Data Mining

Prof. Dr. Stefan Kramer
Johannes Gutenberg-Universität Mainz

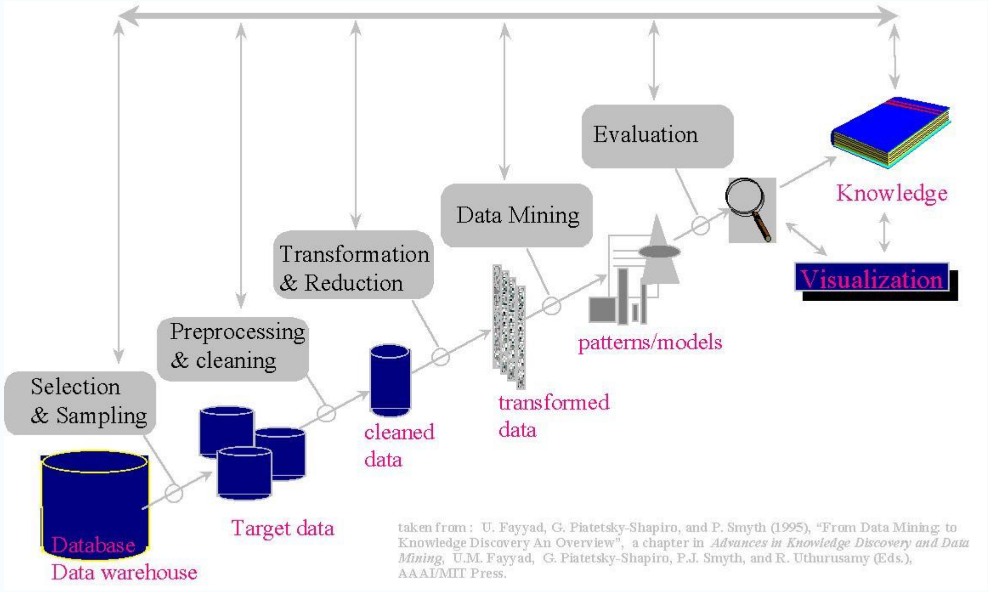
Outline

- A brief introduction to data mining and knowledge discovery in databases
- Organization of course
- A brief look at the WEKA workbench
- Itemsets and APriori

A Brief Introduction to Data Mining and KDD

Knowledge Discovery in Databases

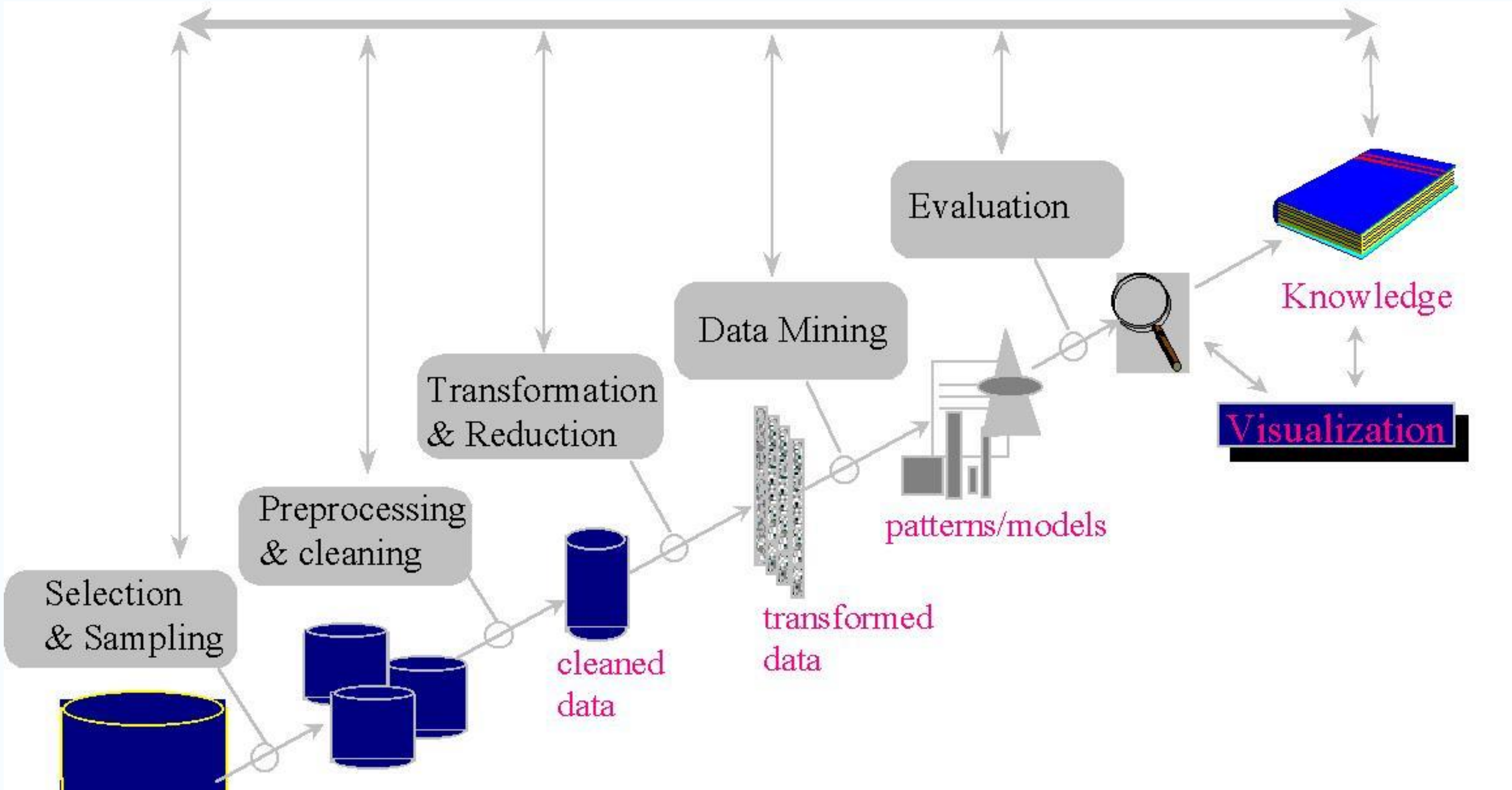
"… is the process of identifying valid, novel, potentially useful and ultimately understandable structure in data."



(Fayyad & Uthurusamy, 1996)

Structure = pattern or model

Knowledge Discovery in Databases and Data Mining



taken from: U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth (1995), "From Data Mining: to Knowledge Discovery An Overview", a chapter in *Advances in Knowledge Discovery and Data Mining*, U.M. Fayyad, G. Piatetsky-Shapiro, P.J. Smyth, and R. Uthurusamy (Eds.), AAAI/MIT Press.

Data Mining

- Knowledge Discovery in Databases (KDD) (Fayyad 96): “KDD is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.“
- Data Mining: **data analysis step within the KDD process**

Machine Learning

- Learning = improving with experience at some task
  - Improve on task T
  - With respect to performance measure P
  - Based on experience E.
- Learn to play checkers:
  - T: Play checkers
  - P: % of games won
  - E: opportunity to play against oneself

Machine Learning

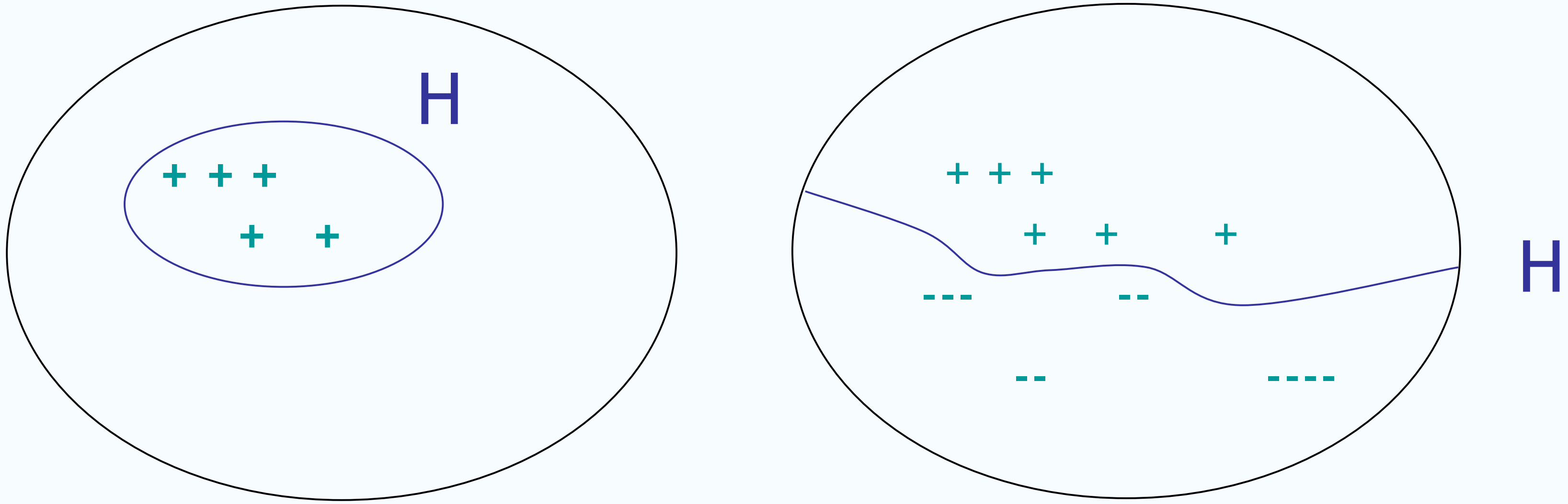
- Learning to classify examples (e.g., gene expression profiles into two subtypes):
  - T: Classifying examples
  - P: % of examples classified correctly
  - E: Training set of examples to learn from
- Machine learning algorithms (such as for classification) often used in Data Mining

Alternative Definitions…

Heikki Mannila:

- *"Knowledge Discovery in Databases is finding the joint probability distribution"*
- "Data Mining is the technology of fast counting"

Descriptive Data Mining, Predictive Data Mining



Organization of Course

Staff / Location / Time

- **Staff:** Andreas Karwwath, Jörg Wicker**Location:** Computerpool des Instituts für Informatik
- **Time:** Thursday, 14:15-17:45
- **Prerequisites (things that make life easier):** programming skills in Java, a scripting language (Python, Perl, …), graph theory, logic programming, basic probability theory, …

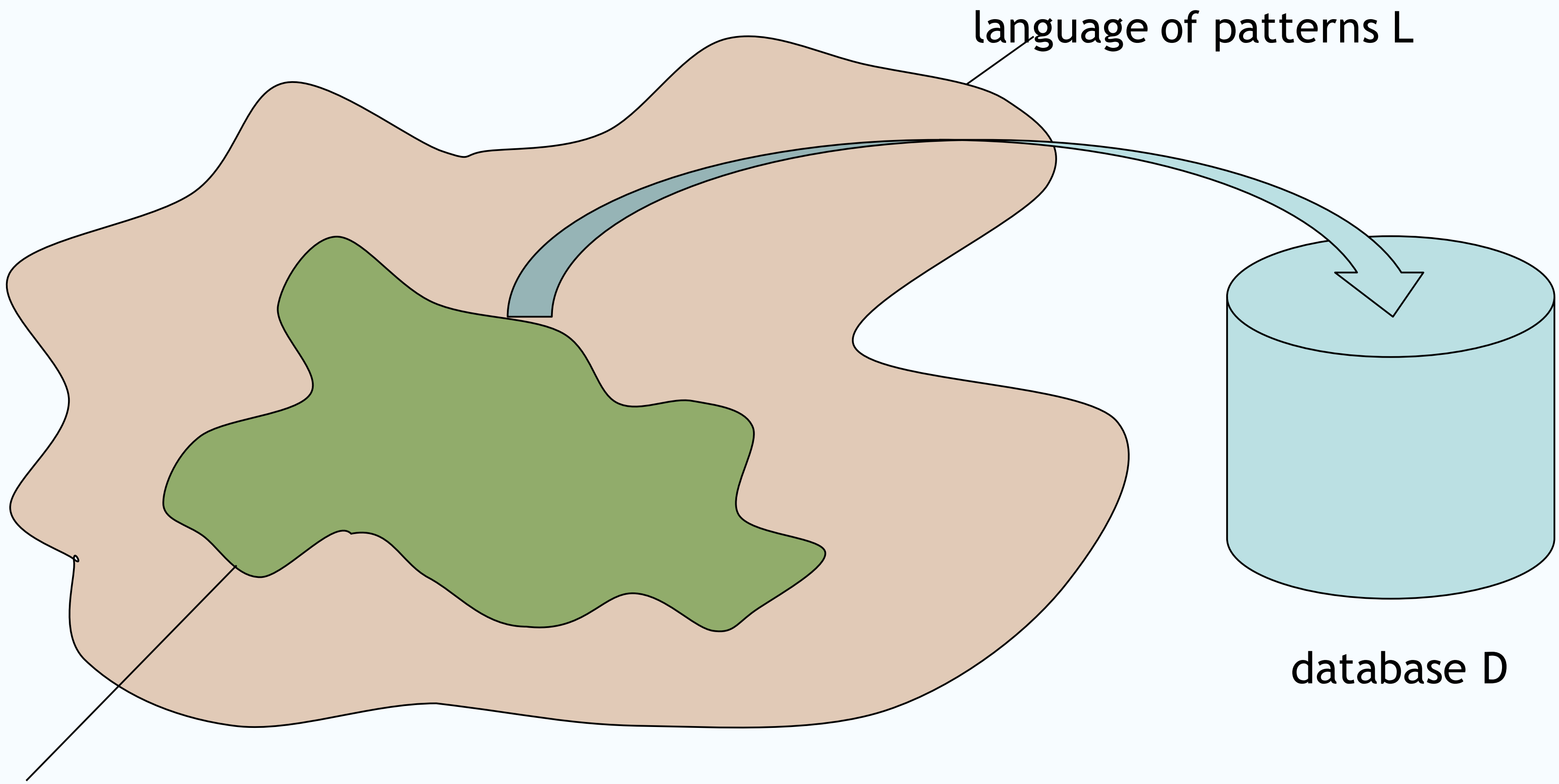
- **Format:**
  - *mix of traditional lectures/exercises/tutorials and flip teaching*: you are asked to **read** book chapter or articles/view video at home and come to lab **prepared**; we then have a few hours to talk and practice ☺
  - *why?*

Staff / Location / Time

- **Format (continued):**
  - Prüfungszulassung: in the sessions, you can gather points which are needed for the admission to the final exam. 50 % of the points and at least 25 % of the points of each session are required to gain admission.
  - groups of three students can work on projects
  - course will involve programming (prototyping), experimenting, …
  - sessions: recapitulating material for preparation, presentation of tasks/exercises, work on tasks/exercises, submission, (breaks)
  - *questions?*

- **Content: focus on 4 topics**

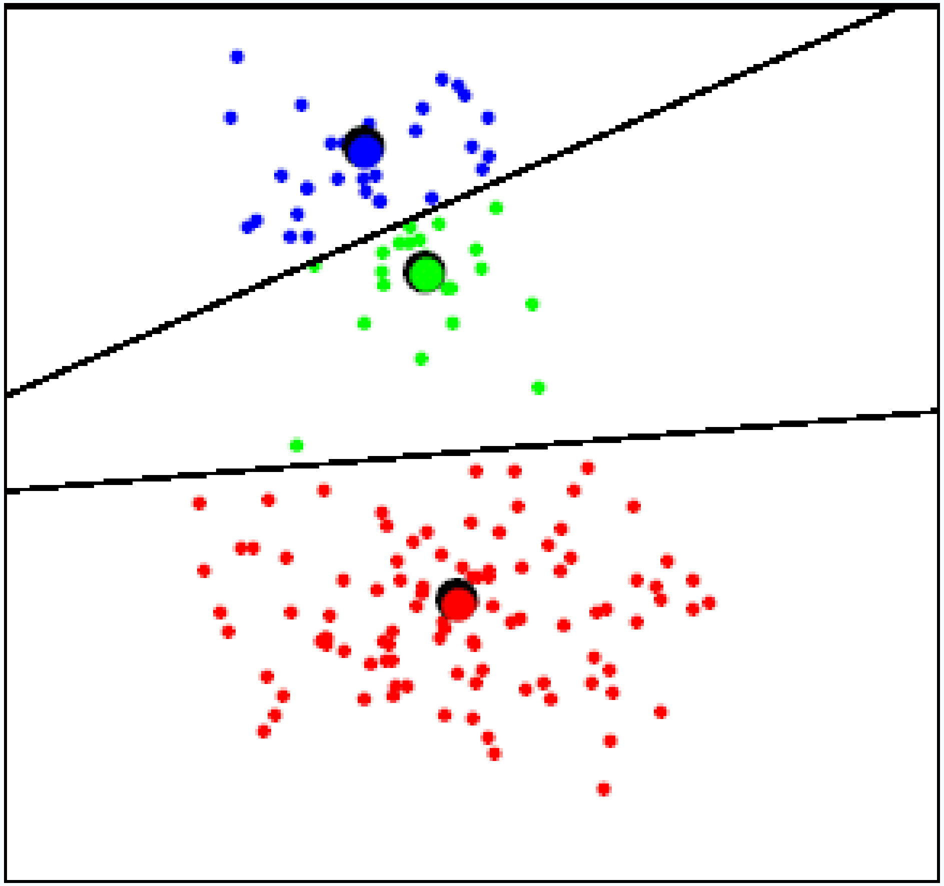
I. Pattern Mining



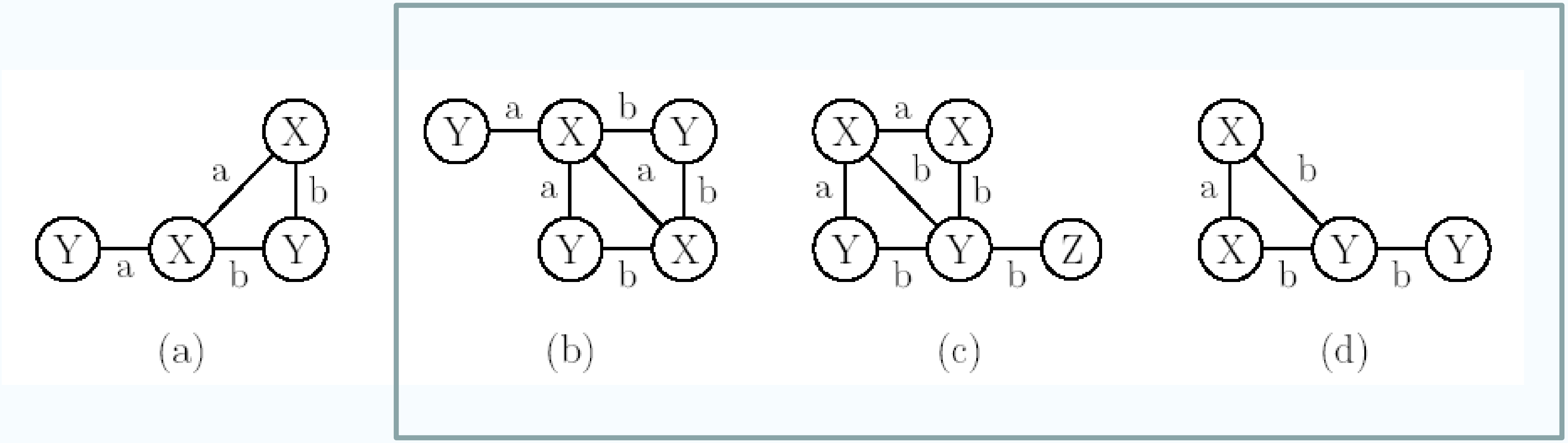
q(p, D) … interestingness predicate: a pattern p from L is interesting wrt. database D

what is interesting? *frequent*, *non-redundant*, *class correlated*, *structurally diverse*, …

II. Clustering

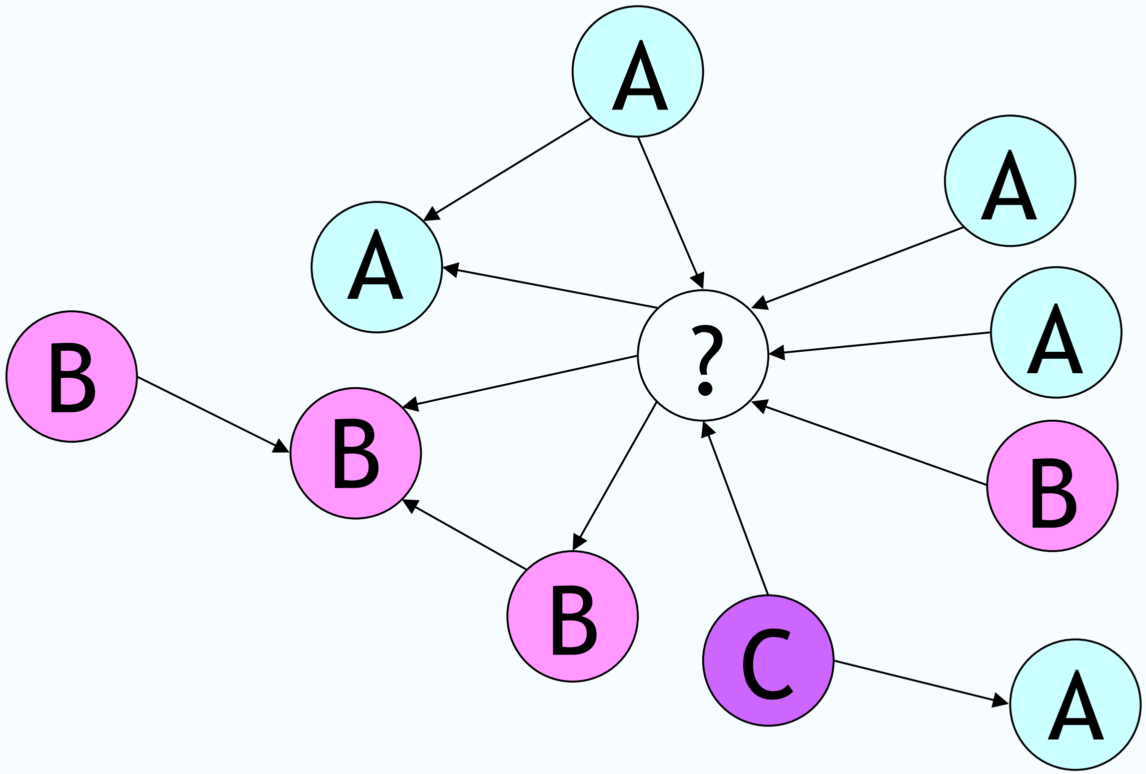


III. Graph Mining



- Graph database D (graphs (b) to (d))
- Find all subgraphs (patterns) that occur in at least two of the three graphs (examples)
- Example subgraph pattern p shown in (a)

III. Graph Mining



IV. Rule Learning

- ***First rule*:**
  if Astigmatism = Yes **and**
  Tear production rate = Normal **and**
  Spectacle prescription = Myope
  then
  Recommendation = Hard
- ***Second rule:***
  if Age = Young **and**
  Astigmatism = Yes **and**
  Tear production rate = Normal
  then
  Recommendation = Hard

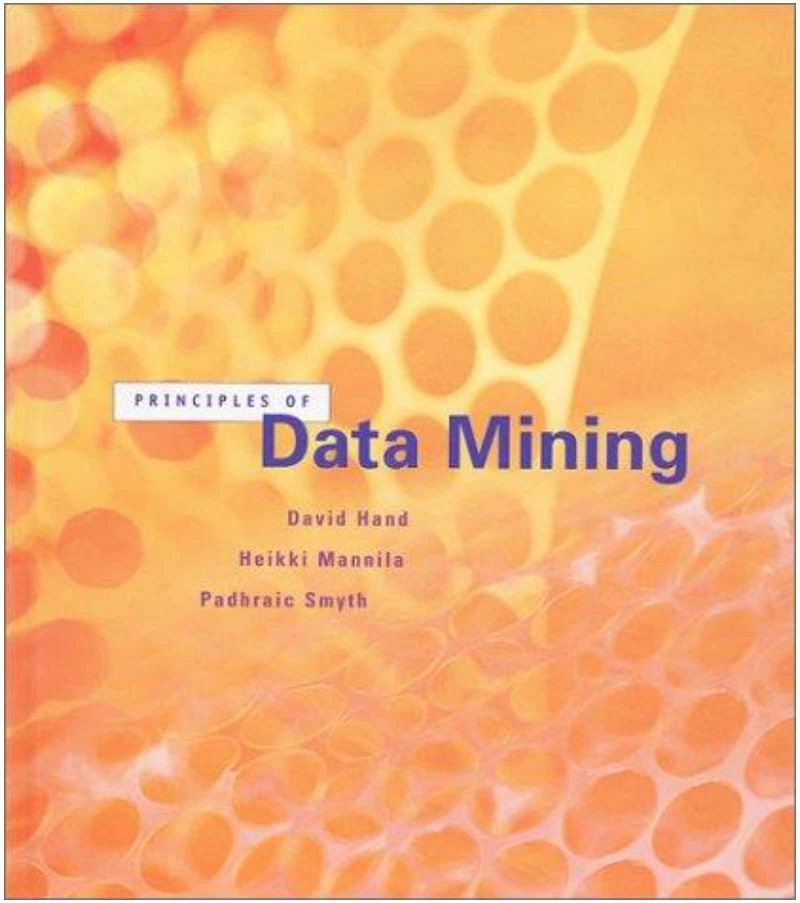
Timeline

27.10. Introduction

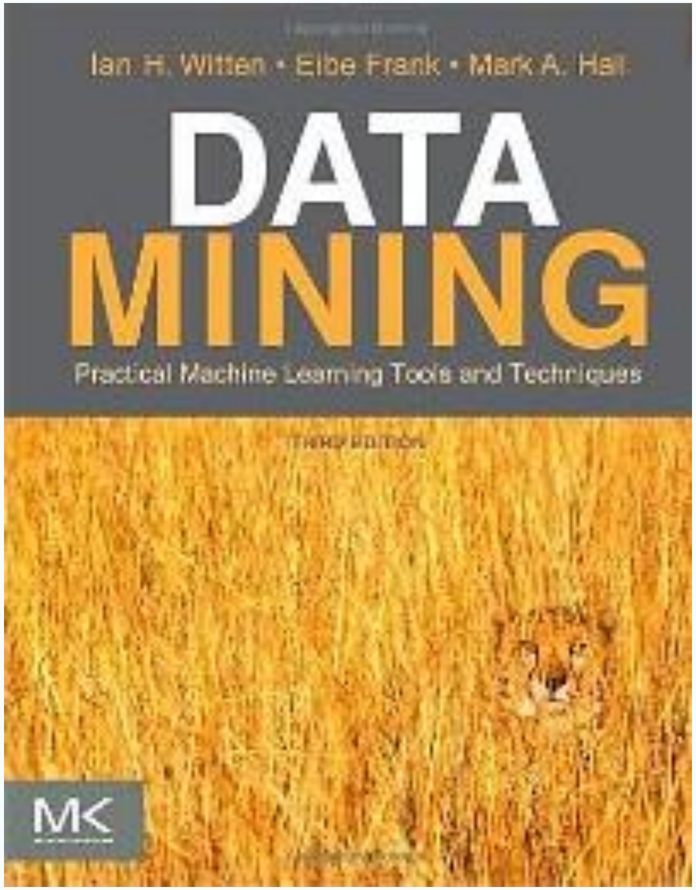
03.11.	Pattern Mining	15.12.	Graph Mining
10.11.	Pattern Mining	22.12.	Graph Mining
17.11.	Pattern Mining	12.01.	Graph Mining
24.11.	Clustering	19.01.	Rule Learning
01.12.	Clustering	26.01.	Rule Learning
08.12.	Clustering	02.02.	Rule Learning

22.03. Exam

Principles of Data Mining, David Hand, Heikki Mannila, Padhraic Smyth, MIT Press, 2001

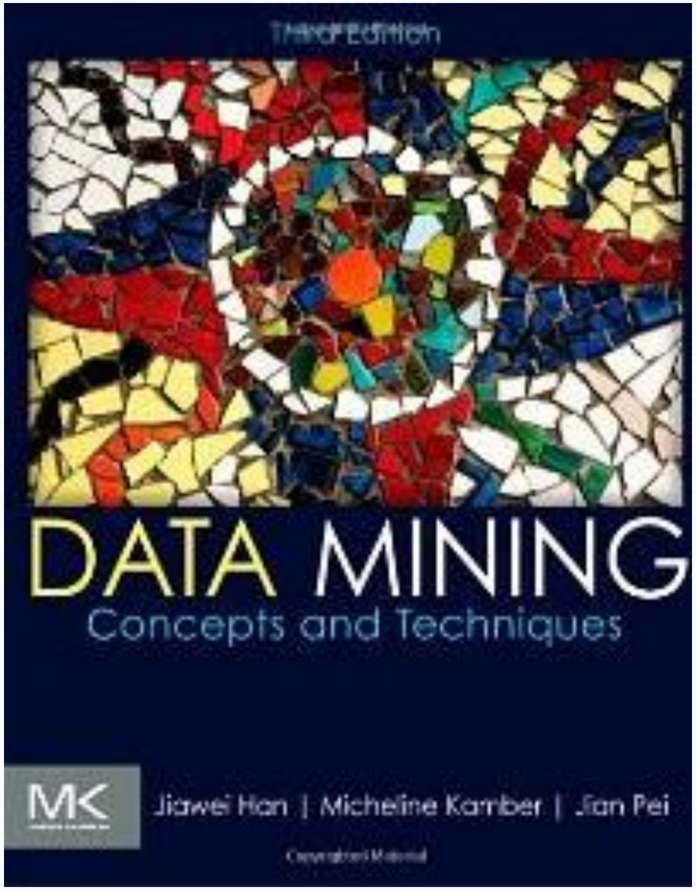


Data Mining: Practical Machine Learning Tools and Techniques with Java Implementions, Ian H. Witten, Eibe Frank, Morgan Kaufmann, 2011



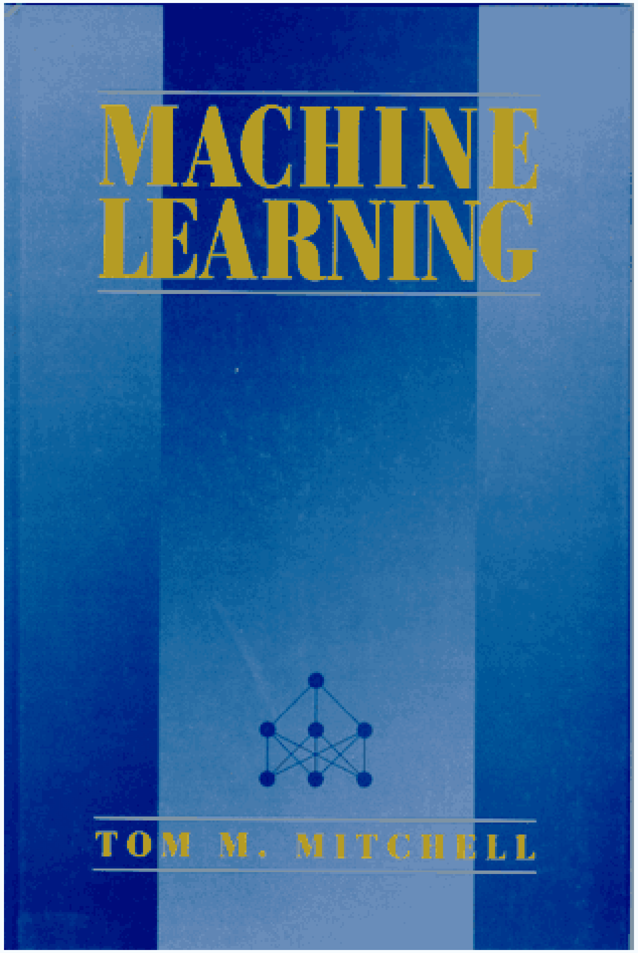
- **Third edition**
- **Parts on clustering and rule learning. Also relevant for exercises (WEKA workbench).**

Data Mining: Concepts and Techniques, Jiawei Han, Micheline Kamber, Jian Pei, Morgan Kaufmann, 2011



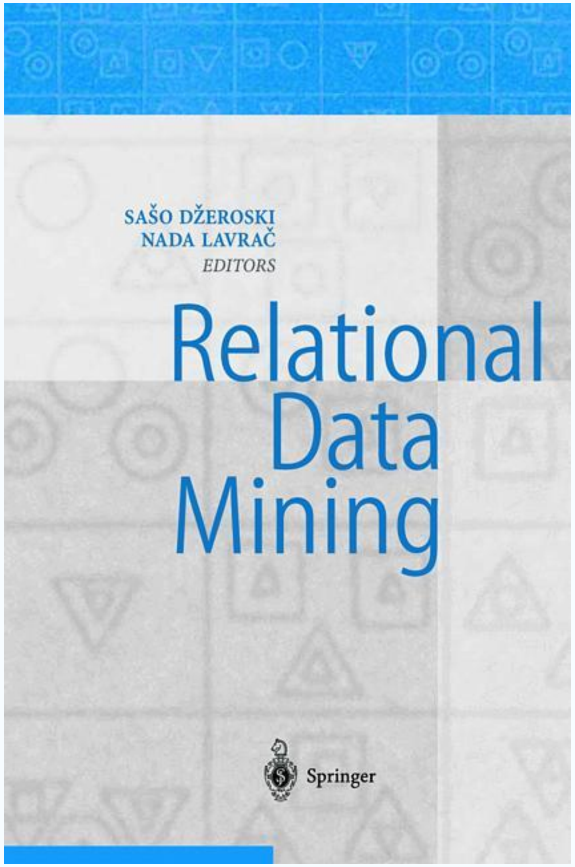
- Third edition
- Parts on pattern mining/association rule mining and clustering

Machine Learning, Tom Mitchell, McGraw Hill, 1997



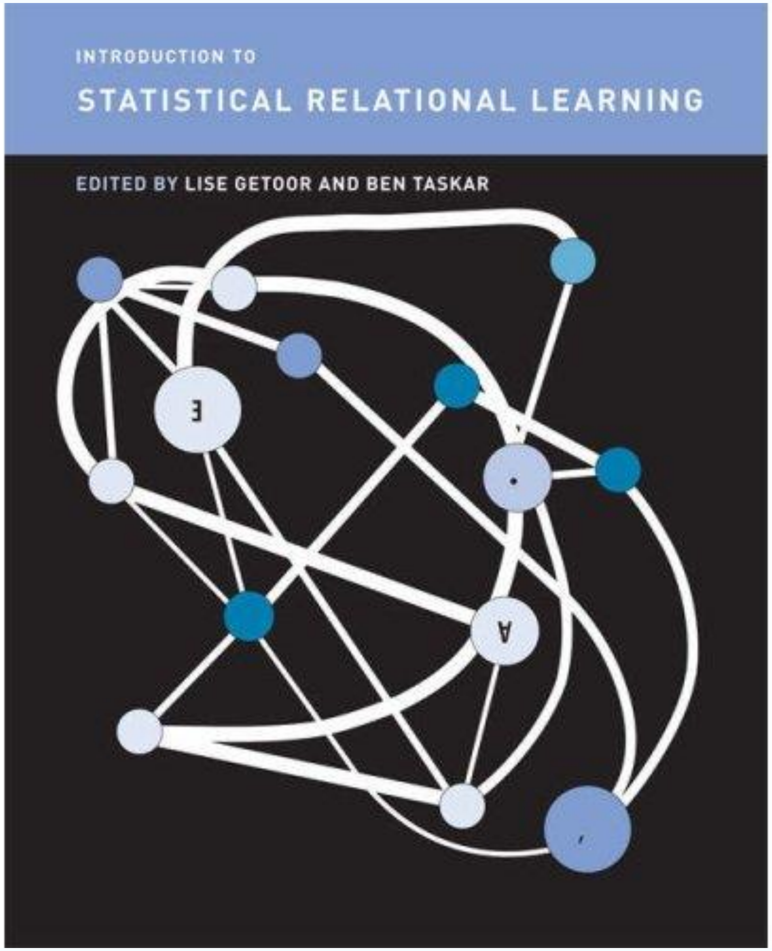
1. Introduction
2. Concept Learning and the General-to-Specific Ordering
3. Decision Tree Learning
4. Artificial Neural Networks
5. Evaluating Hypotheses
6. Bayesian Learning
7. Computational Learning Theory
8. Instance-Based Learning
9. Genetic Algorithms
10. Learning Sets of Rules
11. Analytical Learning
12. Combining Inductive and Analytical Learning
13. Reinforcement Learning

S. Dzeroski, N. Lavrac (eds.), Relational Data Mining, Springer, 2001



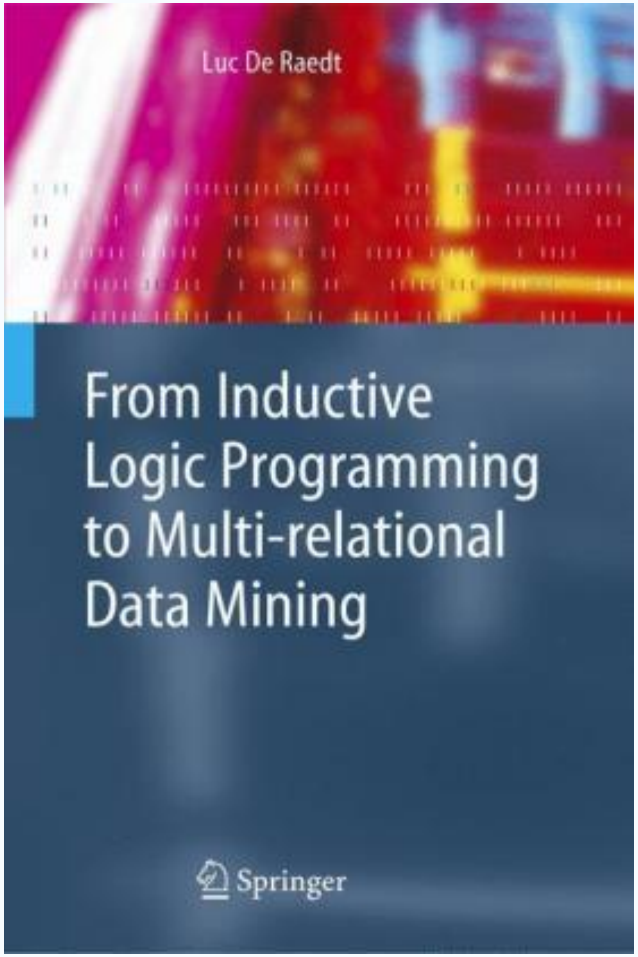
(1) introduction
(2) decision tree learning
(3) rule learning
(4) association rule mining
(5) distance-based learning
(6) subgroup discovery
(7) probabilistic graphical models …
in first-order logic/
for relational data
(8) propositionalization

Statistical Relational Learning



L. Getoor,
B. Taskar (eds.),
Statistical
Relational
Learning,
MIT Press, 2007.

ILP and MRDM Textbook



Luc De Raedt,
From Inductive
Logic Programming
to Multi-Relational
Data Mining,
Springer, 2008.

Tools

- **Generally:**
  - WEKA workbench
  - Stand-alone tools to be extended (e.g., implementing APriori, finding so-called frequent itemsets, borders, free and closed sets …)
- **Pattern mining:** WEKA, stand-alone tools, …
- **Clustering:** WEKA, R, …
- **Graph mining:** gSpan reimplementation, NetKit-SRL

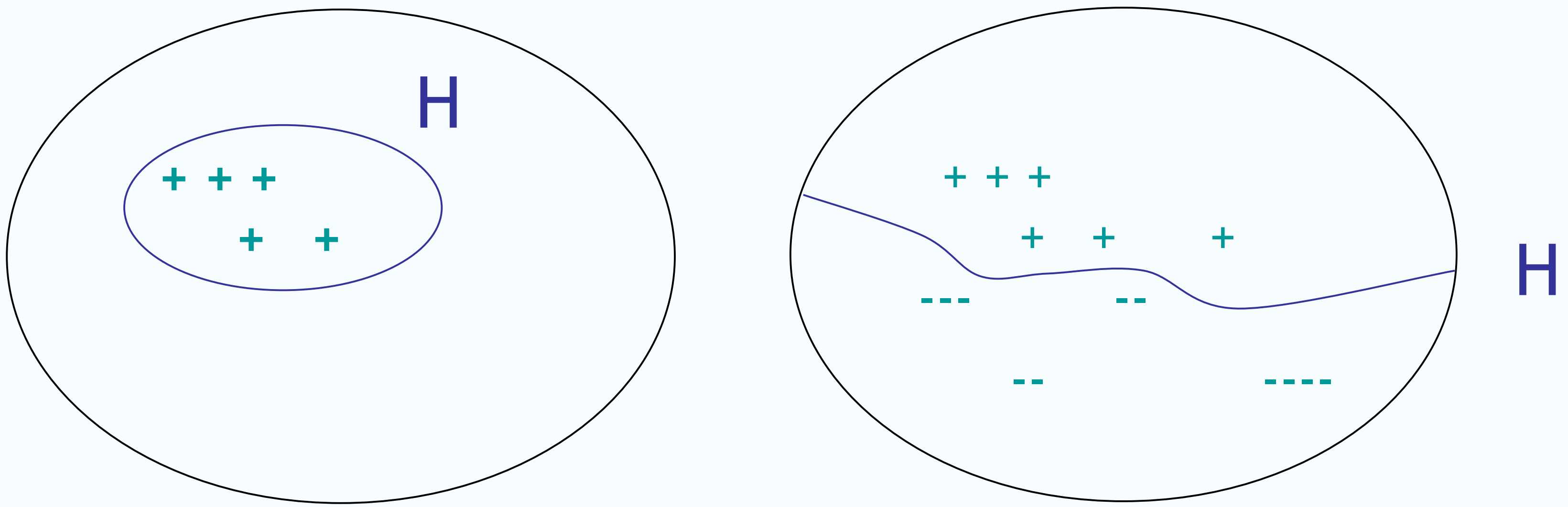
Tools

- **Rule learning:**
  - WEKA
  - FOIL (C implementation)
  - ProbLog

Questions?

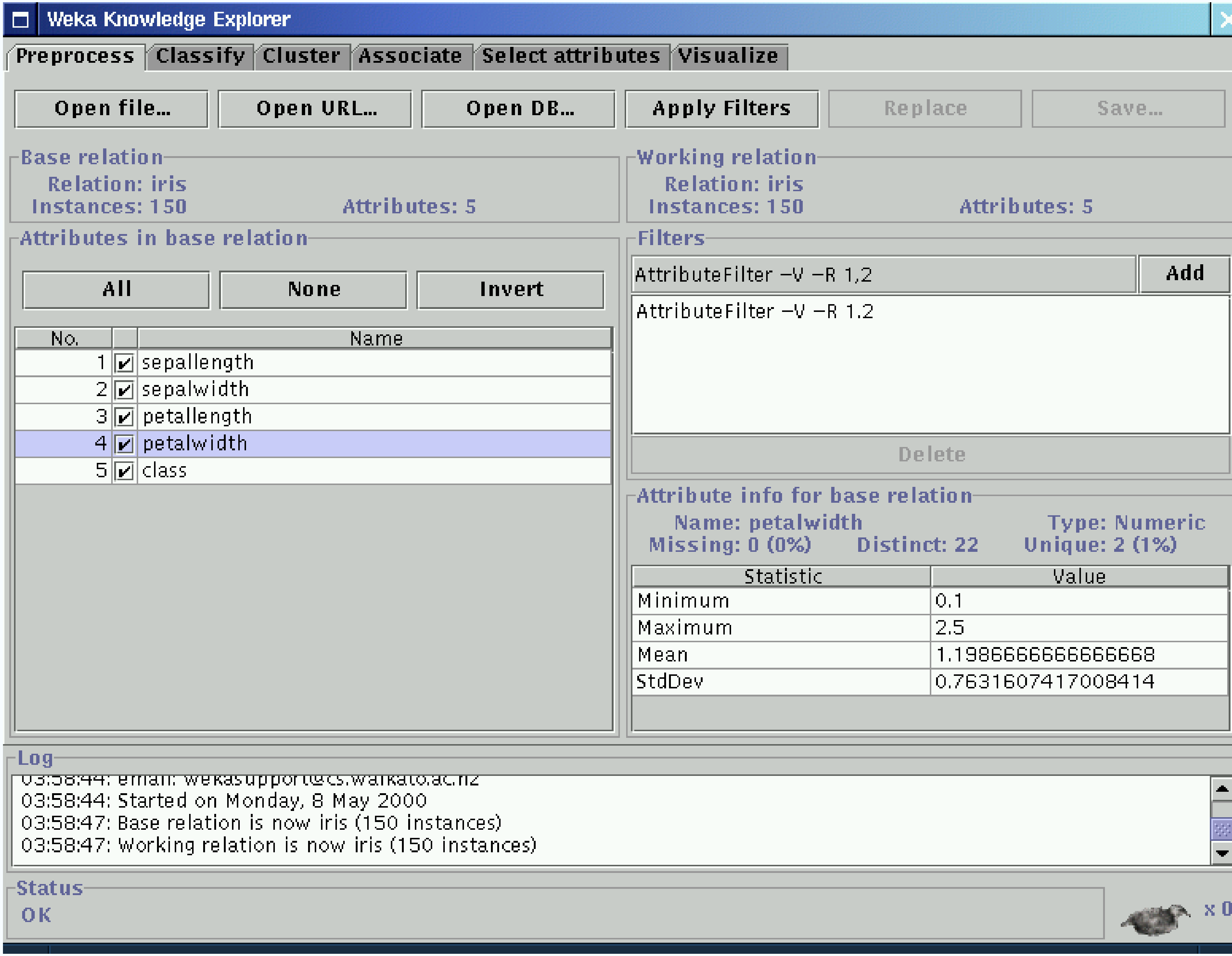
A Brief Look at the WEKA Workbench

Descriptive Data Mining, Predictive Data Mining

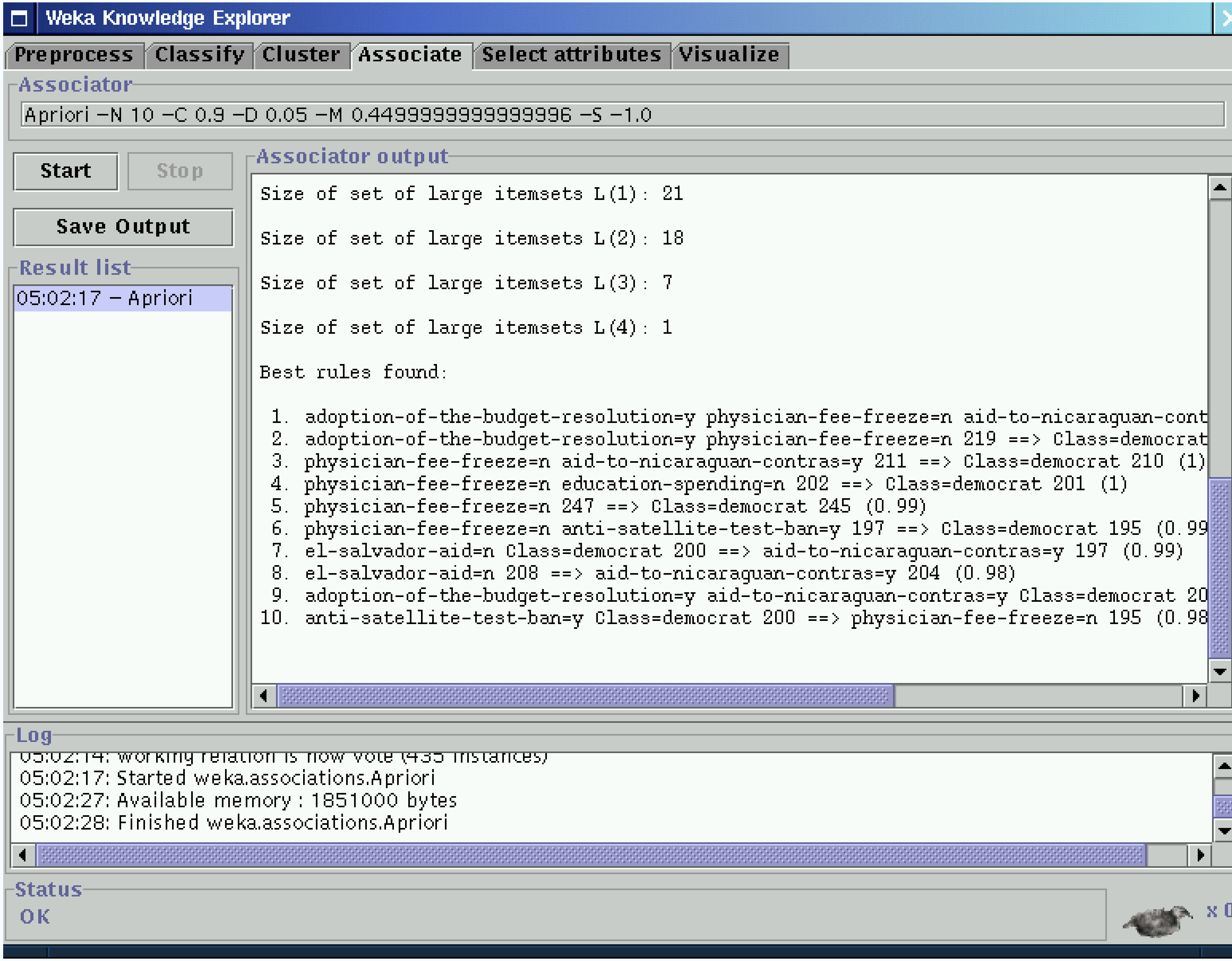


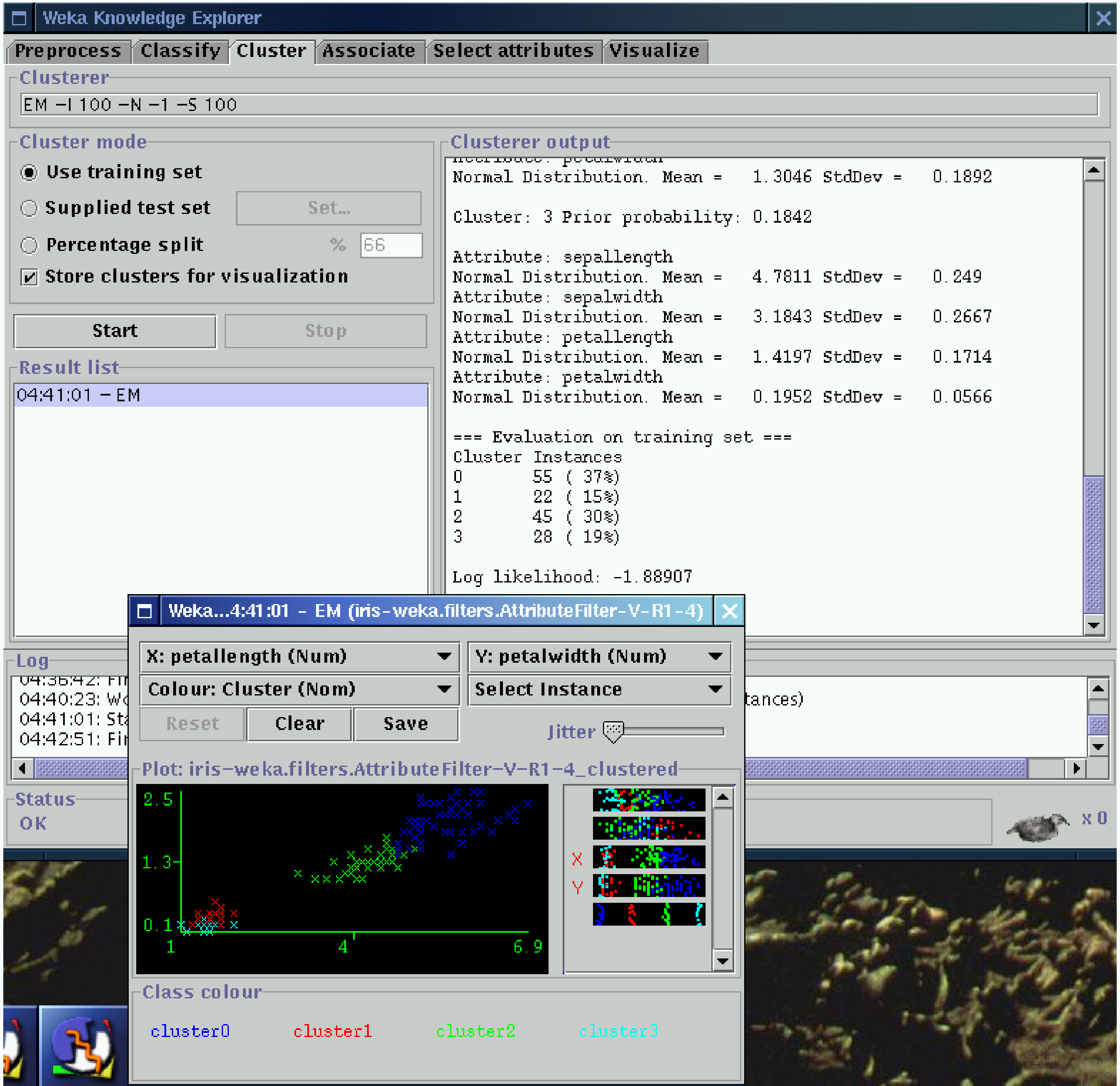
Input Format

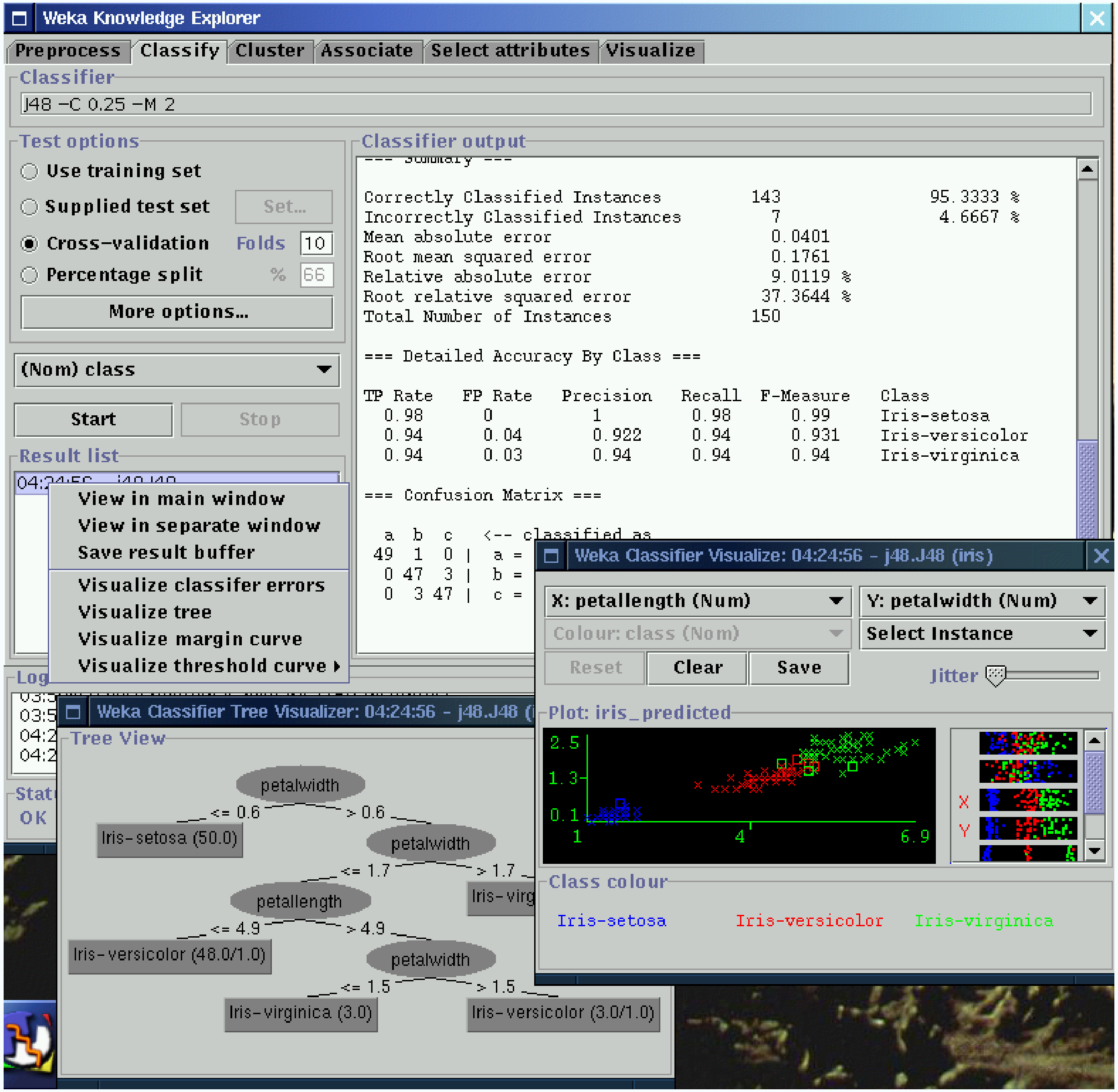
```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```



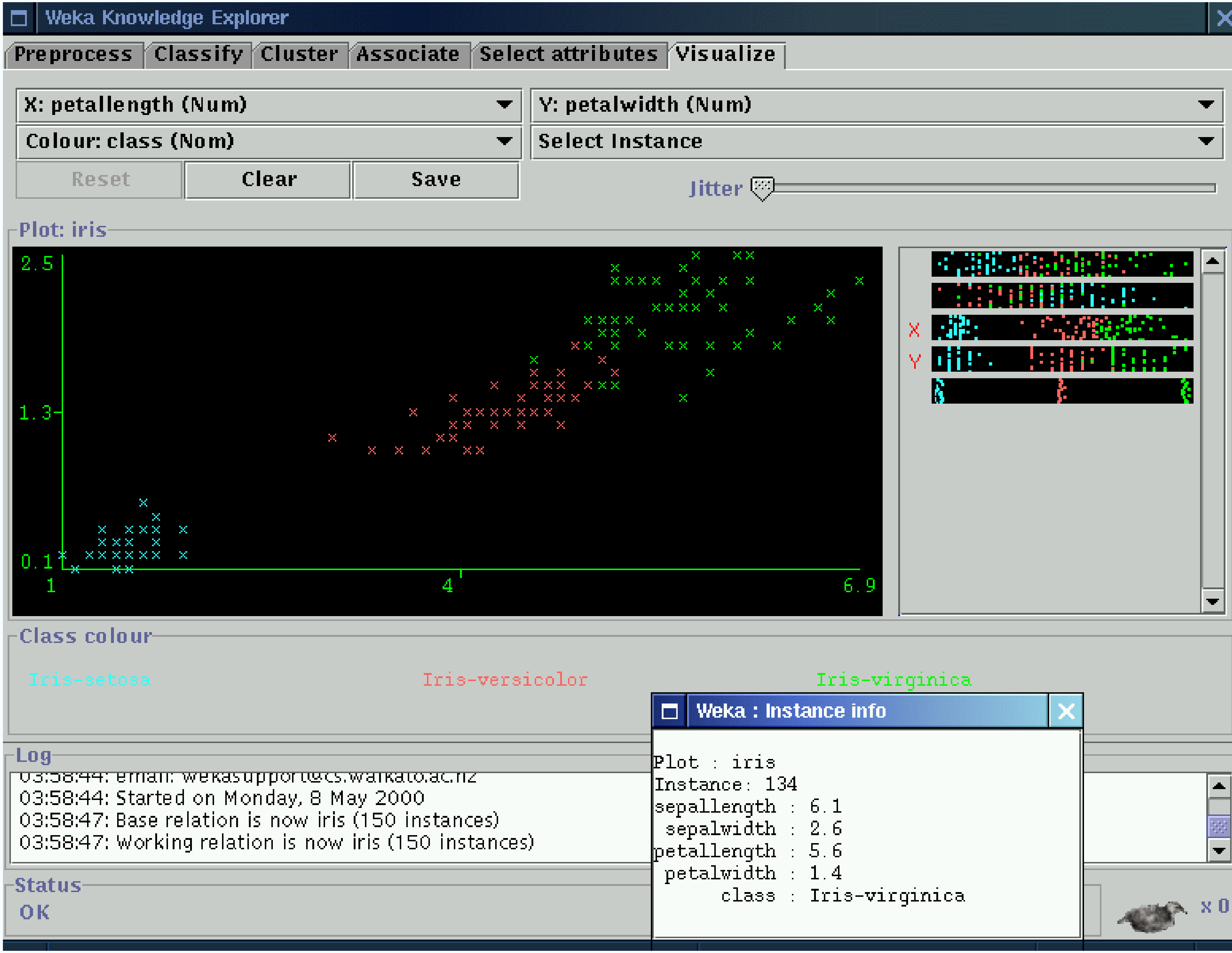
Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Apply Filters
Replace
Save...
Base relation
Relation: iris
Instances: 150
Attributes: 5
Working relation
Relation: iris
Instances: 150
Attributes: 5
Attributes in base relation
All
None
Invert
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Filters
AttributeFilter -V -R 1,2
Add
AttributeFilter -V -R 1,2
Delete
Attribute info for base relation
Name: petalwidth
Type: Numeric
Missing: 0 (0%)
Distinct: 22
Unique: 2 (1%)
Statistic
Value
Minimum 0.1
Maximum 2.5
Mean 1.1986666666666668
StdDev 0.7631607417008414
Log
03:58:44: Started on Monday, 8 May 2000
03:58:47: Base relation is now iris (150 instances)
03:58:47: Working relation is now iris (150 instances)
Status
OK
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Apriori -N 10 -C 0.9 -D 0.05 -M 0.44999999999999996 -S -1.0
Start
Stop
Save Output
Result list
05:02:17 - Apriori
Associator output
Size of set of large itemsets L(1): 21
Size of set of large itemsets L(2): 18
Size of set of large itemsets L(3): 7
Size of set of large itemsets L(4): 1
Best rules found:
1. adoption-of-the-budget-resolution=y physician-fee-freeze=n aid-to-nicaraguan-cont
2. adoption-of-the-budget-resolution=y physician-fee-freeze=n 219 ==> Class=democrat
3. physician-fee-freeze=n aid-to-nicaraguan-contras=y 211 ==> Class=democrat 210 (1)
4. physician-fee-freeze=n education-spending=n 202 ==> Class=democrat 201 (1)
5. physician-fee-freeze=n 247 ==> Class=democrat 245 (0.99)
6. physician-fee-freeze=n anti-satellite-test-ban=y 197 ==> Class=democrat 195 (0.99
7. el-salvador-aid=n Class=democrat 200 ==> aid-to-nicaraguan-contras=y 197 (0.99)
8. el-salvador-aid=n 208 ==> aid-to-nicaraguan-contras=y 204 (0.98)
9. adoption-of-the-budget-resolution=y aid-to-nicaraguan-contras=y Class=democrat 20
10. anti-satellite-test-ban=y Class=democrat 200 ==> physician-fee-freeze=n 195 (0.98
Log
05:02:17: Started weka.associations.Apriori
05:02:27: Available memory : 1851000 bytes
05:02:28: Finished weka.associations.Apriori
Status
OK
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
EM -I 100 -N -1 -S 100
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Store clusters for visualization
Start
Stop
Result list
04:41:01 - EM
Clusterer output
Normal Distribution. Mean = 1.3046 StdDev = 0.1892
Cluster: 3 Prior probability: 0.1842
Attribute: sepallength
Normal Distribution. Mean = 4.7811 StdDev = 0.249
Attribute: sepalwidth
Normal Distribution. Mean = 3.1843 StdDev = 0.2667
Attribute: petallength
Normal Distribution. Mean = 1.4197 StdDev = 0.1714
Attribute: petalwidth
Normal Distribution. Mean = 0.1952 StdDev = 0.0566
=== Evaluation on training set ===
Cluster Instances
0 55 (37%)
1 22 (15%)
2 45 (30%)
3 28 (19%)
Log likelihood: -1.88907
Log
04:40:23: W
04:41:01: St
04:42:51: Fi
tances)
Status
OK
x 0
Weka...4:41:01 - EM (iris-weka.filters.AttributeFilter-V-R1-4)
X: petallength (Num)
Y: petalwidth (Num)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Save
Jitter
Plot: iris-weka.filters.AttributeFilter-V-R1-4_clustered
2.5
1.3
0.1
1
4
6.9
X
Y
Class colour
cluster0
cluster1
cluster2
cluster3



Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list
View in main window
View in separate window
Save result buffer
Visualize classifier errors
Visualize tree
Visualize margin curve
Visualize threshold curve
Classifier output
Correctly Classified Instances 143 95.3333 %
Incorrectly Classified Instances 7 4.6667 %
Mean absolute error 0.0401
Root mean squared error 0.1761
Relative absolute error 9.0119 %
Root relative squared error 37.3644 %
Total Number of Instances 150
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure Class
0.98 0 1 0.98 0.99 Iris-setosa
0.94 0.04 0.922 0.94 0.931 Iris-versicolor
0.94 0.03 0.94 0.94 0.94 Iris-virginica
=== Confusion Matrix ===
a b c <-- classified as
49 1 0 | a =
0 47 3 | b =
0 3 47 | c =
Weka Classifier Visualize: 04:24:56 - j48.J48 (iris)
X: petallength (Num)
Y: petalwidth (Num)
Colour: class (Nom)
Select Instance
Reset
Clear
Save
Jitter
Plot: iris_predicted
2.5
1.3
0.1
1
4
6.9
X
Y
Class colour
Iris-setosa
Iris-versicolor
Iris-virginica
Weka Classifier Tree Visualizer: 04:24:56 - j48.J48
Tree View
petalwidth
<= 0.6
> 0.6
Iris-setosa (50.0)
petalwidth
<= 1.7
> 1.7
petallength
<= 4.9
> 4.9
Iris-versicolor (48.0/1.0)
petalwidth
<= 1.5
> 1.5
Iris-virginica (3.0)
Iris-versicolor (3.0/1.0)
Log
Status
OK


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
X: petallength (Num)
Y: petalwidth (Num)
Colour: class (Nom)
Select Instance
Reset
Clear
Save
Jitter
Plot: iris
2.5
1.3
0.1
1
4
6.9
X
Y
Class colour
Iris-setosa
Iris-versicolor
Iris-virginica
Log
03:58:44: email: wekasupport@cs.waikato.ac.nz
03:58:44: Started on Monday, 8 May 2000
03:58:47: Base relation is now iris (150 instances)
03:58:47: Working relation is now iris (150 instances)
Status
OK
x 0
Weka : Instance info
Plot : iris
Instance: 134
sepallength : 6.1
sepalwidth : 2.6
petallength : 5.6
petalwidth : 1.4
class : Iris-virginica

Itemsets and APriori

Example Microarray Data

	ARG1	ARG4	ARO3	….	LYS1
1	1	1	1	…	0
2	1	1	1	…	1
3	0	1	1	…	1
4	0	1	0	…	1
5	1	1	1	…	0
6	0	0	0	…	0
7	…	….	…	…	…

Before data mining step: data cleaning, sampling, discretization, feature selection, etc.

Another Representation

	ARG1	ARG4	ARO3	…	LYS1
1	1	1	1	…	0
2	1	1	1	…	1
3	0	1	1	…	1
4	0	1	0	…	1
5	1	1	1	…	0
6	0	0	0	…	0
7	…	….	…	…	…

D = {{ARG1, ARG4, ARO3},
{ARG1, ARG4, ARO3, LYS1},
{ARG4, ARO3, LYS1},
{ARG4, LYS1},
{ARG1, ARG4, ARO3},
{},
…}

Multiset of itemsets

Association Rule Mining

Table in relational database

	ARG1	ARG4	ARO3	...	LYS1
1	1	1	1	...	0
2	1	1	1	...	1
3	0	1	1	...	1
4	0	1	0	...	1
5	1	1	1	...	0
6	0	0	0	...	0
7	...	....	...	...	...

Association rules

"**IF** ARG1 and HIS5
THEN LYS1"

support: 54 %
confidence: 93 %

"**IF** YOL118C
THEN ARG1"

support: 53 %
confidence: 88 %

Frequent Itemsets and Association Rules

60 % of observations: ARO3 and LYS1 upregulated

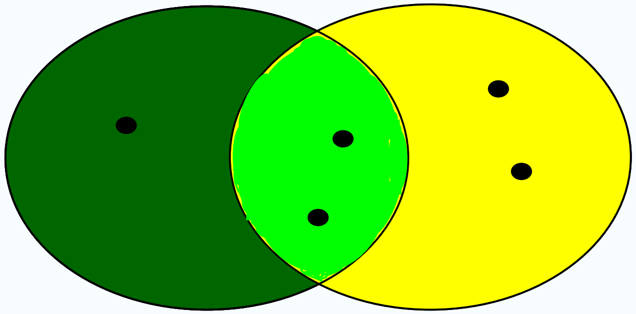
80 % of observations: ARG1 upregulated

40 % of observations: ARO3, LYS1 and ARG1 upregulated

"**IF** ARO3 and LYS1 **THEN** ARG1"

support: 40 %
confidence: 67 %

ARO3 and LYS1 vs. ARG1



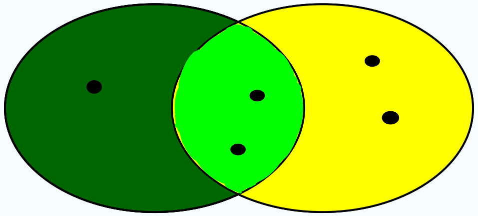
Two-Phased Algorithm

- *First phase*: find frequent itemsets (e.g., {ARO3, LYS1} , {ARG1} , {ARO3, LYS1, ARG1})
- *Second phase*: construct association rules (e.g., if {ARO3, LYS1} then {ARG1})

"**IF** ARO3 and LYS1 **THEN** ARG1"

support: 40 %
confidence: 67 %

{ARO3, LYS1} vs. {ARG1}

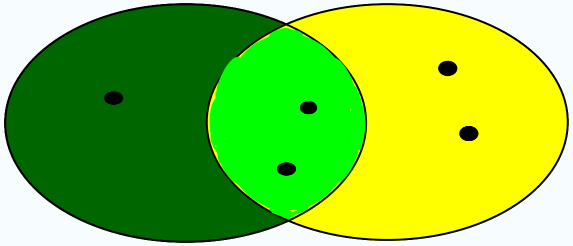


Support and Confidence

"**IF** ARO3 and LYS1
THEN ARG1"

support: 40 %
confidence: 67 %

{ARO3, LYS1} vs.
{ARG1}



"**IF** Y **THEN** X"

Support: $p(X,Y)$

Confidence: $p(X|Y) = \frac{p(X,Y)}{p(Y)}$

Frequent Pattern Discovery

Input:

- table D in relational database
- minimum support threshold: minSupport

Output:

- all patterns (here: itemsets) p for which freq(p, D) $\geq$ minSupport

How?

APriori Algorithm (Agrawal et al., 1993)

$i := 1$
$C_i := \{\{A\} | A \text{ is an item}\}$
while $C_i \neq \{\}$ **do**
    % candidate testing (database scan)
    for each set in C_i test whether it is frequent
    let F_i be the collection of frequent sets from C_i
    % candidate formation
    let C_{i+1} be those sets of size i+1 such that all subsets are in F_i (frequent)
    $i := i + 1$
return $\cup F_j$

Candidate Formation

- By *joining*: union of pairs of frequent itemsets from the previous level
- e.g., {A,B} and {B,C} gives {A,B,C}
- However, {A, C} might still be infrequent
- Thus, additional pruning step checking whether all subsets are known to be frequent

Apriori - Example

min_support = 2

Data

Tid	Items
1	A, C, D
2	B, C, E
3	A, B, C, E
4	B, E

1st scan $\longrightarrow$ C_1

Itemset	support
$\{A\}$	2
$\{B\}$	3
$\{C\}$	3
$\{D\}$	1
$\{E\}$	3

$\longrightarrow L_1$

Itemset	support
$\{A\}$	2
$\{B\}$	3
$\{C\}$	3
$\{E\}$	3

$\longrightarrow C_2$

Itemset
$\{A, B\}$
$\{A, C\}$
$\{A, E\}$
$\{B, C\}$
$\{B, E\}$
$\{C, E\}$

2nd scan $\downarrow$

C_2

Itemset	support
$\{A, B\}$	1
$\{A, C\}$	2
$\{A, E\}$	1
$\{B, C\}$	2
$\{B, E\}$	3
$\{C, E\}$	2

$\longleftarrow L_2$

Itemset	support
$\{A, C\}$	2
$\{B, C\}$	2
$\{B, E\}$	3
$\{C, E\}$	2

$\longleftarrow C_3$

Itemset
$\{B, C, E\}$

3rd scan $\longleftarrow$ L_3

Itemset	support
$\{B, C, E\}$	2

Main Ideas of APriori

- Each iteration consists of two phases
  - candidate formation
  - candidate testing (database scan)
- Minimize database scans

  for each tuple *t* do

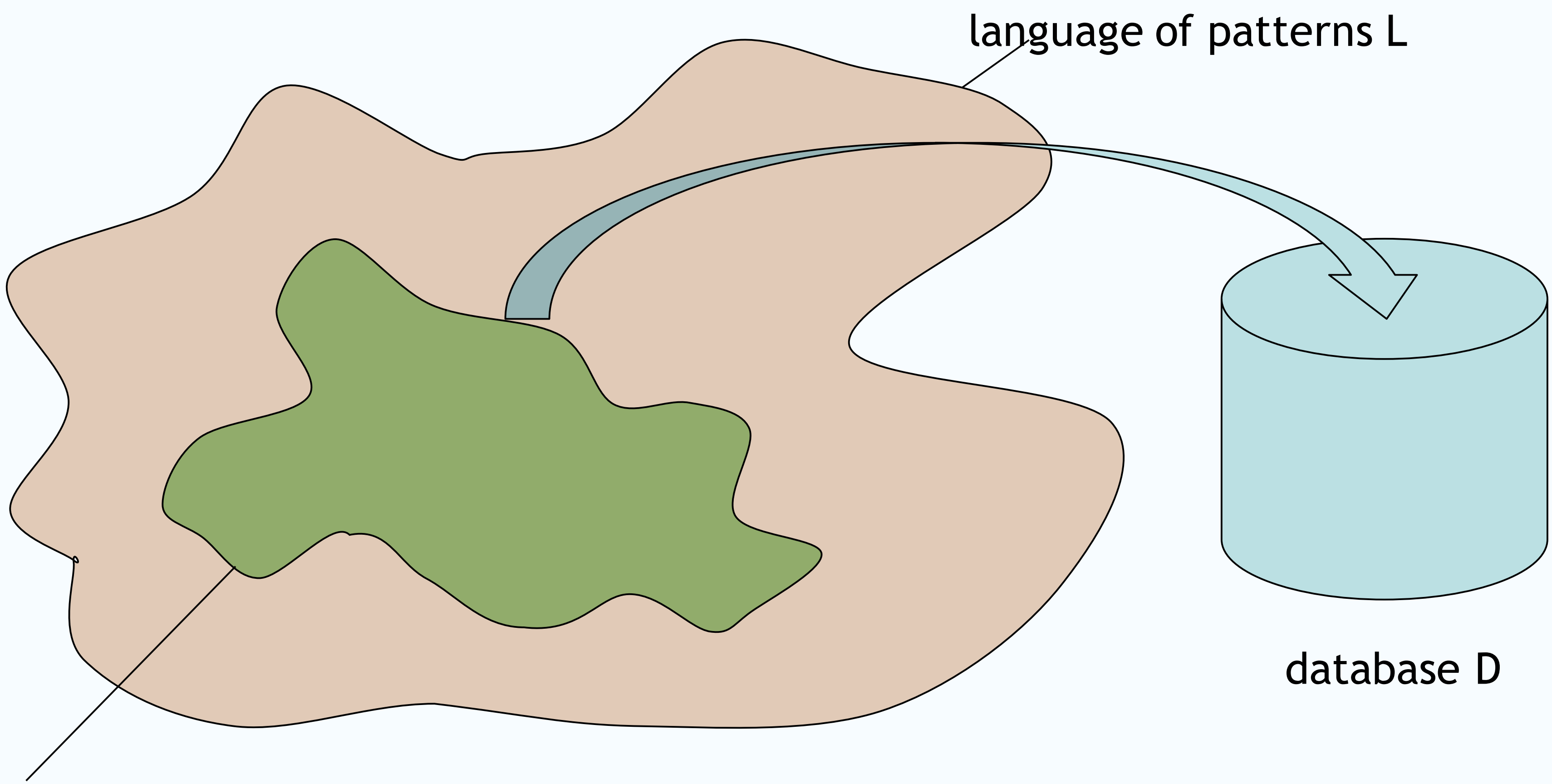
      for each candidate itemset *i* do

          …
- Avoid unnecessary tests on the database (test only those patterns that can, knowing the previous levels, be frequent)

Patterns (Itemsets) and (Association) Rules

- From frequent itemsets c and $c \cup \{i\}$ derive if c then {i}
- Start with the maximally specific frequent itemsets
- Variants possible: only one item in the RHS (very common assumption), only one item in the LHS (not very common)
- Generally: patterns and rules
  frequent patterns p, q such that $p \leq q$
  if p then q (with some confidence)

Formalization of Data Mining



q(p, D) … interestingness predicate: a pattern p from L is interesting wrt. database D

what is interesting? *frequent*, *non-redundant*, *class correlated*, *structurally diverse*, …

Formalization of Data Mining

- Simple formalization/definition of data mining (Mannila & Toivonen, 1997)
- Language L of patterns p
- Database D
- Interestingness predicate q
- Find a theory of the data:
  $Th(L, D, q) = \{p \in L \mid q(p,D) \text{ is true}\}$

Assignment

- Read J. Han *et al.*, chapter 6: Mining Frequent Patterns, Associations, and Correlations: Basic Concepts and Methods, 6.1-6.2.3