

SID – Wykład 10

Systemy uczące się

Dominik Ślęzak

Wydział Matematyki, Informatyki i Mechaniki UW
slezak@mimuw.edu.pl



Uczenie indukcyjne

Obiekty:

dane reprezentujące rzeczywisty stan lub obiekt, tworzą przestrzeń obiektów X

Decyzja:

Funkcja $dec : X \rightarrow V_{dec}$ przypisująca obiektom wartość decyzji z ustalonego zbioru V_{dec}

Zbiór przykładów:

ustalony zbiór obiektów z X z przypisanymi wartościami decyzji:
 $(x_1, dec(x_1)), \dots, (x_m, dec(x_m))$

Problem:

Z danego zbioru przykładów nauczyć się funkcji (hipotezy) $h : X \rightarrow V_{dec}$ aproksymującej decyzję dec tak, żeby możliwie najbardziej poprawnie przypisywała ją obiektom z przestrzeni X , dla których nieznana jest wartość decyzji dec



Uczenie indukcyjne: uprawianie sportu

Obiekty to wektory wartości opisujące bieżące warunki pogodowe

| Przykład | Atrybuty | | | | | | Decyzja |
|----------|--------------|-------------|----------------|--------------|-------------|-----------------|--------------|
| | <i>Niebo</i> | <i>Temp</i> | <i>Wilgotn</i> | <i>Wiatr</i> | <i>Woda</i> | <i>Prognoza</i> | <i>Sport</i> |
| Dzień 1 | Słońce | Ciepło | Normalna | Silny | Ciepła | Bez zmian | Tak |
| Dzień 2 | Słońce | Ciepło | Wysoka | Silny | Ciepła | Bez zmian | Tak |
| Dzień 3 | Deszcz | Zimno | Normalna | Silny | Ciepła | Zmiana | Nie |
| Dzień 4 | Słońce | Ciepło | Wysoka | Silny | Chłodna | Zmiana | Tak |

Problem polega na nauczeniu się podejmowania właściwego wyboru na podstawie bieżących warunków pogodowych.



Rodzaje decyzji i atrybuty

Decyzja może przyjmować wartości:

- rzeczywiste (decyzja ciągła)
- dyskretne
- binarne (*TRUE* lub *FALSE*)
- złożone

Atrybuty: zbiór atrybutów $A = \{a_1, a_2, \dots, a_n\}$

Obiekty: Wektory wartości atrybutów $x = \langle x_1, x_2, \dots, x_n \rangle$

Wartości atrybutów mogą być:

- ciągłe
- dyskretne
- binarne (*TRUE* lub *FALSE*)



Przestrzeń hipotez

Ile jest różnych hipotez (funkcji) binarnych dla n atrybutów binarnych?

= liczba funkcji binarnych dla dziedziny z 2^n obiektami = 2^{2^n}

np. dla 6 atrybutów binarnych jest 18.446.744.073.709.551.616 hipotez

Ograniczanie przestrzeni hipotez:

Przestrzeń hipotez można ograniczyć do ustalonej klasy hipotez

Ile jest czysto koniunkcyjnych funkcji (np. Hungry \wedge \neg Rain?)

Każdy atrybut może wystąpić jako literał pozytywny, jako literał negatywny lub wcale

$\Rightarrow 3^n$ różnych funkcji koniunkcyjnych

Zwiększanie przestrzeni hipotez

Co daje zwiększenie klasy dopuszczalnych hipotez?

- zwiększa szansę, że funkcja docelowa może być wyrażona
- zwiększa liczbę hipotez zgodnych ze zbiorem treningowym

\Rightarrow może spowodować gorszą skuteczność predykcji



Empiryczna miara jakości hipotezy

Dane dzielone są na zbiór treningowy U_{trn} i zbiór testowy U_{tst}

Hipoteza $h : X \rightarrow V_{dec}$ jest indukowana na podstawie zbioru treningowego U_{trn}

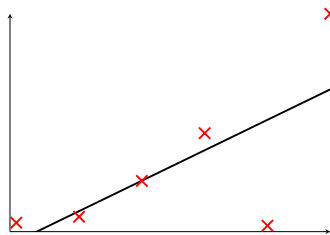
Skuteczność hipotezy $Accuracy(h)$ jest mierzona proporcją poprawnie sklasyfikowanych obiektów ze zbioru testowego

$$Accuracy(h) = \frac{|\{x \in U_{tst} : h(x) = dec(x)\}|}{|U_{tst}|}$$

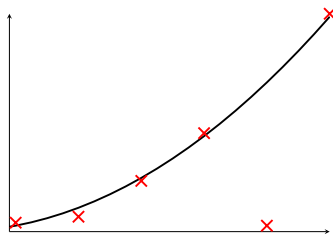


Regresja (decyzja ciągła)

Regresja to aproksymacja decyzji o wartości ciągłej



Regresja liniowa

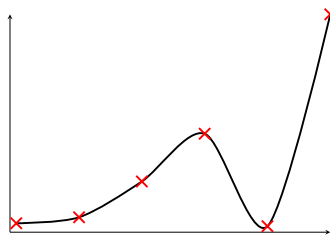


Regresja kwadratowa

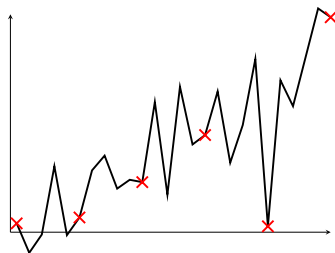


Regresja (decyzja ciągła)

Regresja to aproksymacja decyzji o wartości ciągłej



Regresja wyższego stopnia (spójna)



Regresja kawałkami liniowa (spójna)



Modele decyzyjne

Metody wnioskowania dla danych opisanych przez zbiór atrybutów z decyzją dyskretną:

- Drzewa decyzyjne
- Systemy regułowe
- Sieci neuronowe
- Sieci bayessowskie



Drzewa decyzyjne: reprezentacja

Węzły wewnętrzne

Każdy związany z jednym atrybutem,
reprezentuje test wartości tego atrybutu

Gałęzie

Każda związana z jedną wartością lub z podzbiorem wartości atrybutu
węzła, z którego wychodzi gałąź,
odpowiada obiektom danych z pasującymi wartościami atrybutu

Liście

Każdy związany z decyzją lub rozkładem decyzji,
odpowiada obiektom danych pasującym do ścieżki prowadzącej do
danego liścia



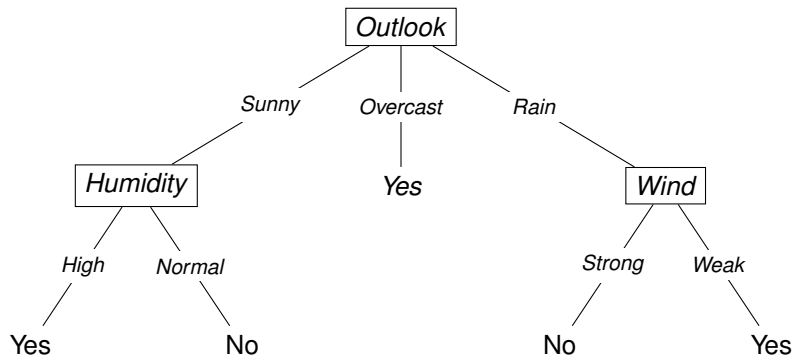
Drzewa decyzyjne: przykład

Reprezentacja danych

| Day | Outlook | Temperature | Humidity | Wind | PlayTennis |
|-----|----------|-------------|----------|--------|------------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |



Drzewa decyzyjne: przykład



Drzewa decyzyjne: moc wyrażania

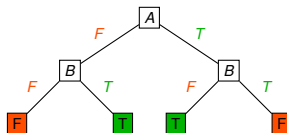
Fakt 1: Dla każdego zbioru treningowego istnieje spójne drzewo decyzyjne

Dowód: zaczynamy od pustego drzewa i dla każdego obiektu danych dokładamy ścieżkę prowadzącą przez wszystkie atrybuty z wartościami krawędzi odpowiadającymi wartościom atrybutów w obiekcie

Fakt 2: Dla każdej funkcji istnieje spójne drzewo decyzyjne

Dowód: można utworzyć zbiór treningowy zawierający obiekty odpowiadające wszystkim kombinacjom wartości atrybutów o decyzji zgodnej z daną funkcją

| A | B | A xor B |
|---|---|---------|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | F |



Drzewa decyzyjne: trenowanie

Cel: znalezienie drzewa spójnego ze zbiorem treningowym

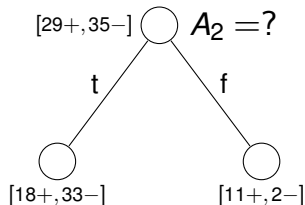
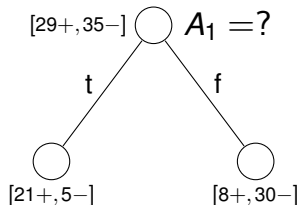
Pomysł: rekurencyjne wybieranie najbardziej znaczącego atrybutu jako korzenia poddrzewa

```
function DTL(examples, attributes, default) returns a decision tree
  if examples is empty then
    return default
  else if all examples have the same classification then
    return the classification
  else if attributes is empty then
    return MODE(examples)
  else
    best ← CHOOSE-ATTRIBUTE(attributes, examples)
    tree ← a new decision tree with root test best
    for each value  $v_i$  of best do
      examplesi ← { elements of examples with best =  $v_i$  }
      subtree ← DTL(examplesi, attributes – best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
    end for
    return tree
  end if
end function
```



Drzewa decyzyjne: wybór atrybutu

Różne atrybuty dają różne rozkłady decyzji w gałęziach



Funkcja CHOOSE-ATTRIBUTE wybiera najlepszy z nich



Entropia

Dany jest rozkład prawdopodobieństwa $\langle p_1, \dots, p_n \rangle$

Miara entropii wyznacza, ile informacji niesie z sobą ten rozkład

$$\text{Entropy}(\langle p_1, \dots, p_n \rangle) = \sum_{i=1}^n -p_i \log_2 p_i$$

S — zbiór obiektów danych

S_d — zbiór obiektów w S z decyzją d

$$\text{Entropy}(S) = \sum_{d \in V_{dec}} -\frac{|S_d|}{|S|} \log_2 \frac{|S_d|}{|S|}$$

Entropia = średnia liczba bitów potrzebna do zakodowania decyzji d dla losowo wybranego obiektu ze zbioru S (przy optymalnym kodowaniu decyzji)

Dlaczego?

Optymalne kodowanie przydziela $-\log_2 p$ bitów do decyzji występującej z prawdopodobieństwem p



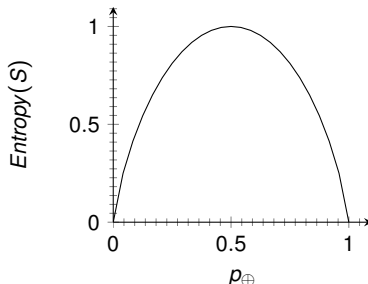
Entropia: dwie decyzje

Dane są dwie decyzje: pozytywna (\oplus) i negatywna (\ominus)

$p_{\oplus} = \frac{|S_{\oplus}|}{|S|}$ — proporcja obiektów z decyzją pozytywną w zbiorze S

$p_{\ominus} = \frac{|S_{\ominus}|}{|S|}$ — proporcja obiektów z decyzją negatywną w zbiorze S

$$Entropy(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$



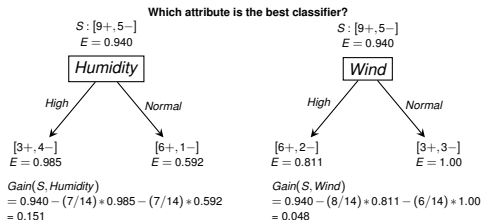
Zysk informacji dla atrybutu symbolicznego

Zysk informacji

$Gain(S, a)$ = redukcja entropii przy podziale zbioru względem atrybutu a

S_v – zbiór obiektów w S z wartością atrybutu $a = v$

$$Gain(S, a) = Entropy(S) - \sum_{v \in Values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$



Drzewo decyzyjne w każdym węźle wybiera atrybut a z największym zyskiem informacji, tzn. z największą wartością $Gain(S, a)$.



Reguły

Warunek

Koniunkcja selektorów, każdy selektor reprezentuje test wartości pojedynczego atrybutu, warunek odpowiada obiektom spełniającym wszystkie selektory.

Decyzja

Każda reguła związana jest z jedną decyzją, przypisywaną obiektom spełniającym warunek reguły.

Przykład

$Wind = Weak \wedge Temp > 20 \wedge Outlook \neq Rain \Rightarrow PlayTennis = Yes$



Reguły: selektory

Atrybuty symboliczne:

- Selektor równościowy $X = v$
- Selektor wykluczający $X \neq v$
- Selektor ogólny $X \in \{v_1, \dots, v_k\}$

Atrybuty numeryczne:

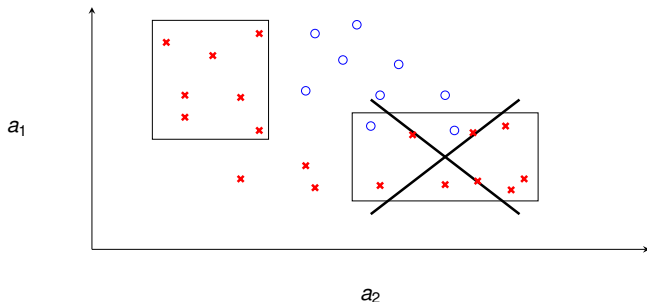
- Selektor przedziałowy $X \in (a, b)$

Przedział może być jednostronnie nieograniczony, może też być jedno- lub obustronnie domknięty.



Reguły spójne

Reguła $\alpha \implies dec = d$ jest spójna ze zbiorem treningowym U_{trn} jeśli każdy przykład $x \in U_{trn}$ spełniający warunek α ma decyzję $dec(x) = d$.



Reguły spójne: przykład

| Day | Outlook | Temperature | Humidity | Wind | PlayTennis |
|-----|----------|-------------|----------|--------|------------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

$Outlook = Overcast \Rightarrow PlayTennis = Yes$

$Humidity = Normal \Rightarrow PlayTennis = Yes$

spójna
niespójna, bo D6 sprzeczne



Systemy regułowe: uczenie i klasyfikacja

Uczenie

Generowanie zbioru reguł na podstawie zbioru przykładów treningowych.

Klasyfikacja

Wyszukiwane są reguły pasujące do klasyfikowanego obiektu x , tzn. te, których warunek jest spełniany przez obiekt x , możliwe są dwie strategie podejmowania decyzji:

1. Najlepszy wygrywa:

regułom przypisana jest miara ważności *Importance*,
decyzja podejmowana jest na podstawie pasującej do x reguły r o najwyższej wartości $Importance(r)$

2. Głosowanie:

reguły mają przypisane wagi *Weight*,
obiekt x klasyfikowany jest decyzją o najwyższej sumie wag reguł pasujących
$$\max \arg_{d_j} \sum_{\alpha \Rightarrow d_j: x \text{ spełnia } \alpha} Weight(\alpha \implies d_j)$$



Generowanie reguł

Bezpośrednio ze zbioru przykładów

- zupełne
- sekwencyjne pokrywanie (CN2, AQ)

Przy użyciu struktur pośrednich

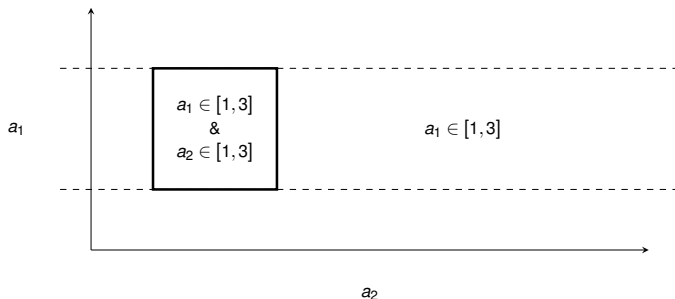
- z reduktu (teoria zbiorów przybliżonych)
- z drzewa decyzyjnego (C4.5rules)



Generowanie reguł – zupełne

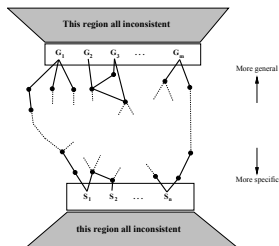
Fakt

Dla dowolnej reguły $s_1 \wedge \dots \wedge s_m \Rightarrow d$ wszystkie obiekty rozpoznawane przez nią rozpoznawane są także przez każdą regułę zbudowaną z podzbioru jej selektorów $s_{i_1} \wedge \dots \wedge s_{i_k} \Rightarrow d$



Generowanie reguł – zupełne

Wniosek: Obszar przestrzeni obiektów pokrywany przez wszystkie maksymalnie ogólne reguły spójne (G_1, \dots, G_m) jest taki sam jak obszar pokrywany przez wszystkie reguły spójne.



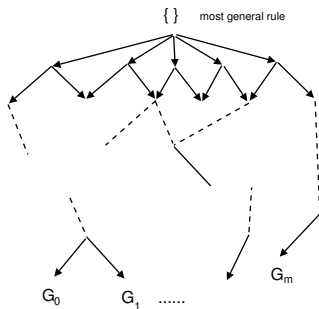
⇒ Wystarczy wyszukać wszystkie reguły spójne o minimalnym zbiorze selektorów, tzn. takim, że usunięcie dowolnego selektora daje regułę niespójną.



Generowanie reguł – zupełne

Jak to robić?

Można przeszukiwać przestrzeń wszystkich reguł zaczynając od reguł najbardziej ogólnych. Dopóki reguły nie są spójne ze zbiorem treningowym, są rozszerzane o selektory wykluczające przykłady powodujące ich niespójność.



Generowanie reguł – zupełne – algorytm

```
function EXHAUSTIVE-RULES(examples, decisions, selectors) returns a rule set
  rules  $\leftarrow$  {}
  for each decision  $d \in$  decisions do
    candidates  $\leftarrow$  {}  $\Rightarrow$   $d$ 
    repeat
      newCandidates  $\leftarrow$  {}
      for each candidate rule  $\alpha \Rightarrow d \in$  candidates do
         $e_{neg} \leftarrow$  a random example matching  $\alpha$  but with a decision  $\neq d$ 
        for each selector  $s \in$  selectors excluding  $e_{neg}$  do
           $r_{new} \leftarrow \alpha \wedge s \Rightarrow d$ 
          if  $r_{new}$  covers one or more objects with decision  $d$  in examples
            and is not subsumed by another rule from  $rules \cup newCandidates$  then
              if  $r_{new}$  is consistent with examples then
                 $rules \leftarrow rules \cup r_{new}$ 
              else
                 $newCandidates \leftarrow newCandidates \cup r_{new}$ 
            end if
          end if
        end for
      end for
      candidates  $\leftarrow newCandidates$ 
    until candidates is empty
  end for
  return rules
end function
```

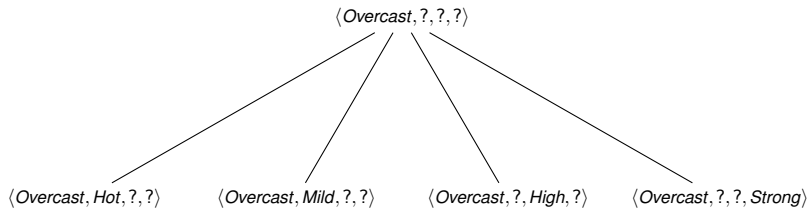


Generowanie reguł – zupełne – przykład

Reguły z decyzją $PlayTennis = Yes$

candidates: $Outlook = Overcast \Rightarrow PlayTennis = Yes$

Kontrprzykład: $\langle Overcast, Cool, Normal, Weak, PlayTennis = No \rangle$



Generowanie reguł – sekwencyjne pokrywanie

Generowanie reguł zupełne przegląda zazwyczaj wykładniczo dużą podprzestrzeń reguł, w praktyce niewykonalne.

Pomysł (heurystyczny): Reguły można generować pojedynczo do momentu pokrycia przez nie wszystkich obiektów treningowych.

```
function SEQUENTIAL-COVERING(examples) returns a rule set
  rules  $\leftarrow$  {}
  uncovered  $\leftarrow$  examples
  repeat
    r  $\leftarrow$  LEARN-ONE-RULE(examples, uncovered)
    rules  $\leftarrow$  rules  $\cup$  r
    remove all examples covered by r from uncovered
  until uncovered is empty
  return rules
end function
```

Funkcja LEARN-ONE-RULE wyszukuje heurystycznie jak najlepszą regułę względem pewnej miary jakości reguł.



Clark, Niblett, 1991

◇ Używa atrybutów symbolicznych

Traktuje wszystkie atrybuty jako symboliczne, atrybuty numeryczne zamieniane są na symboliczne w ten sposób, że zakres wartości każdego atrybutu dzielony jest na równe przedziały, wartości z jednego przedziału zamieniane są na taką samą wartość symboliczną

◇ Używa metody sekwencyjnego pokrywania

Szukanie kolejnej reguły (procedura LEARN-ONE-RULE) podobnie jak generowanie reguł zupełnie rozpoczyna od najbardziej ogólnych reguł (warunków) i uszczegóławia je dodając kolejne selektory, ale:

- zbiór reguł-kandydatów ograniczony jest do rozmiaru k określanego przez użytkownika, do rozszerzania brane są najlepsze kandydujące reguły
- jako wynik zwracana jest najlepsza reguła spośród wygenerowanych kandydatów



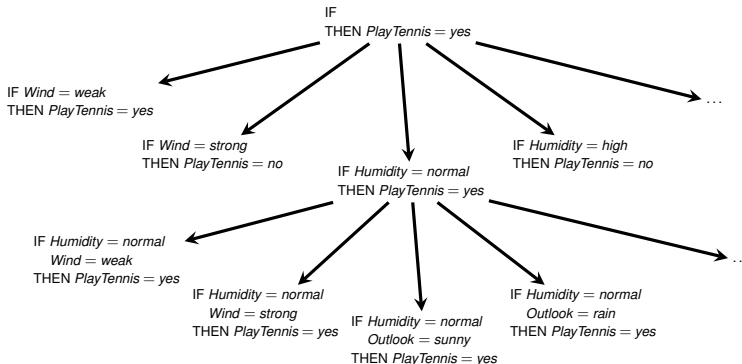
CN2: szukanie najlepszej reguły

```
function LEARN-ONE-RULE-CN2(uncov, k) returns a rule
  inputs: uncov - the examples not covered by the previous rules
           k - the width of searching
  best  $\leftarrow$  the most general empty condition
  candidates  $\leftarrow$  {best}
  repeat
    newCandidates  $\leftarrow$  {}
    for each candidate  $\alpha \in$  candidates do
      for each selector s of the form  $a = v$  or  $a \neq v$  consistent with  $\alpha$  do
        if  $\alpha \wedge s \notin$  candidates  $\cup$  newCandidates then
          newCandidates  $\leftarrow$  newCandidates  $\cup$  { $\alpha \wedge s$ }
          if PERFORMANCE( $\alpha \wedge s$ , uncov) > PERFORMANCE(best, uncov) then
            best  $\leftarrow$   $\alpha \wedge s$ 
          end if
        end if
      end for
    end for
    retain only k best candidates in newCandidates according to PERFORMANCE
    candidates  $\leftarrow$  newCandidates
  until candidates is empty
  return best  $\Rightarrow$  d (the most frequent decision among objects matching best)
end function
```



CN2: szukanie najlepszej reguły, przykład

Rozmiar zbioru kandydatów = 1 \Rightarrow przeszukiwanie zachłanne



CN2: miara jakości reguły

Funkcja $\text{PERFORMANCE}(\alpha, \text{uncov})$ szacuje jakość warunku α na podstawie dotychczas niepokrytych przykładów uncov .

n – liczba przykładów z uncov pasujących do α

n_d – liczba przykładów z uncov pasujących do α z najczęstszą decyzją d

m -estymata prawdopodobieństwa

$$\frac{n_d + mp_d}{n + m}$$

$\langle p_{d_1}, \dots, p_{d_D} \rangle$ – pierwotny rozkład prawdopodobieństwa w danych

m – parametr estymacji

CN2 używa szczególnego przypadku, estymaty Laplace'a:

równomierny rozkład pierwotny $\langle \frac{1}{D}, \dots, \frac{1}{D} \rangle$ i $m = D$ (D – liczba decyzji)

$$\frac{n_d + D \frac{1}{D}}{n + D} = \frac{n_d + 1}{n + D}$$



Inne miary jakości reguły

Funkcja $\text{PERFORMANCE}(\alpha, \text{uncov})$ szacuje jakość warunku α na podstawie dotychczas niepokrytych przykładów uncov

n – liczba przykładów z uncov pasujących do α

n_d – liczba przykładów z uncov pasujących do α z najczęstszą decyzją d

Częstość względna

$$\frac{n_d}{n}$$

Negacja entropii

$$\sum_{d_i} \frac{n_{d_i}}{n} \log_2 \frac{n}{n_{d_i}}$$

n_{d_i} – liczba przykładów z uncov pasujących do α z decyzją d_i



Miary jakości reguły: przykład

α_1 pokrywa 1000 przykładów z decyzją d_1 i 1 przykład z decyzją d_2

α_2 pokrywa 5 przykładów z decyzją d_1 i 0 przykładów z decyzją d_2

α_3 pokrywa 1 przykład z decyzją d_1 i 0 przykładów z decyzją d_2

| | $\alpha_1 \Rightarrow d_1$ | $\alpha_2 \Rightarrow d_1$ | $\alpha_3 \Rightarrow d_1$ |
|-------------------|----------------------------|----------------------------|----------------------------|
| Częstość względna | 99.9% | 100% | 100% |
| Negacja entropii | <0 | 0 | 0 |

Częstość względna i negacja entropii faworyzują reguły $\alpha_2 \Rightarrow d_1$ i $\alpha_3 \Rightarrow d_1$

Wartości estymaty Laplace'a ($D = 2$):

99.8% dla $\alpha_1 \Rightarrow d_1$

85.7% dla $\alpha_2 \Rightarrow d_1$

66.6% dla $\alpha_3 \Rightarrow d_1$



CN2: klasyfikacja pierwszy wygrywa

Lista decyzyjna to lista reguł utworzona przez algorytm sekwencyjnego pokrywania uporządkowana w kolejności takiej, w jakiej reguły były generowane, z dodatkową regułą domyślną na końcu

$$\rightarrow R_0 \rightarrow R_1 \rightarrow R_2 \rightarrow \dots \rightarrow R_m \rightarrow \text{Default}$$

W CN2:

R_0 – reguła wygenerowana ze wszystkich przykładów

R_1 – reguła wygenerowana z przykładów niepokrywanych przez R_0

R_2 – reguła wygenerowana z przykładów niepokrywanych przez R_0, R_1 , itd.

Default – reguła bezwarunkowa zwracająca najczęstszą decyzję w zbiorze treningowym

Obiektowi przypisywana jest decyzja d z pierwszej reguły $\alpha \Rightarrow d$ na liście decyzyjnej, której warunek α pasuje do obiektu.



CN2: klasyfikacja przez głosowanie reguł

rules — zbiór warunków wygenerowany przez algorytm sekwencyjnego pokrywania (bez ustalonych decyzji)

Rozkład decyzyjny warunku $\alpha \in rules$:

$$\langle n_1(\alpha), \dots, n_{|D|}(\alpha) \rangle$$

n_i — liczba przykładów z decyzją d_i spełniających α w zbiorze *uncov*, tzn. tylko tych przykładów, które nie spełniają żadnego z wcześniej wygenerowanych warunków

Wybór decyzji dla obiektu x przez sumowanie rozkładów:

$$\max \arg_{d_i} \sum_{\alpha \in rules: x \text{ spełnia } \alpha} n_i(\alpha)$$



AQ15, Michalski, 1986

◇ Używa atrybutów symbolicznych i numerycznych

Do atrybutów symbolicznych stosuje selektory równościowe i wykluczające, do atrybutów numerycznych stosuje selektory ograniczające ($<$, \leq , $>$, \geq).

◇ Używa metody sekwencyjnego pokrywania, ale oddzielnie dla każdej decyzji

Szukanie kolejnej reguły (procedura LEARN-ONE-RULE) podobnie jak w CN2 przebiega od najbardziej ogólnych do bardziej specyficznych reguł, ale:

- przeszukiwanie sterowane jest wybranym przykładem
- reguły-kandydatki poprawiane są tak długo, dopóki nie osiągną warunku spójności ze zbiorem treningowym, reguła najlepsza wybierana jest spośród końcowych reguł spójnych



AQ: szukanie najlepszej reguły

function LEARN-ONE-RULE-AQ(*examples*, *uncov*, *d*, *k*) **returns** a rule

inputs: *examples* - all training examples

uncov - the examples not covered by the previous rules

d - decision of a return rule

k - the width of searching

$e_{pos} \leftarrow$ a random examples from *uncov* with decision *d*

candidates \leftarrow { the most general empty condition }

repeat

$e_{neg} \leftarrow$ example with decision $\neq d$ covered by one or more conditions in *candidates*
with the maximum number of values = the corresponding values of e_{pos}

selectors \leftarrow all selectors consistent with e_{pos} excluding e_{neg}

candidates \leftarrow { $x \wedge s : x \in \text{candidates}, s \in \text{selectors}$ }

candidates \leftarrow { $x \in \text{candidates} : \neg \exists y \in \text{candidates}$ more general than x }

retain only *k* best candidates in *candidates* according to PERFORMANCE

until *candidates* cover no examples with decision $\neq d$

best \leftarrow the best condition in *candidates* according to PERFORMANCE

return *best* $\Rightarrow d$

end function



AQ: miara jakości reguły

$$\text{PERFORMANCE}(\alpha \Rightarrow d, \text{examples}) = \text{pos}_{\text{included}} + \text{neg}_{\text{excluded}}$$

$\text{pos}_{\text{included}}$ – liczba przykładów w *examples* z decyzją d pasujących do warunku α
tzw. wsparcie reguły

$\text{neg}_{\text{excluded}}$ – liczba przykładów w *examples* z decyzją $\neq d$ wykluczanych przez warunek α

Uwaga

Jeśli reguła jest spójna ze zbiorem treningowym, tzn. warunek reguły wyklucza wszystkie przykłady z decyzją $\neq d$, to miary jakości reguły jest równa wsparciu reguły

$$\text{PERFORMANCE}(\alpha \Rightarrow d, \text{examples}) = \text{pos}_{\text{included}}$$



AQ: klasyfikacja

Klasyfikacja przez głosowanie reguł

Waga pojedynczej reguły:

$$\text{Weight}(\alpha \Rightarrow d) = \frac{|\text{pos}_{included}(\alpha \Rightarrow d)|}{|\text{examples}|}$$

Wybór decyzji dla obiektu x :

$$\max \arg_d \sum_{\alpha \Rightarrow d: x \text{ spelnia } \alpha} \frac{|\text{pos}_{included}(\alpha \Rightarrow d)|}{|\text{examples}|}$$



Generowanie reguł: CN2 vs AQ

Cechy wspólne

- metoda sekwencyjnego pokrywania
- szukanie pojedynczej reguły:
 - metoda pierwszy najlepszy ustalonej szerokości
 - od najbardziej ogólnych w kierunku bardziej specyficznych

Różnice

| | Przeszukiwanie sterowane | Wymaganie spójności | Miara jakości reguł |
|-----|--------------------------|---------------------|---------------------|
| CN2 | całym zbiorem | NIE | estymata Laplace'a |
| AQ | pojedynczym przykładem | TAK | wsparcie |



Dziękuję za uwagę!

