Induction of Rules

Lecturer: JERZY STEFANOWSKI
Institute of Computing Sciences
Poznan University of Technology
Poznan, Poland

Lecture 6
SE Master Course 2008/2009
Update 2010

Źródła

- Wykład częściowo oparty na moim wykładzie szkoleniowym dla COST Action Spring School on Data Mining and MCDA – Troina 2008 oraz wcześniejszych wystąpieniach konferencyjnych.
- Proszę także przeczytać stosowane rozdziały z mojej rozprawy habilitacyjnej – dostępna na mojej stronie www.cs.put.poznan.pl/jstefanowski.

Outline of this lecture

1. Rule representation
2. Basic algorithms for rule induction – idea of „Sequential covering” search strategy
3. MODLEM $\rightarrow$ exemplary algorithm for inducing a minimal set of rules.
4. Classification strategies
5. Descriptive properties of rules and Explore algorithm $\rightarrow$ discovering a richer set of rules
6. Logical relations (ILP) and rule induction
7. Final remarks

Rules - preliminaries

- **Rules** → the most popular symbolic representation of knowledge derived from data;
  - Natural and easy form of representation → possible inspection by human and their interpretation.
  - More comprehensive than any other knowledge representation!
- Standard form of rules
  IF *Conditions* THEN *Class*
- Other forms: Class IF Conditions; Conditions → Class

Example: The set of decision rules induced from PlaySport:

if outlook = overcast **then** Play = yes

if temperature = mild **and** humidity = normal **then** Play = yes

if outlook = rainy **and** windy = FALSE **then** Play = yes

if humidity = normal **and** windy = FALSE **then** Play = yes

if outlook = sunny **and** humidity = high **then** Play = no

if outlook = rainy **and** windy = TRUE **then** Play = no

Rules – more formal notations

- A rule corresponding to class K_j is represented as

if P then Q

where $P = w_1$ and w_2 and … and w_m is a condition part and Q is a decision part (object x satisfying P is assigned to class K_j)

- Elementary condition w_i (a *rel* v), where $a \in A$ and v is its value (or a set of values) and *rel* stands for an operator as $=, <, \leq, \geq, >$.
- [*P*] is a cover of a condition part of a rule → a subset of examples satisfying *P*.
  - *if* ($a2$ = small) *and* ($a3 \leq 2$) *then* (d = C1) {$x1$,$x7$}
- A rule is certain / discriminant in *DT* iff $[P]=\bigcap [w_i] \subseteq [K_j]$, otherwise ($P \cap K_j \neq \emptyset$) the rule is partly discriminating.

An example of rules induced from data table

Minimal set of rules

- *if* (*a*2 = s) ∧ (*a*3 ≤ 2) *then* (*d* = C1) {*x*1,*x*7}
- *if* (*a*2 = n) ∧ (*a*4 = c) *then* (*d* = C1) {*x*3,*x*4}
- *if* (*a*2 = w) *then* (*d* = C2) {*x*2,*x*6}
- *if* (*a*1 = f) ∧ (*a*4 = a) *then* (*d* = C2) {*x*5,*x*8}

Partly discriminating rule:

- *if* (*a1*=m) *then* (*d*=C1) {*x1*,*x3*,*x7* | *x6*} 3/4

id.	a_1	a_2	a_3	a_4	d
x_1	m	s	1	a	C1
x_2	f	w	1	b	C2
x_3	m	n	3	c	C1
x_4	f	n	2	c	C1
x_5	f	n	2	a	C2
x_6	m	w	2	c	C2
x_7	m	s	2	b	C1
x_8	f	s	3	a	C2

Polish contribution – prof. Ryszard Michalski

- Father of Machine Learning and rule induction

Ryszard S. Michalski
(1937 - 2007)

PRC Chaired Professor of Computational Sciences
and Health Informatics
Director of the Center for Discovery Science and Health
Informatics

George Mason University

Interests

Biosketch

Publications

Teaching

Research

Solving problems

Machine Learning and Inference Laboratory

School of Computational Sciences

George Mason University

This page has been visited 15691 *since January 1, 1999*

6/27/06 R.S. Michalski gives a banquet address at the International Conference on Machine Learning, to celebrate the return of the conference to Carnegie-Mell
after 26 years since the very first conference was organized there by Carbonell, Michalski and Mitchell

Articles in *Mason Gazette:*

7/31/07 New Center to Help Investigators Discover New Knowledge in Medical Databases
3/12/03 University Wins 10th Patent for Machine Learning Invention
11/19/02 Spotlight on Research: Grants Support Machine Learning and Inference Research
7/27/00 Michalski Receives Prestigious Science Honor

Interests

Research areas:
Machine Learning, Data Mining and Knowledge Discovery, Inductive Databases and Knowledge Scouts, Non-Darwinian Evolutionary Computation and Plausib
applications of these areas to Bioinformatics, Medicine, User Modeling, Intrusion Detection, and Very Complex System Design.

Rules – more preliminaries

- A set of rules – a disjunctive set of conjunctive rules.
- Also DNF form:
  - *Class* IF *Cond*_1 OR *Cond*_2 OR … *Cond*_m
- Various types of rules in data mining
  - Decision / classification rules
  - Association rules
  - Logic formulas (ILP)
  - Other → action rules, …
- **MCDA** → attributes with some additional preferential information and ordinal classes.

Why Decision Rules?

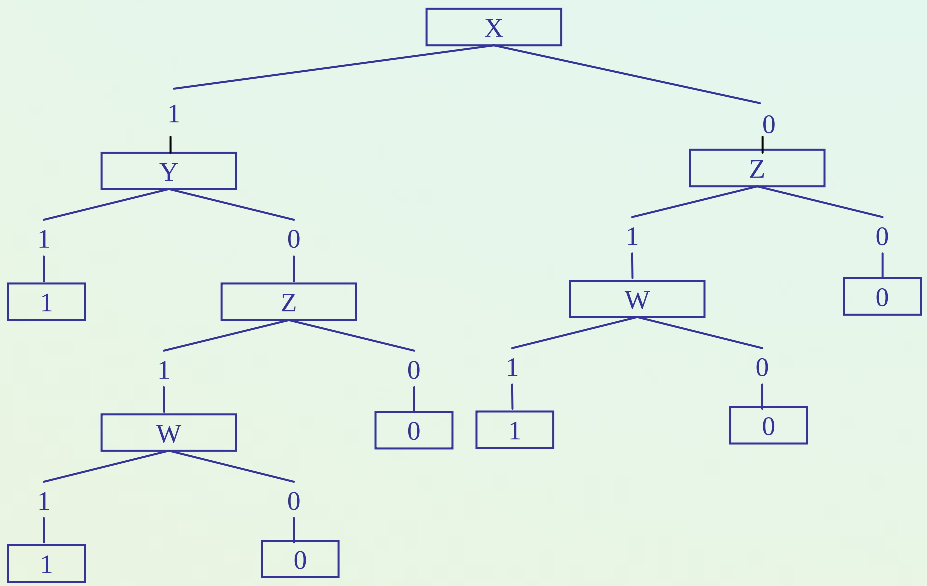
- Decision rules are more compact.
- Decision rules are more understandable and natural for human.
- Better for descriptive perspective in data mining.
- Can be nicely combined with background knowledge and more advanced operations, …

Example: Let $X \in \{0,1\}$, $Y \in \{0,1\}$, $Z \in \{0,1\}$, $W \in \{0,1\}$. The rules are:

if X=1 and Y=1 **then** 1

if Z=1 and W=1 **then** 1

Otherwise 0;



How to learn decision rules?

- Typical algorithms based on the scheme of a sequential covering and heuristically generate a minimal set of rule covering examples:
  - see, e.g., AQ, CN2, LEM, PRISM, MODLEM, Other ideas – PVM, R1 and RIPPER).
- Other approaches to induce „richer” sets of rules:
  - Satisfying some requirements (Explore, BRUTE, or modification of association rules, „Apriori-like”).
  - Based on local „reducts” → boolean reasoning or LDA.
- Specific optimization, eg. genetic approaches.
- Transformations of other representations:
  - Trees → rules.
  - Construction of (fuzzy) rules from ANN.

Covering algorithms

- A strategy for generating a rule set directly from data:
  - for each class in turn find a rule set that covers all examples in it (excluding examples not in the class).
- The main procedure is iteratively repeated for each class.
  - Positive examples from this class vs. negative examples.
- This approach is called a *covering* approach because at each stage a rule is identified that covers some of the examples (then these examples are skipped from consideration for the next rules).
- A sequential approach.
  - For a given class it conducts in a stepwise way a general to specific search for the best rules (learn-one-rule) guided by the evaluation measures.

General schema of inducing minimal set of rules

- The procedure conducts a general to specific (greedy) search for the best rules (**learn-one-rule**) guided by the evaluation measures.
- At each stage add to the current condition part next elementary tests that optimize possible rule's evaluation (no backtracking).

```
Procedure Sequential covering (K_j Class; A attributes; E examples,
τ - acceptance threshold);
begin
    R := ∅;       {set of induced rules}
    r := learn-one-rule(Y_j Class; A attributes; E examples)
    while  evaluate(r,E) > τ  do
    begin
      R := R ∪ r;
      E := E \ [R];        {remove positive examples covered by R}
      r := learn-one-rule(Kj Class; A attributes; E examples);
    end;
return R
end.
```

The contact lenses data

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

Inducing rules by PRISM from contact lens data

- Rule we seek:

```
If ?
 then recommendation = hard
```

- Possible conditions:

Age = Young	2/8
Age = Pre-presbyopic	1/8
Age = Presbyopic	1/8
Spectacle prescription = Myope	3/12
Spectacle prescription = Hypermetrope	1/12
Astigmatism = no	0/12
Astigmatism = yes	4/12
Tear production rate = Reduced	0/12
Tear production rate = Normal	4/12

PRISM - Evaluation of candidates for a rule:

High accuracy P(K|R);

High coverage |[P]I

ACK: slides coming from witten&eibe WEKA

Modified candidate for a rule and covered data

- Condition part of the rule with the best elementary condition added:

```
If astigmatism = yes
    then recommendation = hard
```

- Examples covered by the first condition part:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

Further specialization of conditions

- Current state:

```
If astigmatism = yes
     and ?
   then recommendation = hard
```

- Possible conditions:

Age = Young	2/4
Age = Pre-presbyopic	1/4
Age = Presbyopic	1/4
Spectacle prescription = Myope	3/6
Spectacle prescription = Hypermetrope	1/6
Tear production rate = Reduced	0/6
Tear production rate = Normal	4/6

Two conditions in the rule

- The rule with the next best condition added:

```
If astigmatism = yes
    and tear production rate = normal
  then recommendation = hard
```

- Examples covered by modified rule:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Normal	None

Further specialization of the candidate for a rule

- The current state:

```
If astigmatism = yes
     and tear production rate = normal
     and ?
   then recommendation = hard
```

- Possible conditions:

Age = Young	2/2
Age = Pre-presbyopic	1/2
Age = Presbyopic	1/2
Spectacle prescription = Myope	3/3
Spectacle prescription = Hypermetrope	1/3

- Tie between the first and the fourth test
    - We choose the one with greater coverage

The result for class „hard”

- Final rule:

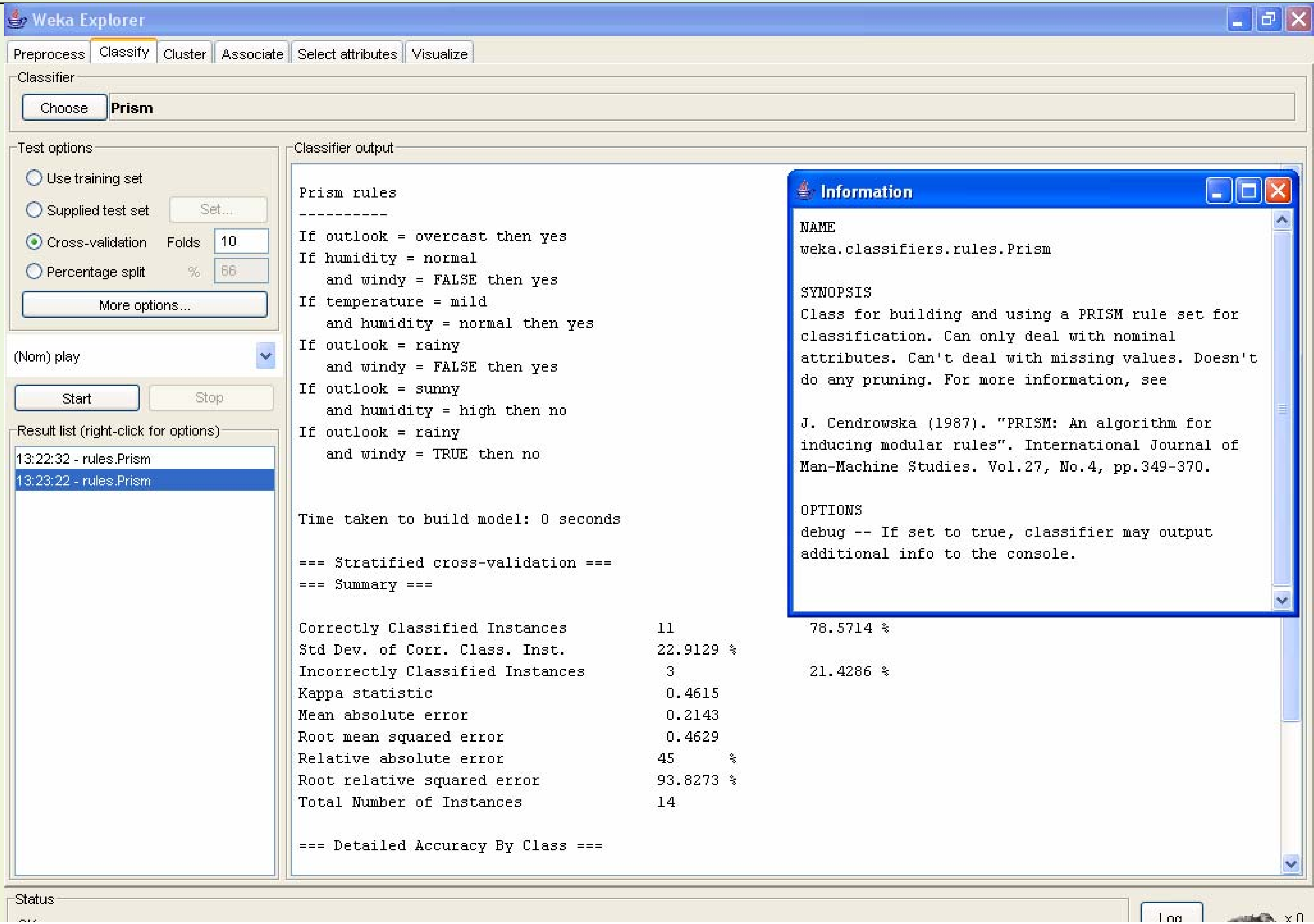
```
If astigmatism = yes
   and tear production rate = normal
   and spectacle prescription = myope
   then recommendation = hard
```

- Second rule for recommending “hard lenses”:
  (built from instances not covered by first rule)

```
If age = young and astigmatism = yes
   and tear production rate = normal
   then recommendation = hard
```

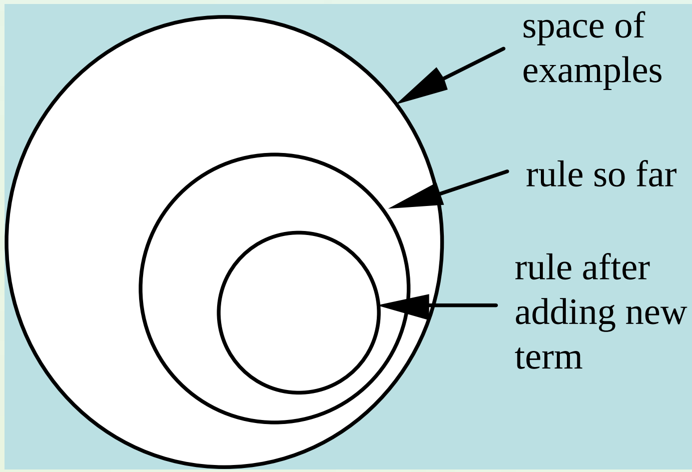
- These two rules cover all “hard lenses”:
  - Process is repeated with other two classes

More on PRISM (WEKA)



A search in a simple covering algorithm

- Generates a rule by adding tests that maximize rule's accuracy
- Similar to situation in decision trees: problem of selecting an attribute to split on
  - But: decision tree inducer maximizes overall purity
- Each new term reduces rule's coverage:



LEM2 algorithm with rough approximations

- Grzymala 92; - induces rules from rough sets approximations of inconsistent decision classes.
- Sequential covering (similar to PRISM but another evaluation criteria)
- A heuristic approach to minimal set of rules; it is based on iterative computing the single local covering **T** (see it as a set of cond. parts) of each concept (approximation) in a decision table
- **T** is a local covering of K iff

  Each member T$\in$ **T** is minimal

$$\bigcup_{T \in \mathbf{T}} [T] = K$$

T is minimal, i.e. contains the smallest number of elements *T*.

LEM2 - the description

Procedure LEM2
(**input:** a set K; **output:** a single local covering T of set K);
begin
G := K; **T**:= ∅;
while $G \neq \emptyset$ **do**
begin
$T := \emptyset$;
$T(G) := \{t \mid [t] \cap G \neq \emptyset\}$;
while $T = \emptyset$ **or not (**$[T] \subseteq B$) **do begin**
select a pair a pair t from $T(G)$ such that $|[t] \cap G|$ is maximum; if another tie occurs, select a pair
$t \in T(G)$ with the smallest cardinality of $[t]$; if a further tie occurs, select first pair;
$T := T \cup \{t\}$;
$G := [t] \cap G$;
$T(G) := \{t \mid [t] \cap G \neq \emptyset\}$;
$T(G) := T(G) - T$;
end; {while}
for each t in T **do if** $[T - \{t\}] \subseteq B$ **then** $T := T - \{t\}$;
T := **T** $\cup \{T\}$;
$G := B - \cup$ **[T]**;
end {while};
for each $T \in$ **T do if** $\cup_{S \in \mathbf{T} - \{T\}} [S] = B$ **then T** := **T** $- \{T\}$;
end {procedure}.

LEM2 – An Example (1)

U	*Headache*	*Nausea*	*Temp.*	*Flu*
x1	no	no	normal	No
x2	yes	no	high	Yes
x3	yes	yes	high	Yes
x4	yes	no	normal	No
x5	no	no	high	No
x6	no	no	high	Yes

IND: {x1}, {x2}, {x3}, {x4}, {x5,x6}
YES: lower appr. {x2,x3}
upper {x2,x3,x5,x6}
NO: lower approx. {x1,x4}
upper {x1,x4,x5,x6}
Inconsistent boundary {x5,x6}

Certajn rules for **(Flue=Yes)**: Concept {x2,x3}

(headache,yes) {x2,x3+ ; x4-}
(nausea,no) {x2+ ; x1,x4,x5,x6-}
(nausea,yes) {x3+ }
(temperature,high) {x2,x3+ ; x5,x6-}

Choose t_1 (headache,yes) but it {x2,x3+ ; x4-} $\not\subset$ {x2,x3}, so look for next, new condition ; Add (temperature,high),
now t1$\cap$t2= {x2,x3+ ; x4-} $\cap$ {x2,x3+ ; x5,x6-} $\subseteq$ {x2,x3}
Finally, the rule **(headache=yes) $\cap$ (temperature=high) $\rightarrow$(Flue=Yes)** describes all examples from this concept

LEM2 – An Example (2)

U	*Headache*	*Nausea*	*Temp.*	*Flu*
x1	no	no	normal	No
x2	yes	no	high	Yes
x3	yes	yes	high	Yes
x4	yes	no	normal	No
x5	no	no	high	No
x6	no	no	high	Yes

IND: {x1}, {x2}, {x3}, {x4}, {x5,x6}
YES: lower appr. {x2,x3}
upper {x2,x3,x5,x6}
NO: lower approx. {x1,x4}
upper {x1,x4,x5,x6}

Certajn rules for **(Flue=No)**: Concept {x1,x4}
(headache,no} {x1+; x5,x6-}
(headache,yes) {x4+ ; x2,x3-}
(nausea,no) {x1,x4+;x2,x5,x6-}
(temperature,normal) {x1,x4+ ; ∅}

Choose t_1 (temperature,normal),
now t1= {x1,x4+ ; ∅-} ⊆ {x1,x4}
Finally, the rule **(temperature=normal) →(Flue=No)** describes all examples from this concept

Evaluation of candidates in Learning One Rule

- When is a candidate for a rule R treated as "good"?
  - High accuracy P(K|R);
  - High coverage |[P]I = *n*.
- Possible evaluation functions:
  - *Relative frequency*: $$\frac{n_K(R)}{n(R)}$$
    - where n_K is the number of correctly classified examples form class K, and *n* is the number of examples covered by the rule → problems with small samples;
  - Laplace estimate:

    Good for uniform prior distribution of k classes $$\frac{n_K(R)+1}{n(R)+k}$$
  - *m-estimate of accuracy*: $(n_K(R)+mp)/(n(R)+m)$,

    where n_K is the number of correctly classified examples, *n* is the number of examples covered by the rule, *p* is the prior probablity of the class predicted by the rule, and *m* is the weight of *p* (domain dependent – more noise / larger *m*).

Other evaluation functions of rule R and class K

Assume rule R specialized to rule R'

- Entropy (Information gain and others versions).
- Accuracy gain (increase in expected accuracy)

  P(K|R') – P(K|R)

- Many others
- Also weighted functions, e.g.

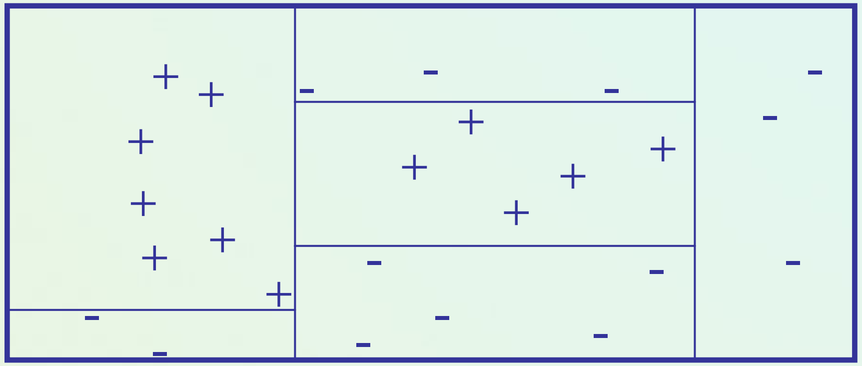
$$WAG(R', R) = \frac{n_K(R')}{n_K(R)} \cdot (P(K \mid R') - P(K \mid R))$$

$$WIG(R', R) = \frac{n_K(R')}{n_K(R)} \cdot (\log_2(K \mid R') - \log_2(K \mid R))$$

Decision rules vs. decision trees → graphical interpretation

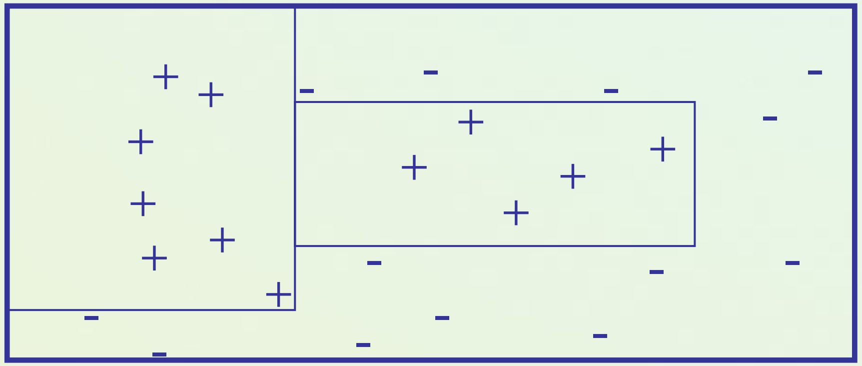
- Trees – splitting the data space (e.g. C4.5)

Decision boundaries of decision trees



- Rules – covering parts of the space (AQ, CN2, LEM)

Decision boundaries of decision rules



Original covering idea (AQ, Michalski 1969, 86)

for each class Ki **do**

Ei := Pi U Ni (Pi positive, Ni negative example)

RuleSet(Ki) := empty

repeat {find-set-of-rules}

find-one-rule R covering some positive examples

and no negative ones

add R to RuleSet(Ki)

delete from Pi all pos. ex. covered by R

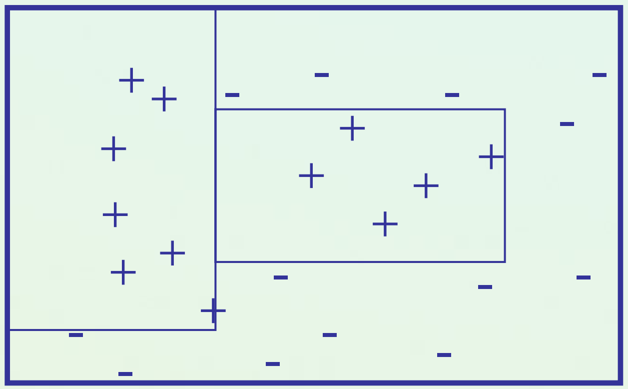
until Pi (set of pos. ex.) = empty

Find one rule:

Choosing a positive example called a seed.

Find a limited set of rules characterizing the seed → **STAR**.

Choose the best rule according to LEF criteria.



Another variant – CN2 algorithm

- Clark and Niblett 1989; Clark and Boswell 1991; Many other improvements
- Combine ideas AQ with TDIDT (search as in AQ, additional evaluation criteria or prunning as for TDIDT).
  - AQ depends on a seed example
  - Basic AQ has difficulties with noise handling
    - Latter solved by rule truncation (pos-pruning)
- Principles:
  - Covering approach (but stopping criteria relaxed).
  - Learning one rule – not so much example-seed driven.
  - Two options:
    - Generating an unordered set of rules (First Class, then conditions).
    - Generating an ordered list of rules (find first the best condition part than determine Class).

MODLEM – Algorithm for rule induction

- MODLEM [Stefanowski 98] generates a minimal set of rules.
- Its extra specificity – handling directly numerical attributes during rule induction; elementary conditions, e.g. $(a \geq v)$, $(a < v)$, $(a \in [v_1, v_2))$ or $(a = v)$.
- Elementary condition evaluated by one of three measures: class entropy, Laplace accuracy or Grzymala 2-LEF.

obj.	*a*1	*a*2	*a*3	*a*4	*D*
*x*1	m	2.0	1	a	C1
*x*2	f	2.5	1	b	C2
*x*3	m	1.5	3	c	C1
*x*4	f	2.3	2	c	C1
*x*5	f	1.4	2	a	C2
*x*6	m	3.2	2	c	C2
*x*7	m	1.9	2	b	C1
*x*8	f	2.0	3	a	C2

if $(a1 = \text{m})$ *and* $(a2 \leq 2.6)$ *then* $(D = \text{C1})$ $\{x1, x3, x7\}$

if $(a2 \in [1.45, 2.4])$ *and* $(a3 \leq 2)$ *then* $(D = \text{C1})$ $\{x1, x4, x7\}$

if $(a2 \geq 2.4)$ *then* $(D = \text{C2})$ $\{x2, x6\}$

if $(a1 = \text{f})$ *and* $(a2 \leq 2.15)$ *then* $(D = \text{C2})$ $\{x5, x8\}$

Mushroom data (UCI Repository)

- Mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms (1981).
- This data set includes descriptions of hypothetical samples corresponding to 23 species of mushrooms in the Agaricus and Lepiota Family. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility.
- Number of examples: 8124.
- Number of attributes: 22 (all nominally valued)
- Missing attribute values: 2480 of them.
- Class Distribution:
  - -- edible: 4208 (51.8%)
  - -- poisonous: 3916 (48.2%)

MOLDEM rule set (Implemented in WEKA)

=== Classifier model (full training set) ===

Rule 1.(odor is in: {n, a, l})&(spore-print-color is in: {n, k, b, h, o, u, y, w})&(gill-size = b) => (class = e); [3920, 3920, 93.16%, 100%]

Rule 2.(odor is in: {n, a, l})&(spore-print-color is in: {n, h, k, u}) => (class = e); [3488, 3488, 82.89%, 100%]

Rule 3.(gill-spacing = w)&(cap-color is in: {c, n}) => (class = e); [304, 304, 7.22%, 100%]

Rule 4.(spore-print-color = r) => (class = p); [72, 72, 1.84%, 100%]

Rule 5.(stalk-surface-below-ring = y)&(gill-size = n) => (class = p); [40, 40, 1.02%, 100%]

Rule 6.(odor = n)&(gill-size = n)&(bruises? = t) => (class = p); [8, 8, 0.2%, 100%]

Rule 7.(odor is in: {f, s, y, p, c, m}) => (class = p); [3796, 3796, 96.94%, 100%]

Number of rules: 7

Number of conditions: 14

Approaches to Avoiding Overfitting

- **Pre-pruning:** stop learning the decision rules before they reach the point where they perfectly classify the training data

- **Post-pruning:** allow the decision rules to overfit the training data, and then post-prune the rules.

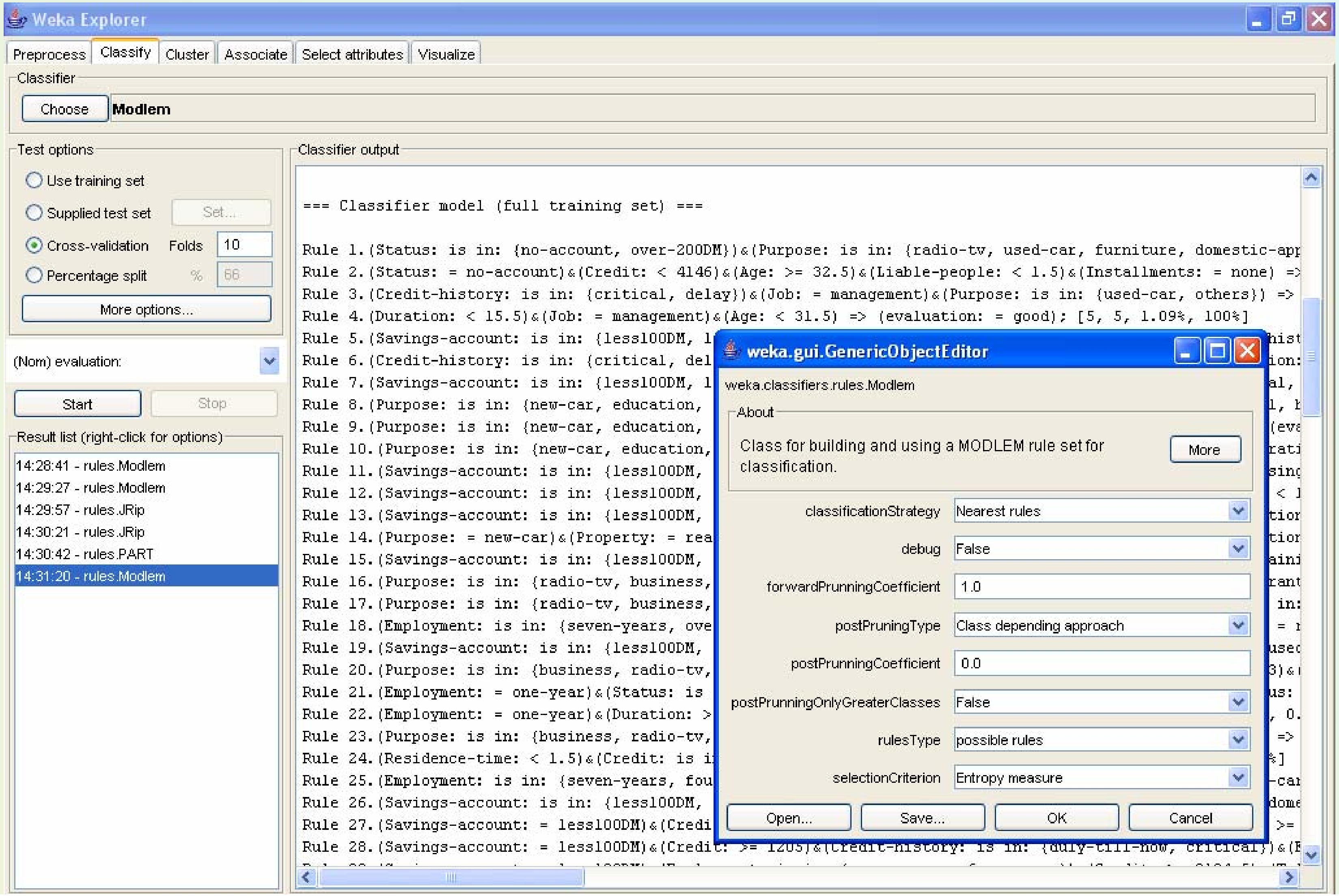
Pre-Pruning

The criteria for stopping learning rules can be:

- **minimum purity** criterion requires a certain percentage of the instances covered by the rule to be positive;
- **significance test** determines if there is a significant difference between the distribution of the instances covered by the rule and the distribution of the instances in the training sets.

Pruning in MODLEM

- Majority class in pre-pruning, Min_supp in post-pruning



Post-Pruning (Grow, IREP)

1. Split instances into *Growing Set* and *Pruning Set*;
2. Learn set *SR* of rules using *Growing Set*;
3. Find the best simplification *BSR* of *SR*.
4. **while** (Accuracy(*BSR*, *Pruning Set*) > Accuracy(*SR*, *Pruning Set*)) **do**

   4.1 *SR* = *BSR*;

   4.2 Find the best simplification *BSR* of *SR*.
5. **return** *BSR*;

JRIP – prune or not (WEKA)

- WEKA impl. of RIPPER runned for WZW data set

14:29:57 - rules.JRip

```
JRIP rules:
===========

(Naklucia: = n) => WZW:=a (251.0/56.0)
(Wiek: <= 42) and (Zywienie: = p) => WZW:=a (10.0/2.0)
 => WZW:=b (241.0/47.0)

Number of Rules : 3


Time taken to build model: 0.21 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         386               76.
Std Dev. of Corr. Class. Inst.           5.8513 %
Incorrectly Classified Instances       116               23.
Kappa statistic                          0.5378
Mean absolute error                      0.3388
Root mean squared error                  0.4227
Relative absolute error                 67.7579 %
Root relative squared error             84.5384 %
Total Number of Instances              502

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.756     0.218      0.775     0.756     0.765    a
  0.782     0.244      0.764     0.782     0.773    b

=== Confusion Matrix ===

   a   b   <-- classified as
 189  61 |   a = a
  55 197 |   b = b
```

14:42:01 - rules.JRip

```
JRIP rules:
===========

(Naklucia: = n) and (Wiek: <= 19) and (Wiek: <= 15) and (Zachorowanie: = jz) 
(Naklucia: = n) and (Wiek: <= 29) and (Wiek: <= 15) and (Kontakt: = t) => WZW
(Naklucia: = n) and (Wiek: <= 29) and (Zaopatrzenie_w_wode: = l) and (Ustep: 
(Naklucia: = n) and (Wiek: <= 33) and (Objawy_rzekomogrypowe: = t) and (Wiek:
(Naklucia: = n) and (Wiek: <= 33) and (Zaburzenia_dyspeptyczne: = t) and (Ust
(Naklucia: = n) and (Zaburzenia_dyspeptyczne: = t) and (Wiek: <= 19) and (Kon
(Naklucia: = n) and (Kontakt: = t) and (Wiek: <= 44) and (Wiek: >= 28) and (O
(Naklucia: = n) and (Zazolcenie: = n) and (Mocz_i_kal: = z) => WZW:=a (4.0/0.
(Wiek: <= 42) and (Zywienie: = p) and (Czystosc: = bc) => WZW:=a (8.0/0.0)
(Naklucia: = n) and (Gamma_globulina: = n) and (Objawy_rzekomogrypowe: = t) a
(Wiek: <= 39) and (Zaburzenia_dyspeptyczne: = t) and (Wiek: <= 11) and (Wiek:
(Wiek: <= 39) and (Zywienie: = p) and (Wiek: >= 18) and (Wiek: <= 23) => WZW:
(Wiek: <= 39) and (Zaburzenia_dyspeptyczne: = t) and (Zaopatrzenie_w_wode: = 
(Wiek: <= 34) and (Zaburzenia_dyspeptyczne: = t) and (Zaopatrzenie_w_wode: = 
(Naklucia: = n) and (Zachorowanie: = wl) and (Czystosc: = b) => WZW:=a (3.0/0
(Naklucia: = n) and (Zachorowanie: = wl) and (Zaburzenia_dyspeptyczne: = t) a
 => WZW:=b (310.0/58.0)

Number of Rules : 17


Time taken to build model: 0.1 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         387               77.0916 %
Std Dev. of Corr. Class. Inst.           8.3994 %
Incorrectly Classified Instances       115               22.9084 %
Kappa statistic                          0.5414
Mean absolute error                      0.2742
Root mean squared error                  0.4354
Relative absolute error                 54.8451 %
Root relative squared error             87.0713 %
Total Number of Instances              502
```

Slajd 38 z 68 | 2_Projekt domyślny | Polski

Applying rule set to classify objects

- **Matching** a new object description *x* to condition parts of rules.
  - Either object's description satisfies all elementary conditions in a rule, or not.

  IF (a1=L) and (a3≥ 3) THEN Class +

  x → (a1=L),(a2=s),(a3=7),(a4=1)

- Two ways of assigning x to class K depending on the set of rules:
  - Unordered set of rules (AQ, CN2, PRISM, LEM)
  - Ordered list of rules (CN2, c4.5rules)

Applying rule set to classify objects

- The rules are ordered into priority decision list!

  Another way of rule induction – rules are learned by first determining Conditions and then Class (CN2)

Notice: mixed sequence of classes K1,…, K in a rule list

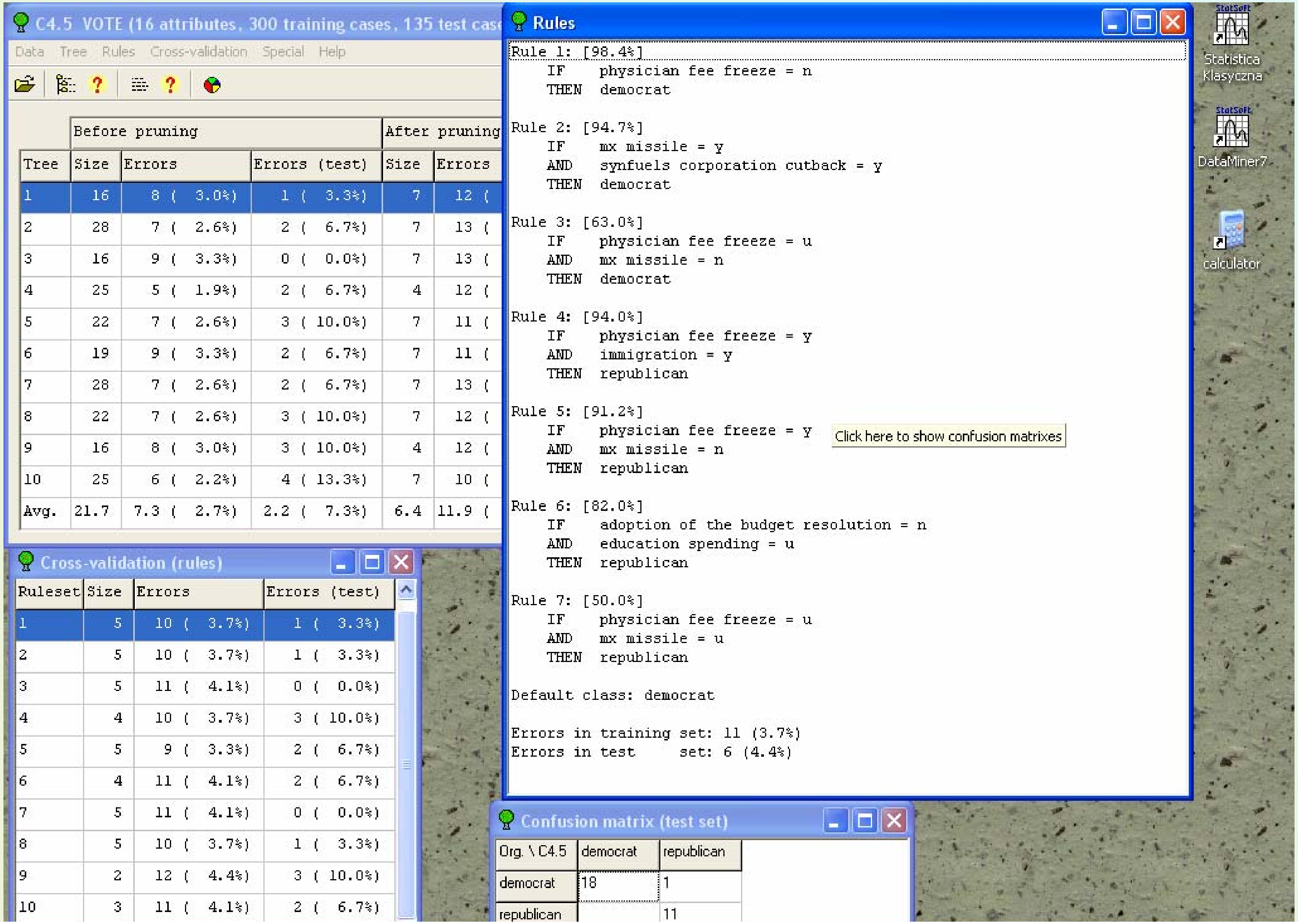
But: ordered execution when classifying a new instance: rules are sequentially tried and the first rule that 'fires' (covers the example) is used for final decision

Decision list {R1, R2, R3, …, D}: rules Ri are

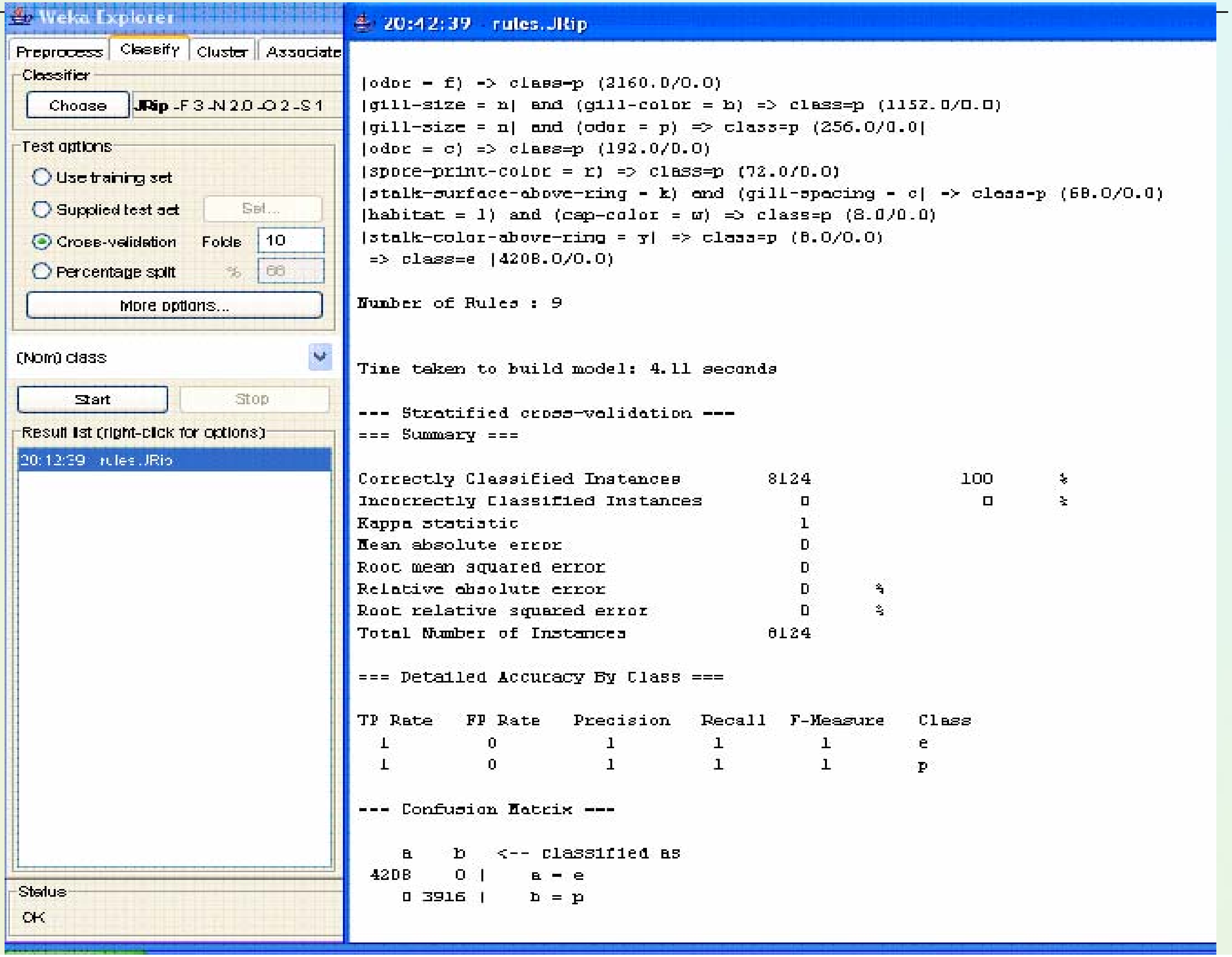
interpreted as **if-then-else** rules

If no rule fires, then DefaultClass (majority class in input data)

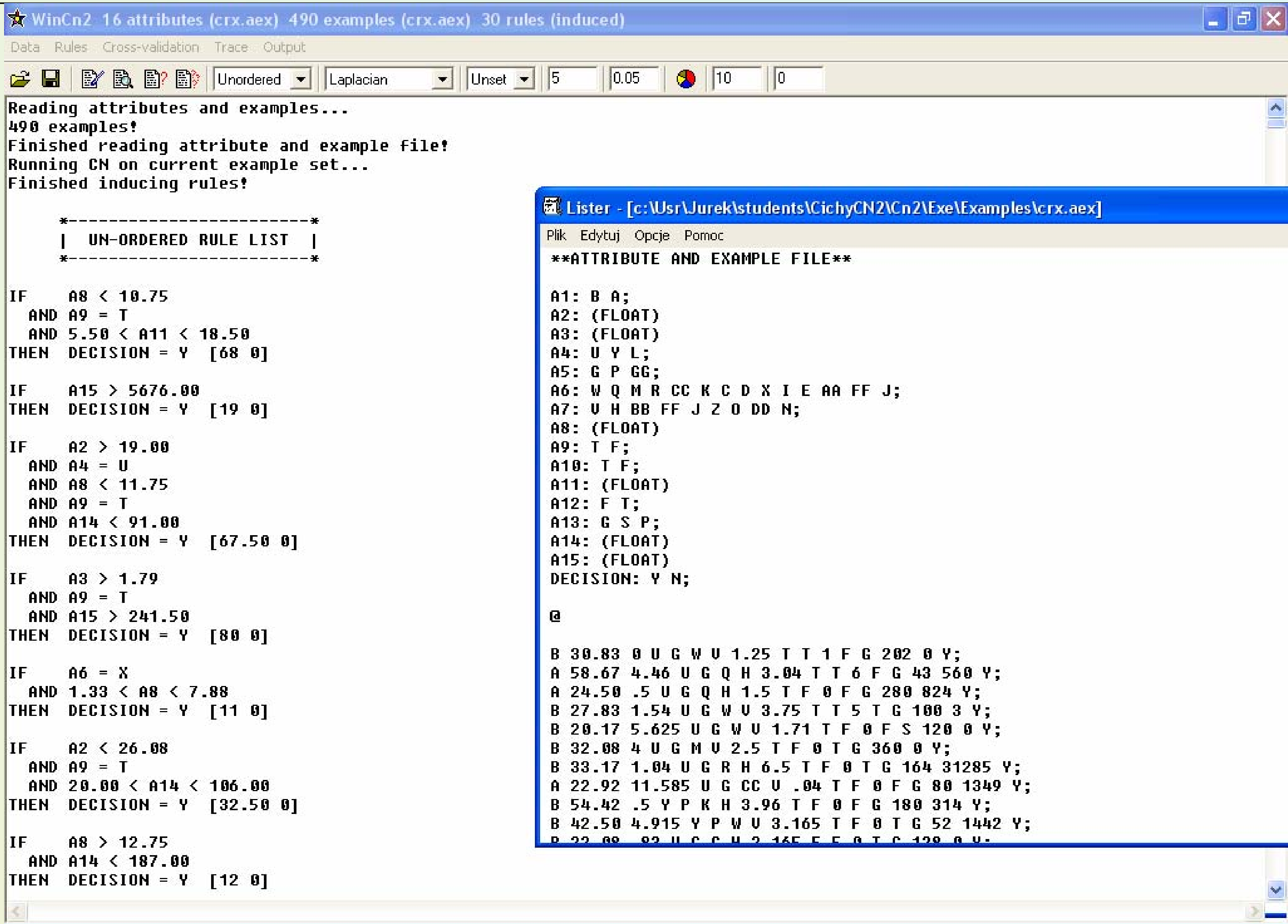
Priority decision list (C4.5 rules)



Specific list of rules - RIPPER (Mushroom data)



CN2 – unordered rule set



Applying unordered rule set to classify objects

- An unordered set of rules → three situations:
  - Matching to rules indicating the same class.
  - Multiple matching to rules from different classes.
  - No matching to any rule.
- <u>An example:</u>
- e1={(Age=m), (Job=p),(Period=6),(Income=3000),(Purpose=K)}
  - rule 3: if (Period∈[3.5,12.5)) then (Dec=d) [2]
  - Exact matching to rule 3. → Class (Dec=d)
- e2={(Age=m), (Job=p),(Period=2),(Income=2600),(Purpose=M)}
  - No matching!

Solving conflict situations

- LERS classification strategy (Grzymala 94)
  - Multiple matching
    - Two factors: *Strength*(*R*) – number of learning examples correctly classified by *R* and final class *Support*(*Y*i):

$$\sum_{\text{matching rules R for Yi}} Strength(R)$$

  - Partial matching
    - Matching factor *MF*(*R*) and

$$\sum_{\text{partially match. rules R for Yi}} MF(R) \cdot Strength(R)$$

- e2={(Age=m), (Job=p), (Period=2),(Income=2600),(Purpose=M)}
  - Partial matching to rules 2 , 4 and 5 for all with MF = 0.5
  - Support(r) = 0.5·2 =1 ; Support(d) = 0.5·2+0.5·2=2
- Alternative approaches – e.g. nearest rules (Stefanowski 95)
- Instead of MF use a kind of normalized distance *x* to conditions of *r*

Some experiments

- Analysing strategies (total accuracy in [%]):

data set	all	multiple	exact
large soybean	87.9	85.7	79.2
election	89.4	79.5	71.8
hsv2	77.1	70.5	59.8
concretes	88.9	82.8	81.0
breast cancer	67.1	59.3	51.2
imidasolium	53.3	44.8	34.4
lymphograpy	85.2	73.6	67.6
oncology	83.8	82.4	74.1
buses	98.0	93.5	90.8
bearings	96.4	90.9	87.3

- Comparing to other classification approaches
  - Depends on the data
  - Generally → similar to decision trees

Different perspectives of rule application

- In a descriptive perspective
  - To present, analyse the relationships between values of attributes, to explain and understand classification patterns
- In a prediction/classification perspective,
  - To predict value of decision class for new (unseen) object)

Perspectives are different;
Moreover rules are evaluated in a different ways!

Evaluating single rules

- rule *r* (*if P then Q*) derived from *DT*, examples *U*.

	Q	$\neg Q$	
P	n_{PQ}	$n_{P\neg Q}$	n_P
$\neg P$	$n_{\neg PQ}$	$n_{\neg P\neg Q}$	$n_{\neg P}$
	n_Q	$n_{\neg Q}$	n

- Reviews of measures, e.g.
- Yao Y.Y, Zhong N., An analysis of quantitative measures associated with rules, In: Proc. the 3rd Pacific-Asia Conf. on Knowledge Discovery and Data Mining, LNAI 1574, Springer, 1999, pp. 479-488.
- Hilderman R.J., Hamilton H.J, Knowledge Discovery and Measures of Interest. Kluwer, 2002.
- Support of rule r $$G(P \wedge Q) = \frac{n_{PQ}}{n}$$ Coverage $$AS(P \mid Q) = \frac{n_{PQ}}{n_Q}$$
- Confidence of rule r $$AS(Q \mid P) = \frac{n_{PQ}}{n_P}$$ and others …

Other descriptive measures

Change of support – confirmation of supporting Q by a premise P (Piatetsky-Shapiro)

$$CS(Q \mid P) = AS(Q \mid P) - G(Q)$$

where $G(Q) = \frac{n_Q}{n}$

Interpretaion: Range between -1 and +1 ; Difference of probabilites a prior i a posterior; A positive number indicates influence of premise P on conslusion Q; a negative values shows no influence**.**

Degree of independence:

$$IND(Q, P) = \frac{G(P \wedge Q)}{G(P) \cdot G(Q)}$$

Aggregated measures

Based on previous measures:

Significance of a rule (propozycja Yao i Liu)

$$S(Q|P) = AS(Q|P) \cdot IND(Q,P)$$

Klosgen's measure of interest

$$K(Q|P) = G(P)^{\alpha} \cdot (AS(Q|P) - G(Q))$$

Michalski's weighted sum

$$WSC(Q|P) = w_1 \cdot AS(Q|P) + w_2 \cdot AS(P|Q)$$

The relative risk (Ali, Srikant):

$$r(Q|P) = \frac{AS(Q|P)}{AS(Q|\neg P)}$$

Descriptive requirements to single rules

- In descriptive perspective users may prefer to discover rules which should be:
  - strong / general – high enough rule coverage $AS(P|Q)$ or support.
  - accurate – sufficient accuracy $AS(Q|P)$.
  - simple (e.g. which are in a limited number and have short condition parts).
  - Number of rules should not be too high.
- Covering algorithms biased towards minimum set of rules - containing only a limited part of potentially `interesting' rules.
  - We need another kind of rule induction algorithms!

Explore algorithm (Stefanowski, Vanderpooten)

- Another aim of rule induction
  - to extract from data set inducing **all rules** that *satisfy* some *user's requirements* connected with *his interest* (regarding, e.g. the strength of the rule, level of confidence, length, sometimes also emphasis on the syntax of rules).
- Special technique of exploration the space of possible rules:
  - Progressively generation rules of increasing size using in the most efficient way some 'good' pruning and stopping condition that reject unnecessary candidates for rules.
- Similar to adaptations of Apriori principle for looking frequent itemsets [AIS94]; Brute [Etzioni]

Various sets of rules (Stefanowski and Vanderpooten 1994)

- A minimal set of rules (LEM2):

rule 1. if $(q_1 = 2) \wedge (q_3 = 1)$ then $(d = 1)$ $\{1, 2, 3, 4, 5\}$ 5/8
rule 2. if $(q_1 = 1)$ then $(d = 1)$ $\{6, 7\}$ 2/8
rule 3. if $(q_3 = 2) \wedge (q_6 = 2)$ then $(d = 1)$ $\{6, 8\}$ 2/8
rule 4. if $(q_1 = 3)$ then $(d = 2)$ $\{9, 10, 11, 13, 14\}$ 5/7
rule 5. if $(q_3 = 3)$ then $(d = 2)$ $\{15\}$ 1/7
rule 6. if $(q_3 = 2) \wedge (q_4 = 1) \wedge (q_6 = 1)$ then $(d = 2)$ $\{12\}$ 1/7

- A „satisfactory” set of rules (Explore):

Let us assume that the user's level of interest to the possible strength of a rule by assigning a value $l = 50\%$ in SC.

Explore gives the following decision rules:

rule 1. if $(q_2 = 3)$ then $(d = 1)$ $\{1, 2, 3, 6, 7\}$ 5/8
rule 2. if $(q_1 = 2) \wedge (q_3 = 1)$ then $(d = 1)$ $\{1, 2, 3, 4, 5\}$ 5/8
rule 3. if $(q_1 = 3)$ then $(d = 2)$ $\{9, 10, 11, 13, 14\}$ 5/7
rule 4. if $(q_4 = 2)$ then $(d = 2)$ $\{10, 13, 14, 15\}$ 4/7

Table 1: The illustrative set of learning exam

No.	q_1	q_2	q_3	q_4	q_5	q_6	d
1	2	3	1	3	1	2	1
2	2	3	1	1	1	1	1
3	2	3	1	3	2	1	1
4	2	1	1	1	1	1	1
5	2	2	1	1	2	2	1
6	1	3	2	3	1	2	1
7	1	3	2	3	2	1	1
8	2	1	2	1	2	2	1
9	3	1	1	3	1	2	2
10	3	1	2	2	2	1	2
11	3	1	1	3	2	2	2
12	2	1	2	1	2	1	2
13	3	2	4	2	1	1	2
14	3	2	4	2	2	1	2
15	2	2	3	2	1	2	2
16	2	2	2	1	1	1	1
17	2	2	2	1	1	1	2

A diagnostic case study

- A fleet of homogeneous 76 buses (AutoSan H9-21) operating in an inter-city and local transportation system.
- The following symptoms characterize these buses :

  s1 – maximum speed [km/h],

  s2 – compression pressure [Mpa],

  s3 – blacking components in exhaust gas [%],

  s4 – torque [Nm],

  s5 – summer fuel consumption [l/100lm],

  s6 – winter fuel consumption [l/100km],

  s7 – oil consumption [l/1000km],

  s8 – maximum horsepower of the engine [km].

Experts' classification of busses:

1. Buses with engines in a good technical state – further use (46 buses),
2. Buses with engines in a bad technical state – requiring repair (30 buses).

MODLEM algorithm – (sequential covering)

- A ***minimal*** set of *discriminating* decision rules

  1. if (*s2*≥2.4 MPa) & (*s7*<2.1 *l*/1000km) then (technical state=good) [46]

  2. if (*s2*<2.4 MPa) then (technical state=bad) [29]

  3. if (*s7*≥2.1 *l*/1000km) then (technical state=bad) [24]

- The prediction accuracy (‘leaving-one-out’ reclassification test) is equal to 98.7%.

Another set of rules (EXPLORE)

All decision rules with min. SC1 threshold (rule coverage > 50%):

1. if (*s1*>85 km/h) then (technical state=good) [34]
2. if (*s8*>134 kM) then (technical state=good) [26]
3. if (*s2*≥2.4 MPa) & (*s3*<61 %) then (technical state=good) [44]
4. if (*s2*≥2.4 MPa) & (*s4*>444 Nm) then (technical state=good) [44]
5. if (*s2*≥2.4 MPa) & (s7<2.1 *l*/1000km) then (technical state=good) [46]
6. if (*s3*<61 %) & (*s4*>444 Nm) then (technical state=good) [42]
7. if (*s1*≤77 km/h) then (technical state=bad) [25]
8. if (*s2*<2.4 MPa) then (technical state=bad) [29]
9. if (*s7*≥2.1 *l*/1000km) then (technical state=bad) [24]
10. if (*s3*≥61 %) & (*s4*≤444 Nm) then (technical state=bad) [28]
11. if (*s3*≥61 %) & (*s8*<120 kM) then (technical state=bad) [27]

The prediction accuracy - 98.7%

Preference ordered data

- MCDA vs. traditional classification (ML & Stat):
  - Attributes with preference ordered domains $\rightarrow$ criteria.
  - Ordinal classes rather than nominal labels.
  - „Semantic correlation” between values of criteria, and classes.
  - For objects x,y if $a(x) \preceq a(y)$ then their labels $\lambda(x) \preceq \lambda(y)$
- Possible inconsistency

Client	Month salary	Account status	Credit risk
A	9000	high	low
B	4000	medium	medium
C	5500	medium	high

- Dominance based rough set approach to handle it
  - Greco S., Matarazzo B., Slowinski R.

Dominance based decision rules

- Induced from rough approximations of unions of classes (upward and downward):
  - *certain* D≥-*decision rules*, supported by objects $\in Cl_t^{\geq}$ without ambiguity:

    if $q_1(x) \succeq_{q1} r_{q1}$ *and* $q_2(x) \succeq_{q2} r_{q2}$ *and …* $q_p(x) \succeq_{qp} r_{qp}$ *then* $x \in Cl_t^{\geq}$
  - *possible* D≥-*decision rules*, supported by objects $\in Cl_t^{\geq}$ and ambiguous ones from its upper approximation:

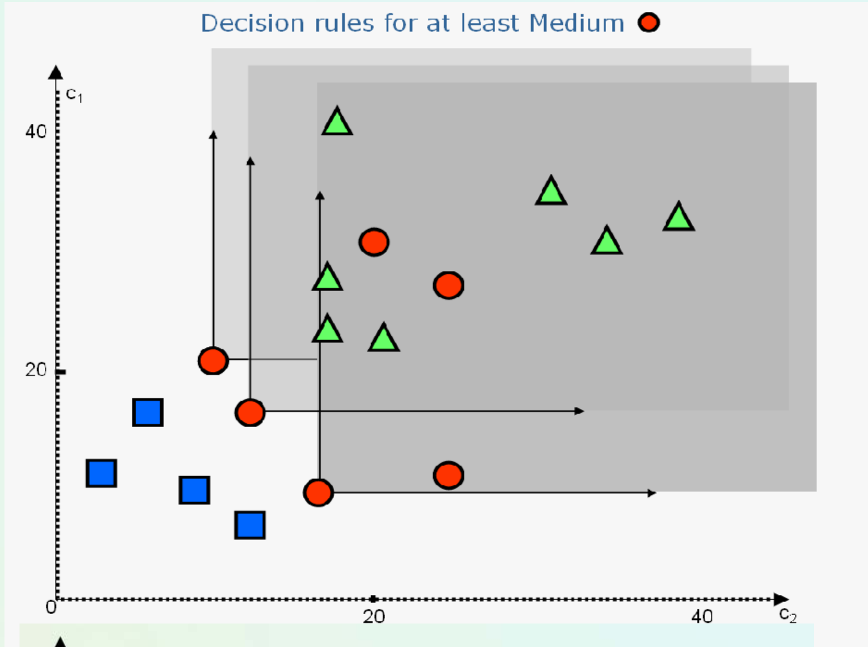
    if $q_1(x) \succeq_{q1} r_{q1}$ *and* $q_2(x) \succeq_{q2} r_{q2}$ *and …* $q_p(x) \succeq_{qp} r_{qp}$, *then* x possibly $\in Cl_t^{\geq}$

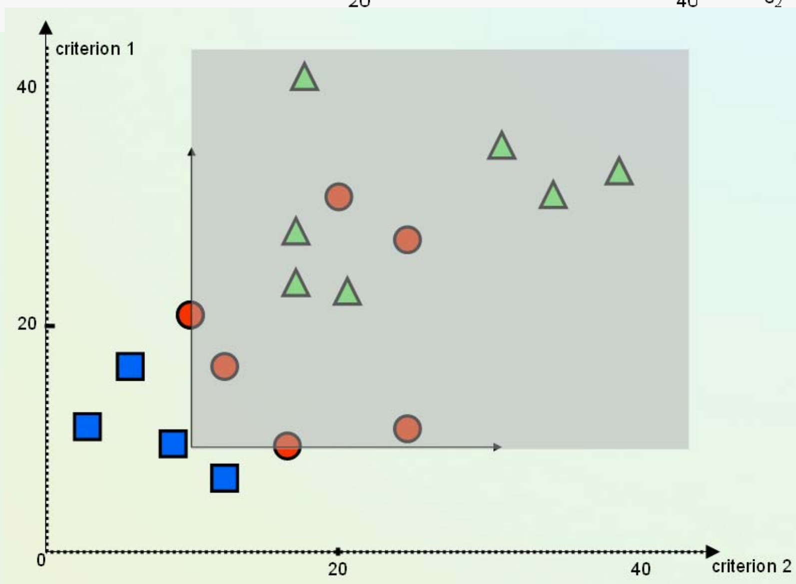
  - *certain* D≤-*decision rules*, supported by objects $\in Cl_t^{\leq}$ without ambiguity:

    if $q_1(x) \preceq_{q1} r_{q1}$ *and* $q_2(x) \preceq_{q2} r_{q2}$ *and …* $q_p(x) \preceq_{qp} r_{qp}$, *then* $x \in Cl_t^{\leq}$

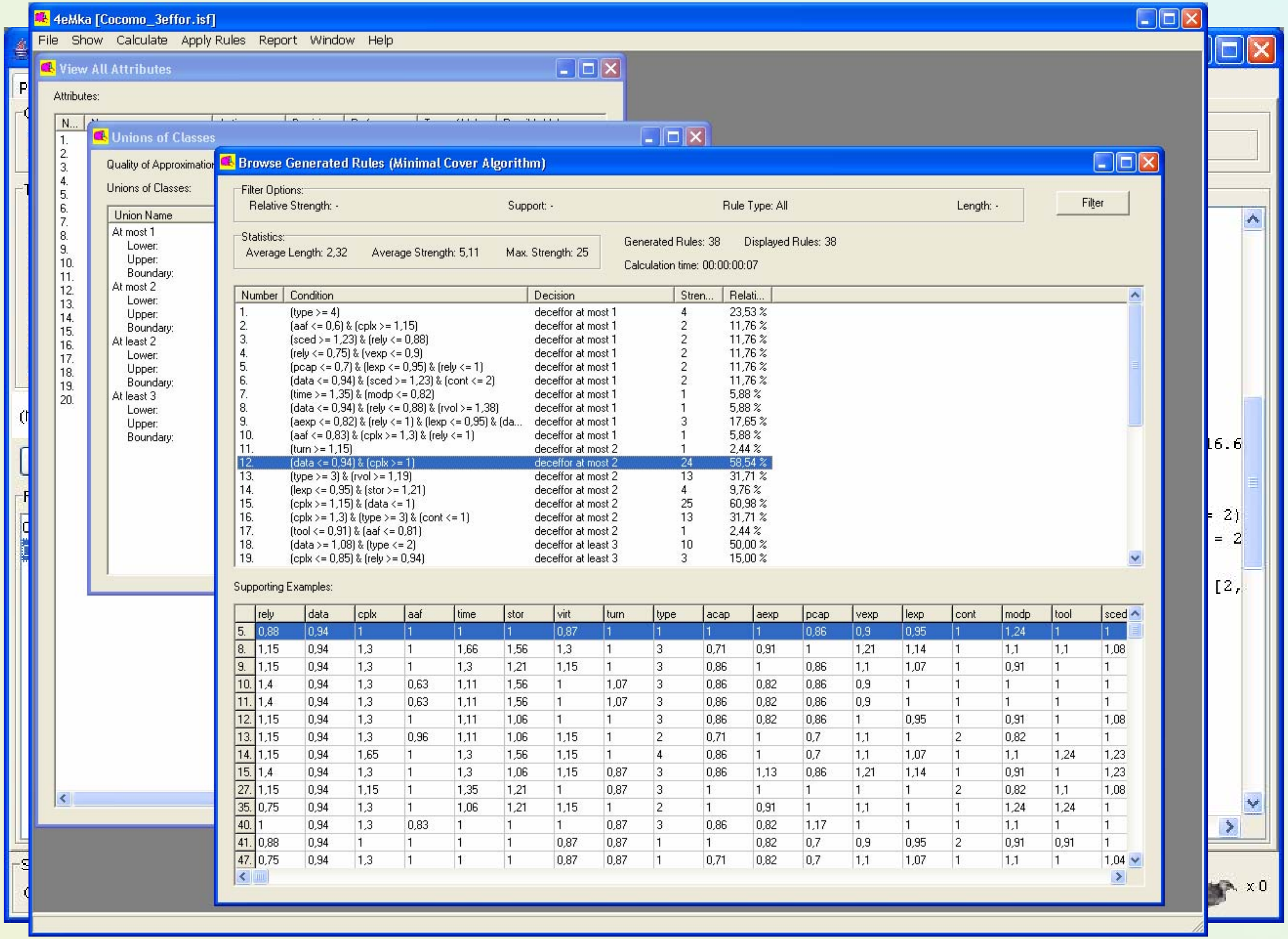
Algorithms for inducing dominance based rules

- Greco, Slowinski, Stefanowski, Blaszczynski, Dembczyński and others
  – a number of proposals
- Minimal sets of rules:
  - DOMLEM → adaptation of ideas behind MODLEM.
- DOMApriori → richer set of rules
- Robust rules → syntax based on an object from data table.
  - All rules → modifications of boolean reasoning
  - Glance → incremental learning.





Software from PUT

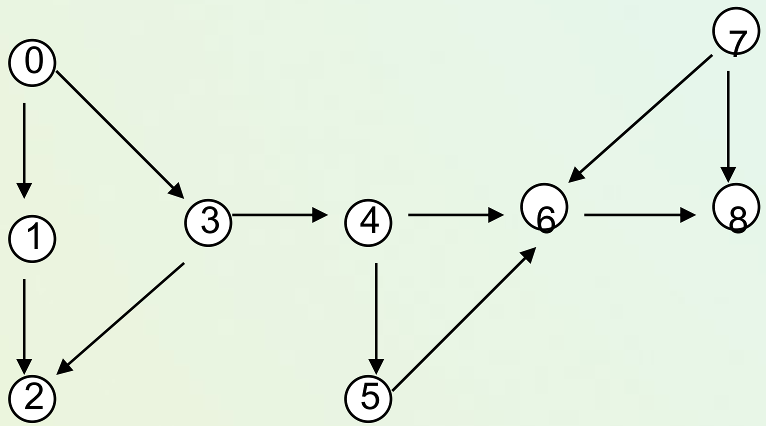


Learning First Order Rules

- Is object/attribute table sufficient data representation?
- Some limitations:
  - Representation expressivness – unable to express relations between objects or object elements. ,
  - *background knowledge* sometimes is quite complicated.
- Can learn sets of rules such as
  - $Parent(x,y) \rightarrow Ancestor(x,y)$
  - $Parent(x,z)$ and $Ancestor(z,y) \rightarrow Ancestor(x,y)$
- Research field of **Inductive Logic Programming**.

Why ILP? (slide of S.Matwin)

- **expressiveness of logic as representation** (Quinlan)



- can’t represent this graph as a fixed length vector of attributes
- can’t represent a “transition” rule:

A can-reach B if A link C, and C can-reach B

without variables

FINITE ELEMENT MESH DESIGN

Given a geometric **structure** and **loadings/boundary conditions**
Find an **appropriate resolution** for a finite element mesh

Examples: ten structures with appropriate meshes (cca. 650 edges)

Background knowledge

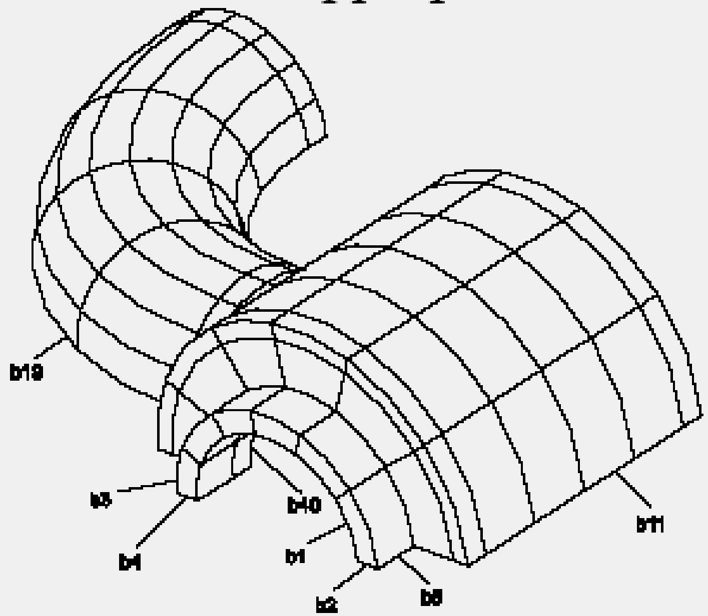
- Properties of edges (short, loaded, two-side-fixed, ...)
- Relations between edges (neighbor, opposite, equal)

ILP systems applied: GOLEM, CLAUDIEN

Many interesting rules discovered (according to expert evaluation)

Finite element mesh design (ctd.)

Example structure with an appropriate mesh



Example rules

$$mesh(Edge, 7) \leftarrow usual\_length(Edge),$$
$$neighbour\_xy(Edge, EdgeY), two\_side\_fixed(EdgeY),$$
$$neighbour\_zx(EdgeZ, Edge), not\_loaded(EdgeZ)$$
$$mesh(Edge, N) \leftarrow equal(Edge, Edge2), mesh(Edge2, N)$$

Application areas

- Medicine
- Economy, Finance
- Environmental cases
- Engineering
  - Control engineering and robotics
  - Technical diagnostics
  - Signal processing and image analysis
- Information sciences
- Social Sciences
- Molecular Biology
- Chemistry and Pharmacy
- …

Where to find more?

- T. Mitchell *Machine Learning* New York: McGraw-Hill, 1997.
- I. H. Witten & Eibe Frank *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations* San Francisco: Morgan Kaufmann, 1999.
- Michalski R.S., Bratko I., Kubat M. *Machine learning and data mining*; J. Wiley. 1998.
- Clark, P., & Niblett, T. (1989). The CN2 induction algorithm.*Machine Learning*, *3*, 261–283.
- Cohen W. Fast effective rule induction. *Proc. of the 12th Int. Conf. on Machine Learning 1995.* 115–123
- R.S. Michalski, I. Mozetic, J. Hong and N. Lavrac, The multi-purpose incremental learning system AQ15 and its testing application to three medical domains, *Proceedings of i4AAI 1986, 1041-1045, (1986).*
- J.W. Grzymala-Busse, LERS-A system for learning from example-s based on rough sets, In Intelligent`*Decision* Support: Handbook *of* Applications and Advances *of Rough Sets Theory,* (Edited by R.Slowinski), pp. 3-18
- Michalski R.S.: A theory and methodology of inductive learning. W Michalski R.S, Carbonell J.G., Mitchell T.M. (red.) *Machine learning: An Artificiall Intelligence Approach,* Morgan Kaufmann Publishers, Los Altos (1983),.
- J.Stefanowski: On rough set based approaches to induction of decision rules, w: A. Skowron, L. Polkowski (red.), *Rough Sets in Knowledge Discovery Vol* 1, Physica Verlag, Heidelberg, 1998, 500-529.
- J.Stefanowski, The rough set based rule induction technique forclassification problems, w: *Proceedings of 6th European Conference on Intelligent Techniques and Soft Computing, Aachen, EUFIT 98*, 1998, 109-113.
- J. Furnkranz . Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.

Where to find more - 2

- P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Proceedings of the 5th European Working Session on Learning (EWSL-91)*, pp. 151–163, 1991.
- Grzymala-Busse J.W.: Managing uncertainty in machine learning from examples. Proceedings of 3rd Int. Symp. on Intelligent Systems, Wigry 1994 .
- Cendrowska J.: PRISM, an algorithm for inducing modular rules. *Int. J. Man-Machine Studies*, 27 (1987), 349-370.
- Frank, E., & Witten, I. H. (1998). Generating accurate rule sets without global optimization. *Proc. of the 15th Int. Conf. on Machine Learning (ICML-98)* (pp. 144–151).
- J. Furnkranz and P. Flach. An analysis of rule evaluation metrics. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 202–209,
- S. M. Weiss and N. Indurkhya. Lightweight rule induction. In *Proc. of the 17th Int. Conference on Machine Learning (ICML-2000)*, pp. 1135–1142,
- J.Stefanowski, D.Vanderpooten: Induction of decision rules in classification and discovery-oriented perspectives, *International Journal of Intelligent Systems*, vol. 16 no. 1, 2001, 13-28.
- J.W.Grzymala-Busse, J.Stefanowski: Three approaches to numerical attribute discretization for rule induction, *International Journal of Intelligent Systems*, vol. 16 no. 1, 2001, 29-38.
- P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24:141–168, 1996.
- R. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

Any questions, remarks?

?