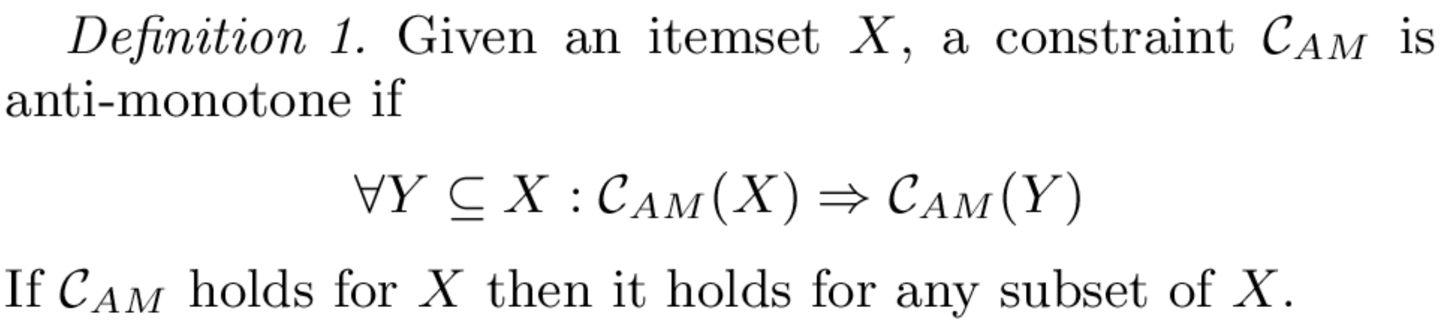
Mining Association Rules with Constraints

- Postprocessing
  - A naïve solution: apply Apriori for finding all frequent sets, and then to test them for constraint satisfaction one by one.
- Optimization
  - Han approach: comprehensive analysis of the properties of constraints and try to push them as deeply as possible inside the frequent set computation.

Apriori property revisited

- Anti-monotonicity: *If a set S violates the constraint, any superset of S violates the constraint.*
- Examples:
  - $\mathit{sum}(S.\mathit{Price}) \leq v$ is anti-monotone
  - $\mathit{sum}(S.\mathit{Price}) \geq v$ is not anti-monotone
  - $\mathit{sum}(S.\mathit{Price}) = v$ is partly anti-monotone
- Application:
  - Push "$\mathit{sum}(S.\mathit{price}) \leq 1000$" deeply into iterative frequent set computation.

Problem Definition: Antimonotone Constraint



Definition 1. Given an itemset X, a constraint $\mathcal{C}_{AM}$ is anti-monotone if

$$\forall Y \subseteq X : \mathcal{C}_{AM}(X) \Rightarrow \mathcal{C}_{AM}(Y)$$

If $\mathcal{C}_{AM}$ holds for X then it holds for any subset of X.

- *Frequency is an antimonotone constraint.*
- *"Apriori trick": if an itemset X does not satisfy C_{freq}, then no superset of X can satisfy C_{freq}.*
- *Other examples of antimonotone constraint:*

  sum(X.prices) ≤ 20 euro

  |X| ≤ 5

Characterization of Anti-Monotonicity Constraints

constraint	antimonotone
v ∈ S	no
S ⊆ V	no
S ⊆ V	yes
S = V	partly
min(S) ≤ v	no
min(S) ≥ v	yes
min(S) = v	partly
max(S) ≤ v	yes
max(S) ≥ v	no
max(S) = v	partly
count(S) ≤ v	yes
count(S) ≥ v	no
count(S) = v	partly
sum(S) ≤ v	yes
sum(S) ≥ v	no
sum(S) = v	partly
avg(S) θ v, θ ∈ { =, ≤, ≥ }	convertible
(frequent constraint)	(yes)

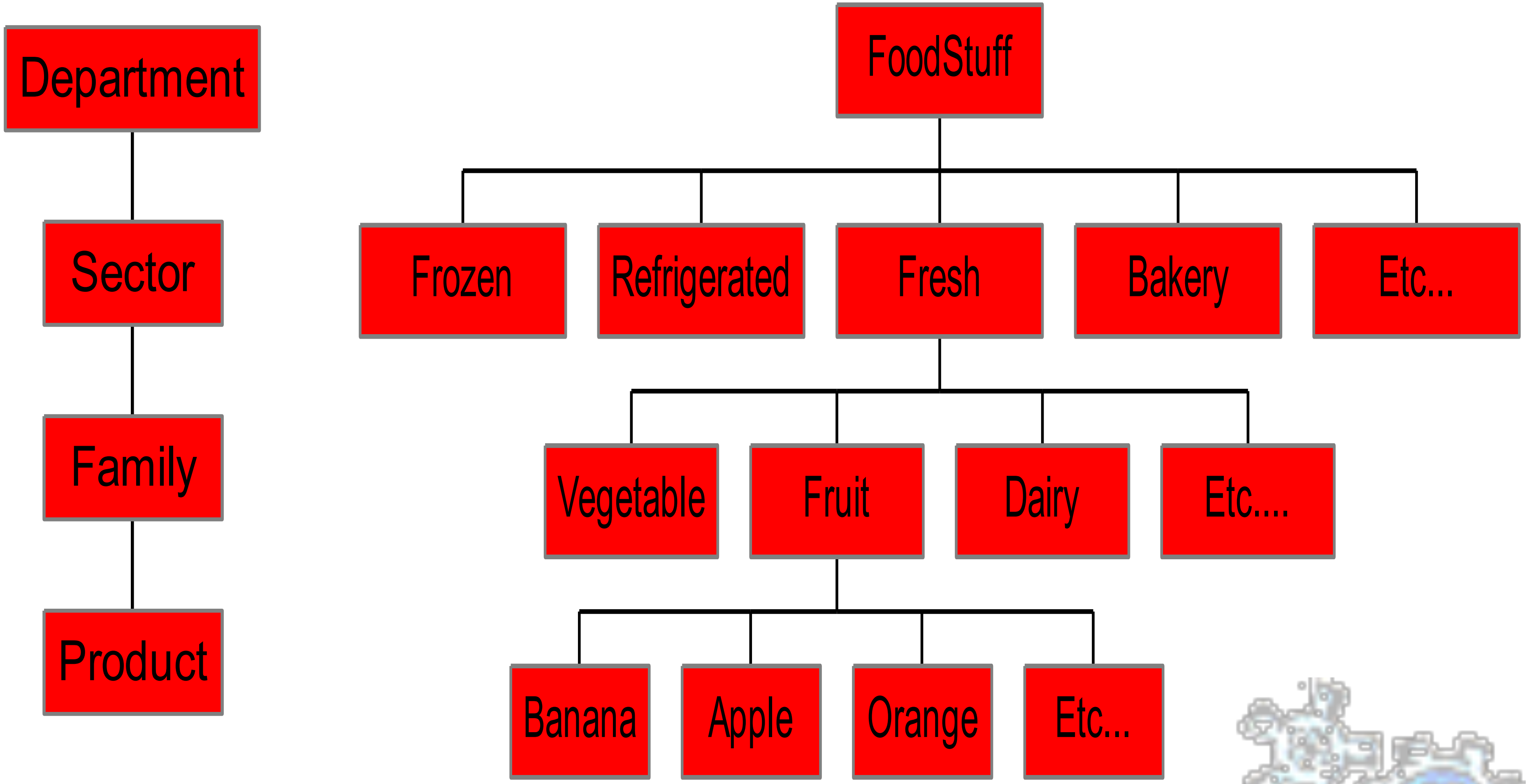
Association rules - module outline

- What are association rules (AR) and what are they used for:
  - The paradigmatic application: Market Basket Analysis
  - The single dimensional AR (intra-attribute)
- How to compute AR
  - Basic Apriori Algorithm and its optimizations
  - Multi-Dimension AR (inter-attribute)
  - Quantitative AR
  - Constrained AR
- How to reason on AR and how to evaluate their quality
  - Multiple-level AR
  - Interestingness
  - Correlation vs. Association

Multilevel AR

- Is difficult to find interesting patterns at a too primitive level
  - high support = too few rules
  - low support = too many rules, most uninteresting
- Approach: reason at suitable level of abstraction
- A common form of background knowledge is that an attribute may be generalized or specialized according to a hierarchy of concepts
- Dimensions and levels can be efficiently encoded in transactions
- Multilevel Association Rules : rules which combine associations with hierarchy of concepts

Hierarchy of concepts



Department

Sector

Family

Product

FoodStuff

Frozen

Refrigerated

Fresh

Bakery

Etc...

Vegetable

Fruit

Dairy

Etc....

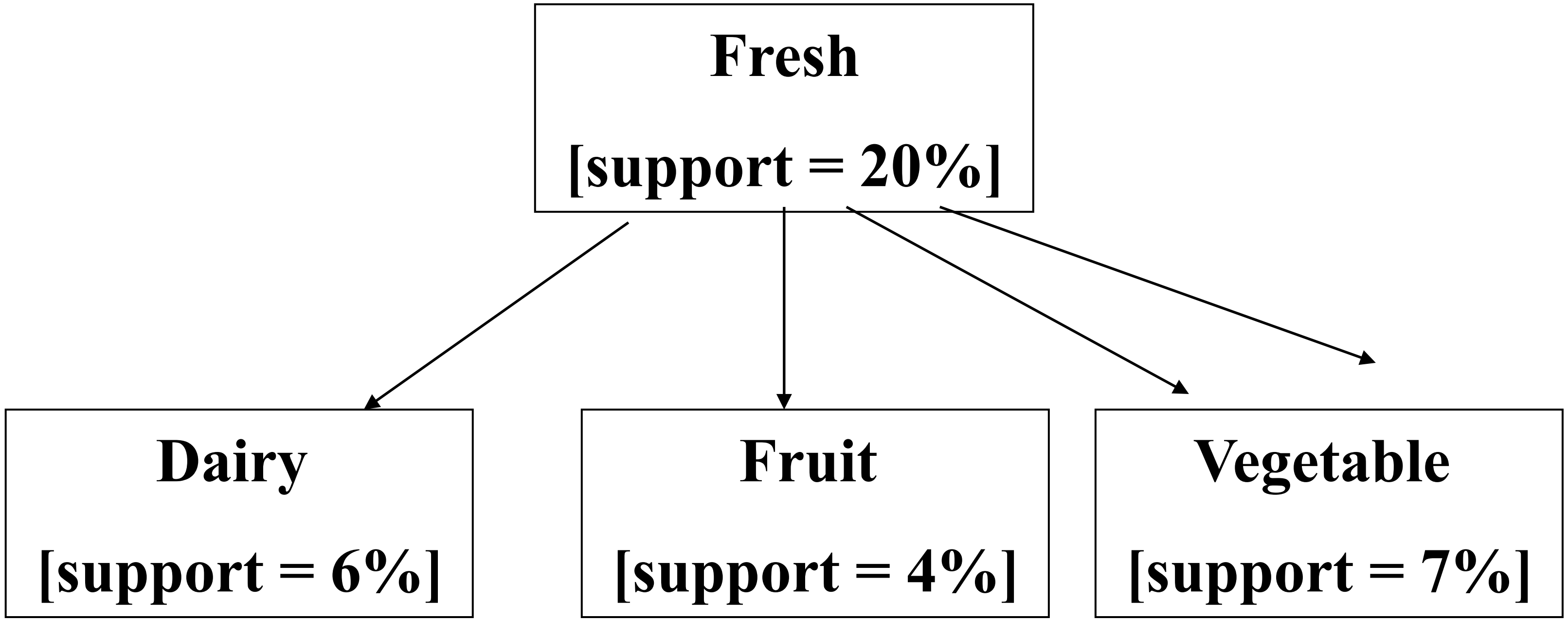
Banana

Apple

Orange

Etc...

Multilevel AR



Fresh ⇒ Bakery [20%, 60%]
Dairy ⇒ Bread [6%, 50%]
Fruit ⇒ Bread [1%, 50%] is not valid

Support and Confidence of Multilevel AR

- from specialized to general: support of rules increases (new rules may become valid)
- from general to specialized: support of rules decreases (rules may become not valid, their support falls under the threshold)
- Confidence is not affected

Reasoning with Multilevel AR

- Too low level => too many rules and too primitive.

  Example: Apple Melinda ⇒ Colgate Tooth-paste

  It is a curiosity not a behavior

- Too high level => uninteresting rules

  Example: Foodstuff ⇒ Varia

- Redundancy => some rules may be redundant due to "ancestor" relationships between items.
  - A rule is redundant if its support is close to the "expected" value, based on the rule's ancestor.
- Example (milk has 4 subclasses)
  - milk ⇒ wheat bread, [support = 8%, confidence = 70%]
  - 2%-milk ⇒ wheat bread, [support = 2%, confidence = 72%]

Mining Multilevel AR

- Calculate frequent itemsets at each concept level, until no more frequent itemsets can be found
- For each level use Apriori
- A top_down, progressive deepening approach:
  - First find high-level strong rules:

    fresh → bakery [20%, 60%].
  - Then find their lower-level "weaker" rules:

    fruit → bread [6%, 50%].
- Variations at mining multiple-level association rules.
  - Level-crossed association rules:

    fruit → *wheat bread*
  - Association rules with multiple, alternative hierarchies:

    fruit → *Wonder bread*

Multi-level Association: Uniform Support vs. Reduced Support

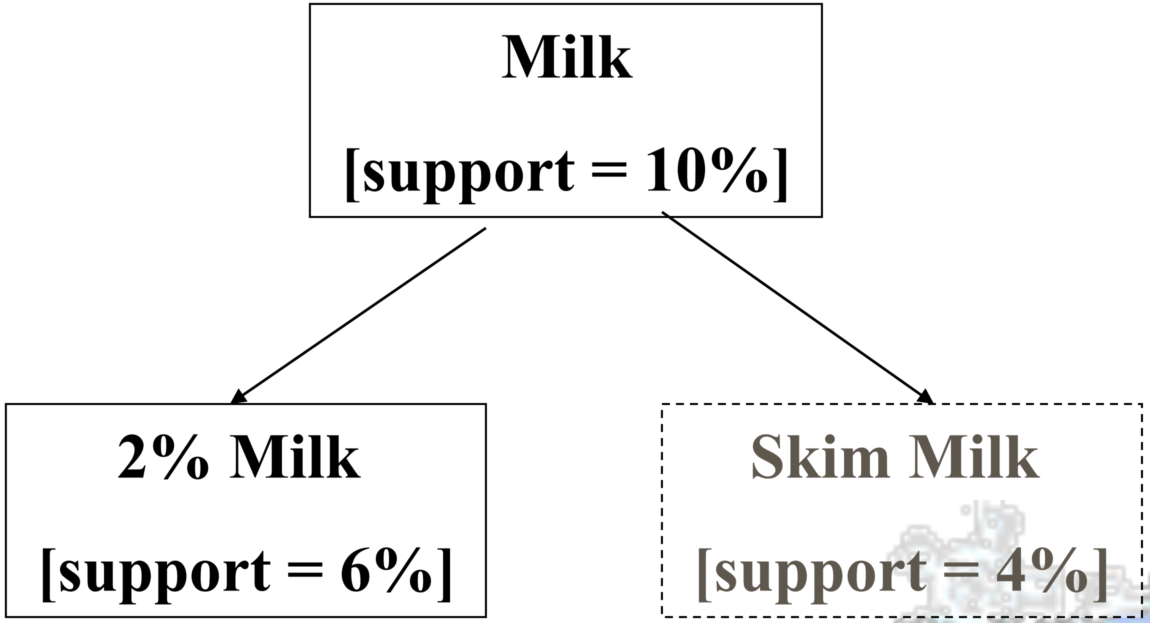
- Uniform Support: the same minimum support for all levels
  - + One minimum support threshold. No need to examine itemsets containing any item whose ancestors do not have minimum support.
  - – If support threshold
    - too high ⇒ miss low level associations.
    - too low ⇒ generate too many high level associations.
- Reduced Support: reduced minimum support at lower levels - different strategies possible

Uniform Support

Multi-level mining with uniform support

Level 1
min_sup = 5%

Level 2
min_sup = 5%



Milk

[support = 10%]

2% Milk

[support = 6%]

Skim Milk

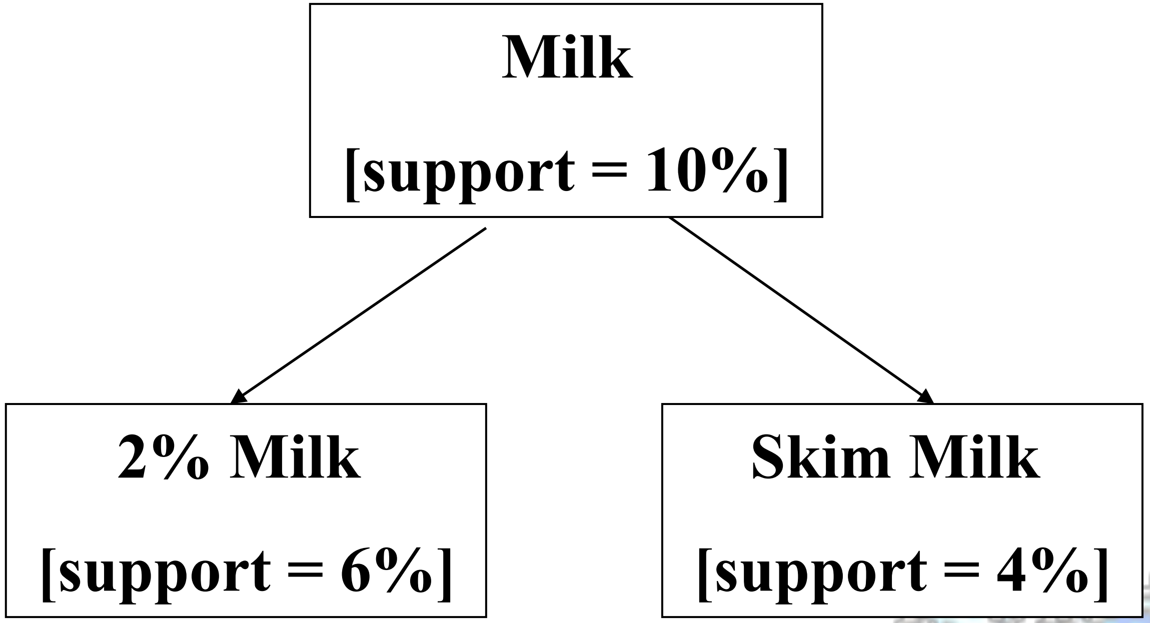
[support = 4%]

Reduced Support

Multi-level mining with reduced support

Level 1
min_sup = 5%

Level 2
min_sup = 3%



Milk

[support = 10%]

2% Milk

[support = 6%]

Skim Milk

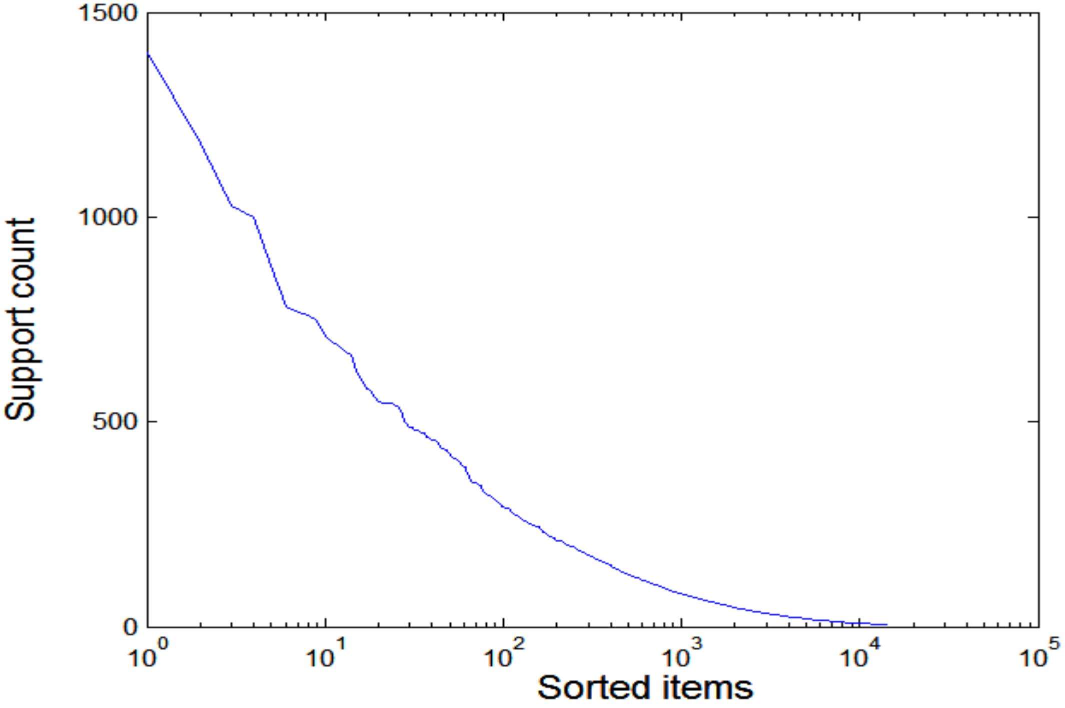
[support = 4%]

Association rules - module outline

- **What are association rules (AR) and what are they used for:**
  - The paradigmatic application: Market Basket Analysis
  - The single dimensional AR (intra-attribute)
- **How to compute AR**
  - Basic Apriori Algorithm and its optimizations
  - Multi-Dimension AR (inter-attribute)
  - Quantitative AR
  - Constrained AR
- **How to reason on AR and how to evaluate their quality**
  - Multiple-level AR
  - Interestingness
  - Correlation vs. Association

Effect of Support Distribution

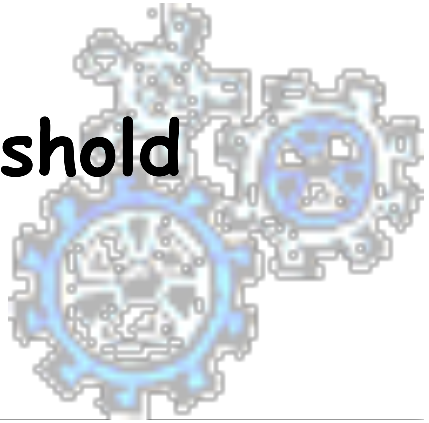
- Many real data sets have skewed support distributior



Support distribution of a retail data set

Effect of Support Distribution

- How to set the appropriate *minsup* threshold?
  - If *minsup* is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)
  - If *minsup* is set too low, it is computationally expensive and the number of itemsets is very large
- Using a single minimum support threshold may not be effective



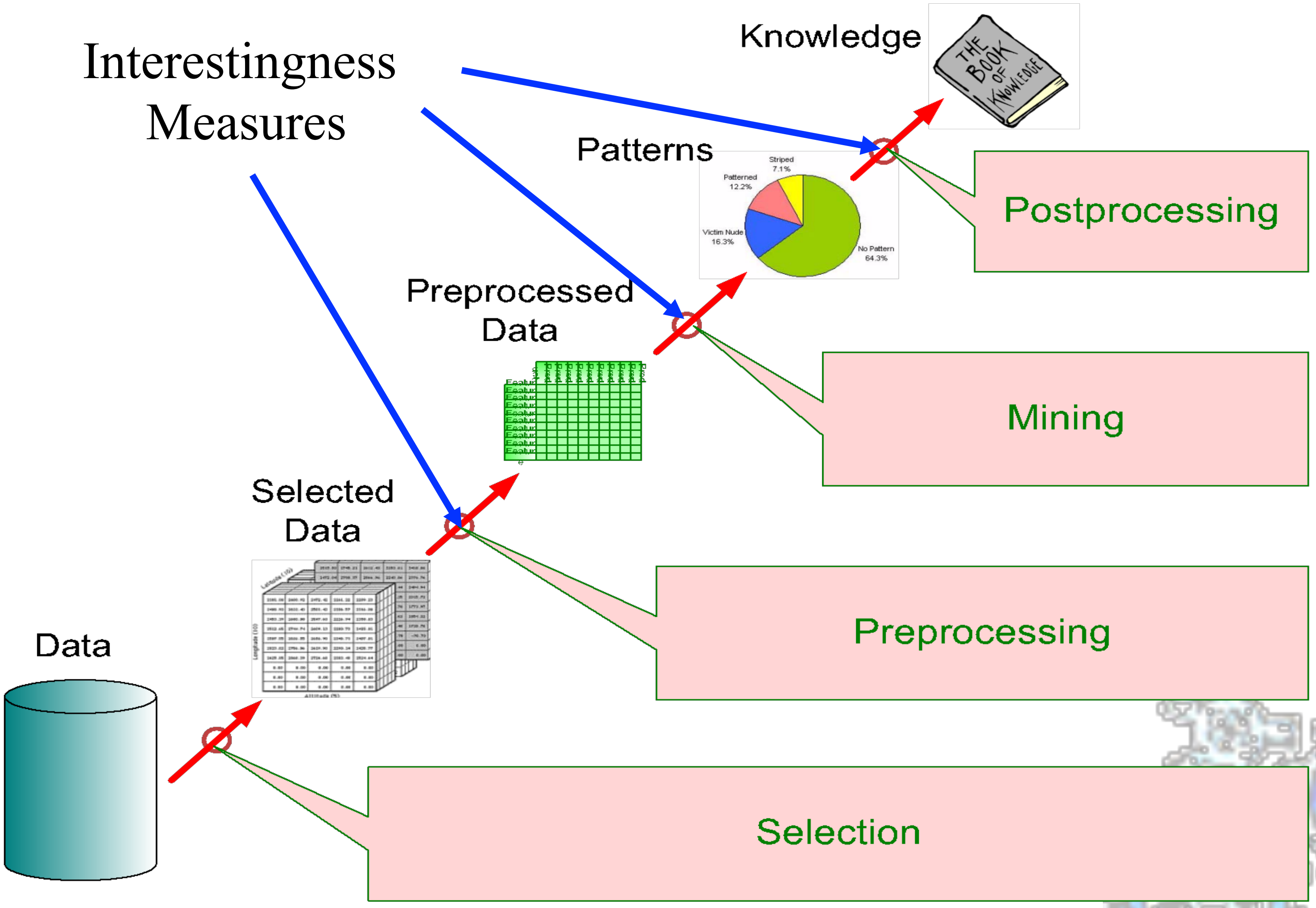
Pattern Evaluation

- 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/rank the derived patterns

- In the original formulation of association rules, support & confidence are the only measures used

Application of Interestingness Measure



Interestingness Measures

Knowledge

Patterns

Preprocessed Data

Selected Data

Data

Postprocessing

Mining

Preprocessing

Selection

Reasoning with AR

- **Redundancy**:

  if {a} ⇒ {b, c} holds, then

  {a, b} ⇒ {c} and {a, c} ⇒ {b} hold also with same support and less or equal confidence. So first rule is stronger.

- **Significance**:

  Example: <1, {a, b}>
  <2, {a} >
  <3, {a, b, c}>
  <4, {b, d}>

  {b} ⇒ {a} has confidence (66%), but is not significant as support({a}) = 75%.

Beyond Support and Confidence

- Example 1: (Aggarwal & Yu, PODS98)

	coffee	not coffee	sum(row)
tea	20	5	25
not tea	70	5	75
sum(col.)	90	10	100

- {tea} => {coffee} has high support (20%) and confidence (80%)
- However, a priori probability that a customer buys coffee is 90%
  - A customer who is known to buy tea is less likely to buy coffee (by 10%)
  - There is a negative correlation between buying tea and buying coffee
  - {~tea} => {coffee} has higher confidence(93%)

Computing Interestingness Measure

- Given a rule X → Y, information needed to compute rule interestingness can be obtained from a contingency table

Contingency table for X → Y

	Y	$\overline{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\overline{X}$	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	$\|T\|$

f_{11}: support of X and Y

f_{10}: support of X and $\overline{Y}$

f_{01}: support of $\overline{X}$ and Y

f_{00}: support of $\overline{X}$ and $\overline{Y}$

Used to define various measures

- support, confidence, lift, Gini, J-measure, etc.

Correlation and Interest

- Two events are independent if $P(A \wedge B) = P(A)*P(B)$, otherwise are correlated.
- Interest = $P(A \wedge B) / P(B)*P(A)$
- Interest expresses measure of correlation
  - = 1 ⇒ A and B are independent events
  - less than 1 ⇒ A and B negatively correlated,
  - greater than 1 ⇒ A and B positively correlated.
  - In our example, I(*buy tea* ∧ *buy coffee*)=0.89 i.e. they are negatively correlated.

Statistical-based Measures

- **Measures that take into account statistical dependence**

$$Lift = \frac{P(Y \mid X)}{P(Y)}$$

$$Interest = \frac{P(X,Y)}{P(X)P(Y)}$$

$$PS = P(X,Y) - P(X)P(Y)$$

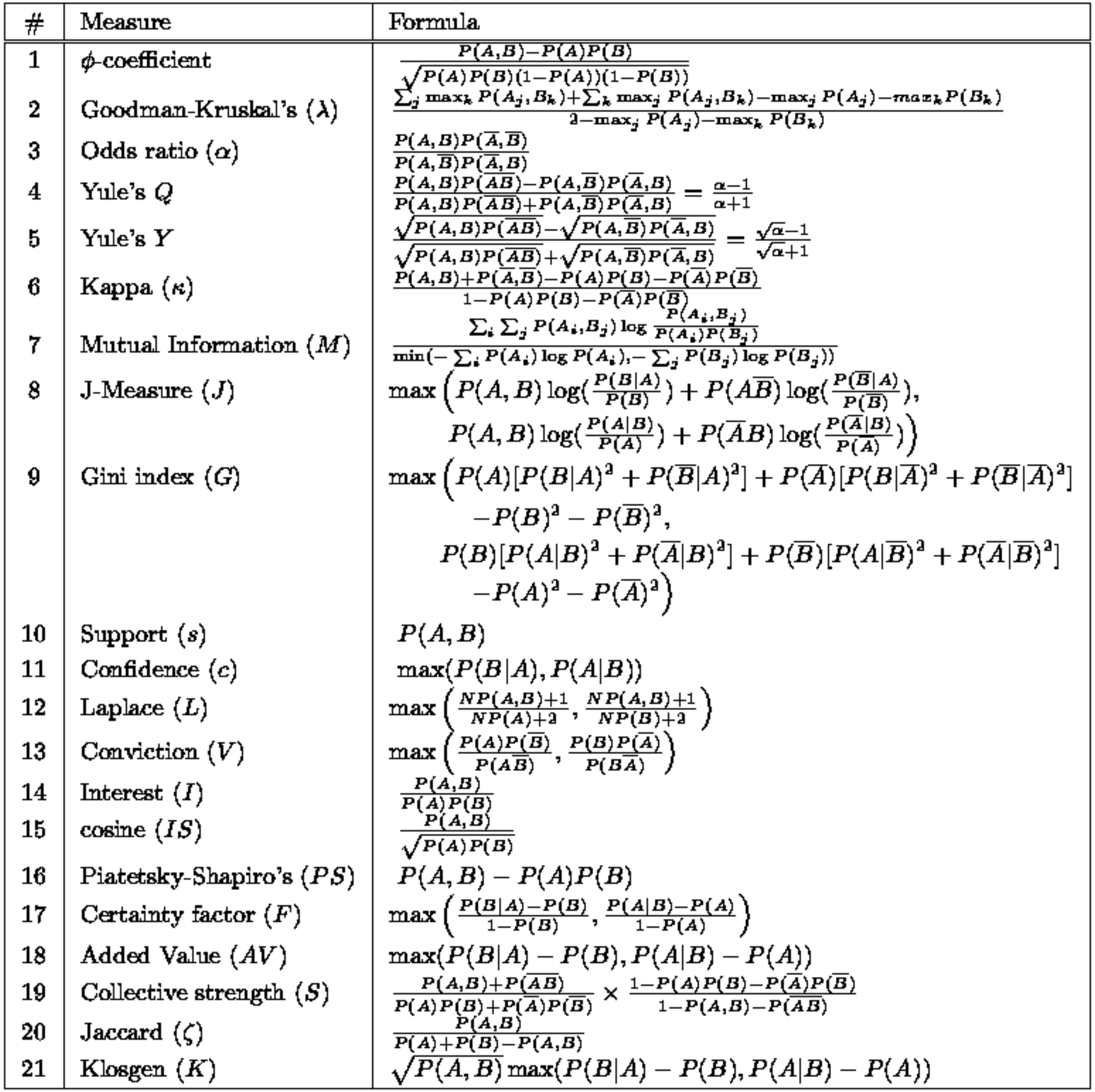
$$\phi - coefficient = \frac{P(X,Y) - P(X)P(Y)}{\sqrt{P(X)[1 - P(X)]P(Y)[1 - P(Y)]}}$$

There are lots of measures proposed in the literature

Some measures are good for certain applications, but not for others

What criteria should we use to determine whether a measure is good or bad?

What about Apriori-style support based pruning? How does it affect these measures?



#	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	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
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)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B) - P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\|A) - P(B), P(A\|B) - P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A) - P(B), P(A\|B) - P(A))$

Properties of A Good Measure

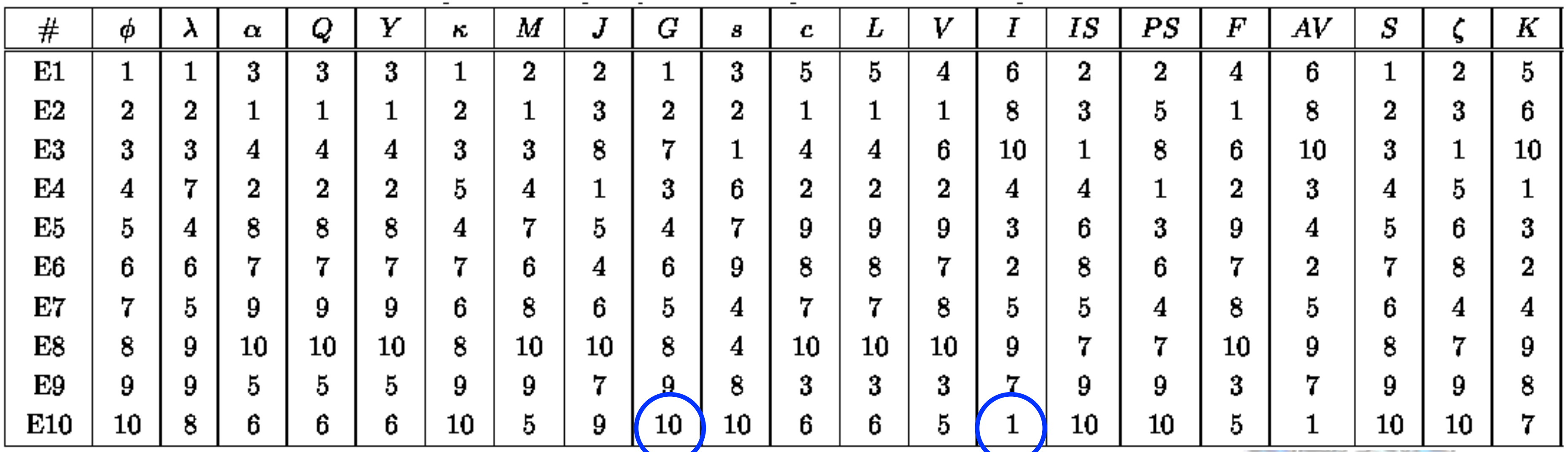
- Piatetsky-Shapiro:
  3 properties a good measure M must satisfy:
  - M(A,B) = 0 if A and B are statistically independent
  - M(A,B) increase monotonically with P(A,B) when P(A) and P(B) remain unchanged
  - M(A,B) decreases monotonically with P(A) [or P (B)] when P(A,B) and P(B) [or P(A)] remain unchanged

Comparing Different Measures

10 examples of contingency tables:

Example	f_{11}	f_{10}	f_{01}	f_{00}
E1	8123	83	424	1370
E2	8330	2	622	1046
E3	9481	94	127	298
E4	3954	3080	5	2961
E5	2886	1363	1320	4431
E6	1500	2000	500	6000
E7	4000	2000	1000	3000
E8	4000	2000	2000	2000
E9	1720	7121	5	1154
E10	61	2483	4	7452

Rankings of contingency tables using various measures:



#	ϕ	λ	α	Q	Y	κ	M	J	G	s	c	L	V	I	IS	PS	F	AV	S	ζ	K
E1	1	1	3	3	3	1	2	2	1	3	5	5	4	6	2	2	4	6	1	2	5
E2	2	2	1	1	1	2	1	3	2	2	1	1	1	8	3	5	1	8	2	3	6
E3	3	3	4	4	4	3	3	8	7	1	4	4	6	10	1	8	6	10	3	1	10
E4	4	7	2	2	2	5	4	1	3	6	2	2	2	4	4	1	2	3	4	5	1
E5	5	4	8	8	8	4	7	5	4	7	9	9	9	3	6	3	9	4	5	6	3
E6	6	6	7	7	7	7	6	4	6	9	8	8	7	2	8	6	7	2	7	8	2
E7	7	5	9	9	9	6	8	6	5	4	7	7	8	5	5	4	8	5	6	4	4
E8	8	9	10	10	10	8	10	10	8	4	10	10	10	9	7	7	10	9	8	7	9
E9	9	9	5	5	5	9	9	7	9	8	3	3	3	7	9	9	3	7	9	9	8
E10	10	8	6	6	6	10	5	9	10	10	6	6	5	1	10	10	5	1	10	10	7

Domain dependent measures

- Together with support, confidence, interest, …, use also (in post-processing) domain-dependent measures
- E.g., use rule constraints on rules
- Example: take only rules which are significant with respect their economic value
- sum(LHS)+ sum(RHS) > 100

MBA in Text / Web Content Mining

- Documents Associations
  - Find (content-based) associations among documents in a collection
  - Documents correspond to items and words correspond to transactions
  - Frequent itemsets are groups of docs in which many words occur in common

	Doc 1	Doc 2	Doc 3	. . .	Doc n
business	5	5	2	. . .	1
capital	2	4	3	. . .	5
fund	0	0	0	. . .	1
⋮	⋮	⋮	⋮	⋮ ⋮ ⋮	⋮
invest	6	0	0	. . .	3

- Term Associations
  - Find associations among words based on their occurrences in documents
  - similar to above, but invert the table (terms as items, and docs as transactions)

MBA in Web Usage Mining

- Association Rules in Web Transactions
  - discover affinities among sets of Web page references across user sessions
- Examples
  - 60% of clients who accessed `/products/`, also accessed `/products/software/webminer.htm`
  - 30% of clients who accessed `/special-offer.html`, placed an online order in `/products/software/`
  - Actual Example from IBM official Olympics Site:
    - ✓ {Badminton, Diving} ==> {Table Tennis} [conf = 69.7%, sup = 0.35%]
- Applications
  - Use rules to serve dynamic, customized contents to users
  - prefetch files that are most likely to be accessed
  - determine the best way to structure the Web site (site optimization)
  - targeted electronic advertising and increasing cross sales

Web Usage Mining: Example

- Association Rules From Cray Research Web Site

Conf	supp	Association Rule
82.8	3.17	/PUBLIC/product-info/T3E ===> /PUBLIC/product-info/T3E/CRAY_T3E.html
90	0.14	/PUBLIC/product-info/J90/J90.html, /PUBLIC/product-info/T3E ===> /PUBLIC/product-info/T3E/CRAY_T3E.html
97.2	0.15	/PUBLIC/product-info/J90, /PUBLIC/product-info/T3E/CRAY_T3E.html, /PUBLIC/product-info/T90, ===> /PUBLIC/product-info/T3E, /PUBLIC/sc.html

- Design “suggestions”
  - from rules 1 and 2: there is something in J90.html that should be moved to th page /PUBLIC/product-info/T3E (why?)

A brief history of AR mining research

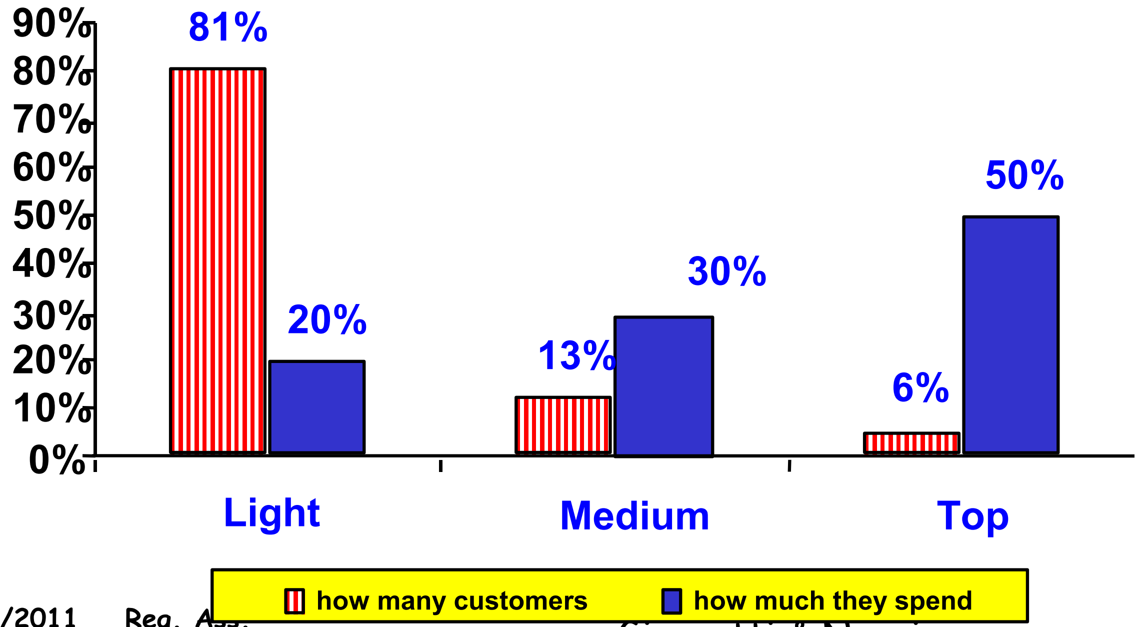
- Apriori (Agrawal et. al SIGMOD93)
- Optimizations of Apriori
    - Fast algorithm (Agrawal et. al VLDB94)
    - Hash-based (Park et. al SIGMOD95)
    - Partitioning (Navathe et. al VLDB95)
    - Direct Itemset Counting (Brin et. al SIGMOD97)
- Problem extensions
    - Multilevel AR (Srikant et. al; Han et. al. VLDB95)
    - Quantitative AR (Srikant et. al SIGMOD96)
    - Multidimensional AR (Lu et. al DMKD'98)
    - Temporal AR (Ozden et al. ICDE98)
- Parallel mining (Agrawal et. al TKDE96)
- Distributed mining (Cheung et. al PDIS96)
- Incremental mining (Cheung et. al ICDE96)

Conclusions

- Association rule mining
  - probably the most significant contribution from the database community to KDD
  - A large number of papers have been published
- Many interesting issues have been explored
- An interesting research direction
  - Association analysis in other types of data: spatial data, multimedia data, time series data, etc.

Conclusion (2)

- MBA is a key factor of success in the competition of supermarket retailers.
- Knowledge of customers and their purchasing behavior brings potentially huge added value.



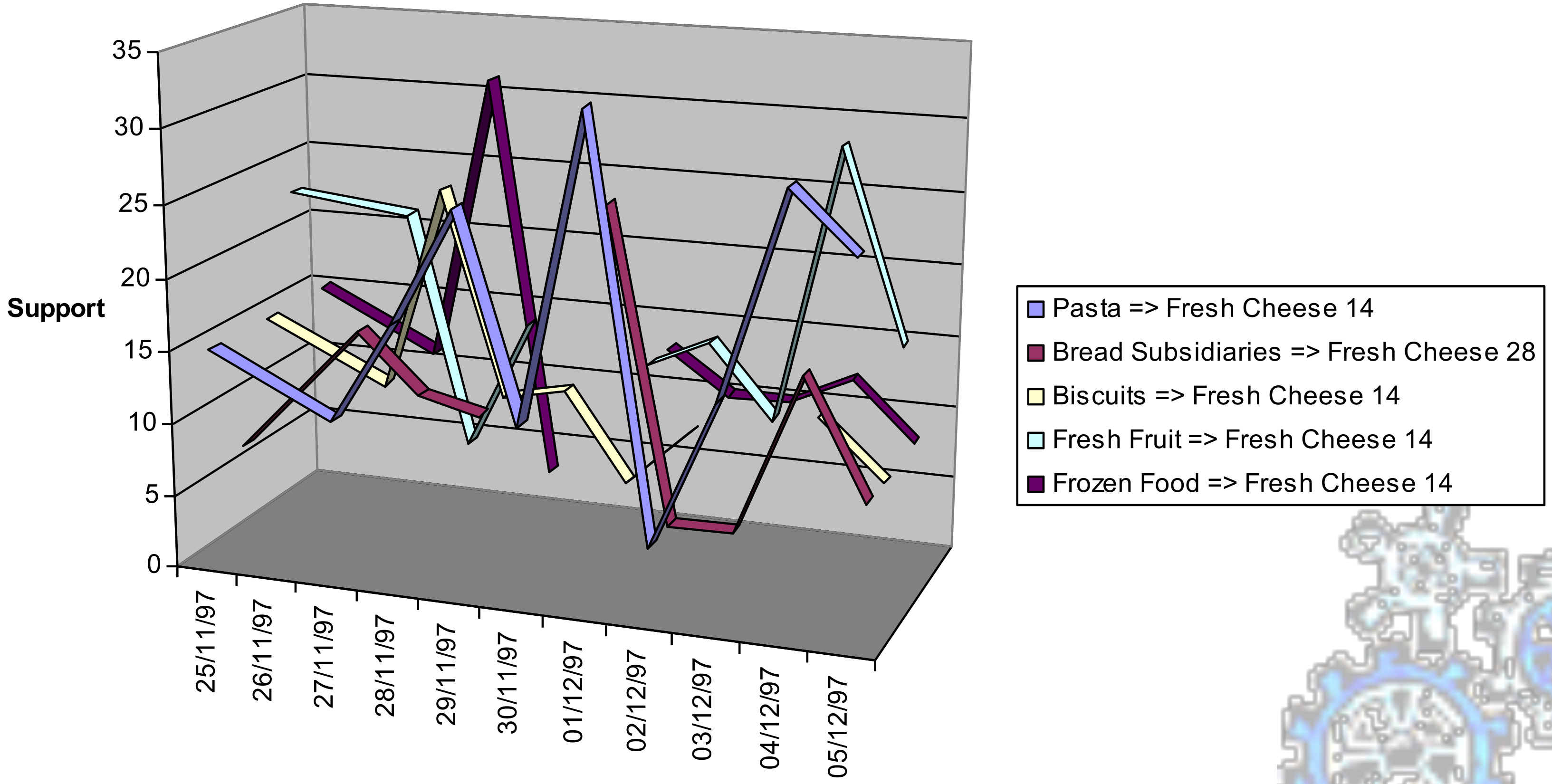
Which tools for market basket analysis?

- Association rule are needed but insufficient

- Market analysts ask for business rules:
  - Is supermarket assortment adequate for the company's target class of customers?
  - Is a promotional campaign effective in establishing a desired purchasing habit?

Business rules: temporal reasoning on AR

- Which rules are established by a promotion?
- How do rules change along time?



References - Association rules

- R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93, 207-216, Washington, D.C.
- R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94 487-499, Santiago, Chile.
- R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95, 3-14, Taipei, Taiwan.
- R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98, 85-93, Seattle, Washington.
- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97, 265-276, Tucson, Arizona..
- D.W. Cheung, J. Han, V. Ng, and C.Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. ICDE'96, 106-114, New Orleans, LA..
- T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. SIGMOD'96, 13-23, Montreal, Canada.
- E.-H. Han, G. Karypis, and V. Kumar. Scalable parallel data mining for association rules. SIGMOD'97, 277-288, Tucson, Arizona.
- J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. VLDB'95, 420-431, Zurich, Switzerland.
- M. Kamber, J. Han, and J. Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. KDD'97, 207-210, Newport Beach, California.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94, 401-408, Gaithersburg, Maryland.
- R. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules. SIGMOD'98, 13-24, Seattle, Washington.
- B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. ICDE'98, 412-421, Orlando, FL.
- J.S. Park, M.S. Chen, and P.S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95, 175-186, San Jose, CA.
- S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. VLDB'98, 368-379, New York, NY.
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98, 343-354, Seattle, WA.

References - Association rules

- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. VLDB'95, 432-443, Zurich, Switzerland.
- C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. VLDB'98, 594-605, New York, NY.
- R. Srikant and R. Agrawal. Mining generalized association rules. VLDB'95, 407-419, Zurich, Switzerland.
- R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. SIGMOD'96, 1-12, Montreal, Canada.
- R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. KDD'97, 67-73, Newport Beach, California.
- D. Tsur, J. D. Ullman, S. Abitboul, C. Clifton, R. Motwani, and S. Nestorov. Query flocks: A generalization of association-rule mining. SIGMOD'98, 1-12, Seattle, Washington.
- B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. ICDE'98, 412-421, Orlando, FL.
- R.J. Miller and Y. Yang. Association rules over interval data. SIGMOD'97, 452-461, Tucson, Arizona.
- J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. ICDE'99, Sydney, Australia.
- F. Giannotti, G. Manco, D. Pedreschi and F. Turini. Experiences with a logic-based knowledge discovery support environment. In Proc. 1999 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (SIGMOD'99 DMKD). Philadelphia, May 1999.
- F. Giannotti, M. Nanni, G. Manco, D. Pedreschi and F. Turini. Integration of Deduction and Induction for Mining Supermarket Sales Data. In Proc. PADD'99, Practical Application of Data Discovery, Int. Conference, London, April 1999.