Algorytm Fuzzification

Algorytm Fuzzification tworzy zbiory rozmyte na podstawie zbiorów wartości atrybutów warunkowych.

Algorithm 1: Fuzzification

Data: zbiór A wszystkich atrybutów warunkowych;
zbiór $k_A = \{k_a : a \in A\}$, którego każdy element określa liczbę zbiorów rozmytych do utworzenia dla danego atrybutu;
Result: zbiór rozmyty $\mathcal{A}_F$ atrybutów warunkowych;
begin
  $\mathcal{A}_F := \emptyset$;
  foreach $a \in A$ **do**
    $A_F := \emptyset$;
    Podzielić zbiór V_a na k_a podzbiorów $V_{a^1}, V_{a^2}, \ldots, V_{a^{k_a}}$;
    foreach $V_{a^i} (1 \leqslant i \leqslant k_a)$ **do**
      Określić funkcję przynależności μ_{A^i};
      $A^i := \{(v, \mu_{A^i}(v)) : v \in V_{a^i}\}$;
      $A_F := A_F \cup \{A^i\}$;
    $\mathcal{A}_F := \mathcal{A}_F \cup \{A_F\}$;
end

Przykład 1.

Dany są:
$A = \{a\}$ – zbiór atrybutów warunkowych, gdzie a =temperatura powietrza;
$k_a = 3$ – liczba zbiorów rozmytych do utworzenia na podstawie atrybutu a;
Niech $V_a = \{14, 16, 17, 18, 20, 21, 23, 25, 26, 27, 29, 30, 32, 33, 35\}$.
$A_F := \emptyset$;
Zbioru V_a może być podzielony na k_a następujących podzbiorów:
$V_{a^1} = \{14, 16, 17, 18, 20, 21\}, V_{a^2} = \{18, 20, 21, 23, 25, 26, 27, 29, 30\}, V_{a^3} = \{27, 29, 30, 32, 33, 35\}$.
Korzystając odpowiednio z funkcji przynależności klasy L, t, γ, których parametrami są minimalne, średnie i maksymalne wartości ze zbiorów V_{a^i} otrzymujemy:
$A^1 = \{(14, 1), (16, 1), (17, 1), (18, 0.86), (20, 0.29), (21, 0)\}$,
$A^2 = \{(18, 0), (20, 0.33), (21, 0.5), (23, 0.83), (25, 0.42), (26, 0.33), (27, 0.25), (29, 0.08), (30, 0)\}$,
$A^3 = \{(27, 0), (29, 0.5), (29, 0.75), (30, 1), (33, 1), (35, 1)\}$.
Atrybut rozmyty jest postaci $A_F = \{A^1, A^2, A^3\}$, a zbiór rozmyty atrybutów warunkowych $\mathcal{A}_F = \{A_F\}$.
Zbiory rozmyte A^1, A^2, A^3 można określić odpowiednio jako "temperatura niska", "temperatura normalna", "temperatura wysoka".

Niech $A \in A_F$. Wyrażenie (x jest A) jest prawdziwe wtedy i tylko wtedy, gdy $\mu_A(x) = max\{\mu_{A'}(x) : A' \in A_F\}$.

Przykład 2.
Dany jest atrybut rozmyty A_F otrzymany w Przykładzie 1. Wyrażenie (20 jest "temperatura normalna") jest prawdziwe, ponieważ $\mu_{A^2}(20) = max\{\mu_{A^1}(20), \mu_{A^2}(20), \mu_{A^3}(20)\} = max\{0.29, 0.33, 0\}$.

Algorytm LEM1

Oznaczenia i definicje:

- U - uniwersum, tj. zbiór obiektów;
- A - zbiór atrybutów warunkowych;
- d - atrybut decyzyjny;
- $IND(B) = \{(x, y) \in U \times U : \forall_{a \in B} a(x) = a(y)\}$ - relacja nierozróżnialności, tj. zbiór wszystkich par obiektów, które mają takie same wartości na wszystkich atrybutach ze zbioru $B \subseteq A$;
- B^* - rodzina wszystkich klas nierozróżnialności wyznaczonych przez relację $IND(B)$, tzn. $B^* = U/IND(B)$;
- $\{d\}^*$ - rodzina wszystkich klas nierozróżnialności wyznaczonych przez relację $IND(\{d\})$, tzn. $\{d\}^* = U/IND(\{d\})$. Inaczej, rodzina wszystkich klas decyzyjnych;
- Zbiór $\{d\}$ *zależy* od zbioru B wtedy i tylko wtedy, gdy $B^* \leqslant \{d\}^*$ (tutaj " $\leqslant$ " oznacza, że każdy zbiór pierwszej rodziny jest podzbiorem pewnego zbioru drugiej rodziny, czyli $\forall_{X \in B^*} \exists_{Y \in \{d\}^*} X \subseteq Y$);
- B jest *globalnym pokryciem* zbioru $\{d\}$ wtedy i tylko wtedy, gdy $\{d\}$ zależy od B i nie istnieje $B' \subset T$ takie, że $\{d\}$ zależy od B';

Algorithm 2: LEM1

Data: zbiór A wszystkich atrybutów warunkowych, rodzina $\{d\}^*$;
Result: pojedyncze globalne pokrycie B;
begin
    $B := \emptyset$; $C := A$;
    if $A^* \leqslant \{d\}^*$ **then**
        foreach $a \in A$ **do**
            $D := C \setminus \{a\}$;
            if $D^* \leqslant \{d\}^*$ **then** $C := D$;
        $B := C$;
end

Opis algorytmu.

Algorytm LEM1 działa jedynie dla danych niesprzecznych, tzn. każde dwa obiekty z U, należące do różnych klas muszą być rozróżnialne na atrybutach z A. Wtedy spełniony jest warunek $A^* \leqslant \{d\}^*$. Algorytm w każdym kroku sprawdza, czy możliwe jest usunięcie jednego atrybutu z rozpatrywanego zbioru atrybutów (tj. $D := C \setminus \{a\}$). Jeżeli tak (tj. $D^* \leqslant \{d\}^*$), to atrybut jest usuwany ze zbioru atrybutów. W efekcie otrzymywany jest najmniejszy podzbiór atrybutów warunkowych zachowujący rozróżnialność obiektów z różnych klas.

Algorytm F-LEM1

Algorytm F-LEM1 jest modyfikacją algorytmu LEM1 polegającą na przystosowaniu go do działania na zbiorach rozmytych.

Algorithm 3: F-LEM1

Data: zbiór A wszystkich atrybutów warunkowych;
zbiór $k_A = \{k_a : a \in A\}$, którego każdy element określa liczbę zbiorów rozmytych do utworzenia dla danego atrybutu;
Result: pojedyncze globalne pokrycie B;
begin
  $\mathcal{A}_F := Fuzzification(A, k_A)$;
  foreach $o \in U$ **do**
    foreach $A_F \in \mathcal{A}_F$ **do**
      Określić zbiór rozmyty $A \in A_F$, do którego obiekt o należy;
  $B := LEM1(A_F)$
end

Zadanie.

Dana jest rozmyta tablica decyzyjna $(U, \mathcal{A}_F \cup \{d\})$,

obiekt	*aura*			*temperatura*			*wilgotnosc*		*wiatr*		*pogoda*
	s	*p*	*d*	*c*	*u*	*z*	*w*	*n*	*sl*	*m*	
1	0.8	0.2	0.1	0.9	0.1	0	0.8	0.2	0.7	0.4	0
2	0.9	0.1	0.1	0.8	0.2	0.1	0.9	0.2	0.1	0.8	0
3	0.3	0.7	0.2	0.9	0.1	0.1	0.9	0.1	0.9	0.1	1
4	0	0.1	0.9	0.1	0.9	0	0.6	0.5	0.8	0.3	1
5	0	0.2	0.9	0	0.1	0.9	0	0.1	0.8	0.2	1
6	0.4	0.3	0.6	0	0.2	0.9	0.1	0.9	0.1	0.9	0

gdzie $\mathcal{A}_F = \{aura, temperatura, wilgotnosc, wiatr\}, d = pogoda$
oraz s=sloneczna, p=pochmurna, d=deszczowa, c=ciepla, u=umiarkowana, z=zimna, w=wysoka, n=normalna, sl=slaby, m=mocny.
Wygenerować reguły za pomocą algorytmu F-LEM1.

Należenie obiektów do poszczególnych zbiorów rozmytych przedstawione jest w poniższej tablicy.

obiekt	*aura*	*temperatura*	*wilgotnosc*	*wiatr*	*pogoda*
1	sloneczna	ciepla	wysoka	slaby	0
2	sloneczna	ciepla	wysoka	mocny	0
3	pochmurna	ciepla	wysoka	slaby	1
4	deszczowa	umiarkowana	wysoka	slaby	1
5	deszczowa	zimna	normalna	slaby	1
6	deszczowa	zimna	normalna	mocny	0

$B := \emptyset$; $C := \mathcal{A}_F$;
$\mathcal{A}_F^* = \{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\}\}, \{d\}^* = \{\{1,2,6\},\{3,4,5\}\}$, zatem $\mathcal{A}_F^* \leqslant \{d\}^*$;
$D := C \setminus \{aura\} = \{temperatura, wilgotnosc, wiatr\}$,
stąd $D^* = \{\{1,3\},\{2\},\{4\},\{5\},\{6\}\}$, więc $D^* \nleqslant \{d\}^*$;
$D := C \setminus \{temperatura\} = \{aura, wilgotnosc, wiatr\}$,
stąd $D^* = \{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\}\}$, więc $D^* \leqslant \{d\}^*$, zatem $C := D$;
$D := C \setminus \{wilgotnosc\} = \{aura, wiatr\}$,
stąd $D^* = \{\{1\},\{2\},\{3\},\{4,5\},\{6\}\}$, więc $D^* \leqslant \{d\}^*$, zatem $C := D$;
$D := C \setminus \{wiatr\} = \{aura\}$,
stąd $D^* = \{\{1,2\},\{3\},\{4,5,6\}\}$, więc $D^* \nleqslant \{d\}^*$, zatem globalne pokrycie wynosi $B := \{aura, wiatr\}$;

Na podstawie globalnego pokrycia konstruowane są reguły przy zastosowaniu techniki „dropping conditions". Na początku konstruowana jest reguła w oparciu o obiekt 1, tj. $(aura, sloneczna) \wedge (wiatr, slaby) \to (pogoda, 0)$. Reguła pokrywa tylko pierwszy obiekt (tj. 1 jako jedyny spełnia część warunkową reguły). Aby zwiększyć pokrycie reguły o kolejne obiekty z tej samej klasy decyzyjnej, rozpatrywane jest opuszczenie kolejnych warunków reguły. Warunek $(aura, sloneczna)$ nie może być opuszczony, gdyż otrzymana w ten sposób reguła $(wiatr, slaby) \to (pogoda, 0)$, oprócz obiektu 1, pokrywa obiekty z innej klasy decyzyjnej, tj. obiekty $3, 4, 5$. Usunięcie warunku $(wiatr, slaby)$ powoduje, że reguła $(aura, sloneczna) \to (pogoda, 0)$ pokrywa obiekty z jednej klasy, tj. 1 i 2.
Dla pozostałych obiektów reguły generowane są w analogiczny sposób. Zbiór reguł wygenerowanych przez algorytm F-LEM1 składa się z następujących reguł (po dwukropku podane pokrycie reguły):
$(aura, sloneczna) \to (pogoda, 0) : 1, 2$,
$(wiatr, silny) \to (pogoda, 0) : 2, 6$,
$(aura, deszczowa) \wedge (wiatr, slaby) \to (pogoda, 1) : 4, 5$,
$(aura, pochmurna) \to (pogoda, 1) : 3$.