

## Sztuczna Inteligencja – Projekt

**Temat: Algorytm SQ**

**Liczba osób realizujących projekt: 2**

1. Zaimplementować algorytm *SQ*.
2. Zaimplementować klasyfikator *Classifier*.
3. Za pomocą algorytmu *SQ* wygenerować dla każdej klasy decyzyjnej reguły decyzyjne z 2/3 losowo wybranych obiektów zbioru *iris.tab*.
4. Za pomocą klasyfikatora *Classifier* dokonać klasyfikacji pozostałych obiektów zbioru *iris.tab*.

Ad 1.

1. Zaimplementować funkcję *Satisfied*, która sprawdzi dla dowolnego obiektu czy spełnia dowolną formułę postaci  $(a_1, v_1) \wedge (a_2, v_2) \wedge \dots \wedge (a_k, v_k)$ .
2. Zaimplementować funkcję *SatisfiedSet*, która wyznaczy zbiór  $\|\alpha\|_S$  obiektów należących do *S* i spełniających formułę  $\alpha$ , określoną jak wyżej.
3. Zaimplementować funkcję *CreatePremiseSet* (patrz opis algorytmu *SQ*).
4. Zaimplementować funkcję *BestPremise* (patrz opis algorytmu *SQ*).

Ad 2.

Klasyfikator *Classifier* ma przypisać każdy obiekt do klasy tej reguły, którą obiekt spełnia. Jeżeli występuje konflikt klas, tzn. obiekt spełnia reguły dotyczące różnych klas, to jest on przypisywany do klasy tych reguł, których spełnia więcej. W przypadku, gdy obiekt spełnia tyle samo reguł każdej ze spornych klas, to nie jest on przypisywany do żadnej klasy.

## Charakterystyka algorytmu SQ

Oznaczenia i definicje:

- $U$  – uniwersum, tj. zbiór obiektów;
- $A$  – zbiór atrybutów warunkowych;
- $d$  – atrybut decyzyjny;
- $V_a$  – zbiór wartości atrybutu  $a$ ;
- $DT = (U, A \cup \{d\})$  – tablica decyzyjna;
- $\|\alpha\|_S$  – zbiór obiektów ze zbioru  $S$  spełniających formułę  $\alpha$ ;
- $r = \varphi \rightarrow \psi$  – reguła decyzyjna, gdzie  $\varphi = (a_1, v_1) \wedge (a_2, v_2) \wedge \dots \wedge (a_k, v_k)$  jest poprzednikiem reguły, a  $\psi = (d, v)$  jej następnikiem oraz  $a_i \in A, v_i \in V_{a_i}, v \in V_d$  ( $1 \leq i \leq k$ );
- $p = (a, v)$  – przesłanka reguły, tj. jeden ze składników poprzednika reguły.

---

### Algorytm 1: SQ

---

**Data:** tablica decyzyjna  $DT$ , wartość atrybutu decyzyjnego  $v_d$ , typ aproksymacji  $app$ ;

**Result:** zbiór reguł decyzyjnych  $RS$ ;

**begin**

$RS := \emptyset; \psi := (d, v_d); C := \|\psi\|_{DT}; NC := U \setminus C;$

**while**  $C \neq \emptyset$  **do**

$\varphi := NULL; b := true;$

**while**  $\|\varphi\|_{NC} \neq \emptyset$  **do**

$P := CreatePremiseSet(\|\varphi\|_C);$

$p := BestPremise(\varphi, P, C, NC); \varphi' := \varphi \wedge p;$

**if**  $\|\varphi'\|_{NC} = \|\varphi\|_{NC}$  **then**  $b := false;$  exit while;

$\varphi := \varphi';$

**if**  $\neg(b = false \wedge app = lower)$  **then**

$r := \varphi \rightarrow \psi; RS := RS \cup \{r\}; C := C \setminus \|\varphi\|_C;$

**end**

---

Funkcja  $CreatePremiseSet(\|\varphi\|_C)$ , na podstawie zbioru  $\|\varphi\|_C$  obiektów rozpatrywanej klasy spełniających formułę  $\varphi$ , tworzy zbiór składający się z przesłanek postaci  $(a, v)$ , gdzie  $a \in A, v \in V_a$ .

---

**Funkcja 2:** BestPremise

---

**Data:** poprzednik reguły  $\varphi$ , zbiór  $P$  możliwych przesłanek, zbiór  $C$  obiektów rozpatrywanej klasy, zbiór  $NC$  obiektów spoza rozpatrywanej klasy;

**Result:** najlepsza przesłanka  $p$ ;

**begin**

$val := 0$ ;

**foreach**  $p' \in P$  **do**

$\varphi' := \varphi \wedge p'$ ;

$val' := \frac{\|\varphi'\|_C}{\|\varphi'\|_{NC} + 1}$ ;

**if**  $val' > val$  **then**  $val := val'$ ;  $p := p'$ ;

**end**

---

### Opis działania algorytmu

Algorytm SQ generuje zbiór reguł decyzyjnych dotyczących danej klasy, wykorzystując metodę sekwencyjnego pokrywania. Wygenerowany zbiór reguł odpowiada aproksymacji rozpatrywanej klasy decyzyjnej. Reguły są generowane, dopóki istnieją obiekty danej klasy (tj. **while**  $C \neq \emptyset$ ), dla których nie zostały wygenerowane jeszcze reguły. Inaczej mówiąc, w zbiorze  $C$  pozostają te obiekty, które nie spełniają żadnej reguły ze zbioru  $RS$ . Pojedyncza reguła jest generowana, dopóki obiekty spoza rozpatrywanej klasy spełniają tę regułę (tj. **while**  $\|\varphi\|_{NC} \neq \emptyset$ ). W każdym kroku dodawana jest najlepsza przesłanka (tj.  $p := BestPremise(\varphi, P, C, NC)$ ), czyli taka, po dodaniu której regułę będzie spełniać jak największa liczba obiektów z rozpatrywanej klasy oraz jak najmniejsza liczba obiektów spoza rozpatrywanej klasy. Jeżeli dodanie kolejnej przesłanki nie zmniejszyłoby liczby obiektów spoza rozpatrywanej klasy spełniających regułę (tj. **if**  $\|\varphi'\|_{NC} = \|\varphi\|_{NC}$ ), to proces generowania reguły jest przerywany. Jeżeli rozpatrywana jest górna aproksymacja, to wszystkie reguły są dodawane do zbioru reguł, jeżeli dolna, to tylko te, które są spełniane tylko przez obiekty z rozpatrywanej klasy (tj. **if**  $\neg(b = false \wedge app = lower)$ ).

### Przykład

Dana jest tablica decyzyjna  $DT = (U, A \cup \{d\})$ :

obiekt	<i>temperatura</i>	<i>bol_glowy</i>	<i>oslabienie</i>	<i>nudnosci</i>	<i>grypa</i>
1	b._wysoka	tak	tak	nie	tak
2	wysoka	tak	nie	tak	tak
3	normalna	nie	nie	nie	nie
4	normalna	tak	tak	tak	tak
5	wysoka	nie	tak	nie	tak
6	wysoka	nie	nie	nie	nie
7	normalna	nie	tak	nie	nie

gdzie  $U = \{1, \dots, 7\}$ ,

$A = \{temperatura, bol\_glowy, oslabienie, nudnosci\}$ ,  $d = grypa$ .

Wyznaczyć reguły dotyczące klasy decyzyjnej  $grypa = tak$ .

Niech  $app = lower$ .

$RS := \emptyset$ ;  $\psi := (grypa, tak)$ ;  $C := \{1, 2, 4, 5\}$ ;  $NC := \{3, 6, 7\}$ ;

Pętla **while**  $C \neq \emptyset$  (krok 1):

$\varphi := NULL$ ;  $b := true$ ; (W tym przypadku  $\|\varphi\|_{NC} = NC$ .)

Pętla **while**  $\|\varphi\|_{NC} \neq \emptyset$  (krok 1):

$P := \{(temperatura, b\_wysoka), (temperatura, wysoka), (temperatura, normalna), (bol\_glowy, tak), (bol\_glowy, nie), (oslabienie, tak), (oslabienie, nie), (nudnosci, tak), (nudnosci, nie)\}$ ;

(Zbiór  $P$  składa się z tych przesłanek, które są spełnione przez te obiekty należące do  $C$ , które spełniają  $\varphi$ . Gdy  $\varphi = NULL$ , to  $\|\varphi\|_C = C$ .)

Jakość reguły w zależności od nowej przesłanki:

$(temperatura, b\_wysoka) : 1/1 = 1$ ,  $(temperatura, wysoka) : 2/2 = 1$ ,

$(temperatura, normalna) : 1/3 = 0.33$ ,  $(bol\_glowy, tak) : 3/1 = \mathbf{3}$ ,

$(bol\_glowy, nie) : 1/4 = 0.25$ ,  $(oslabienie, tak) : 3/2 = 1.5$ ,  $(oslabienie, nie) :$

$1/3 = 0.33$ ,  $(nudnosci, tak) : 2/1 = 2$ ,  $(nudnosci, nie) : 2/4 = 0.5$ ;

Zatem  $p := (bol\_glowy, tak)$  oraz  $\varphi := (bol\_glowy, tak)$ .

Otrzymujemy  $\|\varphi\|_{NC} = \emptyset$ , więc wychodzimy z pętli wewnętrznej.

$RS := \{(bol\_glowy, tak) \rightarrow (grypa, tak)\}$ ;

Otrzymujemy  $C = \{5\}$ .

Pętla **while**  $C \neq \emptyset$  (krok 2):

$\varphi := NULL$ ;  $b := true$ ;

Pętla **while**  $\|\varphi\|_{NC} \neq \emptyset$  (krok 1):

$P := \{(temperatura, wysoka), (bol\_glowy, nie), (oslabienie, tak), (nudnosci, nie)\}$ ;

Jakość reguły w zależności od nowej przesłanki:

$(temperatura, wysoka) : 1/2 = \mathbf{0.5}$ ,  $(bol\_glowy, nie) : 1/4 = 0.25$ ,

$(oslabienie, tak) : 1/2 = \mathbf{0.5}$ ,  $(nudnosci, nie) : 1/4 = 0.25$ ;

Wybieramy pierwszą przesłankę o najwyższej jakości,

zatem  $p := (temperatura, wysoka)$  oraz  $\varphi := (temperatura, wysoka)$ ;

Otrzymujemy  $\|\varphi\|_{NC} = \{6\}$ .

Pętla **while**  $\|\varphi\|_{NC} \neq \emptyset$  (krok 2):

$(bol\_glowy, nie) : 1/2 = 0.5$ ,  $(oslabienie, tak) : 1/1 = \mathbf{1}$ ,  $(nudnosci, nie) :$

$1/2 = 0.5$ ;

Zatem  $p := (oslabienie, tak)$  oraz  $\varphi := (temperatura, wysoka) \wedge (oslabienie, tak)$ ;

Otrzymujemy  $\|\varphi\|_{NC} = \emptyset$ , więc wychodzimy z pętli wewnętrznej.

$RS := \{(bol\_glowy, tak) \rightarrow (grypa, tak),$

$(temperatura, wysoka) \wedge (oslabienie, tak) \rightarrow (grypa, tak)\}$ ;

Otrzymujemy  $C = \emptyset$ , więc wychodzimy z pętli zewnętrznej.

Zbiór reguł wygenerowanych przez algorytm  $SQ$  składa się z następują-

cych reguł (po dwukropku podane pokrycie reguły):

$(bol\_glowy, tak) \rightarrow (grypa, tak) : 1, 2, 4,$

$(temperatura, wysoka) \wedge (oslabienie, tak) \rightarrow (grypa, tak) : 5,$

$(temperatura, normalna) \wedge (bol\_glowy, nie) \rightarrow (grypa, nie) : 3, 7,$

$(bol\_glowy, nie) \wedge (oslabienie, nie) \rightarrow (grypa, nie) : 3, 6.$