Indukcja reguł

JERZY STEFANOWSKI

Instytut Informatyki
Politechnika Poznańska
aktualizacja 2020

Źródła

- Wykład częściowo oparty na moich poprzednich wykładach na szkołach doktorskich.
- Proszę także przeczytać stosowane rozdziały z mojej rozprawy habilitacyjnej – dostępna na mojej stronie www.cs.put.poznan.pl/jstefanowski.
- Warto także przeczytać różne anglojęzyczne teksty:
  - J. Furnkranz: Separate-and-conquer rule learning. Artificial Intelligence Review.
  - J. Furnkranz, D. Gamberger, N. Lavrac: Foundations of Rule Learning, Springer 2012
  - P. Flach: Machine Learning (książka)

Reguły klasyfikacyjne

Zadanie klasyfikacji nadzorowanej

Wiek	Zawód	dochód	…	Decyzja
21	Prac. fiz.	1220	…	Nie kupi
26	Menedżer	2900	…	Kupuje
44	Inżynier	2600	…	Kupuje
23	Student	1100	…	Kupuje
56	Nauczyciel	1700	…	Nie kupi
…	…	…	…	…
45	Lekarz	2200	…	Nie kupi
25	Student	800	…	Kupuje

Przykłady uczące

Algorytm indukcji

Reprezentacja wiedzy
reguły
R1. Jeżeli student to kupuje komputer
R2. Jeżeli dochód > 2400 …

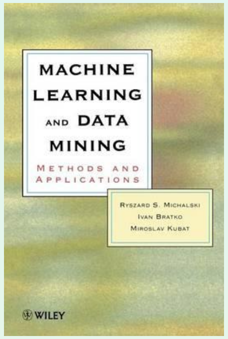
Reguły – "why?"

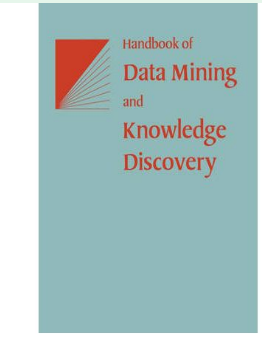
- Prosta reprezentacja symboliczna (CNF logic)

  IF Conditions THEN Class

- Łatwiejsza i naturalna dla zrozumienia przez ludzi
  → possible inspection and interpretation
  → descriptive perspective
  - Individual rules constitute "blocks" of knowledge
  - Rules directly related to facts in the training data
- Predykcja → łatwiejsza dla uzasadnienia
- Lepsza integracja z tzw. Background knowledge
- Knowledge representations in AI / Intelligent Systems
  - Expert systems, Inference in IS
- Ogólna idea – indukcja bezpośrednio z przykładów (lecz więcej algorytmów niż dla drzew)

IF Sex = male AND Age > 46 AND Number_of_painful_joints > 3 AND Skin_manif. = psoriasis THEN Diagnosis = Crystal_induced_synovitis





R. Michalski I.Bratko: Machine Learning and Data Mining.; W.Klosgen, J.Zytkow: Handbook of Data Mining and Knowledge Disc; J.Stefanowski: Algorytmy indukcji reguł decyzyjnych; C.Aggarwal: Data Classification 2015.

Pionierzy– prof. Ryszard Michalski

Father of machine learning and rule induction

Interests

Biosketch

Publications

Teaching

Research

Solving problems

Machine Learning and Inference Laboratory

School of Computational Sciences

George Mason University

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

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.

Reguły - przykład

- Standardowa forma reprezentacji
  IF *Conditions* THEN *Class*
- Inne: Class IF Conditions; Conditions → Class

Outlook	Temper.	Humid.	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

Przykład: reguły dla zbioru PlaySport:

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

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

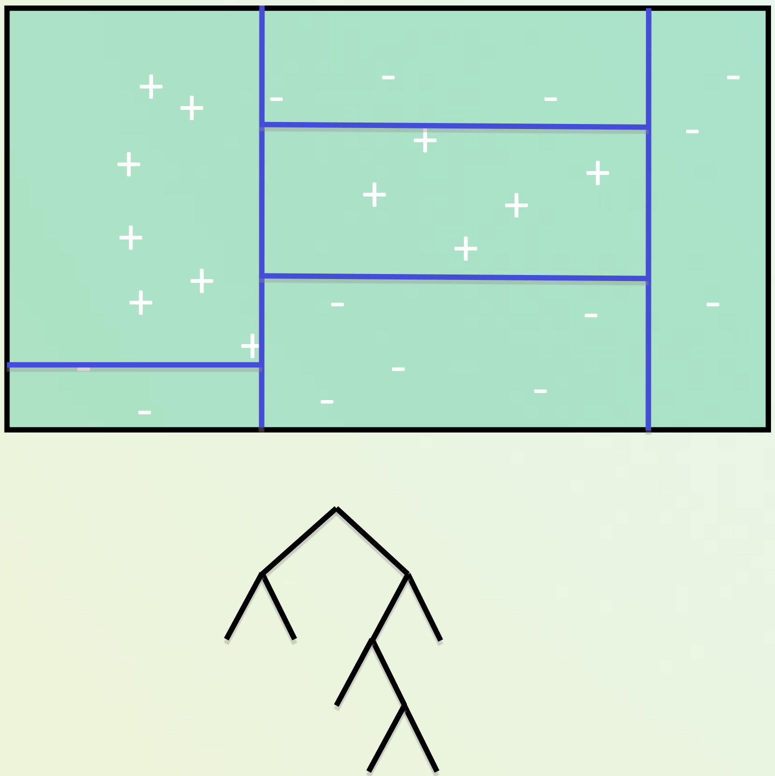
if outlook = rainy **and** windy = FALSE **then** Play = yes

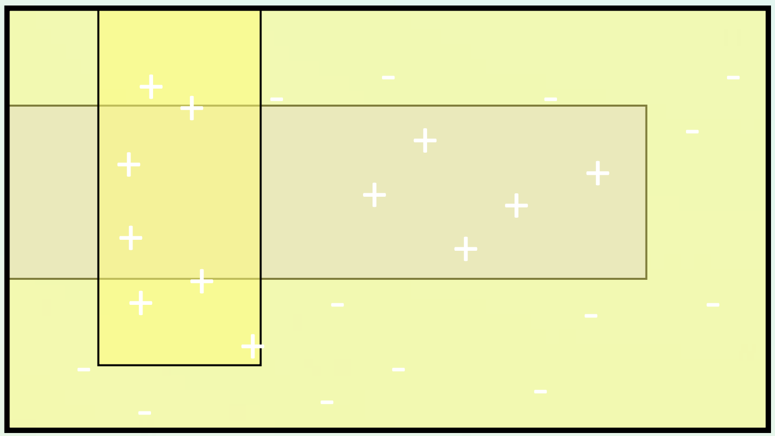
if humidity = normal **and** windy = FALSE **then** Play = yes

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

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

Decision Boundaries – interpretacje graficzne

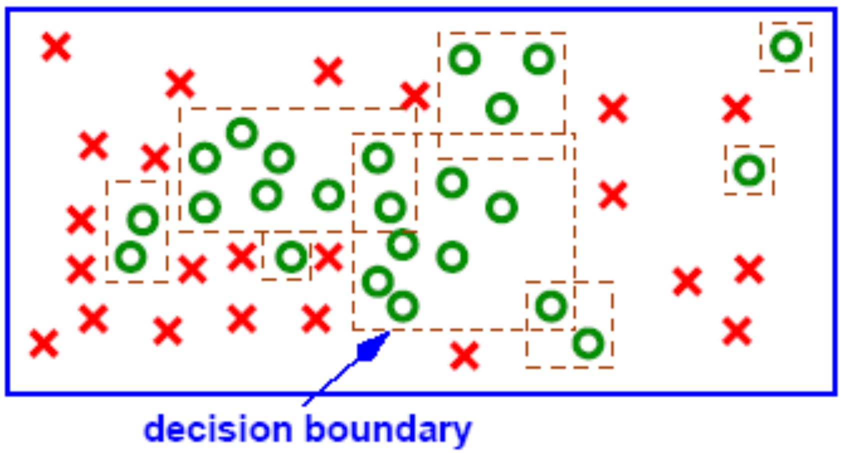




if A and B then pos
if C and D then pos

Dokładniej o pokryciu regułami

- Instances x_i in dataset D mapped to feature space:



Classes associated with instances: X, O

- Classification:

$$f(x_i) = c \in \{X, O\}$$

  - with $x_{i,j} \in \{\top, \bot\}$, and f classifier
  - dataset D is a multiset

- Objective: learn f (supervised)

- Reguły pokrywają pod-zbiór przykładów
- Rules are **local patterns!**

Reguły – trochę notacji

- 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 =,<, ≤, ≥ , >.
- [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) {$x1$,$x7$}

Oczekiwania wobec reguł

- B $\rightarrow$ a set of examples from K_j.
- A rule if P then Q is discriminant in *DT* iff $[P]=\bigcap [w_i] \subseteq$ B,
- otherwise ($P \cap B \neq \emptyset$) the rule is partly discriminating
  - Rule accuracy (or confidence) $|[P \cap K]|/|[P]|$
- Rule cannot not have a redundant condition part, i.e. there is no other $P^* \subset P$ such that $[P^*] \subseteq$ B.

- Rule sets induced from DT
  - Minimal set of rules
  - Other sets of rules (all rules, satisfactory)

Przykłady zbiorów reguł

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:

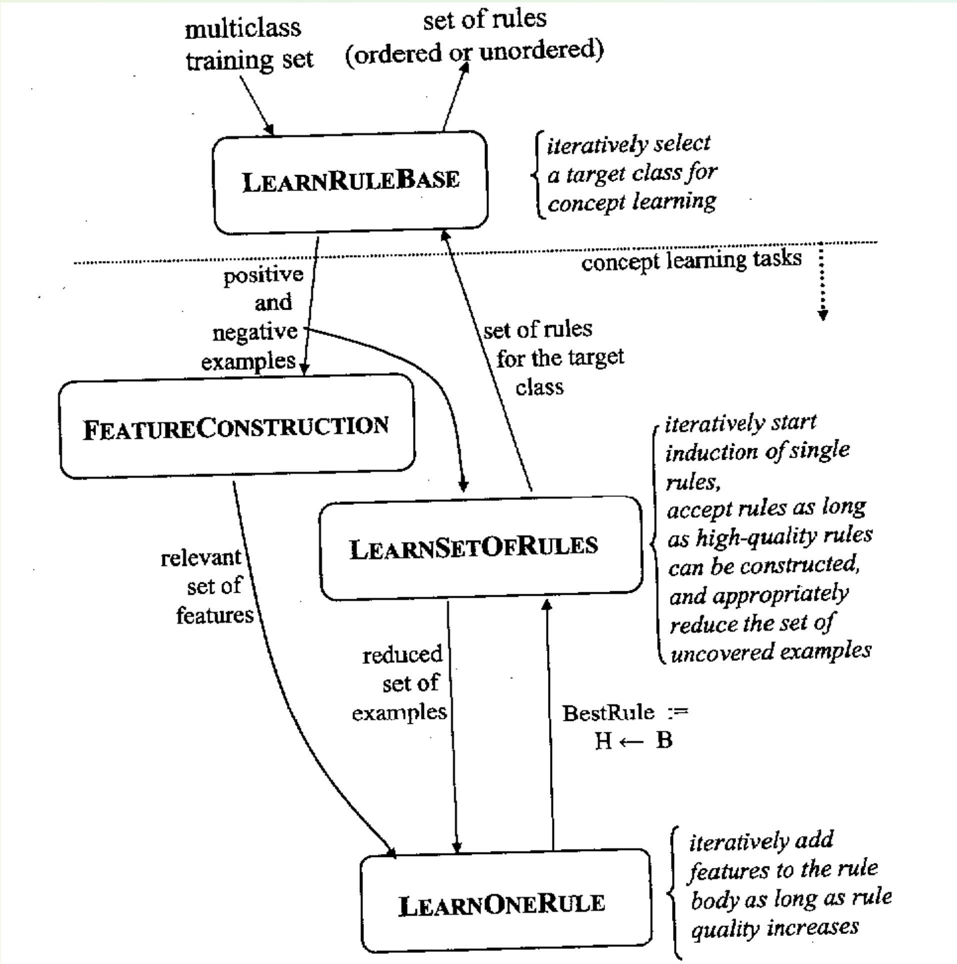
- *if* (*a1*=m) *then* (*d*=C1) {*x1*,*x3*,*x7* | *x6*} 3/4

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

Jak poszukiwać reguł?

- Typowo – sekwencyjne pokrywania – minimalny zbiór reguł:
  - np., AQ, CN2, LEM, PRISM, MODLEM, Inne – PVM, R1 lub RIPPER).
- Poszukiwanie innych zbiorów reguł
  - Satisfying some requirements (Explore, BRUTE, or modification of association rules, „Apriori-like”).
  - Based on local „reducts” → boolean reasoning or LDA.
- Klasyfikatory genetyczne
- Transformcje:
  - Trees → rules.
  - Construction of (fuzzy) rules from ANN.

Ogólne spojrzenia na popularne algorytmy



Given

- Data (set of learning examples) + hypotheses description language

Find a hypothesis in a form of a rule set, which is

- Complete, i.e. covers all or a subset of learning examples
- Consistent, i.e. predicts the correct class for all examples

- Search space, search strategy
- Rule quality measure
- Acceptance/stopping conditions
- Making prediction decisions

Problem indukcji reguł jako opisu pojęć

Wejście:

- Data (set of learning examples) + hypotheses description language

Znajdź

A hypothesis in a form of rule set R which is

- Complete, i.e. covers all the examples
- Consistent, i.e. predicts the correct class for all examples

Uwaga1 – Rozwiązanie optymalne - kosztowne!

Uwaga2 – dla rzeczywistych, niedoskonałnych danych uczących -> osłabienie wymagań

Indukcja reguł metodą generowania kolejnych pokryć

Sequential covering (*Xj* klasa; *A* atrybuty; *E* przykłady, τ próg akceptacji);

begin

    R := $\emptyset$;   {zbiór poszukiwanych reguł}

    r := *learn-one-rule*(klasa *Xj*; atrybuty *A*; przykłady *E*)

    while zbiór E niepusty and *evaluate*(*r*,*E*) > τ **do**

    begin

      R := $R \cup r$;

      E := $E \setminus [R]$;   {usuń przykłady pozytywne pokryte przez *R*}

      r := *learn-one-rule*(klasa *Xj*; atrybuty *A*; przykłady *E*);

    end;

return *R*

end.

- Funkcja *learn-one-rule* dla danego zbioru przykładów znajduje jedną regułę pokrywającą możliwie jak najwięcej przykładów pozytywnych i jak najmniej negatywnych.

Przykład -Learning One Rule

```
function LearnOneRule(Target, Attrs, Examples):
   NewRule := "IF true THEN Class-pos"
   NewRuleNeg := Neg
   while NewRuleNeg not empty, do
      // add a new elementary condition to the rule
         Candidates := generate candidate conditions L
         Best_cond := argmax_{L∈Candidates} performance(Specialise(NewRule,L))
         NewRule := Specialise(NewRule, Best_cond)
         NewRuleNeg := {x∈Neg | x covered by NewRule}
   return NewRule

function Specialise(Rule, L):
   let Rule = "IF conditions THEN pos"
   return "IF conditions and L THEN pos"
```

Look for one rule that has

High accuracy When it predicts something, it should be correct

Any coverage Does not make a prediction for all examples, just for some of them

The contact lenses data – alg. PRISM

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

Przykład: contact lens data 2

- Rule we seek:
- Possible conditions:

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

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

ACK: slides coming from witten&eibe WEKA

Specjalizacja koniunkcji

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

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

- Examples covered by 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

Dalsza specjalizacja, 2

- 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

Dwa warunki elementarne

- 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

Kolejny wybór warunku, 4

- 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

Znaleziona reguła

- 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

Reguły z PRISM dla wszystkich klas

1. If astigmatism = no
   and tear-prod-rate = normal
   and spectacle-prescrip = hypermetrope
   then soft
2. If astigmatism = no
   and tear-prod-rate = normal
   and age = young then soft
3. If age = pre-presbyopic
   and astigmatism = no
   and tear-prod-rate = normal then soft
4. If astigmatism = yes
   and tear-prod-rate = normal
   and spectacle-prescrip = myope then
   hard
5. If age = young
   and astigmatism = yes
   and tear-prod-rate = normal then hard
6. If tear-prod-rate = reduced then none
7. If age = presbyopic
   and tear-prod-rate = normal
   and spectacle-prescrip = myope
   and astigmatism = no then none
8. If spectacle-prescrip = hypermetrope
   and astigmatism = yes
   and age = pre-presbyopic then none
9. If age = presbyopic
   and spectacle-prescrip = hypermetrope
   and astigmatism = yes then none

Ocena kandydatów w funkcji „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: $$\frac{n_K(R)+1}{n(R)+k}$$

    Good for uniform prior distribution of k classes
  - *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*).

Inne funkcje oceny reguły R i klasy 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.

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

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

Samodzielne zdania (LEM2 albo PRISM)

- Poszukaj sam reguł dla każdego pojęcia {+,-}

Zbiór przykładów uczących - J.R. Quinlan

no.	**Height**	**Hair**	**Eyes**	**Attractiveness**
1	short	blond	blue	+
2	tall	blond	brown	-
3	tall	red	blue	+
4	short	dark	blue	-
5	tall	dark	blue	-
6	tall	blond	blue	+
7	tall	dark	brown	-
8	short	blond	brown	-

Zbiór reguł z alg. LEM2:

r1: (Hair=Dark) (Attractiveness,-)

Pokryte przykłady {4,5,7}

r2: (Hair=Red) (Attractiveness,+)

Pokryte przykłady{3}

r3: (Hair=Blond)(Eyes=Blue) (Attractiveness,+)

Pokryte przykłady {1,6}

r4: (Hair=blond)(Eyes=Brown) (Attractiveness,-)

Pokryte przykłady {2,8}

MODLEM – algorytm indukcji reguł

- MODLEM [Stefanowski 98] → minimalny zbiór reguł.
- Ocena warunków elementarnych → entropia.
- Niespójne dane → zbiory przybliżone (**rough sets)** → przybliżenia klas decyzyjnych → indukcja pewnych i możliwych reguł.
- Przetwarzanie atrybutów nominalnych i liczbowych.
- W połączeniu ze strategiami klasyfikacyjnymi → skuteczny klasyfikator
  - Dopasowanie opisu obiektu do części warunkowych.
  - Niejednoznaczności → decyzja większościowa

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

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

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

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

Przykład dla MODLEM-a (1)

No.	Age	Job	Period	Income	Purpose	Dec.
1	m	u	0	500	K	r
2	sr	p	2	1400	S	r
3	m	p	4	2600	M	d
4	st	p	16	2300	D	d
5	sr	p	14	1600	M	p
6	m	u	0	700	W	r
7	sr	b	0	600	D	r
8	m	p	3	1400	D	p
9	sr	p	11	1600	W	d
10	st	e	0	1100	D	p
11	m	u	0	1500	D	p
12	m	b	0	1000	M	r
13	sr	p	17	2500	S	p
14	m	b	0	700	D	r
15	st	p	21	5000	S	d
16	m	p	5	3700	M	d
17	m	b	0	800	K	r

Class (Decision = r)

E = {1, 2, 6, 7, 12, 14, 17}

List of candidates

(Age=m) {1,6,12,14,17+; 3,8,11,16-}
(Age=sr) {2,7+; 5,9,13-}

(Job=u) {1,6+; 11-}
(Job=p) {2+, 3,4,8,9,13,15,16-}
(Job=b) {7,12,14,17+; ∅}

(Pur=K) {1,17+; ∅}
(Pur=S) {2+;13,15-}
{Pur=W} {6+, 9-}
{Pur=D} {7,14+; 4,8,10,11-}
{Pur=M} {12+;5,16-}

MODLEM (2)

- Numerical attributes: Income

500	600	700	800	1000	1100	1400	1500	1600	2300	2500	2600	3700	5000
1+	7+	6+ 14+	17+	12+	10-	2+ 8-	11-	9- 5-	4-	13-	3-	10-	15-

(Income < 1050) {1,6,7,12,14,17+;∅}

(Income < 1250) {1,6,7,12,14,17+;10-}

(Income < 1450) {1,2,6,7,12,14,17+;8,10-}

Period

(Period < 1) {1,6,7,14,17+;10,11-}

(Period < 2.5) {1,2,6,7,12,14,17+;10,11-}

MODLEM (3) – minimalny zbiór reguł

- if (Income<1050) then (Dec=r) [6]
- if (Age=sr) and (Period<2.5) then (Dec=r) [2]
- if (Period∈[3.5,12.5)) then (Dec=d) [2]
- if (Age=st) and (Job=p) then (Dec=d) [3]
- if (Age=m) and (Income∈[1050,2550)) then (Dec=p) [2]
- if (Job=e) then (Dec=p) [1]
- if (Age=sr) and (Period≥12.5) then (Dec=p) [2]
- Niespójne (sprzeczne dane):
  - Przybliżenia klas decyzyjnych (rough sets)
  - „Rule post-processing (a kind of post-pruning) or extra testing and earlier acceptance of rules” → częściej stosowane.

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

Analiza diagnostycznej bazy danych

- Bada się stan techniczny 76 autobusów tego samego typu (dokładnie ich silników) na podstawie symptomów stanu technicznego - parametrów pochodzących z okresowych badań diagnostycznych
  - Autobusy są podzielone na dwie klasy: dobry i zły stan techniczny pojazdu
- Cel analizy
  - Ocenia się jakość diagnostyczną symptomów stanu technicznego
  - Poszukuje się zależności pomiędzy wartościami najistotniejszych w tych symptomów a przydziałem do klas
  - Konstruuje się klasyfikator stanu technicznego

Rozważane symptomy

s1 – prędkość maksymalna [km/h],

s2 – ciśnienie sprężania [Mpa],

s3 – zawartość elementów smołowatych w spalinach wylotowych [%],

s4 – moment obrotowy silnika [Nm],

s5 – letnie zużycie paliwa [l/100lm],

s6 – zimowe zużycie paliwa [l/100km],

s7 – zużycie oleju [l/1000km],

s8 – aktualna moc silnika [KM].

Dwie klasy decyzyjne:

1. Autobusy z silnikami w dobrym stanie – dalsza eksploatacja (46),
2. Autobusy z silnikami w złym stanie – konieczność napraw (30).

Minimalny zbiór reguł klasyfikujących

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

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

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

Oszacowana trafność klasyfikowania
(‘leaving one out’ test) 98.7%.

Poszukiwanie zbioru reguł silnych

Próg satysfakcji (51%):

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]

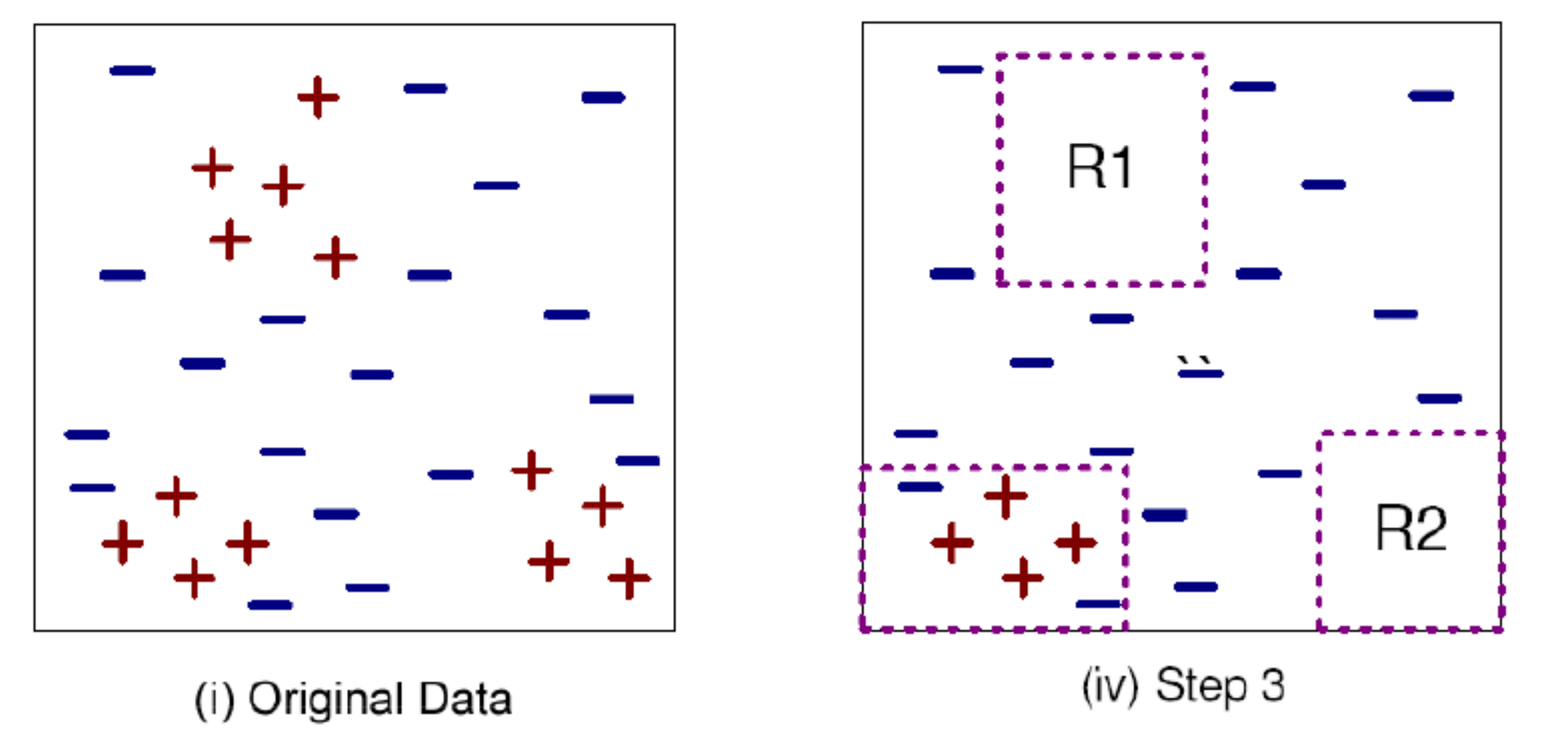
9. if (*s7*≥2.1 *l*/1000km) then (technical state=bad) [24]

10. if (*s3*≥61 %) & (*s4*≤444 Nm) then (technical state=bad) [28]

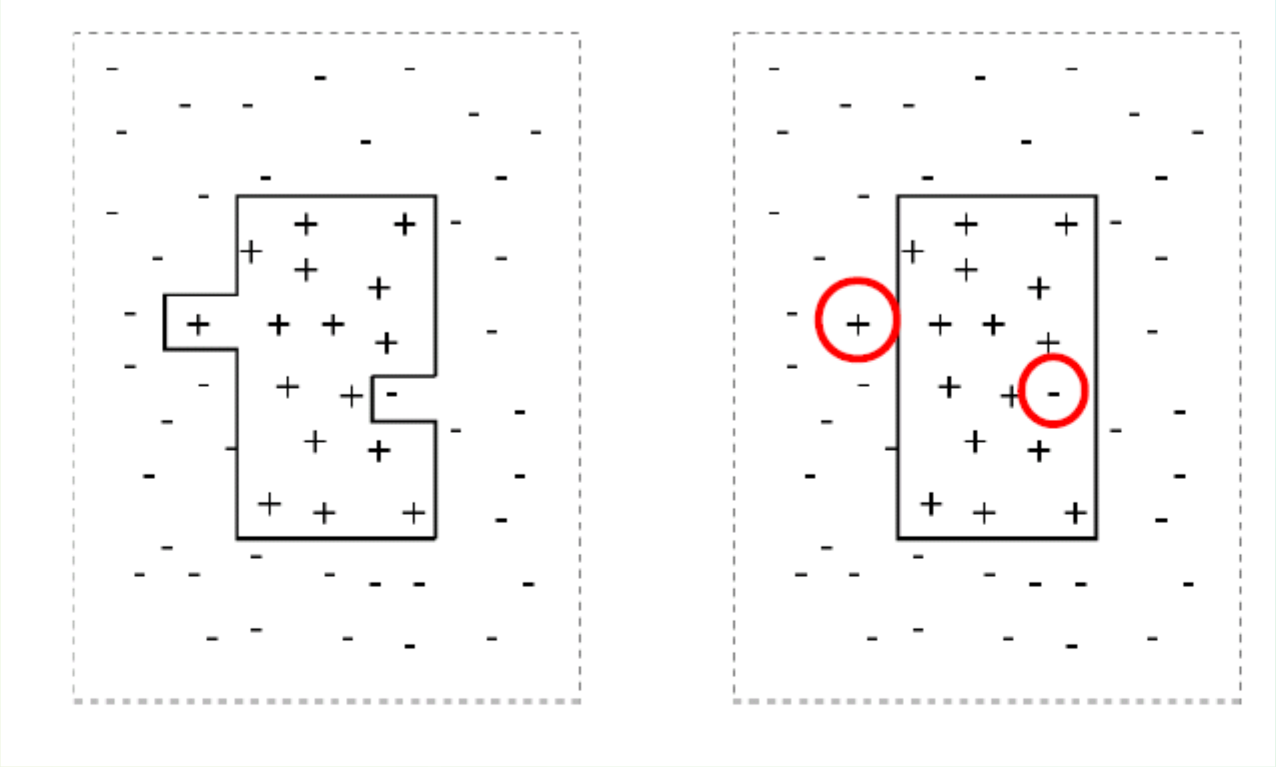
11. if (*s3*³61 %) & (*s8*<120 kM) then (technical state=bad) [27]

Czy zawsze kompletny i dokładny zbiór reguł?

- Nie zawsze, zależy od przykładów oraz dążenia do mniejszego zbioru reguł → osłabienie warynków completeness and consistency



Unikaj przeuczenia



- *R* nie pokrywa wszystkich przykładów pozytywnych
- Pojedyncza reguła *r* może pokrywać ograniczoną liczbę przykładów negatywnych

Znane techniki upraszczania zbioru reguł

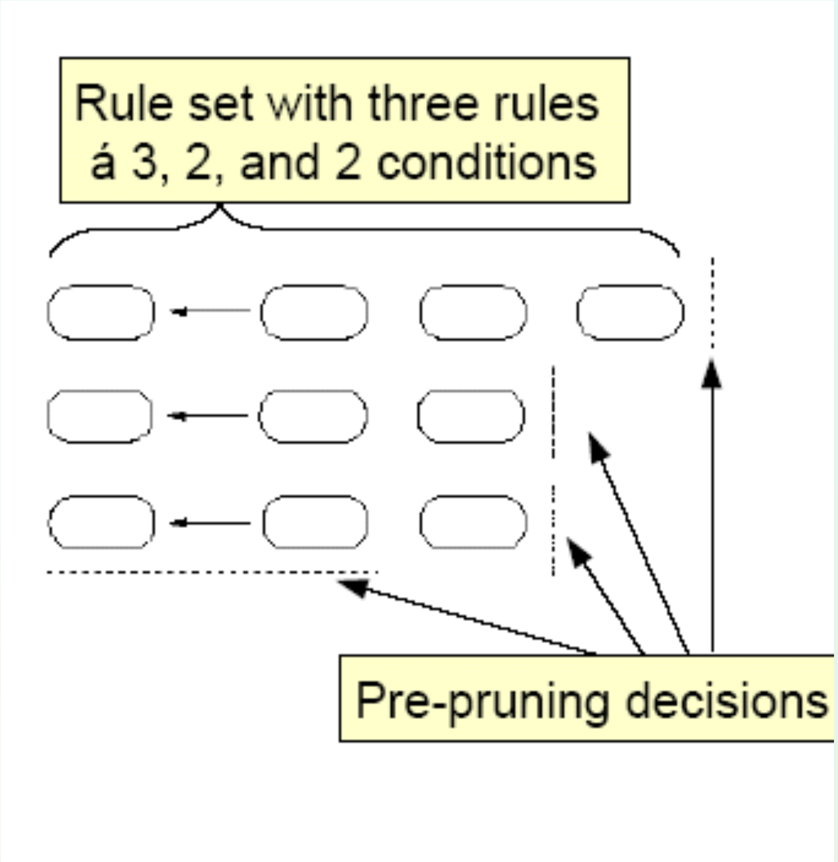
- Bias the learning towards simple concept descriptions
  - a short rule that covers many positive examples (but possibly also a few negatives) is often better than a long rule that covers only a few positive examples
- **Pre-pruning:** stop learning the decision rules before they reach the point where they too „perfectly re-classify” the training data
- **Post-pruning:** allow the rules to overfit the training data (induce complete and „consistent” set of rules), and then post-prune the rules

Pre-pruning

Ogólna idea

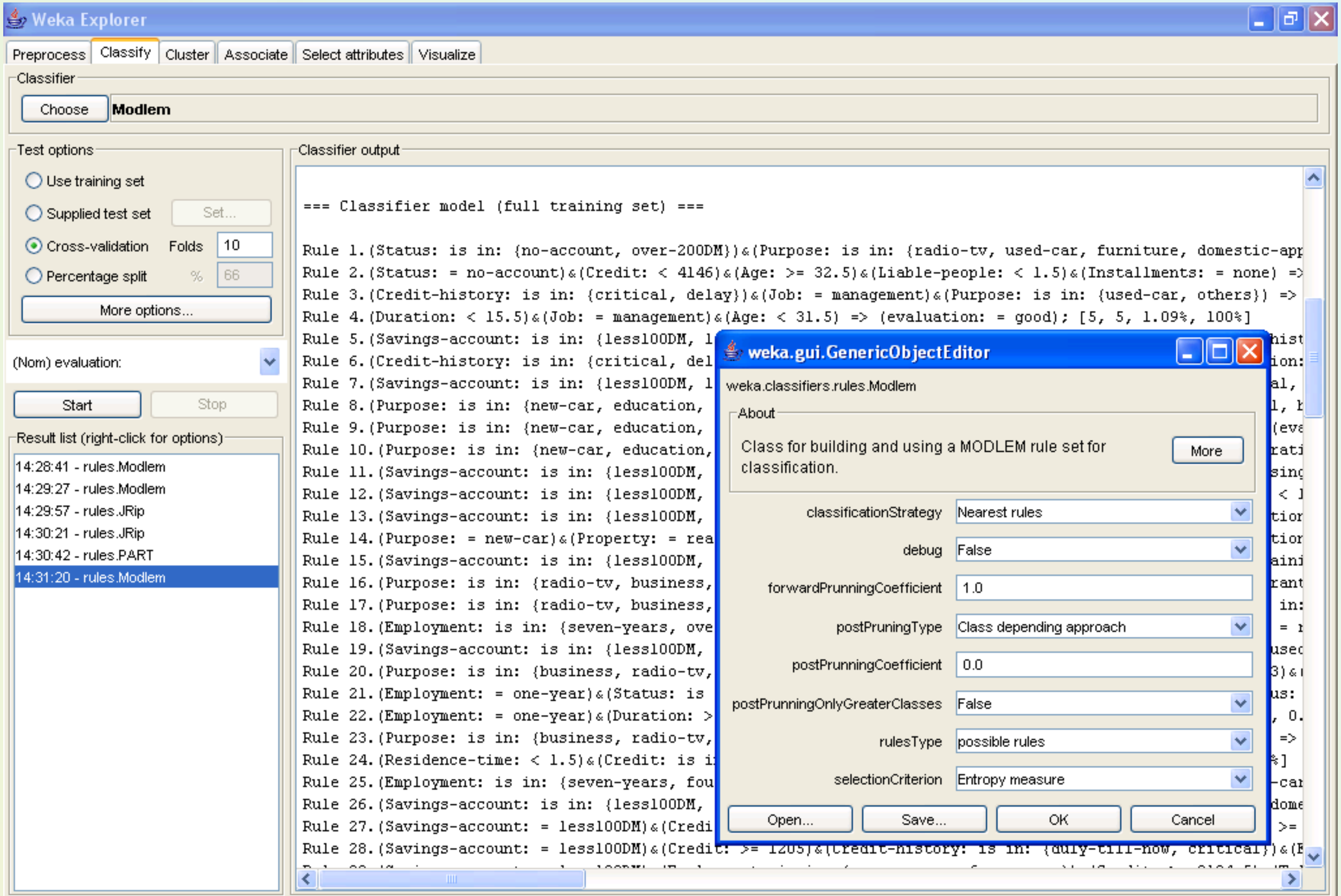
- decide when to stop adding conditions to a rule (*relax consistency* constraint)
- decide when to stop adding rules to a rule set (*relax completeness* constraint)

Computationally efficient but less accurate

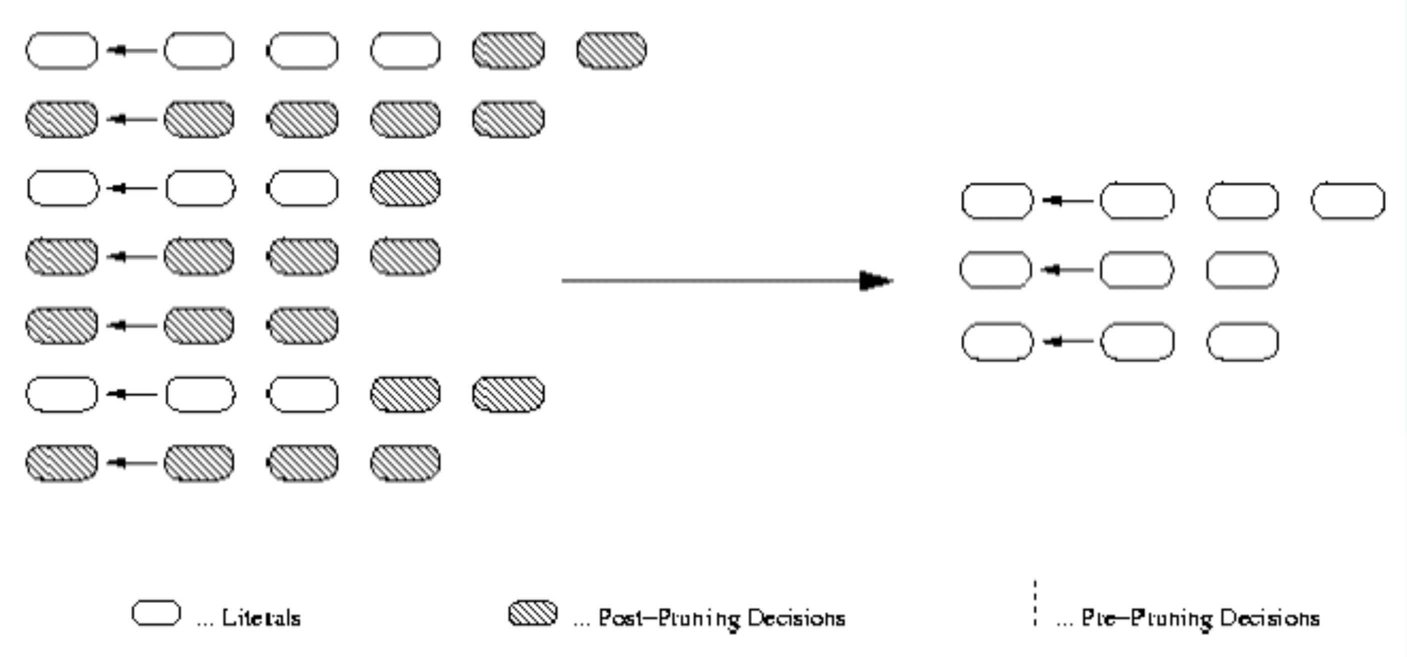


Przykład z MODLEM

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



Post-pruning – różne możliwości



- Delete conditions (any, last, worse)
- Delete a rule or other pruning operators

Lecz – jakie kryteria oceny I heurystyki redukcji kosztów obliczeniowych

Post-Pruning (REP)

1. Split all training examples into *Growing* (validation) *Set* (2/3) and *Pruning Set* (1/3);
2. Learn a complete and consistent 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*;

REP has a time complexity of $O(n^4)$ + inital rule induction of $O(n^2)$; alternative concept of generalization in GROW – more accurate but still costly

RIPPER

1. IREP* is used to obtain a rule set
2. Rule optimization takes place
3. IREP* is used to cover remaining positive examples

→ **R**epeated **I**ncremental **P**runing to **P**roduce **E**rror **R**eduction

Fast Effective Rule Induction: William W. Cohen

Important → rules re-ordered in a special multiclass list

Klasyfikowanie nowych obiektów

- Dopasowanie obiektu do części warunkowej reguły – pełne vs. częściowe.
- Klasyfikowanie za pomocą uporządkowanego zbioru reguł - wykorzystanie listy decyzyjnej (obecność reguły domyślnej).
- Klasyfikowanie obiektu za pomocą nieuporządkowanego zbioru reguł:
  - Część warunkowa dokładnie jednej reguły jest całkowicie dopasowana do obiektu, który jest zaklasyfikowany do klasy wskazywanej przez tę regułę.
  - Część warunkowa więcej niż jednej reguły jest całkowicie dopasowana do opisu obiektu.
  - Część warunkowa żadnej reguły nie jest dopasowana do obiektu.

Priority decision list (C4.5 rules)

C4.5 VOTE (16 attributes, 300 training cases, 135 test case

Data Tree Rules Cross-validation Special Help

	Before pruning			After pruning	
Tree	Size	Errors	Errors (test)	Size	Errors
1	16	8 (3.0%)	1 (3.3%)	7	12 (
2	28	7 (2.6%)	2 (6.7%)	7	13 (
3	16	9 (3.3%)	0 (0.0%)	7	13 (
4	25	5 (1.9%)	2 (6.7%)	4	12 (
5	22	7 (2.6%)	3 (10.0%)	7	11 (
6	19	9 (3.3%)	2 (6.7%)	7	11 (
7	28	7 (2.6%)	2 (6.7%)	7	13 (
8	22	7 (2.6%)	3 (10.0%)	7	12 (
9	16	8 (3.0%)	3 (10.0%)	4	12 (
10	25	6 (2.2%)	4 (13.3%)	7	10 (
Avg.	21.7	7.3 (2.7%)	2.2 (7.3%)	6.4	11.9 (

Cross-validation (rules)

Ruleset	Size	Errors	Errors (test)
1	5	10 (3.7%)	1 (3.3%)
2	5	10 (3.7%)	1 (3.3%)
3	5	11 (4.1%)	0 (0.0%)
4	4	10 (3.7%)	3 (10.0%)
5	5	9 (3.3%)	2 (6.7%)
6	4	11 (4.1%)	2 (6.7%)
7	5	11 (4.1%)	0 (0.0%)
8	5	10 (3.7%)	1 (3.3%)
9	2	12 (4.4%)	3 (10.0%)
10	3	11 (4.1%)	2 (6.7%)

Rules

```
Rule 1: [98.4%]
    IF    physician fee freeze = n
    THEN  democrat

Rule 2: [94.7%]
    IF    mx missile = y
    AND   synfuels corporation cutback = y
    THEN  democrat

Rule 3: [63.0%]
    IF    physician fee freeze = u
    AND   mx missile = n
    THEN  democrat

Rule 4: [94.0%]
    IF    physician fee freeze = y
    AND   immigration = y
    THEN  republican

Rule 5: [91.2%]
    IF    physician fee freeze = y
    AND   mx missile = n
    THEN  republican

Rule 6: [82.0%]
    IF    adoption of the budget resolution = n
    AND   education spending = u
    THEN  republican

Rule 7: [50.0%]
    IF    physician fee freeze = u
    AND   mx missile = u
    THEN  republican

Default class: democrat

Errors in training set: 11 (3.7%)
Errors in test     set: 6 (4.4%)
```

Click here to show confusion matrixes

Confusion matrix (test set)

Org. \ C4.5	democrat	republican
democrat	18	1
republican		11

Statistica Klasyczna

DataMiner7

calculator

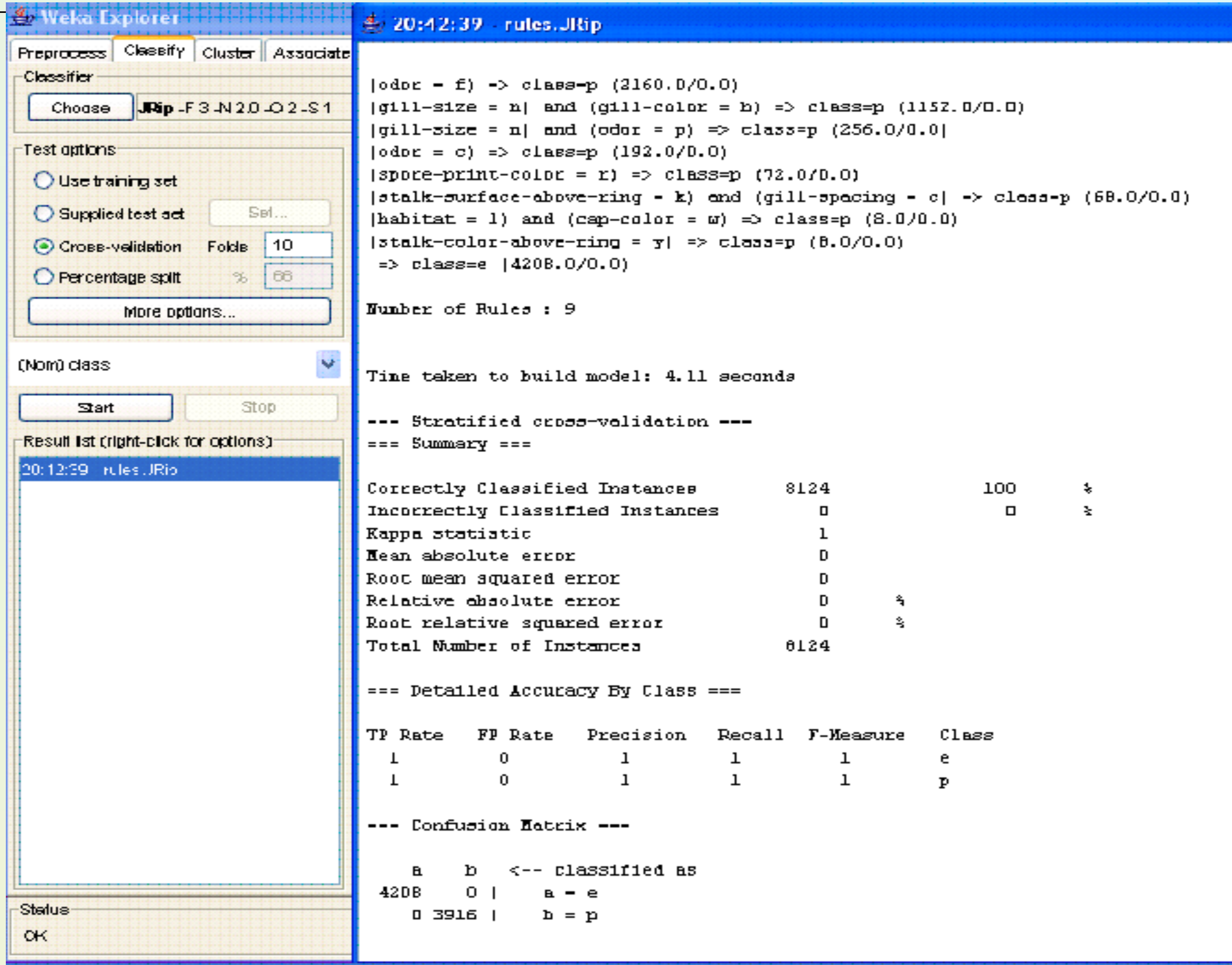
Klasyfikacja z listą uporzadowanych reguł

- Rules are rank ordered according to their priority
  - An ordered rule set is known as a decision list
- When a test record is presented to the classifier
  - It is assigned to the class label of the highest ranked rule which has matched
  - If none of the rules fired, it is assigned to the default class

R1: (Give Birth = no) $\wedge$ (Can Fly = yes) $\rightarrow$ Birds
R2: (Give Birth = no) $\wedge$ (Live in Water = yes) $\rightarrow$ Fishes
R3: (Give Birth = yes) $\wedge$ (Blood Type = warm) $\rightarrow$ Mammals
R4: (Give Birth = no) $\wedge$ (Can Fly = no) $\rightarrow$ Reptiles
R5: (Live in Water = sometimes) $\rightarrow$ Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

Specific list of rules - RIPPER (Mushroom data)



CN2 – unordered rule set

WinCn2 16 attributes (crx.aex) 490 examples (crx.aex) 30 rules (induced)

Data Rules Cross-validation Trace Output

Unordered | Laplacian | Unset | 5 | 0.05 | 10 | 0

```
Reading attributes and examples...
490 examples!
Finished reading attribute and example file!
Running CN on current example set...
Finished inducing rules!

     *-----------------------*
     |  UN-ORDERED RULE LIST  |
     *-----------------------*

IF    A8 < 10.75
  AND A9 = T
  AND 5.50 < A11 < 18.50
THEN  DECISION = Y  [68 0]

IF    A15 > 5676.00
THEN  DECISION = Y  [19 0]

IF    A2 > 19.00
  AND A4 = U
  AND A8 < 11.75
  AND A9 = T
  AND A14 < 91.00
THEN  DECISION = Y  [67.50 0]

IF    A3 > 1.79
  AND A9 = T
  AND A15 > 241.50
THEN  DECISION = Y  [80 0]

IF    A6 = X
  AND 1.33 < A8 < 7.88
THEN  DECISION = Y  [11 0]

IF    A2 < 26.08
  AND A9 = T
  AND 20.00 < A14 < 106.00
THEN  DECISION = Y  [32.50 0]

IF    A8 > 12.75
  AND A14 < 187.00
THEN  DECISION = Y  [12 0]
```

Lister - [c:\Usr\Jurek\students\CichyCN2\Cn2\Exe\Examples\crx.aex]

Plik Edytuj Opcje Pomoc

```
**ATTRIBUTE AND EXAMPLE FILE**

A1: B A;
A2: (FLOAT)
A3: (FLOAT)
A4: U Y L;
A5: G P GG;
A6: W Q M R CC K C D X I E AA FF J;
A7: V H BB FF J Z O DD N;
A8: (FLOAT)
A9: T F;
A10: T F;
A11: (FLOAT)
A12: F T;
A13: G S P;
A14: (FLOAT)
A15: (FLOAT)
DECISION: Y N;

@

B 30.83 0 U G W V 1.25 T T 1 F G 202 0 Y;
A 58.67 4.46 U G Q H 3.04 T T 6 F G 43 560 Y;
A 24.50 .5 U G Q H 1.5 T F 0 F G 280 824 Y;
B 27.83 1.54 U G W V 3.75 T T 5 T G 100 3 Y;
B 20.17 5.625 U G W V 1.71 T F 0 F S 120 0 Y;
B 32.08 4 U G M V 2.5 T F 0 T G 360 0 Y;
B 33.17 1.04 U G R H 6.5 T F 0 T G 164 31285 Y;
A 22.92 11.585 U G CC V .04 T F 0 F G 80 1349 Y;
B 54.42 .5 Y P K H 3.96 T F 0 F G 180 314 Y;
B 42.50 4.915 Y P W V 3.165 T F 0 T G 52 1442 Y;
```

Użycie nieuporządkowanych reguł (no list)

- 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.
- An example:
- 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!

Rozwiązywanie konfliktowych sytuacji

- Przykład LERS classification strategy (Grzymala)
  - 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)
- Instead of MF use a kind of normalized distance *x* to conditions of *r*

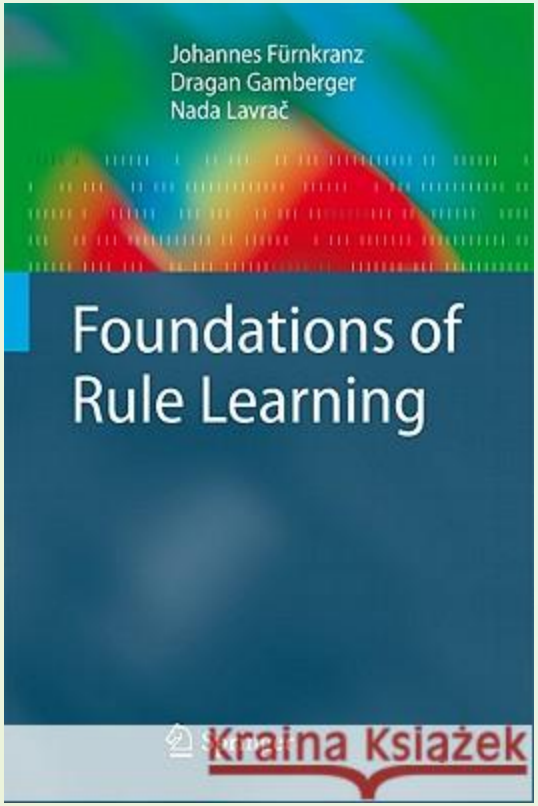
Ocena eksperymentalna LERS

- 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

Książki



- Johannes Furnkranz, Draqan Gamberger, Nada Lavrac Foundations of Rule Learning, Springer 2012

Literatura

- 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.

Literatura- 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.

Pytania?

?