

Ćwiczenia z algorytmów wyliczania reguł decyzyjnych

March 2, 2025

Ćwiczenie 1 (1pkt)

Algorytmy wyliczania reguł decyzyjnych

Zadanie do wykonania

1. Czytamy teorię, analizujemy przykłady, w razie problemu ze zrozumieniem wyliczamy reguły na kartce.
2. Wczytujemy plik SystemDecyzyjny.txt . Ostatni atrybut (kolumna) określa klasę próbki
3. Implementujemy algorytm w Python, wyliczając następujące reguły:
 - reguły pokrywające obiekty (covering) (1pkt),

Teoria do ćwiczeń z przykładami

Sposoby zapisu deskryptora:

$$(a = a(v))$$

$$(a = v)$$

$$(a, a(v))$$

$$(a, v)$$

Znaczenie: $(a = a(v))$, atrybut a ma wartość v

System Informacyjny: (U, A)

U - zbiór obiektów;

A - zbiór atrybutów warunkowych;

Przykład: $(U, A), U = ob1, ob2, ob3, A = a1, a2, a3$

	a_1	a_2	a_3
ob_1	1	2	3
ob_2	3	2	5
ob_3	10	2	17

System Decyzyjny: (U, A, d) U - zbiór obiektów;

A - zbiór atrybutów warunkowych;

d - atrybut decyzyjny

$d \notin A$

Przykład System decyzyjny zapisujemy jako (U, A, d) , przyjmijmy,

$U = ob1, ob2, ob3$

$A = a1, a2, a3$

$d \in D = 1, 2$

Przykładowy system decyzyjny zgodny z opisem powyżej, może wyglądać następująco,

	a_1	a_2	a_3	d
ob_1	7	2	3	1
ob_2	3	3	5	2
ob_3	10	45	4	1

Zdefiniujmy reguły decyzyjne wzajemnie niesprzeczne

$$(a_1 = 1) \implies (d = 1)$$

$$(a_1 = 2) \wedge (a_2 = 7) \implies (d = 1)$$

$$(\text{pogoda} = \text{słoneczna}) \wedge (\text{żona} = \text{w pracy}) \wedge (\text{czas} = \text{wolny}) \implies (\text{decyzja} = \text{park})$$

$$(\text{pogoda} = \text{słoneczna}) \wedge (\text{żona} = \text{w domu}) \wedge (\text{czas} = \text{wolny}) \implies (\text{decyzja} = \text{dom})$$

Reguły decyzyjne wzajemnie sprzeczne

$$(a_1 = 1) \implies (d = 1)$$

$$(a_1 = 1) \implies (d = 2)$$

$$(\text{pogoda} = \text{słoneczna}) \wedge (\text{żona} = \text{w pracy}) \wedge (\text{czas} = \text{wolny}) \implies (\text{decyzja} = \text{park})$$

$$(\text{pogoda} = \text{słoneczna}) \wedge (\text{żona} = \text{w pracy}) \wedge (\text{czas} = \text{wolny}) \implies (\text{decyzja} = \text{dom})$$

Reguła z alternatywną decyzją

$$(\text{pogoda} = \text{słoneczna}) \wedge (\text{żona} = \text{w pracy}) \wedge (\text{czas} = \text{wolny}) \implies (\text{decyzja} = \text{park}) \vee (\text{decyzja} = \text{dom})$$

1 Algorytm z rodziny sekwencyjnie pokrywających

(sequential covering) (1pkt)

Idea algorytmu pokrywającego obiekty

Szukamy w obiektach systemu decyzyjnego, poczynając od pierwszego, a skończywszy na ostatnim reguł długości jeden, które są niesprzeczne. Po znalezieniu reguły niesprzecznej, dany obiekt wyrzucamy z rozważań, pamiętając o tym, że dalej bierze udział w sprawdzaniu sprzeczności i może wspierać inne reguły. Jeżeli po przeszukaniu wszystkich obiektów, pozostają obiekty niewyrzucone z rozważań, szukamy w nich kombinacji niesprzecznej długości dwa i postępujemy analogicznie jak w przypadku reguł pierwszego rzędu. Wyszukiwanie reguł niesprzecznych jest kontynuowane do momentu wyeliminowania wszystkich obiektów niesprzecznych. Jeżeli w systemie pojawią się obiekty, które są sprzeczne na wszystkich deskryptorach, nie kreujemy z nich reguł.

Przykładowe wyliczanie reguł pokrywających obiekty:

Dany mamy system decyzyjny (U, A, d) , gdzie $U = o_1, o_2, \dots, o_7, o_8$, $A = a_1, a_2, \dots, a_6$

d – atrybut decyzyjny

	a_1	a_2	a_3	a_4	a_5	a_6	d
o_1	1	1	1	1	3	1	1
o_2	1	1	1	1	3	2	1
o_3	1	1	1	3	2	1	0
o_4	1	1	1	3	3	2	1
o_5	1	1	2	1	2	1	0
o_6	1	1	2	1	2	2	1
o_7	1	1	2	2	3	1	0
o_8	1	1	2	2	4	1	1

Rząd I:

z o_1 brak

z o_2 ($a_6 = 2$) $\implies (d = 1)[3]$, wyrzucamy z rozważań obiekty o_2, o_4, o_6 .

z o_3 brak

z o_5 brak

z o_7 brak

z o_8 ($a_5 = 4$) $\implies (d = 1)$, wyrzucamy z rozważań obiekt o_8 .

Rząd II:

z o_1 ($a_3 = 1$) \wedge ($a_4 = 1$) $\implies (d = 1)[2]$, wyrzucamy z rozważań obiekt o_1 .

z o_3 ($a_3 = 1$) \wedge ($a_5 = 2$) $\implies (d = 0)$, wyrzucamy z rozważań obiekt o_3 .

z o_5 ($a_5 = 2$) \wedge ($a_6 = 1$) $\implies (d = 0)[2]$, wyrzucamy z rozważań obiekt o_5 .

z o_7 ($a_3 = 2$) \wedge ($a_5 = 3$) $\implies (d = 0)$, wyrzucamy z rozważań obiekt o_7 .