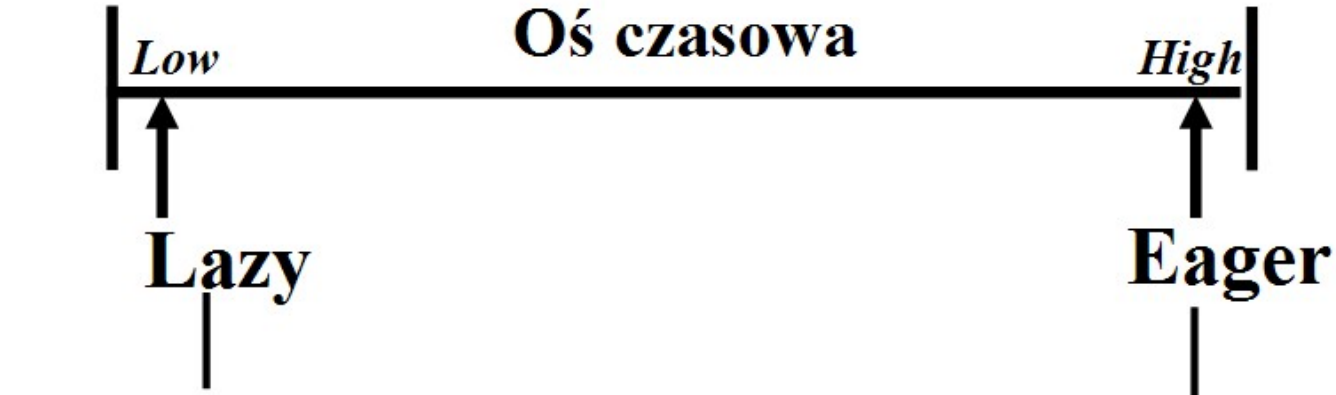
Wykład 8: Leniwe metody klasyfikacji

Nguyen Hung Son

Wydział MIM, Uniwersytet Warszawski

Outline

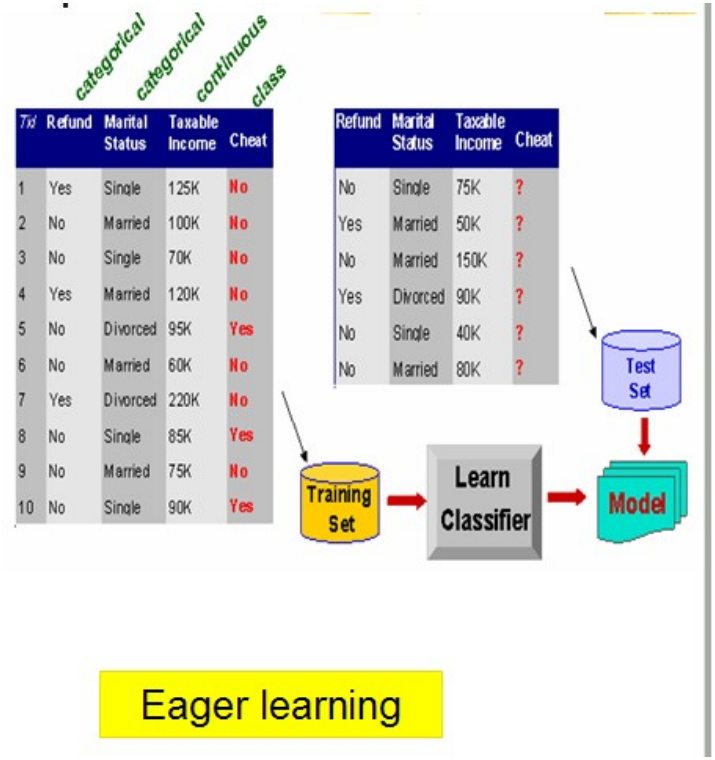
LAZY VS. EAGER LEARNING



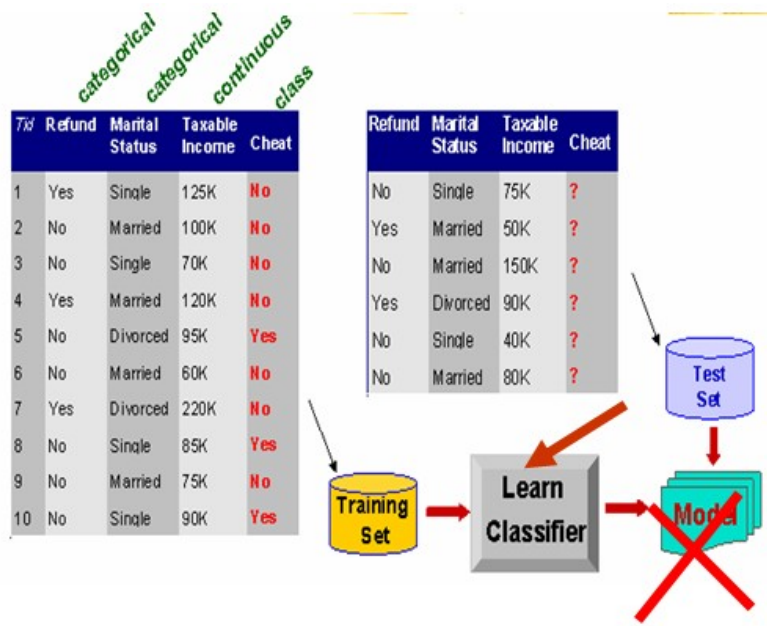
1. **Uczenia się**: Brak. Nie generuje się modelu (hipotezy) dla funkcji docelowej
2. **Klasyfikacja**: Odpowiedź jest kombinowana bezpośrednio ze danych treningowych

1. **Uczenie się**: Buduje się pełny model opisujący funkcję docelującą
2. **Klasyfikacja**: Odpowiedź jest oparte na generowanym modelu

LAZY VS. EAGER LEARNING



Eager learning



Lazy learning

Kiedy stosować leniwe techniki?

- Eager learning: Buduje globalną hipotezę
  - Zaleta:
    Prosty opis zbiór danych
    Szybki czas klasyfikacji
  - Wada:
    Czas uczenia się jest obciążliwy
    Jakość klasyfikacji nie wysoka
- Lazy learning: Buduje lokalną hipotezę
  - Zaleta:
    Szybki czas uczenia się
    Można projektować algorytmy klasyfikacji on-line
    Wysoka dokładność klasyfikacji
  - Wada:
    Czas klasyfikacji długi
    Wymagania: funkcja odległości, strategia głosowania

Metody leniwego uczenia się

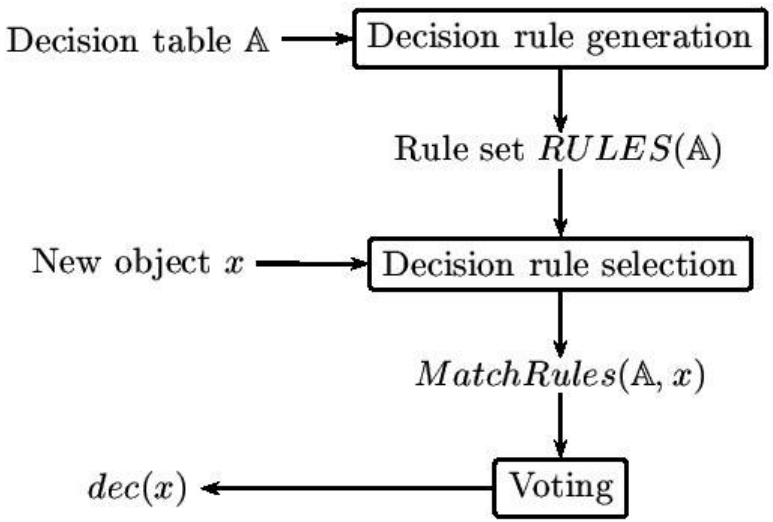
- k-NN
- Połączenie k-NN z drzewem decyzyjnym
- Generowanie podzbioru reguł, które mogą klasyfikować obiekt
- Generowanie poddrzewa decyzyjnego, które może klasyfikować obiekt

Outline

Regułowy klasyfikator - eager learning

Ogólny schemat:

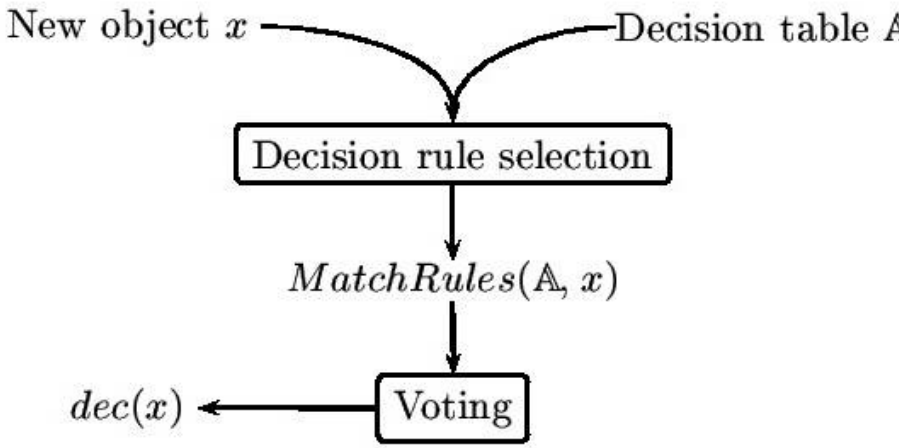
- **Uczenie się:** Generuj zbiór reguł $Rules(T)$ dla tablicy danych T
- **Selekcja reguł:** Wybierz zbiór $MatchRules(T, x)$ zawierający reguły, które pasują do nowego obiektu x.
- **klasyfikacja:** Wyznacz klasę decyzyjną dla x za pomocą głosowania na zbiorze $MatchRules(D, x)$



Regułowy klasyfikator - lazy learning

Ogólny schemat:

- **Generowanie reguł:** Generuj zbiór $MatchRules(T, x)$ zawierający reguły, które pasują do nowego obiektu x.
- **klasyfikacja:** Wyznacz klasę decyzyjną dla x za pomocą głosowania na zbiorze $MatchRules(T, x)$



DESCRIPTION LANGUAGE (JĘZYK DESKRYPTORÓW)

Let A be a set of attributes. The description language for A is a triple

$$\mathcal{L}(A) = (\mathbf{D}, \{\vee, \wedge, \neg\}, \mathbf{F})$$

where

- $\mathbf{D}$ is the set of *descriptors*

$$\mathbf{D} = \{(a = v) : a \in A \text{ and } v \in Val_a\}$$

- $\{\vee, \wedge, \neg\}$ is a set of standard Boolean operators
- $\mathbf{F}$ is a set of boolean expressions defined on $\mathbf{D}$ called *formulas*.
- For any $B \subseteq A$ we denote by $\mathbf{D}|_B$ the set of descriptors restricted to B where $\mathbf{D}|_B = \{(a = v) : a \in B \text{ and } v \in Val_a\}$ We also denote by $\mathbf{F}|_B$ the set of formulas build from $\mathbf{D}|_B$.

The semantics

Let $\mathbb{S} = (U, A)$ be an information table describing a sample $U \subset \mathbb{X}$. The semantics of any formula $\phi \in \mathbf{F}$, denoted by $[[\phi]]_{\mathbb{S}}$, is defined by induction as follows:

$$[[(a = v)]]_{\mathbb{S}} = \{x \in U : a(x) = v\} \quad (1)$$

$$[[\phi_1 \vee \phi_2]]_{\mathbb{S}} = [[\phi_1]]_{\mathbb{S}} \cup [[\phi_2]]_{\mathbb{S}} \quad (2)$$

$$[[\phi_1 \wedge \phi_2]]_{\mathbb{S}} = [[\phi_1]]_{\mathbb{S}} \cap [[\phi_2]]_{\mathbb{S}} \quad (3)$$

$$[[\neg\phi]]_{\mathbb{S}} = U \setminus [[\phi]]_{\mathbb{S}} \quad (4)$$

We associate with every formula ϕ the following numeric features:

- $length(\phi)$ = the number of descriptors that occur in ϕ;
- $support(\phi) = |[[\phi]]_{\mathbb{S}}|$ = the number of objects that match the formula;

Reguły decyzyjne

Definicja reguł decyzyjnych

Let $\mathbb{S} = \{U, A \cup \{dec\}\}$ be a decision table. Any implication of a form

$$\phi \Rightarrow \delta$$

where $\phi \in \mathbf{F}_A$ and $\delta \in \mathbf{F}_{dec}$, is called *the decision rule* in $\mathbb{S}$. The formula ϕ is called *the premise* of the decision rule $\mathbf{r}$ and δ is called *the consequence* of $\mathbf{r}$. We denote the premise and the consequence of the decision rule $\mathbf{r}$ by $prev(\mathbf{r})$ and $cons(\mathbf{r})$, respectively.

Generic decision rule

The decision rule $\mathbf{r}$ whose the premise is a boolean monomial of descriptors, i.e.,

$$\mathbf{r} \equiv (a_{i_1} = v_1) \wedge ... \wedge (a_{i_m} = v_m) \Rightarrow (dec = k) \qquad (5)$$

is called *the generic decision rule*.

We will consider generic decision rules only. For a simplification, we will talk about decision rules keeping in mind the generic ones.

Reguły ...

Every decision rule $\mathbf{r}$ of the form (5) can be characterized by the following featured:

$$
\begin{aligned}
length(\mathbf{r}) &= \text{the number of descriptor on the assumption of } \mathbf{r} \text{ (i.e. the left hand side of implication)} \\
[\mathbf{r}] &= \text{the carrier of } \mathbf{r}\text{, i.e. the set of objects from } U \text{ satisfying the assumption of } \mathbf{r} \\
support(\mathbf{r}) &= \text{the number of objects satisfying the assumption of } \mathbf{r}\text{: } support(\mathbf{r}) = card([\mathbf{r}]) \\
confidence(\mathbf{r}) &= \frac{|[\mathbf{r}] \cap DEC_k|}{|[\mathbf{r}]|}
\end{aligned}
$$

The decision rule $\mathbf{r}$ is called *consistent* with $\mathbb{A}$ if

$$confidence(\mathbf{r}) = 1$$

Minimal consistent rules

For a given decision table $\mathbb{S} = (U, A \cup \{dec\})$, the consistent rule:

$$\mathbf{r} = \phi \Rightarrow (dec = k)$$

is called the *minimal consistent decision rule* if any decision rule $\phi' \Rightarrow (dec = k)$ where ϕ' is a shortening of ϕ is not consistent with $\mathbb{S}$.

Metoda Oparta o Wnioskowanie Boolowskie

- Każda reguła powstaje poprzez skracanie opisu jakiegoś obiektu.
- Redukty lokalne
- Te same heurystyki dla reduktów decyzyjnych.

Przykład tablicy decyzyjnej

Hurt.	Jakość obsługi	Jakość towaru	Obok autostrady?	Centrum?	de
ID	a_1	a_2	a_3	a_4	
1	dobra	dobra	nie	nie	s
2	dobra	dobra	nie	tak	s
3	bdb	dobra	nie	nie	
4	slaba	super	nie	nie	
5	slaba	niska	tak	nie	
6	slaba	niska	tak	tak	s
7	bdb	niska	tak	tak	
8	dobra	super	nie	nie	s
9	dobra	niska	tak	nie	
10	slaba	super	tak	nie	
11	dobra	super	tak	tak	
12	bdb	super	nie	tak	
13	bdb	dobra	tak	nie	
14	slaba	super	nie	tak	

Macierz i funkcja lokalnych odróżnialności

$\mathbb{M}$	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

$$f_{u_3} = (\alpha_1)(\alpha_1 \vee \alpha_4)(\alpha_1 \vee \alpha_2 \vee \alpha_3 \vee \alpha_4)(\alpha_1 \vee \alpha_2) = \alpha_1$$

Reguły:

$$(a_1 = \text{bdb}) \implies dec = \text{zysk}$$

Macierz i funkcja lokalnych odróżnialności

$\mathbb{M}$	1	2	6	8
3	a_1	a_1, a_4	a_1, a_2, a_3, a_4	a_1, a_2
4	a_1, a_2	a_1, a_2, a_4	a_2, a_3, a_4	a_1
5	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_4	a_1, a_2, a_3
7	a_1, a_2, a_3, a_4	a_1, a_2, a_3	a_1	a_1, a_2, a_3, a_4
9	a_2, a_3	a_2, a_3, a_4	a_1, a_4	a_2, a_3
10	a_1, a_2, a_3	a_1, a_2, a_3, a_4	a_2, a_4	a_1, a_3
11	a_2, a_3, a_4	a_2, a_3	a_1, a_2	a_3, a_4
12	a_1, a_2, a_4	a_1, a_2	a_1, a_2, a_3	a_1, a_4

$$
\begin{aligned}
f_{u_8} &= (\alpha_1 \vee \alpha_2)(\alpha_1)(\alpha_1 \vee \alpha_2 \vee \alpha_3)(\alpha_1 \vee \alpha_2 \vee \alpha_3 \vee \alpha_4)(\alpha_2 \vee \\
&\quad (\alpha_1 \vee \alpha_3)(\alpha_3 \vee \alpha_4)(\alpha_1 \vee \alpha_4) \\
&= \alpha_1(\alpha_2 \vee \alpha_3)(\alpha_3 \vee \alpha_4) = \alpha_1\alpha_3 \vee \alpha_1\alpha_2\alpha_4
\end{aligned}
$$

Reguły:

- $(a_1 = \text{dobra}) \wedge (a_3 = \text{nie}) \implies dec = \text{strata}$
- $(a_1 = \text{dobra}) \wedge (a_2 = \text{super}) \wedge (a_4 = nie) \implies dec = \text{strata}$

Algorytm: rule selection

ALGORITHM: Rule selection

Input: The object x, the maximal length $\lambda_{\max}$, the minimal support $\sigma_{\min}$, and the minimal confidence $\alpha_{\min}$.

Output: The set $MatchRules(\mathbb{A}, x)$ of decision rules from $MinRules(\mathbb{A}, \lambda_{\max}, \sigma_{\min}, \alpha_{\min})$ matching x.

BEGIN
  $\mathbf{C}_1 := \mathbf{P}_1$; $i := 1$;
  WHILE $(i \leq \lambda_{\max})$ AND $(\mathbf{C}_i$ IS NOT EMPTY$)$ DO
    $\mathbf{F}_i := \emptyset$; $\mathbf{R}_i := \emptyset$;
    FOR $C \in \mathbf{C}_i$ DO
      $(s_1, \ldots, s_d) := GetClassDistribution(C)$;
      $support = s_1 + \ldots + s_d$;
      IF $support \geq \sigma_{\min}$ THEN
        IF $(\max\{s_1, \ldots, s_d\} \geq \alpha_{\min} * support)$ THEN
          $\mathbf{R}_i := \mathbf{R}_i \cup \{C\}$;
        ELSE
          $\mathbf{F}_i := \mathbf{F}_i \cup \{C\}$;
    ENDFOR
    $\mathbf{C}_{i+1} := AprGen(\mathbf{F}_i)$; $i := i + 1$;
  ENDWHILE
  RETURN $\bigcup_i \mathbf{R}_i$
END

Przykład

$\mathbb{A}$	a_1	a_2	a_3	a_4	dec
ID	outlook	temp.	hum.	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no
x	sunny	mild	high	TRUE	?

$\Rightarrow$

$\mathbb{A}\|_x$	d_1		d_2		d_3		d_4		dec
ID	a_1	x	a_2	x	a_3	x	a_4	x	dec
1	1		0		1		0		no
2	1		0		1		1		no
3	0		0		1		0		yes
4	0		1		1		0		yes
5	0		0		0		0		yes
6	0		0		0		1		no
7	0		0		0		1		yes
8	1		1		1		0		no
9	1		0		0		0		yes
10	0		1		0		0		yes
11	1		1		0		1		yes
12	0		1		1		1		yes
13	0		0		0		0		yes
14	0		1		1		1		no

Zbiór wsztstkich reguł

rules	supp
outlook(overcast)⇒play(yes)	4
humidity(normal) AND windy(FALSE)⇒play(yes)	4
outlook(sunny) AND humidity(high)⇒play(no)	3
outlook(rainy) AND windy(FALSE)⇒play(yes)	3
outlook(sunny) AND temperature(hot)⇒play(no)	2
outlook(rainy) AND windy(TRUE)⇒play(no)	2
outlook(sunny) AND humidity(normal)⇒play(yes)	2
temperature(cool) AND windy(FALSE)⇒play(yes)	2
temperature(mild) AND humidity(normal)⇒play(yes)	2
temperature(hot) AND windy(TRUE)⇒play(no)	1
outlook(sunny) AND temperature(mild) AND windy(FALSE)⇒play(no)	1
outlook(sunny) AND temperature(cool)⇒play(yes)	1
outlook(sunny) AND temperature(mild) AND windy(TRUE)⇒play(yes)	1
temperature(hot) AND humidity(normal)⇒play(yes)	1

$MatchRules(\mathbb{A}, x)$ zawiera 2 reguły:

- (outlook $=$ sunny) AND (humidity $=$ high) $\Rightarrow play = no$ (rule nr 3)
- (outlook $=$ sunny) AND (temperature $=$ mild) AND (windy $=$ TRUE) $\Rightarrow play = yes$ (rule nr 13)

Przykład algorytmu leniwego

$\lambda_{max} = 3; \sigma_{\min} = 1; \alpha_{\min} = 1$

$i = 1$				$i = 2$			
$\mathbf{C}_1$	check	$\mathbf{R}_1$	$\mathbf{F}_1$	$\mathbf{C}_2$	check	$\mathbf{R}_2$	$\mathbf{F}_2$
$\{d_1\}$	(3,2)		$\{d_1\}$	$\{d_1, d_2\}$	(1,1)		$\{d_1, d_2\}$
$\{d_2\}$	(4,2)		$\{d_2\}$	$\{d_1, d_3\}$	(3,0)	$\{d_1, d_3\}$	
$\{d_3\}$	(4,3)		$\{d_3\}$	$\{d_1, d_4\}$	(1,1)		$\{d_1, d_4\}$
$\{d_4\}$	(3,3)		$\{d_4\}$	$\{d_2, d_3\}$	(2,2)		$\{d_2, d_3\}$
				$\{d_2, d_4\}$	(1,1)		$\{d_2, d_4\}$
				$\{d_3, d_4\}$	(2,1)		$\{d_3, d_4\}$

$i = 3$			
$\mathbf{C}_3$	check	$\mathbf{R}_3$	$\mathbf{F}_3$
$\{d_1, d_2, d_4\}$	(0,1)	$\{d_1, d_2, d_4\}$	
$\{d_2, d_3, d_4\}$	(1,1)		$\{d_2, d_3, d_4\}$

$MatchRules(\mathbb{A}, x) = \mathbf{R}_2 \cup \mathbf{R}_3$:

(outlook = sunny) AND (humidity = high) $\Rightarrow play = no$

(outlook = sunny) AND (temperature = mild) AND (windy = TRUE) $\Rightarrow play = yes$