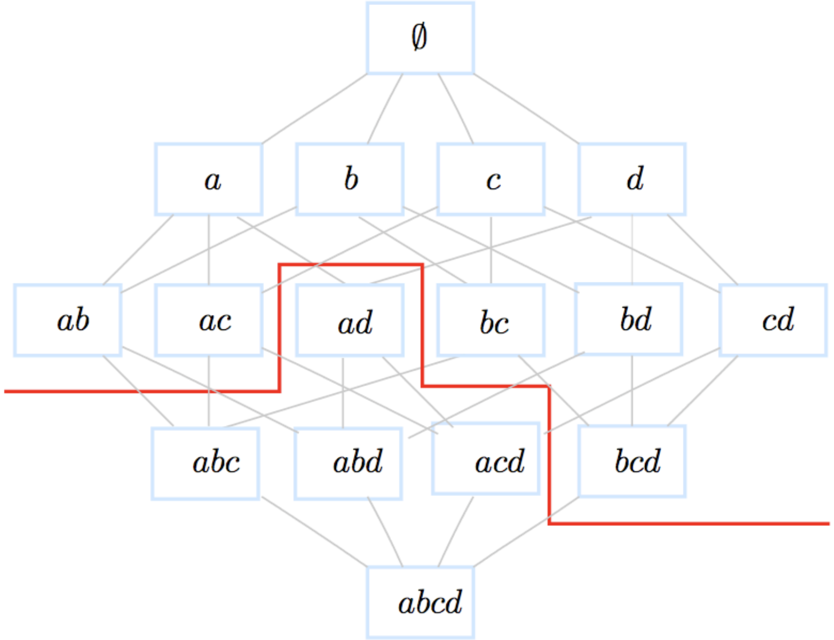
# Artificial Intelligence

## Machine Learning
## Association Rules



# Outline

1. Introduction
2. A two-step process
3. Applications
4. Definitions and examples
5. Frequent patterns: Apriori algorithm
6. Example
7. Representation of $\mathcal{D}$
8. Definitions cont'd
9. Association rules algorithm
10. Example
11. A probabilistic framework Association Rules
12. Support-confidence cons
13. Quantitative association rules

# Introduction

- Unsupervised task.
- R. Agrawal, T. Imielinski and A.N. Swami Mining Association Rules between sets of items in large databases. Proceedings of SIGMOD 1993.
- Highly cited work because of its wide applicability.

# Applications

- Market Basket Analysis: cross-selling (ex. Amazon), product placement, affinity promotion, customer behavior analysis
- Collaborative filtering
- Web organization
- Symptoms-diseases associations
- Supervised classification

# A two-step process

Given a transaction dataset $\mathcal{D}$

1. Mining **frequent** patterns in $\mathcal{D}$
2. Generation of **strong** association rules

**Example**:

$\{Bread,\ Butter\}$ is a frequent pattern (itemset)

$Bread \rightarrow Butter$ is a strong rule

# Definitions

- **Item**: an object belonging to $\mathcal{I} = \{x_1, x_2, ..., x_m\}$.
- **Itemset**: any subset of $\mathcal{I}$.
- $k$-**itemset**: an itemset of cardinality $k$.
- We define a total order $(\mathcal{I}, <)$ on the items.
- $\mathcal{P}(\mathcal{I})$ is a **lattice** with $\bot = \emptyset$ and $\top = \mathcal{I}$.
- **Transaction**: itemset identified by a unique identifier **tid**.
- $\mathcal{T}$: the set of all transactions ids. **Tidset**: a subset of $\mathcal{T}$.
- **Transaction dataset**: $\mathcal{D} = \{(tid, X_{tid})\ /tid \in \mathcal{T},\ X_{tid} \subseteq \mathcal{I}\}$

# Example

| item | name |
|---|---|
| a | coffee |
| b | milk |
| c | butter |
| d | bread |

| $\mathcal{D}$ | |
|---|---|
| tid | transaction |
| 1 | $a\ b$ |
| 2 | $a\ c$ |
| 3 | $c\ d$ |
| 4 | $b\ c\ d$ |
| 5 | $a\ b\ c\ d$ |

$\mathcal{I}=$

$\mathcal{T}=$

$\mathcal{D}=$

# Example

| item | name |
|---|---|
| a | coffee |
| b | milk |
| c | butter |
| d | bread |

| $\mathcal{D}$ | |
|---|---|
| **tid** | **transaction** |
| 1 | $a\ b$ |
| 2 | $a\ c$ |
| 3 | $c\ d$ |
| 4 | $b\ c\ d$ |
| 5 | $a\ b\ c\ d$ |

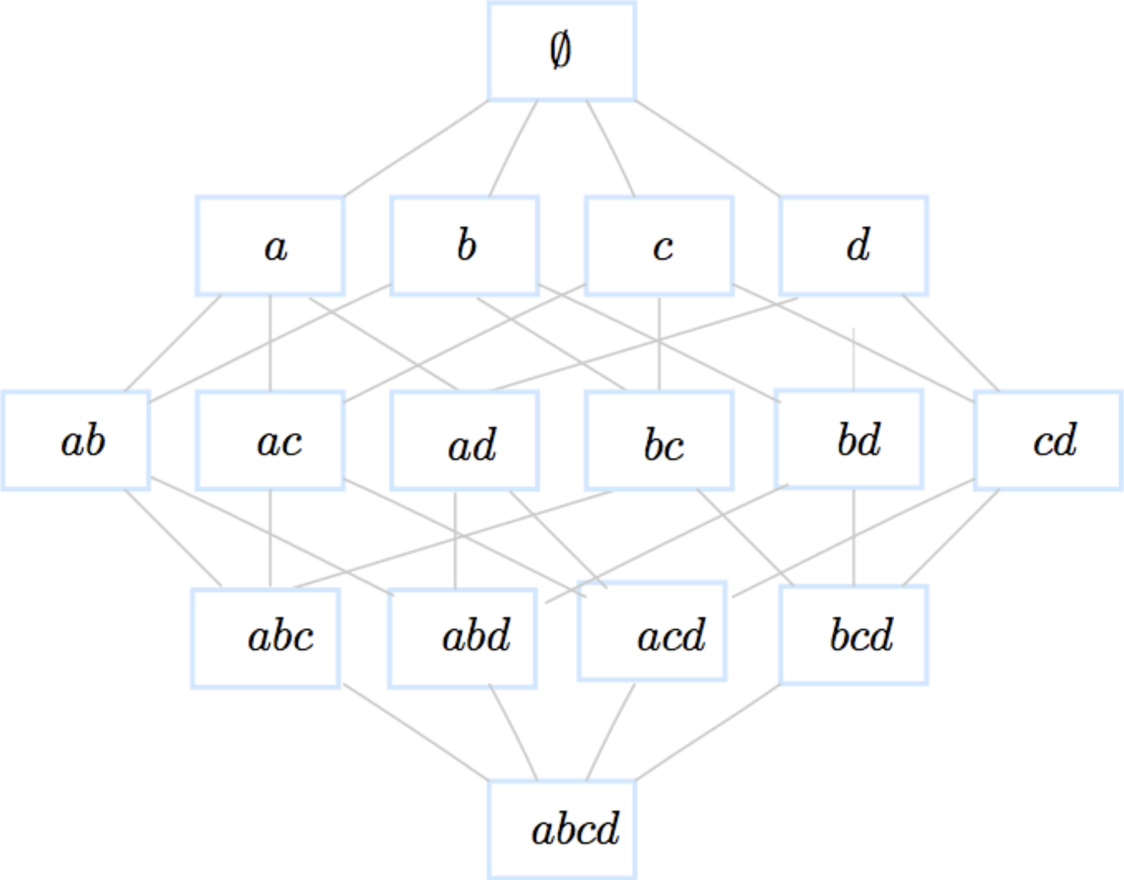
$$\mathcal{I} = \{a,\ b,\ c,\ d\}$$

$$\mathcal{T} = \{1,\ 2,\ 3,\ 4,\ 5\}$$

$$\mathcal{D} = \{(1,\ ab), (2,\ ac), (3,\ cd), (4,\ bcd), (5,\ abcd)\}$$

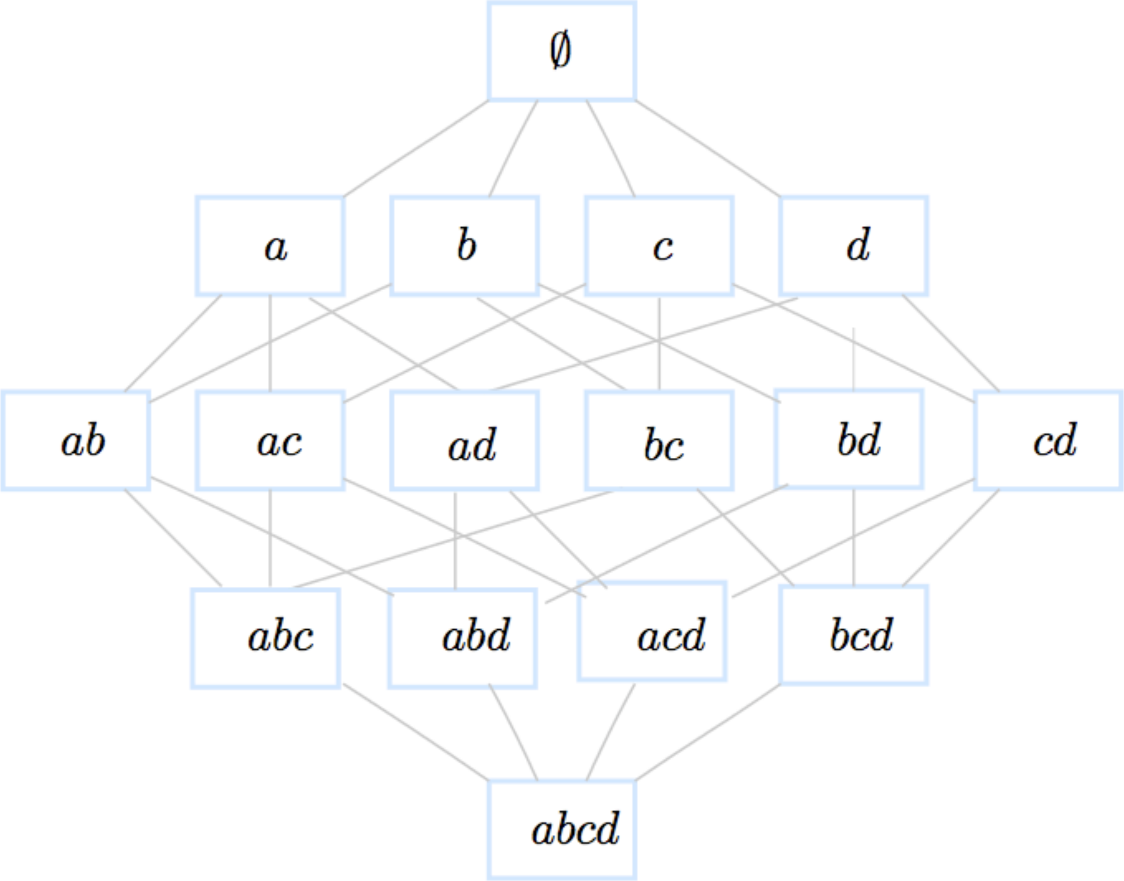
E.g., $\{b, c\}$ is a 2-itemset, for writing simplification we will give up the braces and write $bc$. $\{3, 4, 5\}$ is a tidset similarly let's abandon the braces here too and write 345.

# Example



Lattice of itemsets of size . . .

# Example



Lattice of itemsets of size $2^{|\mathcal{I}|} = 16$.

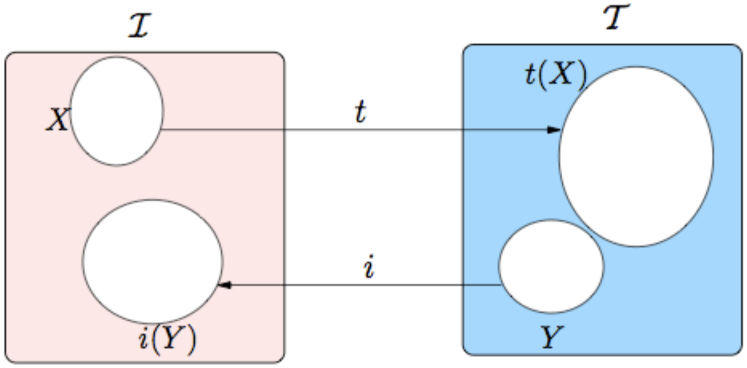
# Definitions cont'd

- **Mapping** $t$:

$$\begin{aligned} t &: \mathcal{P}(\mathcal{I}) \rightarrow \mathcal{P}(\mathcal{T}) \\ X &\mapsto t(X) = \{tid \in \mathcal{T} | \exists X_{tid}, (tid, X_{tid}) \in \mathcal{D} \wedge X \subseteq X_{tid}\} \end{aligned}$$

- **Mapping** $i$:

$$\begin{aligned} i &: \mathcal{P}(\mathcal{T}) \rightarrow \mathcal{P}(\mathcal{I}) \\ Y &\mapsto i(Y) = \{x \in \mathcal{I} | \forall (tid, X_{tid}) \in \mathcal{D},\ \ tid \in Y \ \ \Rightarrow x \in X_{tid}\} \end{aligned}$$

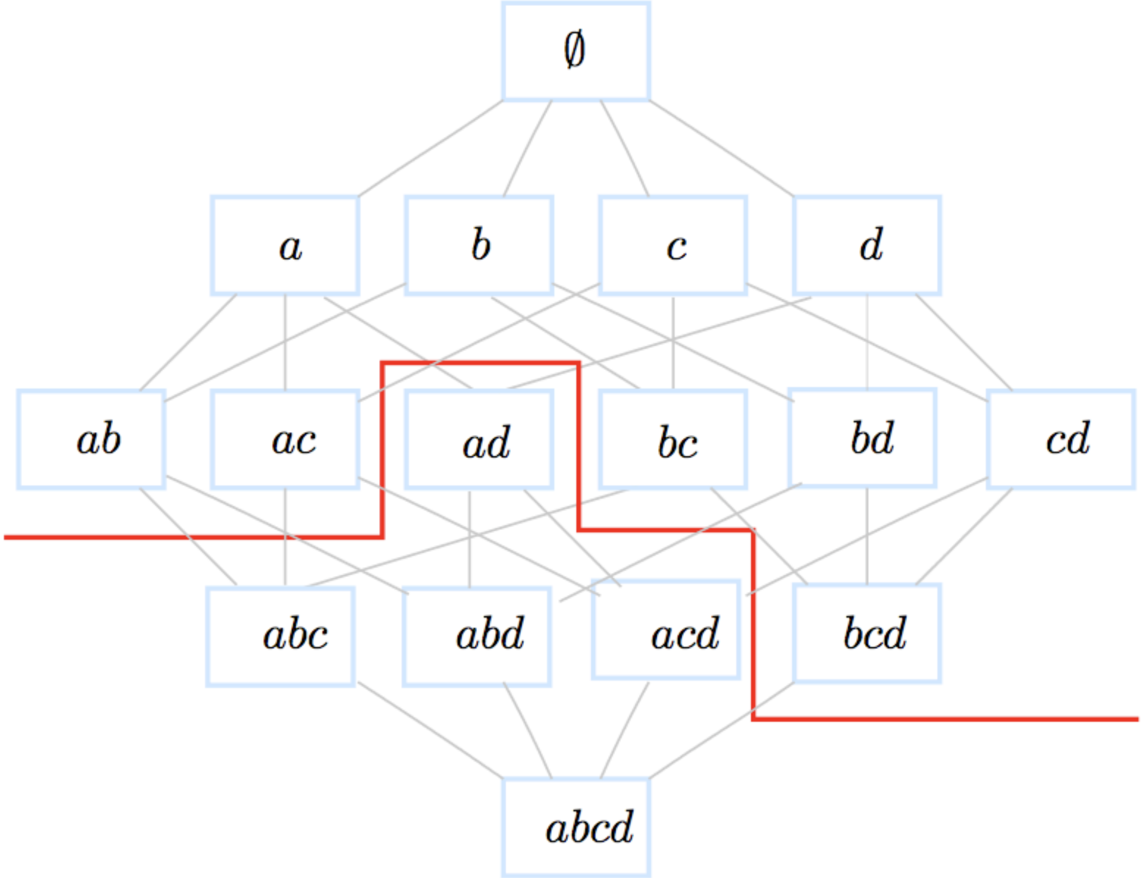


# Definitions cont'd

- **Frequency**: $freq(X) = |\{(tid, X_{tid}) \in \mathcal{D} / X \subseteq X_{tid}\}| \; = \; |t(X)|$
- **Support**: $supp(X) \; = \; \frac{|t(X)|}{|\mathcal{D}|}$
- **Frequent itemset**: $X$ is frequent iff supp$(X) \geq$ MinSupp
- **Property (Support downward closure)** : if an itemset is frequent then all its subsets also are frequent.
- **Mining Frequent Itemsets**:

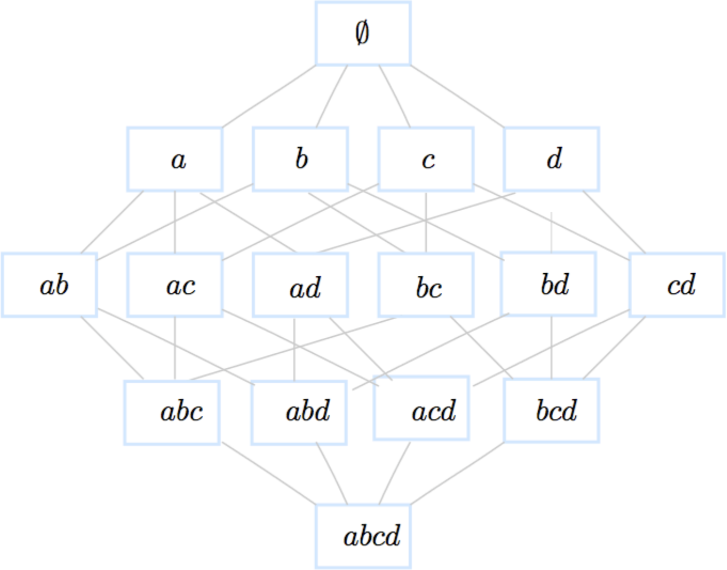
$$\mathcal{F} = \{ \; X \subseteq \mathcal{I} | \; \text{supp(X)} \geq MinSupp\}$$

# Example

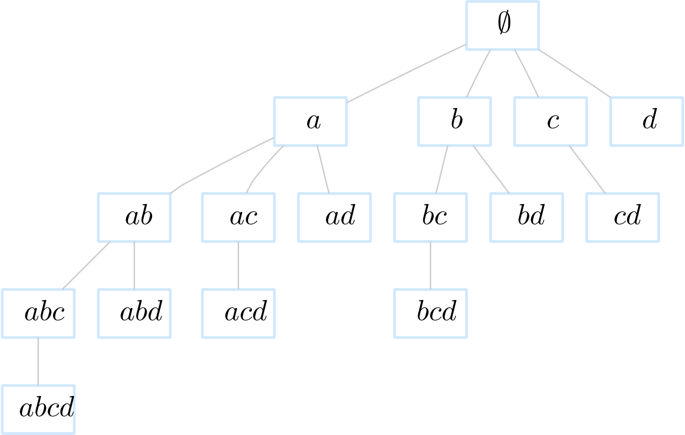


MinSupp=40%

# BFS and DFS



Breadth First Search



Depth First Search

# Apriori pseudo-algorithm

Level-wise algorithm – lattice explored with a Breath First Search approach (BFS). Start at level 1 in the lattice: $k = 1$

- Generate candidates of size $k$

$$\mathcal{C}_k = \{(c_k,\ supp(c_k))|\forall X \subset c_k,\ X \neq \emptyset,\ support(X) \geq MinSupp\}$$

- Scan the dataset to compute the support of each candidate and keep the frequent ones

$$\mathcal{F}_k = \{(l_k,\ supp(l_k))|\ supp(l_k) \geq Minsupp\}$$

- Go the the next level $k = k + 1$ and redo the process.

# Example

Minsupp=2/5 (40%)

| $\mathcal{D}$ | |
|---|---|
| tid | transaction |
| 1 | $a\ b$ |
| 2 | $a\ c$ |
| 3 | $c\ d$ |
| 4 | $b\ c\ d$ |
| 5 | $a\ b\ c\ d$ |

| $\mathcal{C}_1$ |
|---|
| Itemset |
| $a$ |
| $b$ |
| $c$ |
| $d$ |

$\xrightarrow[of\ \mathcal{D}]{Scan}$

| $\mathcal{C}_1$ | |
|---|---|
| Itemset | Support |
| $a$ | 3/5 |
| $b$ | 3/5 |
| $c$ | 4/5 |
| $d$ | 3/5 |

$\rightarrow$

| $\mathcal{F}_1$ | |
|---|---|
| Itemset | Support |
| $a$ | 3/5 |
| $b$ | 3/5 |
| $c$ | 4/5 |
| $d$ | 3/5 |

| $\mathcal{C}_2$ |
|---|
| Itemset |
| $ab$ |
| $ac$ |
| $ad$ |
| $bc$ |
| $bd$ |
| $cd$ |

$\xrightarrow[of\ \mathcal{D}]{Scan}$

| $\mathcal{C}_2$ | |
|---|---|
| Itemset | Support |
| $ab$ | 2/5 |
| $ac$ | 2/5 |
| $ad$ | 1/5 |
| $bc$ | 2/5 |
| $bd$ | 2/5 |
| $cd$ | 3/5 |

$\rightarrow$

| $\mathcal{F}_2$ | |
|---|---|
| Itemset | Support |
| $ab$ | 2/5 |
| $ac$ | 2/5 |
| $bc$ | 2/5 |
| $bd$ | 2/5 |
| $cd$ | 3/5 |

| $\mathcal{C}_3$ |
|---|
| Itemset |
| $abc$ |
| $bcd$ |

$\xrightarrow[of\ \mathcal{D}]{Scan}$

| $\mathcal{C}_3$ | |
|---|---|
| Itemset | Support |
| $abc$ | 1/5 |
| $bcd$ | 2/5 |

$\rightarrow$

| $\mathcal{F}_3$ | |
|---|---|
| Itemset | Support |
| $bcd$ | 2/5 |

# Apriori bottleneck

Characteristics of real-life datasets:

1. Billions of transactions,
2. Tens of thousands of items,
3. Tera-bytes of data.

This leads to:

1. Multiple scans of the dataset residing in the disk (costly I/O operations)
2. A HUGE number of candidates sets.

# Representation of $\mathcal{D}$

**Row-wise**

| | | | | |
|---|---|---|---|---|
| 1 | $a$ | $b$ | | |
| 2 | $a$ | $c$ | | |
| 3 | $c$ | $d$ | | |
| 4 | $b$ | $c$ | $d$ | |
| 5 | $a$ | $b$ | $c$ | $d$ |

**Column-wise**

| $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|
| 1 | 1 | 2 | 3 |
| 2 | 4 | 3 | 4 |
| 5 | 5 | 4 | 5 |
| | | 5 | |

**Boolean**

| | $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 1 |

# Definitions cont'd

- Given $\mathcal{F}$ and a Minimum confidence threshold MinConf

- Generate rules:

$$(l - C) \rightarrow C$$

$$conf((l - C) \rightarrow C) = \frac{supp(l)}{supp(l - C)} \geq MinConf$$

- From a $k - itemset$ $(k > 1)$, one can generate $2^k - 1$ rules.

**Property**

Let $l$ be a large (frequent) itemset:

$$\forall C \subset l,\ C \neq \emptyset,\ [(l - C) \rightarrow C]\ is\ strong\ \Rightarrow \forall \tilde{C} \subset C,\ \tilde{C} \neq \emptyset,\ [(l - \tilde{C}) \rightarrow \tilde{C}]\ is\ strong$$

# Example

Minconf=60%

| Itemset | Rule# | Rule | Confidence | Strong? |
|---|---|---|---|---|
| $ab$ | 1 | $a \to b$ | $2/3 = 66.66\%$ | yes |
| | 2 | $b \to a$ | $2/3 = 66.66\%$ | yes |
| $ac$ | 3 | $a \to c$ | $2/3 = 66.66\%$ | yes |
| | 4 | $c \to a$ | $2/4 = 50.00\%$ | no |
| $bc$ | 5 | $b \to c$ | $2/3 = 66.66\%$ | yes |
| | 6 | $c \to b$ | $2/4 = 50.00\%$ | no |
| $bd$ | 7 | $b \to d$ | $2/3 = 66.66\%$ | yes |
| | 8 | $d \to b$ | $2/3 = 66.66\%$ | yes |
| $cd$ | 9 | $c \to d$ | $3/4 = 75.00\%$ | yes |
| | 10 | $d \to c$ | $3/3 = 100.00\%$ | yes |

# Example

Minconf=60%

| Itemset | Rule# | Rule | Confidence | Strong? |
|---|---|---|---|---|
| $bcd$ | 11 | $cd \rightarrow b$ | $2/3 = 66.66\%$ | yes |
| | 12 | $bd \rightarrow c$ | $2/2 = 100.00\%$ | yes |
| | 13 | $bc \rightarrow d$ | $2/2 = 100.00\%$ | yes |

| Itemset | Rule# | Rule | Confidence | Strong? |
|---|---|---|---|---|
| $bcd$ | 14 | $d \rightarrow bc$ | $2/3 = 66.66\%$ | yes |
| | 15 | $c \rightarrow bd$ | $2/4 = 50.00\%$ | no |
| | 16 | $b \rightarrow cd$ | $2/3 = 66.66\%$ | yes |

# Probabilistic Interpretation

Brin et al. 97

$$R: \quad A \longrightarrow C$$

- $R$ measures the distribution of $A$ and $C$ in the finite space $\mathcal{D}$.
- The sets $A$ and $C$ are 2 events
- $P(A)$ and $P(C)$ the probabilities that events $A$ and $C$ happen resp. estimated by the the frequency of $A$ and $C$ resp. in $\mathcal{D}$

$$supp(A \rightarrow C) = supp(A \cup C) = P(A \wedge C)$$

$$conf(A \rightarrow C) = P(C|A) = \frac{P(A \wedge C)}{P(A)}$$

# Support-Confidence: cons

- Example (Brin et al. 97)

| | $coffee$ | $\overline{coffee}$ | $\sum rows$ |
|---|---|---|---|
| $tea$ | 20 | 5 | 25 |
| $\overline{tea}$ | 70 | 5 | 75 |
| $\sum columns$ | 90 | 10 | 100 |

$$tea \to coffee \quad (supp = 20\%, conf = 80\%)$$

Strong rule?

# Support-Confidence: cons

- Example (Brin et al. 97)

|  | $coffee$ | $\overline{coffee}$ | $\sum rows$ |
|---|---|---|---|
| $tea$ | 20 | 5 | 25 |
| $\overline{tea}$ | 70 | 5 | 75 |
| $\sum columns$ | 90 | 10 | 100 |

$$tea \rightarrow coffee \quad (supp = 20\%, conf = 80\%)$$

Strong rule? Yes but a misleading one!

$Support(coffee) = 90\%$ is a bias that the confidence cannot detect because it ignores support(coffee).

# Other evaluation Measures

- Interest (Piatetsky-Shapiro 91) or Lift (Bayardo et al. 99)

$$Interest(A \to C) = \frac{P(A \wedge C)}{P(A) \times P(C)} = \frac{supp(A \cup C)}{supp(A) \times supp(C)}$$

Interest is between 0 and $+\infty$:

1. If $Interest(\mathcal{R}) = 1$ then $A$ and $C$ are independent;
2. If $Interest(\mathcal{R}) > 1$ then $A$ and $C$ are positively dependent;
3. If $Interest(\mathcal{R}) < 1$ then $A$ and $C$ are negatively dependent.

$$Interest(A \to C) = \frac{conf(A \to C)}{supp(C)} = \frac{conf(C \to A)}{supp(A)}$$

# Other evaluation Measures

|  | $coffee$ | $\overline{coffee}$ | $\sum rows$ |
|---|---|---|---|
| $tea$ | 20 | 5 | 25 |
| $\overline{tea}$ | 70 | 5 | 75 |
| $\sum columns$ | 90 | 10 | 100 |

$$Interest(tea \to coffee) = \frac{P(tea \wedge coffee)}{P(tea) \times P(coffee)} = \frac{0.2}{0.25 * 0.9} = 0.89 < 1$$

|  | $coffee$ | $\overline{coffee}$ |
|---|---|---|
| $tea$ | 0.89 | 2 |
| $\overline{tea}$ | 1.03 | 0.66 |

# Multi-dimensional rules

- One-dimensional rules:

$$buy(x, \text{``}Bread'') \longrightarrow buy(x, \text{``}Butter'')$$

- Multi-dimensional rules:

$$buy(x, \text{``}Pizza'') \wedge age(x, \text{``}Young'') \longrightarrow buy(x, \text{``}Coke'')$$

- Construct k-predicatesets instead of k-itemsets

- How about numerical features?

$$buy(x, \text{``}Pizza'') \wedge age(x, \text{``}\mathsf{18} - \mathsf{22}'') \longrightarrow buy(x, \text{``}Coke'')$$

# Post-processing of AR

- AR framework may lead to a large number of rules.

- How one can reduce the number of rules?

1. Use many evaluation measures

2. Increase minimum support

3. Increase minimum confidence

4. use rule templates (define constraints on max rule length, exclude some items, include in the rules specific items) (Agrawal et al. 1995, Salleb et al. 2007)

# Implementations

- **FIMI** Frequent Itemset Mining Implementations Repository `http://fimi.cs.helsinki.fi/` FIMI'03 and FIMI'04 workshop, Bayardo, Goethals & Zaki

- **Apriori** `http://www.borgelt.net/apriori.html` developed by Borgelt

- **Weka** `http://www.cs.waikato.ac.nz/ml/weka/` by Witten & Frank

- **ARMADA** Data Mining Tool version 1.3.2 in matlab available at Mathworks, by Malone

# FP algorithms

According to the strategy to traverse the search space:

- Breadth First Search (ex: Apriori, AprioriTid, Partition, DIC)

- Depth First Search (ex: Eclat, Clique, Depth project)

- Hybrid (ex: AprioriHybrid, Hybrid, Viper, Kdci)

- Pattern growth, i.e. no candidate generation (ex: Fpgrowth, HMine, Cofi)

# Uniform notion of item

- Apriori has been initially designed for **boolean tables** (transactional datasets) thus propositional logic was sufficient to express: items, itemsets and rules.

$$milk \rightarrow cereals$$

- For **relational tables**, one need to extend the notion of items to literals:

$$item \equiv (attribute, value)$$

  An attribute could be:

  1. categorical, for ex. $(color, blue)$,
  2. quantitatif with a few numerical values, for ex. $(\#cars, 2)$,
  3. quantitatif with a large domain values, for ex. $(age, [20, 40])$.

# Example

| $\mathcal{D}$ : people | | | |
|---|---|---|---|
| **id** | **age** | **married?** | **#cars** |
| 1 | 23 | no | 1 |
| 2 | 25 | yes | 1 |
| 3 | 29 | no | 0 |
| 4 | 34 | yes | 2 |
| 5 | 38 | yes | 2 |

| Examples of frequent itemsets | |
|---|---|
| **itemset** | **support** |
| (age, 20..29) | 3 |
| (age, 30..39) | 2 |
| (married?, yes) | 3 |
| (married?, no) | 2 |
| (#cars, 1) | 2 |
| (#cars, 2) | 2 |
| (age, 30..39),(married?, yes) | 2 |

| Examples of rules | | |
|---|---|---|
| **rule** | **support** | **confidence** |
| (age, 30..39) et (married?, yes) ⟶ (#cars, 2) | 40% | 100% |
| (age, 20..29) ⟶ (#cars, 1) | 60% | 66.6% |

# Quantitative AR

**Question:** Mining Quantitative AR is not a simple extension of mining categorical AR. why?

- **Infinite search space:** In Boolean AR, the Ariori property allows to prune the search space efficiently, but we do explore the whole space of hypothesis (lattice of itemsets), which is IMPOSSIBLE for Quantitative AR.

- **The support-confidence tradeoff:** Choosing intervals is quite sensitive to support and confidence.

  - intervals too small, not enough support;
  - intervals too large, not enough confidence.

- What is the difference between supervised and **unsupervised discretization**?

# Approaches to mine QARs

- Discretization-based approaches
- Distribution-based approaches
- Optimization-based approaches

# Approaches to mine QARs

**Discretization-based approaches**

- A pre-processing step
- Use equi-depth, equi-width, domain-knowledge
- Lent et al., 1997; Miller and Yang, 1997; Srikant and Agrawal, 1996; Wang et al., 1998
- Discretization combined with clustering or interval merging.
- Problems: univariate, sensitive to outliers, loss of information.

# Approaches to mine QARs

**Distribution-based approaches**

$$Sex = female \rightarrow Height : mean = 168 \wedge Weight : mean = 68$$

- Aumann and Lindell, 1999, Webb 2001.
- Restricted form of rules:
  1. A set of categorical attributes on the left-hand side and several distributions on the right-hand side,
  2. A single discretized numeric attribute on the left-hand side and a single distribution on the right-hand side.

# Approaches to mine QARs

**Optimization-based approaches**

- Numerical attributes are optimized during the mining process

- Fukuda et al., 96, Rastogi and Shim 99, Brin et al. 2003. Techniques inspired from image segmentation.

  $$Gain(A \to B) = Supp(AB) - MinConf * Supp(A)$$

  Form of the rules restricted to 1 or 2 numerical attributes.

- Mata et al. 2002 Use genetic algorithms to optimize the support of itemsets with non instantiated intervals.

  $$\mathsf{Fitness} = \mathsf{cov} - (\psi * \mathsf{ampl}) - (\omega * \mathsf{mark}) + (\mu * \mathsf{nAtr})$$

  Apriori-like algorithm to mine association rules.

# Approaches to mine QARs

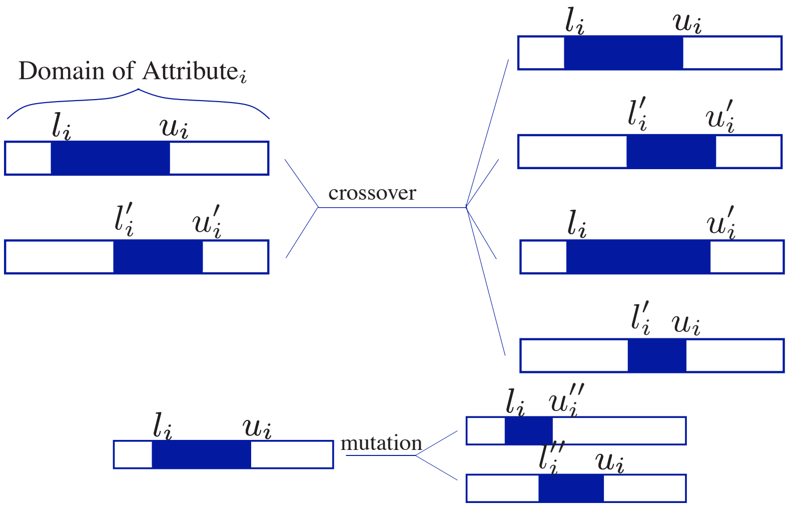
## Optimization-based approaches

- Ruckert et al. 2004 use half-spaces to mine such rules like:

$$x_1 > 20 \rightarrow 0.5x_3 + 2.3x_6 \geq 100$$

  Cannot handle categorical attributes.

- Salleb et al 2007: QuantMiner Optimize the *Gain* of rules templates using a genetic algorithm.



# Approaches to mine QARs

**Optimization-based approaches**: QuantMiner cont'd.

Example UCI Iris dataset:

$$\begin{array}{l} \texttt{Species=} \\ \texttt{value} \end{array} \Rightarrow \left\{ \begin{array}{ll} \texttt{PW} \in [l_1, u_1] & \texttt{SW} \in [l_2, u_2] \\ \texttt{PL} \in [l_3, u_3] & \texttt{SL} \in [l_4, u_4] \end{array} \right\} \begin{array}{l} \texttt{supp\%} \\ \texttt{conf\%} \end{array}$$

$$\begin{array}{l} \texttt{Species=} \\ \texttt{setosa} \end{array} \Rightarrow \left\{ \begin{array}{ll} \texttt{PW} \in [1, 6] & \texttt{SW} \in [31, 39] \\ \texttt{PL} \in [10, 19] & \texttt{SL} \in [46, 54] \end{array} \right\} \begin{array}{l} 23\% \\ 70\% \end{array}$$

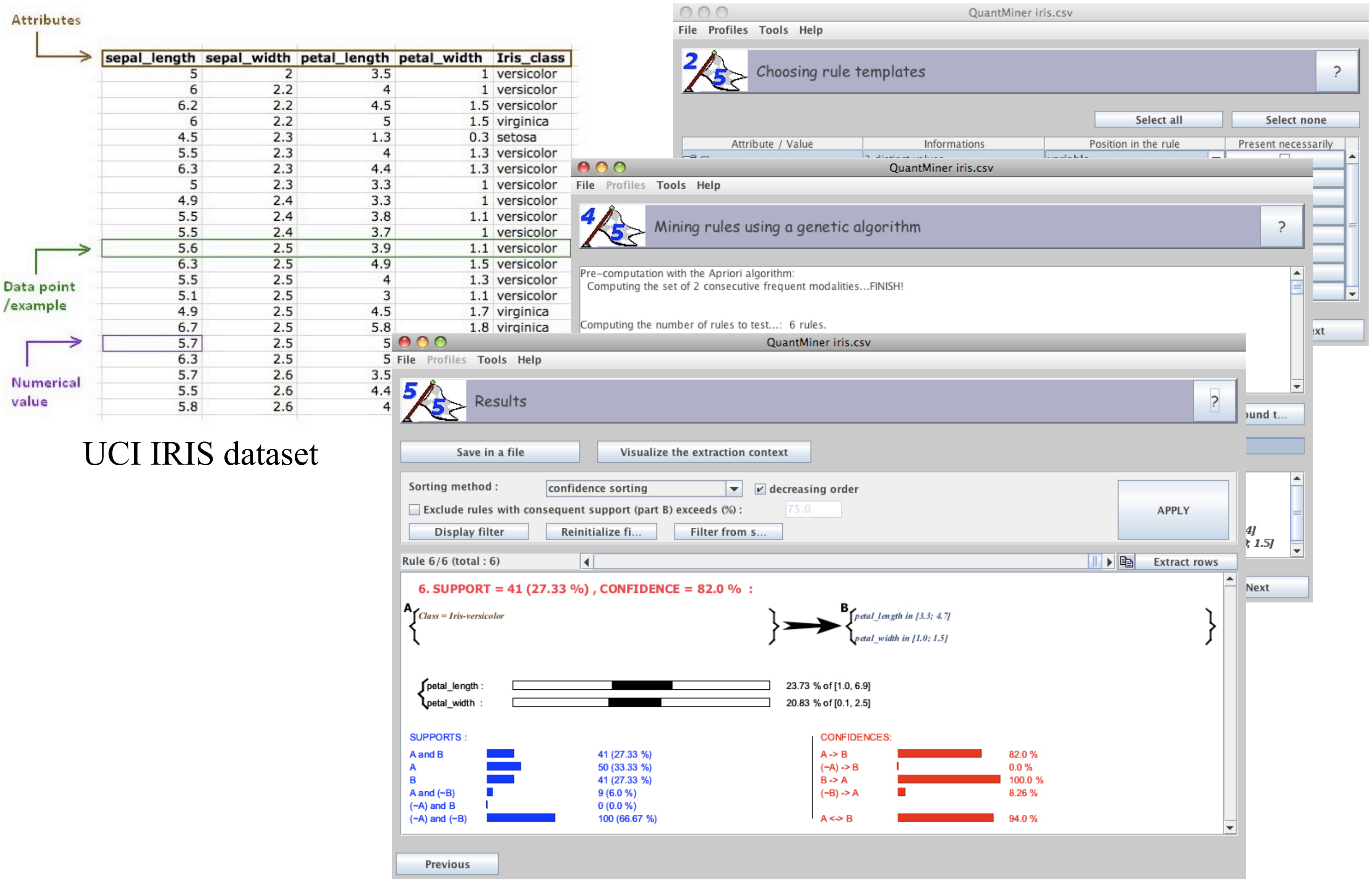
$$\begin{array}{l} \texttt{Species=} \\ \texttt{versicolor} \end{array} \Rightarrow \left\{ \begin{array}{ll} \texttt{PW} \in [10, 15] & \texttt{SW} \in [22, 30] \\ \texttt{PL} \in [35, 47] & \texttt{SL} \in [55, 66] \end{array} \right\} \begin{array}{l} 21\% \\ 64\% \end{array}$$

$$\begin{array}{l} \texttt{Species=} \\ \texttt{virginica} \end{array} \Rightarrow \left\{ \begin{array}{ll} \texttt{PW} \in [18, 25] & \texttt{SW} \in [27, 33] \\ \texttt{PL} \in [48, 60] & \texttt{SL} \in [58, 72] \end{array} \right\} \begin{array}{l} 20\% \\ 60\% \end{array}$$

# QuantMiner

`http://quantminer.github.io/QuantMiner/`

# QuantMiner



| sepal_length | sepal_width | petal_length | petal_width | Iris_class |
|---|---|---|---|---|
| 5 | 2 | 3.5 | 1 | versicolor |
| 6 | 2.2 | 4 | 1 | versicolor |
| 6.2 | 2.2 | 4.5 | 1.5 | versicolor |
| 6 | 2.2 | 5 | 1.5 | virginica |
| 4.5 | 2.3 | 1.3 | 0.3 | setosa |
| 5.5 | 2.3 | 4 | 1.3 | versicolor |
| 6.3 | 2.3 | 4.4 | 1.3 | versicolor |
| 5 | 2.3 | 3.3 | 1 | versicolor |
| 4.9 | 2.4 | 3.3 | 1 | versicolor |
| 5.5 | 2.4 | 3.8 | 1.1 | versicolor |
| 5.5 | 2.4 | 3.7 | 1 | versicolor |
| 5.6 | 2.5 | 3.9 | 1.1 | versicolor |
| 6.3 | 2.5 | 4.9 | 1.5 | versicolor |
| 5.5 | 2.5 | 4 | 1.3 | versicolor |
| 5.1 | 2.5 | 3 | 1.1 | versicolor |
| 4.9 | 2.5 | 4.5 | 1.7 | virginica |
| 6.7 | 2.5 | 5.8 | 1.8 | virginica |
| 5.7 | 2.5 | 5 | | |
| 6.3 | 2.5 | 5 | | |
| 5.7 | 2.6 | 3.5 | | |
| 5.5 | 2.6 | 4.4 | | |
| 5.8 | 2.6 | 4 | | |

UCI IRIS dataset

Pre-computation with the Apriori algorithm:
 Computing the set of 2 consecutive frequent modalities...FINISH!

Computing the number of rules to test...: 6 rules.

Sorting method : confidence sorting
☑ decreasing order
□ Exclude rules with consequent support (part B) exceeds (%) : 75.0

Rule 6/6 (total : 6)

6. SUPPORT = 41 (27.33 %) , CONFIDENCE = 82.0 % :

A {Class = Iris-versicolor} → B {petal_length in [3.3; 4.7], petal_width in [1.0; 1.5]}

petal_length : 23.73 % of [1.0, 6.9]
petal_width : 20.83 % of [0.1, 2.5]

| SUPPORTS : | |
|---|---|
| A and B | 41 (27.33 %) |
| A | 50 (33.33 %) |
| B | 41 (27.33 %) |
| A and (~B) | 9 (6.0 %) |
| (~A) and B | 0 (0.0 %) |
| (~A) and (~B) | 100 (66.67 %) |

| CONFIDENCES: | |
|---|---|
| A -> B | 82.0 % |
| (~A) -> B | 0.0 % |
| B -> A | 100.0 % |
| (~B) -> A | 8.26 % |
| A <-> B | 94.0 % |

# References

- R. Agrawal, T. Imielinski and A.N. Swami "Mining Association Rules between sets of items in large databases". SIGMOD 1993.
- R. Agrawal, R. Srikant "Fast algorithms for mining association rules" VLDB 1994.
- B. Goethals "Survey on Frequent Pattern Mining" Technical report, Helsinki Institute for Information Technology, 2003.
- S. Brin et al. "Beyond Market Baskets: Generalizing Association Rules to Correlations". SIGMOD 1997.
- R. Agrawal et al. "Mining association rules with item constraints". KDD 1997.
- A. Salleb et al. "QuantMiner: A Genetic Algorithm for Mining Quantitative Association Rules", IJCAI 2007.
- U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. "From Data Mining to Knowledge Discovery: An Overview". In Advances in Knowledge Discovery and Data Mining, 1996.