SKRYPT DO PROJEKTU Z PRZEDMIOTU

SZTUCZNA INTELIGENCJA W MEDYCYNIE

autorzy:

dr inż. Piotr Szczuko

prof. dr hab. inż. Bożena Kostek,

Gdańsk, 2015

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt *„Przygotowanie i realizacja kierunku inżynieria biomedyczna – studia międzywydziałowe”*
współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Spis treści

1 Wprowadzenie

Celem projektu jest wykonanie systemu wnioskowania, wykorzystującego teorię zbiorów przybliżonych, dedykowanego wybranemu zadaniu klasyfikacji. Dane treningowe i testowe pozyskać można z ogólnodostępnych baz danych przypadków medycznych lub wykorzystać pliki przykładowe, dostarczone razem z aplikacją RSES [5]. W ramach projektu student zapoznaje się z metodami wstępnego przygotowania danych, dyskretyzacji danych, generowania reduktów kilkoma metodami, tworzeniem reguł i filtracją zbioru reguł, klasyfikacją zbioru testowego. Uzyskana wiedza i praktyczne posługiwanie się zbiorami danych może być wykorzystywane w tworzeniu i obsłudze różnych systemów wnioskowania, nie tylko zbiorów przybliżonych.

Przebieg zajęć obejmuje:

- Przedstawienie wymagań dotyczących opracowania dokumentacji projektowej.
- Opracowanie teoretyczne dotyczące wybranego zagadnienia projektowego.
- Wybór bazy danych, przygotowanie danych do treningu oraz do testowania.
- Testowanie, walidacja i weryfikacja.
- Opracowanie i prezentacja wyników.

W ramach projektu Sztuczna inteligencja w medycynie przydatne mogą być bazy medyczne zawierające przykłady rekordów pacjentów, dotyczących danej jednostki chorobowej:

- http://archive.ics.uci.edu/ml/datasets/Echocardiogram
- http://archive.ics.uci.edu/ml/datasets/Heart+Disease
- http://archive.ics.uci.edu/ml/datasets/Hepatitis
- http://archive.ics.uci.edu/ml/datasets/Liver+Disorders
- http://archive.ics.uci.edu/ml/datasets/Lung+Cancer
- http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29
- http://archive.ics.uci.edu/ml/datasets/Thyroid+Disease

2 Teoria zbiorów przybliżonych

W kolejnych podrozdziałach przedstawiono najważniejsze informacje teoretyczne dotyczące zbiorów przybliżonych.

2.1 Historia zbiorów przybliżonych

Teoria zbiorów przybliżonych (ang. *Rough Set*) stworzona została przez Polaka, prof. Zdzisława Pawlaka. Pierwsza praca na ten temat [1] wprowadziła postawy teorii, która bardzo szybko znalazła praktyczne zastosowanie w systemach informatycznych [2] i uzyskała dojrzałość – dostrzeżono jej liczne zalety i związki z innymi metodami wnioskowania [3].

2.2 System informacyjny i decyzyjny

Teoria zbiorów przybliżonych zajmuje się **klasyfikacją danych** zorganizowanych w postaci tabel. Dane uzyskane mogą być z pomiarów, testów lub od ekspertów. Głównym celem analizy danych wejściowych jest wyznaczenie aproksymacji (przybliżenia) badanej idei (koncepcji, np. pacjenta zdrowego, chorego, itd.), tak aby dokonać dokładnej **analizy problemu**, związków i zależności między atrybutami i decyzjami oraz uzyskania narzędzia **klasyfikującego** nowe przypadki.

W przeciwieństwie do innych metod wnioskowania, w teorii zbiorów przybliżonych dopuszcza się: **nieprecyzyjne dane, sprzeczność danych, niekompletność danych.**

Wprowadza się następujący formalizm:

System informacyjny jest to zorganizowany zestaw danych, w postaci tabeli, w której wiersze reprezentują indywidualne **obiekty**, np. pomierzone w eksperymencie, obserwowane, historyczne, itd., a **atrybuty** obiektów zapisane są w osobnych kolumnach tej tabeli.

U – zbiór obiektów (od słowa „uniwersum”), np. $U=\{x_1, x_2, x_3, x_4\}$, gdzie x_i to obiekt i-ty,

A – zbiór odwzorowań, pomiarów, oznaczanych jako a, z obiektu na wartość jego cechy, tj. a: $U \rightarrow V_a$ dla każdego $a \in A$. Obiektowi x_n z U w wyniku pomiaru a_n, przypisywana jest wartość mierzonej cechy V_a, dla całej puli A metod pomiaru a_n.

Oznaczyć można system informacyjny jako (1):

$$SI = (U, A) \tag{1}$$

Przykład. Obiekty oraz metody ich pomiaru i wartości atrybutów, występujące w systemie informacyjnym różnić się będą w zależności od zastosowań. Przykładowo, czujnik światła w prostej lampie, która ma się sama włączyć, mierzy napięcie na fotodiodzie. Obiektem jest aktualna sytuacja w pobliżu lampy, atrybutem jest poziom oświetlenia zewnętrznego, metodą pomiaru jest pomiar

napięcia. Inteligentniejsza lampa, która włącza się przy słabym oświetleniu oraz po stwierdzeniu ruchu w jej pobliżu, musi mierzyć jeszcze jeden atrybut – obecność ruchu.

Medyczny system informacyjny, w którym obiektami są pacjenci z urazem kręgosłupa, może mieć następującą postać (patrz tab. 1).

Tab. 1. Przykład systemu informacyjnego

	Age	$LEMS$
x_1	16-30	50
x_2	16-30	0
x_3	31-45	1-25
x_4	31-45	1-25
x_5	46-60	26-49
x_6	16-30	26-49
x_7	46-60	26-49

Tab. 1 zawiera dane 7 pacjentów w różnym wieku (*Age*) z różnym wynikiem testu motoryki kończyn dolnych – *Lower-Extremity Motor Score* (LEMS).

Pytanie. W tabeli znajdują się dane pacjentów anonimowych, o których nie wiadomo nic, ponad te dwa atrybuty. Czy przy tym stanie wiedzy, są obiekty nierozróżnialne (tożsame ze sobą)?

Na temat obiektów systemu informacyjnego, w wyniku obserwacji, dedukcji, badań lub doświadczenia eksperta, można powiedzieć coś jeszcze – **określić wynikową decyzję**, dla każdego z obiektów. Powstaje w ten sposób **system decyzyjny**:

$$SD = (U, A, \{d\}) \tag{2}$$

gdzie *d* to zbiór możliwych decyzji, adekwatnych dla obiektów danego uniwersum *U*.

System decyzyjny dla powyższego przykładu medycznego przyjmuje postać:

Tab. 2. Przykład systemu decyzyjnego

	Age	$LEMS$	$Walk$
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

W ostatniej kolumnie tabeli 2 zapisano efekt terapii pacjenta – czy odzyskał on (*Yes*) czy nie (*No*) zdolność chodzenia (*Walk*).

Jak wskazano powyżej, występują obiekty tożsame, pod względem wartości atrybutów, tj. para x_3 i x_4 oraz para x_5 i x_7, jednakże dla tej pierwszej pary **decyzje nie są identyczne** – występuje **sprzeczność danych**.

Poniżej zostaną rozwinięte zagadnienia decyzji, tożsamości obiektów i modelowania zbiorów przybliżających określone koncepcje w oparciu o sprzeczne dane (w tym przypadku zbiór pacjentów chodzących i nie).

2.3 Reguły decyzyjne

Powyższy system decyzyjny odczytywać można jako zbiór siedmiu reguł logicznych, przykładowo dla pacjenta x_1:

JEŻELI Age=”16-30” **I** LEMS=”50” **TO** Walk=”Yes” (3)

Reguły takie wykorzystane mogą być **do klasyfikacji** przypadków nowych, dla których nie jest znana wartość decyzji i celem jest jej prognoza. Algorytm wnioskujący musi wówczas przeanalizować poprzedniki reguł (JEŻELI … I …) (3), znaleźć regułę o wartościach atrybutów odpowiadających nowemu przypadkowi i odczytać następnik reguły (TO …) i zwrócić go jako wynik. Ze względu na występujące sprzeczności w systemie decyzyjnym, nie zawsze możliwe jest podanie jednoznacznej odpowiedzi. Teoria zbiorów przybliżonych podaje rozwiązanie tego problemu niejednoznaczności i sprzeczności.

2.4 Tożsamość obiektów

Dla podanego przykładu pod względem wartości atrybutów pary x_3 i x_4 oraz x_5 i x_7 są nierozróżnialne. Należy podkreślić, że dodanie nowego atrybutu, np. płci, wagi lub historii choroby pacjenta, może spowodować, że obiekty te staną się rozróżnialne – tożsamość określa się względem wybranego podzbioru atrybutów $B \subseteq A$ (zawiera się, czyli może także być równy A – wszystkim atrybutom). Wybranie podzbioru B = {*Age*} sprawia, że tożsame stają się obiekty x_1, x_2 i x_6 (wszystkie mają tę samą wartość *Age*), gdyż przy takim jednoelementowym zbiorze atrybutów B nic innego ich od siebie nie rozróżnia.

Pytanie. Jakie obiekty są tożsame, jeżeli B kolejno równe jest {*Age*}, {*LEMS*}, {*Age, LEMS*}?

2.4.1 Relacja równoważności

Z każdym podzbiorem atrybutów: $B \subseteq A$ związana jest relacja IND (ang. *indiscernibility* – nierozróżnialność, tożsamość) (4):

$$IND(B) = \{(x, x') \in U^2 \mid \forall a \in B \;\; a(x)=a(x')\} \quad (4)$$

Zapis wzoru (4) można odczytać: zbiór takich par (x, x') z uniwersum, że dla każdego ich atrybutu a z podzbioru atrybutów B, wartości atrybutu dla obu obiektów są równe.

Jeżeli: $(x, x') \in IND(B)$, to x i x' są tożsame względem relacji $IND(B)$, tj. nierozróżnialne względem atrybutów B.

W teorii zbiorów (ogólnej teorii, nie chodzi o zbiory przybliżone), mówi się o relacjach obiektów. Między innymi wyróżnia się tzw. **relację równoważności** dwóch obiektów. Aby sprawdzić czy dana relacja R jest relacją równoważności, należy sprawdzić, czy dla każdego x, y, z zachodzą trzy warunki:

(xRx), co można zapisać jako $(x,x) \in R$ – **zwrotność** relacji

Z faktu (xRy) wynika (yRx) – **symetryczność** relacji

Z faktu, że (yRx) i (yRz) wynika (xRz) – **przechodniość** relacji.

Okazuje się, że relacja IND(B) jest taką relacją równoważności.

Przykład. Niech x, y, z będą liczbami naturalnymi a relacja R będzie równością „=”. Prosto wykazać można, że „=” jest zwrotna, bo $x=x$; symetryczna (jeśli ($x=y$) to ($y=x$)); przechodnia (jeśli $x=y$ i $y=z$ to $x=z$).

Sprawdzić można, że relacja większości „>” nie jest relacją równoważności, ale jest przechodnia.

2.4.2 Klasa abstrakcji

Dla każdego dowolnego obiektu x odnalezione w uniwersum inne obiekty o tych samych wartościach atrybutów B tworzą zbiór nazywany **klasą abstrakcji**, oznaczany jako $[x]_B$. Zwrócić należy uwagę na indeks $_B$ w tym zapisie, który sugeruje, że dla innych $B \subseteq A$ te klasy mogą być inne.

Zauważamy, że wewnątrz klasy abstrakcji obiekty są tożsame ze sobą. Aby coś powiedzieć o klasie abstrakcji, wystarczy zbadać (określić atrybuty) jeden z tych obiektów.

Przykład. Jeżeli B to kolor dominujący widzianego przedmiotu, to czerwony przedmiot x generuje klasę abstrakcji zawierającą wszystkie czerwone obiekty z uniwersum. Dla innych B, klasami abstrakcji mogą być np. obiekty o identycznym *rozmiarze i kolorze, wadze i płci,* itd.

2.4.3 Zbiory elementarne

Dla przykładu pacjentów w zbiorze atrybutów A występują trzy niepuste podzbiory:

$$B_1=\{Age\}$$

$$B_2=\{LEMS\}$$

$$B_3=\{Age,LEMS\}$$

Dla każdego z nich uzyskujemy inne relacje równoważności:

$$IND(\{Age\}) = \{\{x_1; x_2; x_6\}; \{x_3; x_4\}; \{x_5; x_7\}\}$$

$$IND(\{LEMS\}) = \{\{x_1\}; \{x_2\}; \{x_3; x_4\}; \{x_5; x_6; x_7\}\}$$

$$IND(\{Age,LEMS\}) = \{\{x_1\}; \{x_2\}; \{x_3; x_4\}; \{x_5; x_7\}; \{x_6\}\}$$

Każda taka relacja równoważności prowadzi do podziału uniwersum na tzw. **zbiory elementarne**. Można zwrócić uwagę, że uwzględnienie większej liczby atrybutów ({*Age*} w porównaniu do {*Age, LEMS*}) zwykle może prowadzić (nie musi) do uzyskania „drobniejszego” podziału uniwersum na zbiory elementarne.

2.5 Aproksymacja zbioru

Każda suma zbiorów elementarnych tworzy tzw. **zbiór definiowalny**. Rodzina zbiorów definiowalnych (wszystkie możliwe zbiory definiowalne, wszystkie możliwe sumy zbiorów elementarnych) oznaczana jest jako *Def*(*B*). Podziały uniwersum na zbiory elementarne za pomocą relacji równoważności służą do tworzenia podzbiorów uniwersum, które wykorzystuje się w zadaniach klasyfikacji i wnioskowania. Zwykle poszukiwane są podzbiory definiowalne charakteryzujące się taką samą wartością atrybutu decyzyjnego.

Tab. 3. Przykład obiektów tożsamych w systemie decyzyjnym o sprzecznych decyzjach

	Age	*LEMS*	*Walk*
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

W tabeli 3 zaznaczono obiekty, które są tożsame, czyli są w tym samym zbiorze elementarnym, jednakże posiadają różne atrybuty decyzyjne. Korzystając ze zbiorów elementarnych nie da się w tym przypadku stworzyć **zbioru definiowalnego** pacjentów o decyzji jednoznacznej *Yes* lub decyzji *No.* Poszukiwany zbiór pacjentów jest **niedefiniowalny**. Należy posłużyć się jego **aproksymacją**.

2.5.1 Dolna i górna aproksymacja zbioru

Pomimo wykazanej powyżej niejednoznaczności możliwe jest określenie, które obiekty **na pewno należą** do poszukiwanego zbioru, które na pewno do niego **nie należą**, a które leżą **częściowo** w tym zbiorze (na jego granicy). Jeżeli w opisywanym zbiorze jakiekolwiek obiekty leżą na granicy, mamy do czynienia ze zbiorem przybliżonym.

Możliwa jest aproksymacja rozpatrywanego zbioru *X* (w naszym przypadku zbioru z decyzją *Walk=Yes*) wyłącznie przez wykorzystanie atrybutów ze zbioru *B*, poprzez określenie *B-dolnej* (5) i *B-górnej* (6) aproksymacji zbioru *X*:

$$\underline{B}X = \{ x \mid [x]_B \subseteq X \} \tag{5}$$

Zapis ten odczytuje się: do *B-dolnej* aproksymacji należą te *x*, których klasy abstrakcji $[x]_B$ w całości zawierają się w rozpatrywanym zbiorze (ich elementy nie mogą mieć innych decyzji).

$$\bar{B}X = \{ x \mid [x]_B \cap X \neq \varnothing \} \tag{6}$$

Zapis ten odczytuje się: do *B-górnej* aproksymacji należą te *x*, których klasy abstrakcji $[x]_B$ mają część wspólną z rozpatrywanym zbiorem niepustą (co najmniej jeden ich element ma właściwą decyzję).

2.5.2 Przykład

W tabeli 4 zaznaczono elementy na pewno należące do poszukiwanego zbioru *Walk=Yes* (niebieski), czyli takie, których zbiory elementarne i klasy abstrakcji mają decyzję *Walk=Yes* oraz elementy należące do granicy zbioru *Walk=Yes* (czerwony), czyli takie, których klasy abstrakcji mają obiekty o decyzji *Walk=Yes* ale także obiekty o decyzji *Walk=No.*

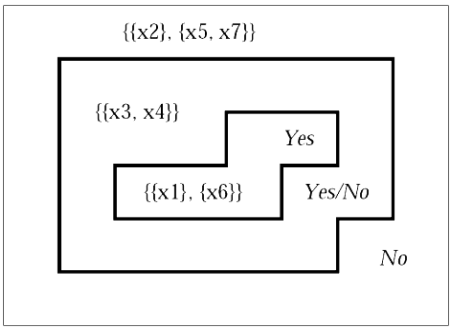
Tab. 4. Podzbiór obiektów wskazujących na decyzję *Yes* oraz ich obiekty tożsame (x_3 i x_4)

	Age	*LEMS*	*Walk*
x_1	16-30	50	Yes
x_2	16-30	0	No
x_3	31-45	1-25	No
x_4	31-45	1-25	Yes
x_5	46-60	26-49	No
x_6	16-30	26-49	Yes
x_7	46-60	26-49	No

B-dolną aproksymacją jest $\{x_1, x_6\}$ – niebieskie, na pewno należą, decyzja jednoznaczna *Walk=Yes*.

B-górną aproksymacją jest $\{x_1, x_3, x_4, x_6\}$ – czerwone i niebieskie, te które mają decyzję jednoznaczną, ale też i te, które mają decyzję jednocześnie *Yes* i *No*.

Wygodna graficzna reprezentacja tego zbioru przedstawiona jest poniżej (rys. 1).



Rys. 1. Graficzna interpretacja zależności między przybliżeniami rozpatrywanego zbioru

Należy zauważyć, że posługiwanie się tylko B_1={*Age*} lub B_2={*LEMS*} prowadzi do innego przybliżenia zbioru *Walk=Yes* niż w powyższym przypadku, gdzie B_3={*Age*, *LEMS*}. Z tego właśnie powodu mówi się „be-górna”, a nie po prostu „górna” aproksymacja.

2.5.3 Obszar graniczny i zewnętrzny

Przybliżenia $\overline{B}X$ i $\underline{B}X$ są zbiorami, na których elementach wykonywać można działania. Między innymi:

$$\mathrm{BND}_B(X) = \overline{B}X - \underline{B}X \tag{7}$$

to **obszar graniczny**, ang. *boundary*, zawiera te obiekty *x*, co do których nie można jednoznacznie zdecydować czy należą czy też nie do zbioru *X.* W rozpatrywanym przykładzie są to obiekty x_3, x_4.

$$\mathrm{EXT}\ B(X) = U - \overline{B}X \tag{8}$$

to **obszar zewnętrzny**, dopełnienie, ang. *exterior*, te *x*, które z całą pewnością nie należą do zbioru *X.* Obiekty x_2, x_5, x_7.

2.5.4 Dokładność przybliżenia

Aproksymację zbioru wyznacza się w oparciu o wiedzę zawartą w *B*. Zakłada się, że rozszerzanie liczby atrybutów, poprzez pomiar innych cech obiektów prowadzić będzie do uzyskania lepszej wiedzy o nich i do zmniejszenia mocy (liczby elementów) obszaru granicznego, czyli niejednoznacznych decyzji. Aby liczbowo wyrażać wpływ doboru atrybutów na dokładność przybliżenia rozpatrywanego zbioru obiektów, oblicza się miarę dokładności (9):

$$\alpha_B(X) = \frac{|\underline{B}X|}{|\overline{B}X|} \tag{9}$$

gdzie |…| oznacza moc zbioru, tj. liczbę jego elementów. Miara ta przyjmuje wartości $0 \le \alpha_B(X) \le 1$, gdzie $\alpha_B(X)$ =1 zachodzi dla zbioru tradycyjnego, który ma pusty obszar graniczny $\mathrm{BND}_B(X)$, a $\alpha_B(X)$ <1 dla zbiorów przybliżonych. Podstawiając ze wzoru (7) na obszar graniczny uzyskuje się (10):

$$\alpha_B(X) = \frac{|\underline{B}X|}{|\underline{B}X| + |\mathrm{BND}_B(X)|} \tag{10}$$

Z powyższego wyraźnie widać, że dla $|BND_B(X)|$ równego zero miara przyjmuje wartość 1, zbiór jest w pełni określony, tradycyjny, a im większe $|BND_B(X)|$, tym większy mianownik i mniejsza miara $\alpha_B(X)$. W końcu dla zbioru, który nie ma jednoznacznych obiektów, czyli jego $|\underline{B}X|$=0 miara ta przyjmuje wartość 0.

2.6 Własności zbiorów przybliżonych

Zachodzą następujące właściwości.

1) $\underline{B}X \subseteq X \subseteq \overline{B}X$
2) $\underline{B}U = U = \overline{B}U$
3) $\underline{B}\varnothing = \varnothing = \overline{B}\varnothing$
4) $\underline{B}\underline{B}X = \overline{B}\underline{B}X = \underline{B}X$
5) $\overline{B}\overline{B}X = \underline{B}\overline{B}X = \overline{B}X$
6) $\overline{B}(-X) = -\underline{B}X$
7) $\underline{B}(-X) = -\overline{B}X$

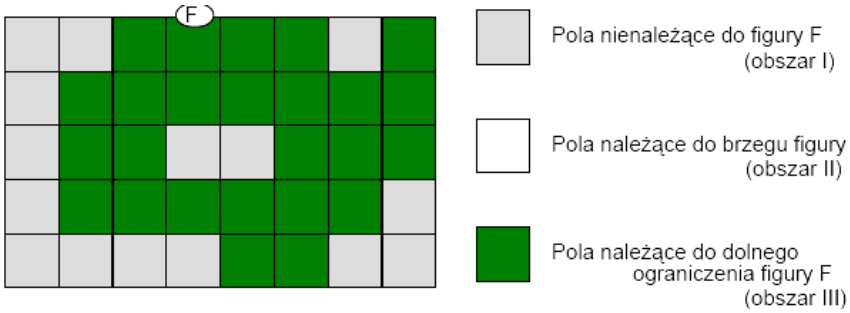
$-X$ oznacza $U-X$

8) $\overline{B}(X \cup Y) = \overline{B}X \cup \overline{B}Y$
9) $\overline{B}(X \cap Y) = \underline{B}X \cap \underline{B}Y$
10) $\underline{B}(X \cup Y) \supseteq \underline{B}X \cup \underline{B}Y$
11) $\underline{B}(X \cap Y) \subseteq \underline{B}X \cap \underline{B}Y$
12) $BN(X \cup Y) \subseteq BNX \cup BNY$
13) $BNX = BN(-X)$
14) $\underline{B}(BNX) = \varnothing$
15) $\overline{B}(BNX) = BNX$
16) $BNX = \varnothing \Leftrightarrow \underline{B}X = X$

2.7 Kategorie zbiorów przybliżonych

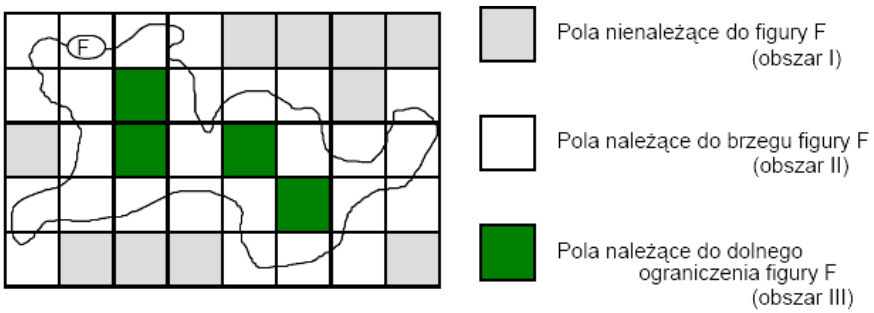
W zależności od liczności elementów w przybliżeniach górnych i dolnych określa się kilka typów zbiorów przybliżonych. Kategorie te można przedstawiać w sposób graficzny (rys. 2-6). Figura F oznacza hipotetyczny poszukiwany zbiór, małe kwadraty to zbiory elementarne, wewnątrz których znajdują się obiekty *x* (jeden kwadrat to cała klasa abstrakcji, nie określa się tu ile obiektów jest wewnątrz).

X jest zbiorem klasycznym, definiowalnym, gdy $\underline{B}(X) \neq \varnothing$ i $\overline{B}(X) = \underline{B}(X)$, $BND_B(X) = \varnothing$



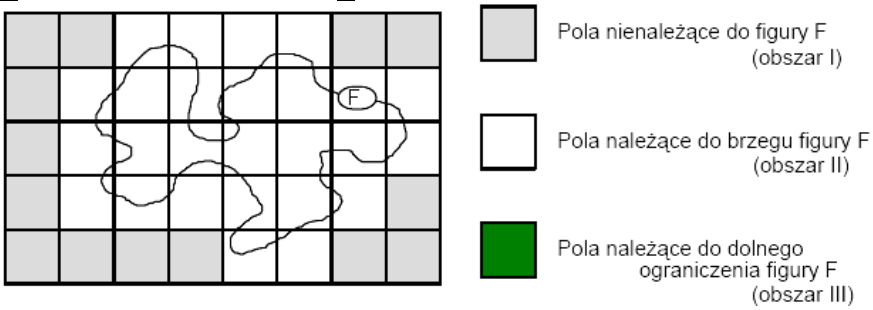
Rys. 2. Przykład zbioru klasycznego

X jest w przybliżeniu *B-definiowalny*, gdy: $\underline{B}(X) \neq \varnothing$ i $\overline{B}(X) \neq U$



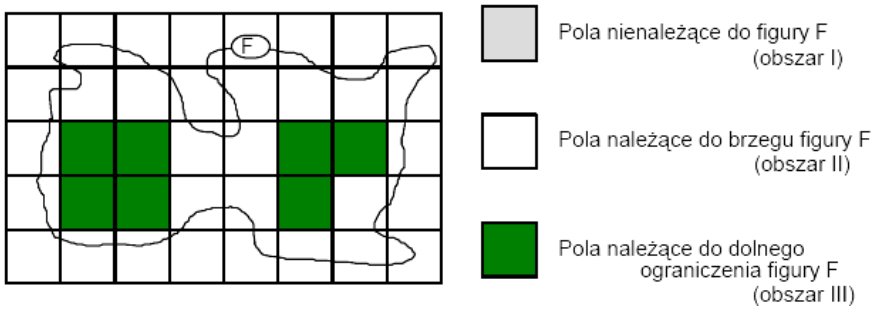
Rys. 3. Przykład zbioru *B*-definiowalnego

X jest wewnętrznie *B*-***nie****definiowalny*, gdy: $\underline{B}(X) = \emptyset$ i $\overline{B}(X) \neq U$



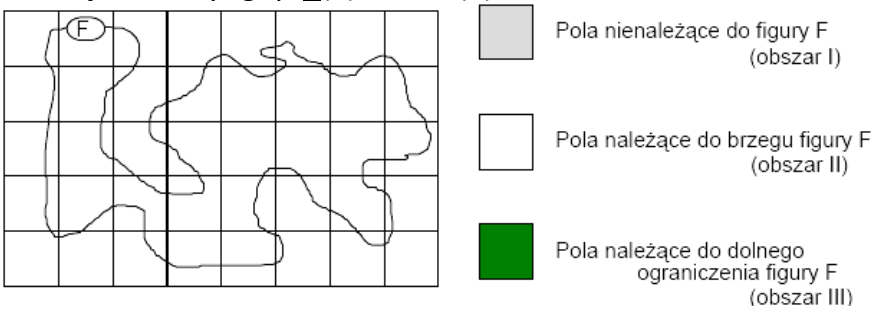
Rys. 4. Przykład zbioru wewnętrznie *B*-niedefiniowalnego

X jest zewnętrznie *B*-***nie****definiowalny*, gdy: $\underline{B}(X) \neq \emptyset$ i $\overline{B}(X) = U$



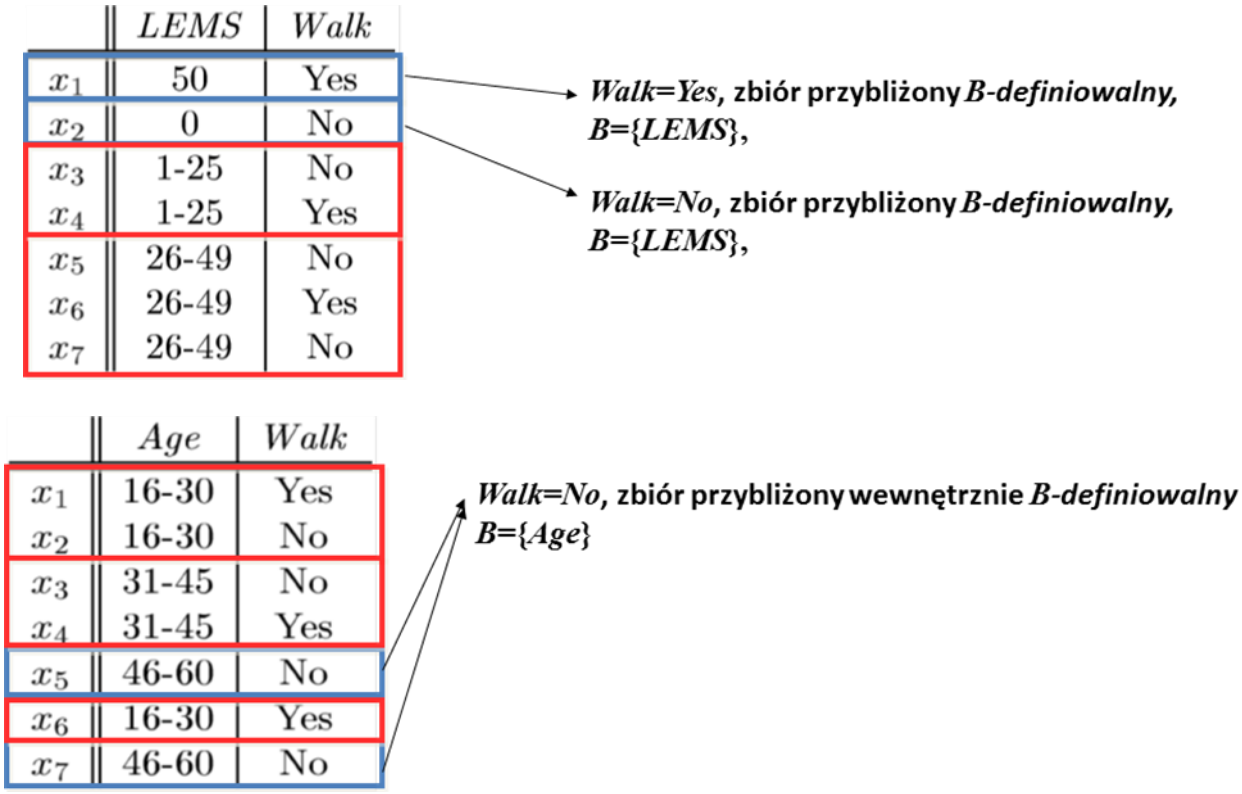
Rys. 5. Przykład zbioru zewnętrznie *B*-niedefiniowalnego

X jest całkowicie *B*-***nie****definiowalny*, gdy: $\underline{B}(X) = \emptyset$ i $\overline{B}(X) = U$



Rys. 6. Przykład zbioru całkowicie *B*-niedefiniowalnego

Dla przypadku medycznego można poszukać różnych typów zbiorów przybliżonych w podanym systemie decyzyjnym (rys. 7).



Rys. 7. Przykłady zbiorów przybliżonych

2.8 Redukty

Różne podzbiory atrybutów $B_1 \subseteq A$ i $B_2 \subseteq A$ mogą prowadzić do identycznych podziałów uniwersum, czyli takich, że $IND_{B1}(X)=IND_{B2}(X)$. Oznacza to, że zbiory elementarne uzyskane z tych podziałów są identyczne, a w konsekwencji także i aproksymacja jest równie precyzyjna.

Redukt B to taki podzbiór atrybutów, który ma minimalną ilość atrybutów, a ponadto $IND_B(X)= IND_A(X)$ (generuje taki sam podział jak cały zbiór atrybutów).

Wyznaczenie reduktu polega na pozostawieniu w podzbiorze tylko tych atrybutów, które zachowują relację równoważności, czyli nie zmieniają aproksymacji zbioru i na usunięciu pozostałych.

Zwykle dla danego systemu decyzyjnego istnieć może wiele reduktów. Czasami ze względów praktycznych nie korzysta się z reduktu najkrótszego (z najmniejszą liczbą atrybutów), tylko z takiego, którego implementacja jest np. najprostsza, a sposób pomiaru wartości atrybutu najmniej kosztowny, najszybszy, itd. Przykładowo w prostych przypadkach przeziębienia, pacjenta można pytać o samopoczucie (dobre, złe) i uzyskać te same wyniki, jak przy użyciu drogiego, czasochłonnego pomiaru temperatury, ciśnienia krwi, morfologii.

Wyznaczanie reduktów jest problem NP-trudnym. Liczba możliwych reduktów wyraża się dwumianem $\binom{m}{\lfloor \frac{m}{2} \rfloor}$, gdzie *m* – liczba atrybutów, a $\lfloor \ \rfloor$ oznacza zaokrąglenie w dół do liczby całkowitej.

Proces wyznaczania reduktów uważany jest za „wąskie gardło” w systemach wnioskowania opartych na zbiorach przybliżonych. Często stosowane są metody genetyczne do wyznaczania reduktów dla dużych systemów decyzyjnych z dziesiątkami lub setkami atrybutów.

2.9 Wykorzystanie reguł decyzyjnych w klasyfikacji

Po wyznaczeniu reduktu (reduktów) możliwe jest wygenerowanie z danych dostępnych w systemie decyzyjnym reguł logicznych o postaci **JEŻELI** … **I** … **TO** …. Przykładowa uproszczona funkcja logiczna dla zagadnienia rekrutacji do pracy posiada dwa redukty: $f_A(d,e,f,r)=(d \wedge e) \vee (e \wedge r)$.

Dla każdego z nich należy odczytywać z danych z systemu decyzyjnego (z tabeli) występujące tam wartości atrybutów i odpowiadające im decyzje, tworząc w ten sposób zestaw reguł, opisujących dostępne przypadki, np. **JEŻELI** *Diploma*=MBA **I** *Experience*=Medium **TO** *Decision*=Accept.

Reguł utworzyć można tyle, ile wynosi liczba obiektów rozróżnialnych (nie tożsamych) pomnożona przez liczbę reduktów. Nie każdą z reguł trzeba jednak wykorzystywać czy implementować w procesie decyzyjnym. Często istotnym elementem procesu selekcji reguł jest wiedza ekspercka wykorzystywana do weryfikacji reguł.

2.9.1 Klasyfikacja

Proces klasyfikacji – nadawania decyzji – nowemu obiektowi przebiega w kilku typowych krokach:

- Obliczenie atrybutów nowego obiektu, czyli pomiar jego cech, tych, które są istotne dla decyzji, powiązanych z treścią reduktów,
- Poszukiwanie reguł pasujących do wartości atrybutów,
- Jeżeli **brak pasujących reguł**, to wynikiem jest **najczęstsza** decyzja w systemie decyzyjnym, lub decyzja **najmniej kosztowna**, np. czasem mniejsze może być ryzyko odrzucenia dobrego kandydata do pracy, lub mniejsze może być ryzyko przyjęcia niepewnego kandydata na okres próbny,

- Jeżeli **pasuje wiele reguł**, to mogą one wskazywać na różne decyzje, wówczas przeprowadzane jest głosowanie – wybierana jest odpowiedź pojawiająca się najczęściej.

W procesie klasyfikacji napotkać można kilka przypadków:

- nowy obiekt pasuje dokładnie do jednej **deterministycznej** reguły - przypadek najbardziej pożądany - uzyskuje się wiadomość, iż obiekt należy do zadanej klasy, do **dolnego przybliżenia zbioru**,
- nowy obiekt pasuje dokładnie do jednej, **niedeterministycznej** reguły - przypadek taki jest nadal pozytywny, choć tym razem uzyskuje się jedynie wiadomość, iż obiekt prawdopodobnie należy do zbioru - a wiec, że należy do jego **górnego przybliżenia**,
- nowy obiekt pasuje do **więcej niż jednej** reguły - kilka potencjalnych przynależności obiektu, a więc decyzja nie jest jednoznaczna; zazwyczaj w takim przypadku stosuje się dodatkowe kryteria dla oceny, do której z klas z największym prawdopodobieństwem należy obiekt. Należy zauważyć, że problem ten nie występowałby, gdyby wszystkie klasy obiektów były parami rozłączne, lecz jest to warunek trudny do spełnienia i jest rzadko stosowany.

2.9.2 Aktualizacja systemu wnioskującego

W wyniku napotykania nowych przypadków i konfrontowania decyzji systemu z decyzją eksperta, kiedy to „życie weryfikuje” poprawność wcześniejszych decyzji, system decyzyjny i bazę reguł można aktualizować. Dodanie nowego przypadku jest szczególnie wskazane, jeżeli:
- jest on opisany wartościami atrybutów, które wcześniej w systemie nie występowały,
- był niewłaściwie sklasyfikowany a jego dodanie wygeneruje nową regułę, poprawiającą decyzję,
- dodanie obiektu i powtórne generowanie reduktu ujawni nowe, przydatne atrybuty.

2.9.3 Jakość decyzji

Zwykle dysponuje się tzw. danymi testowymi, dla których znana jest decyzja a sprawdzane jest, czy odpowiedź zwracana przez system jest z nią identyczna. Wówczas można wyznaczyć jakość decyzji i dokonać oceny systemu. W tym celu wykorzystuje się pojęcie obszaru B-pozytywnego.

2.9.4 Klasa decyzyjna i obszar B-pozytywny

Wprowadzony zostaje następujący formalizm.

r – liczba decyzji w systemie decyzyjnym,

v^i_d – wartość decyzji dla *i*-tego obiektu, np. *Yes*, *No, Accept, Reject,... i*=1,...,*r*

V_d={v^1_d, ... $v^{r(d)}_d$} – zbiór wszystkich wartości decyzji w systemie decyzyjnym, np. {*Yes*,*No*}

X^k_A = {$x \in U$ / $d(x)$= v^k_d} – *k*-ta klasa decyzyjna, *k*=1,...,*r*, czyli zbiór tych wszystkich obiektów x, dla których decyzja $d(x)$ ma wartość *k*-tą v^k_d.

Wówczas analizując wszystkie *r* klas otrzymuje się zbiór klas decyzyjnych $CLASS(d)=\{X^1_A,...X^r_A\}$. Decyzja *d* determinuje podział uniwersum na *r* zbiorów, np. $U = X^{Yes} \cup X^{No}$, oznacza podział na obiekty o decyzji *Yes* i decyzji *No*. W idealnym przypadku są to decyzje jednoznaczne, a w przypadku zbioru przybliżonego o decyzjach niedeterministycznych taka suma może być różna od *U*.

W jednoznaczny sposób jakość decyzji określa się poprzez wyliczenie i posumowanie przybliżeń dolnych wszystkich klas decyzyjnych X^i:

$$POS_B(d) = \underline{B}X^1 \cup \underline{B}X^2 \cup ... \cup \underline{B}X^r \qquad (19)$$

co nazywane jest obszarem B-pozytywnym. System decyzyjny jest **deterministyczny** (zgodny), jeżeli $POS_B(d) = U$ (do dolnych przybliżeń należą wszystkie obiekty uniwersum, inaczej: dla każdego obiektu uniwersum istnieje klasa decyzyjna, w której dolnym przybliżeniu jest ten obiekt) w przeciwnym wypadku jest **niedeterministyczny**.

2.10 Dyskretyzacja parametrów

Obiekty w systemie decyzyjnym mogą w ogólności być opisane atrybutami liczbowymi o dowolnej dokładności z ciągłej dziedziny, co przedstawiono na poniższym przykładzie (rys. 10). Nie ma pewności, że nieznane przypadki w przyszłości będą miały podobne wartości – czy cecha *a* może mieć wartość mniejszą od 0,8 lub większą od 1,6?; czy dokładność do jednego miejsca po przecinku jest tu wystarczająca?; jakie reguły należy wówczas utworzyć dla brakujących wartości?; czy cecha *b* dla drugiego obiektu tylko w wyniku błędu pomiaru jest wyrażona ułamkiem, może wskazane jest zaokrąglenie do liczy całkowitej? Te i podobne wątpliwości można zaniedbać, jeżeli dokona się prawidłowej dyskretyzacji parametrów – zamiany wartości ciągłych na całkowite (lub na etykiety słowne, nazwy przedziałów). W poniższym przykładzie zauważyć można, że u_3 i u_5 stały się tożsame po dyskretyzacji, jednak mają tę samą decyzję *d*, podobnie jak u_4 i u_7. W wyniku dyskretyzacji utracono możliwość rozróżniania obiektów między sobą, jednak bez obniżenia miary jakości aproksymacji zbiorów.

$\mathbf{A}$	a	b	d
u_1	0.8	2	1
u_2	1	0.5	0
u_3	1.3	3	0
u_4	1.4	1	1
u_5	1.4	2	0
u_6	1.6	3	1
u_7	1.3	1	1

$\mathbf{A^P}$	a^P	b^P	d
u_1	0	2	1
u_2	1	0	0
u_3	1	2	0
u_4	1	1	1
u_5	1	2	0
u_6	2	2	1
u_7	1	1	1

Rys. 9. System decyzyjny przed i po dyskretyzacji

Zasadę dyskretyzacji (użyteczną m.in. na etapie wstępnego przetwarzania danych wejściowych przez algorytm) zapisuje się w postaci zbioru cięć (**P** oznacza partycjonowanie), np. **P** = {(*a*; 0,9); (*a*; 1,5); (*b*; 0,75); (*b*; 1,5)}.

Taki zbiór cięć odczytuje się następująco: dziedzinę atrybutu *a*, należy podzielić w punkcie 0,9 i w punkcie 1,5, uzyskując trzy przedziały (–∞; 0,9), <0,9; 1,5), <1,5; ∞). W powyższym przypadku przedziały te zostały nazwane liczbami całkowitymi 0, 1, 2 (rys. 9), jednak można także użyć etykiet słownych, lub innej numeracji.

Powyższe cięcia uzyskano w następujący sposób:

Krok 1. dla atrybutu poszeregowano wartości i utworzono z nich przedziały:

<0.8;1); <1; 1.3); <1.3; 1.4); <1.4; 1.6) dla *a*

<0.5; 1); <1; 2); <2; 3) dla *b*

Krok 2. środki przedziałów przyjmowane są za miejsca cięć:

(*a*; 0.9); (*a*; 1.15); (*a*; 1.35); (*a*; 1.5);

(*b*; 0.75); (*b*; 1.5); (*b*; 2.5)

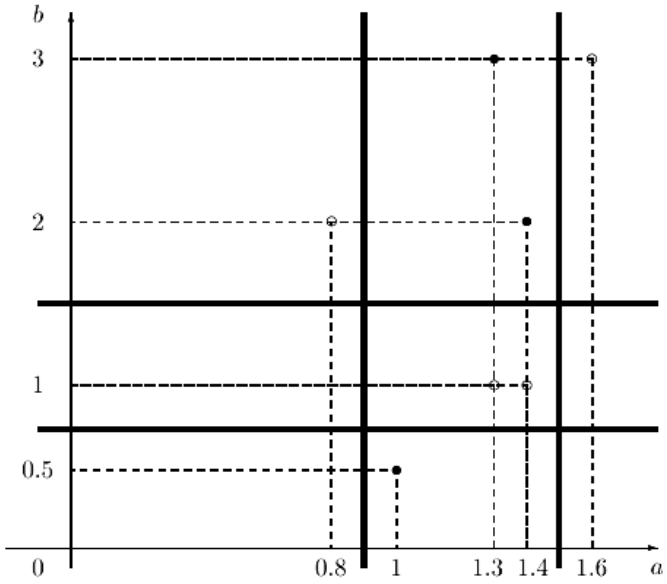
Krok 3. usunięto te cięcia, które nie prowadzą do rozróżnienia choć jednej pary obiektów:

(*a*; 0.9); ~~(*a*; 1.15);~~ ~~(*a*; 1.35);~~ (*a*; 1.5);

(*b*; 0.75); (*b*; 1.5); ~~(*b*; 2.5);~~

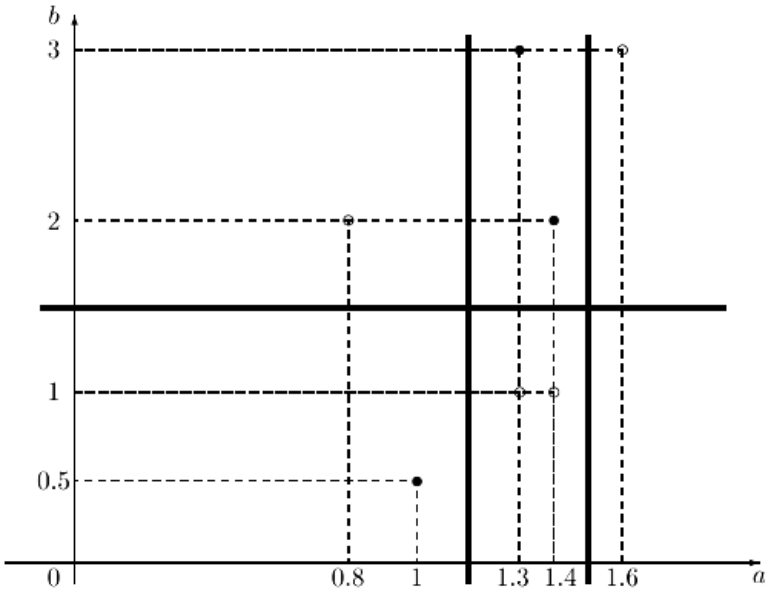
Uzyskane cięcia tworzą zbiór **P** = {(*a*; 0.9); (*a*; 1.5); (*b*; 0.75); (*b*; 1.5)}.

Graficzna interpretacja przedstawiona jest poniżej (rys. 10), punkty czarne (koła) to obiekty o decyzji „1”, punkty białe (okręgi) – o decyzji „0”. Grube linie to cięcia. Usunięte cięcie (*b*; 2.5) nie prowadziło do rozróżniania kół od okręgów, nie było potrzebne.



Rys. 10. Przykładowa dyskretyzacja dziedzin dwóch atrybutów

Należy zwrócić uwagę, że w wyniku takiego podziału uzyskuje się 9 obszarów, co dla innych cięć wyliczane jest jako (*n*+1)(*k*+1), gdzie *n* i *k* to liczba cięć dla atrybutów *a* i *b*. Niestety taki podział nie jest optymalny – występują **obszary bez decyzji** (w tym przykładzie cztery obszary bez obiektów). Nowy obiekt o wartościach atrybutów, lokujących go w jednym z tych przedziałów nie zostanie sklasyfikowany. Zamiast powyższego podejścia stosuje się często algorytm MD-Heuristics [4], wyznaczający dyskretyzację o mniejszej liczbie cięć i obszarów niepewnych (rys. 11).

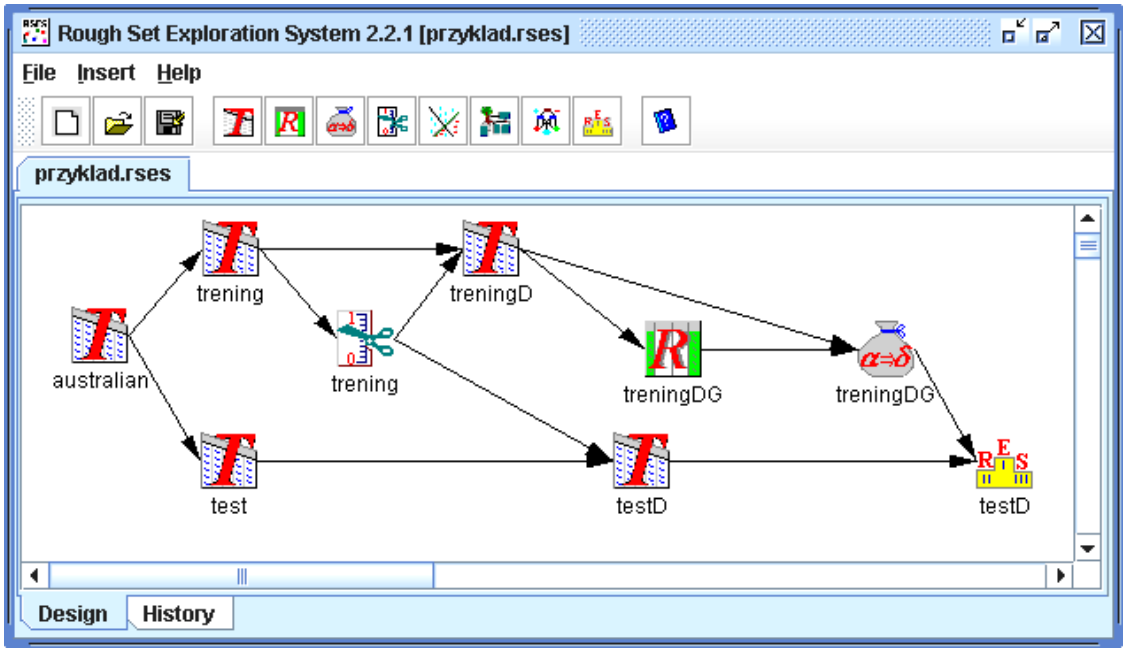


Rys. 11. Dyskretyzacja o mniejszej liczbie cięć i mniejszej liczbie obszarów bez decyzji

Dla tych samych danych jest mniej obszarów i cięć (łatwiejsza implementacja) oraz liczba obszarów pustych (bez decyzji), zmalała do jednego.

3 Funkcje oprogramowania RSES

Rough Set Exploration System to aplikacja autorstwa naukowców z Uniwersytetu Warszawskiego (http://logic.mimuw.edu.pl/~rses/). Udostępnia wszystkie najważniejsze operacje na danych, dotyczące teorii zbiorów przybliżonych i jej zastosowań. Praca z programem polega na budowaniu grafu przetwarzania danych (rys. 12).



Rys. 12. Graf przetwarzania danych: T to tabele, zbiór cięć oznacza ikona z nożyczkami, R to redukty, reguły zapisane są w strukturze dostępnej pod ikoną worka, wyniki pod ikoną podium

Standardowy sposób budowania klasyfikatora zrealizować można następująco:

- wczytać tabelę z danymi
- dokonać jej podziału na część trenującą i testującą (z menu kontekstowego – prawy przycisk myszy na ikonie tabeli danych)
- na części treningowej wykonać generowanie cięć (Generate Cuts z menu kontekstowego)
- na części treningowej wykonać dyskretyzację (Discretize z menu kontekstowego), wskazując w oknie dialogowym źródło danych i źródło cięć (w grafie symbolizowane jest to początkami strzałek)
- na danych dyskretnych przeprowadzić wyliczanie reduktu
- na redukcie wykonać polecenie generowania reguł, jako źródło podając zbiór danych treningowych dyskretyzowanych
- na danych testujących wykonać dyskretyzację tym samym zbiorem cięć

- na danych testujących dyskretyzowanych wykonać klasyfikację z użyciem wygenerowanych reguł.

Aplikacja RSES dostarcza wygodnych graficznych narzędzi do przeglądania danych, cięć, reguł i wyników (rys. 13-18).

Table: trening

A8	A9	A10	A11	A12	A13	A14	CLASS
0	0	0	1	2	100	1213	0
0	0	0	0	2	160	1	0
0	0	0	1	2	280	1	0
1	1	11	1	2	0	1	1
1	1	14	0	2	60	159	1
1	1	6	0	2	43	561	1
0	0	0	0	2	176	538	0
1	1	3	1	2	100	51	0
1	1	4	1	2	253	858	1
1	1	6	1	2	470	1	1
1	1	6	1	2	0	1001	1

Rys. 13. Przykładowa tabela z danymi trenującymi. Stosować można dane binarne, liczby naturalne, rzeczywiste oraz etykiety słowne

Cut set: trening

(1-14)	Attribute	Size	Description
1	A1	0	*
2	A2	8	21.04; 21.21; 23.04; 24.0; 27.915; 31.125; 37.25; 45.58
3	A3	5	0.555; 2.23; 4.48; 6.02; 8.54
4	A4	0	*
5	A5	0	*
6	A6	0	*
7	A7	3	0.1875; 1.02; 3.1675
8	A8	0	*
9	A9	0	*
10	A10	2	0.5; 2.5
11	A11	0	*
12	A12	0	*
13	A13	4	75.5; 111.0; 172.0; 296.0
14	A14	2	13.5; 309.0

Rys. 14. Przykładowy zbiór cięć: gwiazdka oznacza, że dany atrybut nie będzie dyskretyzowany, gdyż obecnie jego wartości mają odpowiednią postać, np. są binarne, lub odpowiednio mało różnych wartości

Table: treningD

A8	A9	A10	A11	A12	A13	A14	CLASS
0	0	"(-Inf,0.5)"	1	2	"(75.5,111.0...	"(309.0,Inf)"	0
0	0	"(-Inf,0.5)"	0	2	"(111.0,172....	"(-Inf,13.5)"	0
0	0	"(-Inf,0.5)"	1	2	"(172.0,296....	"(-Inf,13.5)"	0
1	1	"(2.5,Inf)"	1	2	"(-Inf,75.5)"	"(-Inf,13.5)"	1
1	1	"(2.5,Inf)"	0	2	"(-Inf,75.5)"	"(13.5,309.0...	1
1	1	"(2.5,Inf)"	0	2	"(-Inf,75.5)"	"(309.0,Inf)"	1
0	0	"(-Inf,0.5)"	0	2	"(172.0,296....	"(309.0,Inf)"	0
1	1	"(2.5,Inf)"	1	2	"(75.5,111.0...	"(13.5,309.0...	0
1	1	"(2.5,Inf)"	1	2	"(172.0,296....	"(309.0,Inf)"	1
1	1	"(2.5,Inf)"	1	2	"(296.0,Inf)"	"(-Inf,13.5)"	1
1	1	"(2.5,Inf)"	1	2	"(-Inf,75.5)"	"(309.0,Inf)"	1
0	0	"(-Inf,0.5)"	0	2	"(-Inf,75.5)"	"(309.0,Inf)"	1
0	0	"(-Inf,0.5)"	0	2	"(111.0,172....	"(-Inf,13.5)"	0

Rys. 15. Tabela z danymi po dyskretyzacji. Zauważyć można zastąpienie liczb naturalnych lub rzeczywistych nazwami podprzedziałów, np. „(-Inf, 0.5)” to liczby od $-\infty$ do 0,5

Reduct set: treningDG

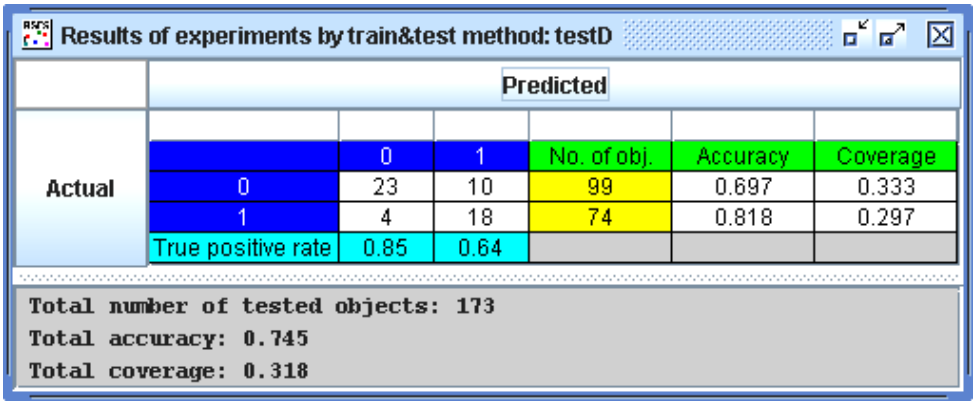
(1-10)	Size	Pos.Reg.	SC	Reducts
1	6	1	1	{ A2, A3, A7, A10, A13, A14 }
2	7	1	1	{ A2, A3, A5, A7, A8, A10, A13 }
3	7	1	1	{ A1, A2, A3, A4, A5, A7, A13 }
4	7	1	1	{ A1, A2, A3, A5, A7, A8, A13 }
5	7	1	1	{ A1, A2, A3, A5, A7, A9, A13 }
6	7	1	1	{ A1, A2, A3, A5, A7, A10, A13 }
7	7	1	1	{ A2, A3, A4, A5, A7, A10, A13 }
8	7	1	1	{ A2, A3, A5, A10, A12, A13, A14 }
9	7	1	1	{ A2, A3, A5, A8, A10, A13, A14 }
10	7	1	1	{ A2, A3, A5, A8, A9, A13, A14 }

Rys. 16. Lista 10 przykładowych reduktów. *Size* oznacza liczbę atrybutów, *Pos.Reg*. to region B-pozytywny

Rule set: treningDG

...	Match	Decision rules
1	1	(A2="(21.21,23.04)")&(A3="(8.54,Inf)")&(A7="(1.02,3.1675)")&(A10="(-Inf,0.5)")&(A13="(75.5,111.0)")&(A14="(309.0,Inf)")=>(CLASS={0[1]})
2	1	(A2="(21.21,23.04)")&(A3="(6.02,8.54)")&(A7="(-Inf,0.1875)")&(A10="(-Inf,0.5)")&(A13="(111.0,172.0)")&(A14="(-Inf,13.5)")=>(CLASS={0[1]})
3	2	(A2="(27.915,31.125)")&(A3="(0.555,2.23)")&(A7="(1.02,3.1675)")&(A10="(-Inf,0.5)")&(A13="(172.0,296.0)")&(A14="(-Inf,13.5)")=>(CLASS={0[2]})
4	1	(A2="(21.21,23.04)")&(A3="(8.54,Inf)")&(A7="(-Inf,0.1875)")&(A10="(2.5,Inf)")&(A13="(-Inf,75.5)")&(A14="(-Inf,13.5)")=>(CLASS={1[1]})
5	1	(A2="(-Inf,21.04)")&(A3="(6.02,8.54)")&(A7="(1.02,3.1675)")&(A10="(2.5,Inf)")&(A13="(-Inf,75.5)")&(A14="(13.5,309.0)")=>(CLASS={1[1]})
6	1	(A2="(45.58,Inf)")&(A3="(2.23,4.48)")&(A7="(1.02,3.1675)")&(A10="(2.5,Inf)")&(A13="(-Inf,75.5)")&(A14="(309.0,Inf)")=>(CLASS={1[1]})
7	1	(A2="(24.0,27.915)")&(A3="(0.555,2.23)")&(A7="(1.02,3.1675)")&(A10="(-Inf,0.5)")&(A13="(172.0,296.0)")&(A14="(309.0,Inf)")=>(CLASS={0[1]})
8	1	(A2="(45.58,Inf)")&(A3="(6.02,8.54)")&(A7="(3.1675,Inf)")&(A10="(2.5,Inf)")&(A13="(75.5,111.0)")&(A14="(13.5,309.0)")=>(CLASS={0[1]})
9	1	(A2="(31.125,37.25)")&(A3="(0.555,2.23)")&(A7="(3.1675,Inf)")&(A10="(2.5,Inf)")&(A13="(172.0,296.0)")&(A14="(309.0,Inf)")=>(CLASS={1[1]})
10	2	(A2="(37.25,45.58)")&(A3="(4.48,6.02)")&(A7="(3.1675,Inf)")&(A10="(2.5,Inf)")&(A13="(296.0,Inf)")&(A14="(-Inf,13.5)")=>(CLASS={1[2]})
11	1	(A2="(31.125,37.25)")&(A3="(4.48,6.02)")&(A7="(3.1675,Inf)")&(A10="(2.5,Inf)")&(A13="(-Inf,75.5)")&(A14="(309.0,Inf)")=>(CLASS={1[1]})
12	1	(A2="(45.58,Inf)")&(A3="(6.02,8.54)")&(A7="(-Inf,0.1875)")&(A10="(-Inf,0.5)")&(A13="(-Inf,75.5)")&(A14="(309.0,Inf)")=>(CLASS={1[1]})
13	1	(A2="(-Inf,21.04)")&(A3="(0.555,2.23)")&(A7="(-Inf,0.1875)")&(A10="(-Inf,0.5)")&(A13="(111.0,172.0)")&(A14="(-Inf,13.5)")=>(CLASS={0[1]})
14	1	(A2="(27.915,31.125)")&(A3="(0.555,2.23)")&(A7="(-Inf,0.1875)")&(A10="(-Inf,0.5)")&(A13="(172.0,296.0)")&(A14="(309.0,Inf)")=>(CLASS={0[1]})
15	1	(A2="(-Inf,21.04)")&(A3="(0.555,2.23)")&(A7="(0.1875,1.02)")&(A10="(-Inf,0.5)")&(A13="(111.0,172.0)")&(A14="(-Inf,13.5)")=>(CLASS={0[1]})
16	4	(A2="(37.25,45.58)")&(A3="(0.555,2.23)")&(A7="(-Inf,0.1875)")&(A10="(-Inf,0.5)")&(A13="(111.0,172.0)")&(A14="(-Inf,13.5)")=>(CLASS={0[4]})
17	1	(A2="(37.25,45.58)")&(A3="(0.555,2.23)")&(A7="(0.1875,1.02)")&(A10="(-Inf,0.5)")&(A13="(111.0,172.0)")&(A14="(-Inf,13.5)")=>(CLASS={0[1]})
18	1	(A2="(-Inf,21.04)")&(A3="(8.54,Inf)")&(A7="(0.1875,1.02)")&(A10="(-Inf,0.5)")&(A13="(75.5,111.0)")&(A14="(309.0,Inf)")=>(CLASS={0[1]})
19	1	(A2="(31.125,37.25)")&(A3="(0.555,2.23)")&(A7="(3.1675,Inf)")&(A10="(2.5,Inf)")&(A13="(-Inf,75.5)")&(A14="(-Inf,13.5)")=>(CLASS={1[1]})
20	2	(A2="(21.21,23.04)")&(A3="(-Inf,0.555)")&(A7="(-Inf,0.1875)")&(A10="(-Inf,0.5)")&(A13="(172.0,296.0)")&(A14="(13.5,309.0)")=>(CLASS={0[2]})

Rys. 17. Treść przykładowych reguł. *Match* oznacza ile obiektów ze zbioru treningowego pasuje do danej reguły

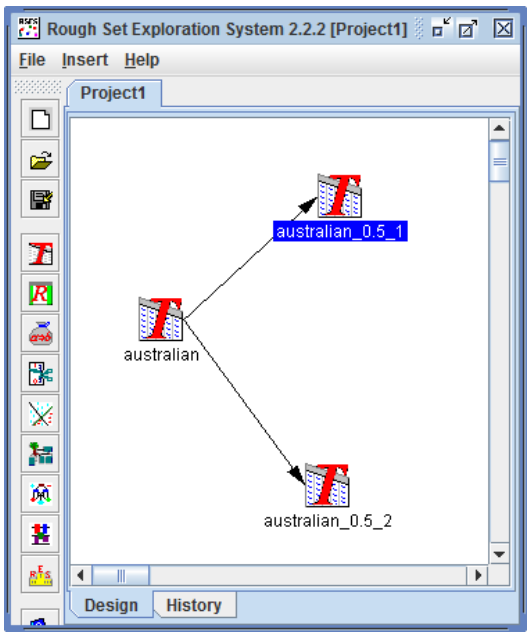


Rys. 18. Wynik klasyfikacji przedstawiony w postaci tzw. macierzy pomyłek. *Actual* to wartości właściwe i oczekiwane, które były dostępne w tabeli testowej, *Predicted* to odpowiedź klasyfikatora

4 Zadania do wykonania

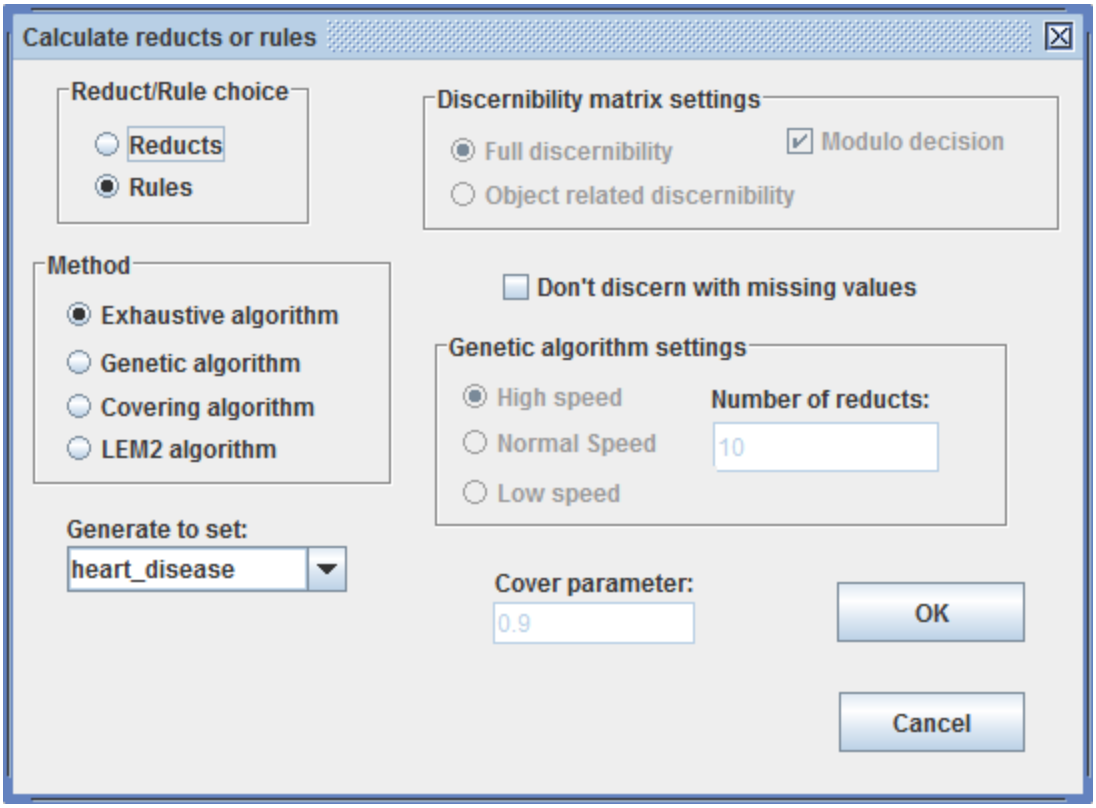
Po uruchomieniu aplikacji należy utworzyć nowy projekt, korzystając z zakładki *File->New Project*. Aby umieścić nową tabelę decyzyjną, należy w pasku bocznym wybrać ikonę tabeli, a następnie kliknąć prawym przyciskiem myszy na nowo powstałym obiekcie i załadować dane (*load*).

Zbiór danych można podzielić na mniejsze tabele, klikając prawym przyciskiem myszy na obiekcie i wybierając opcję ”*Split in two*” podziału na dwie części w proporcjach podanych w oknie dialogowym.



Rys. 19. Wynik podziału tabeli na dwie rozłączne części

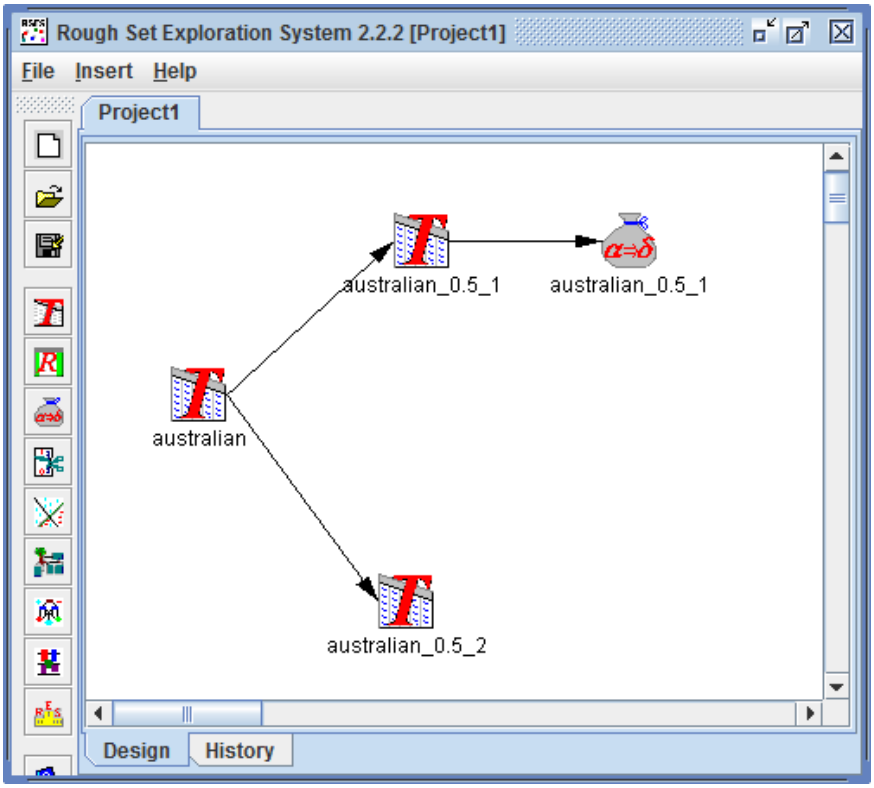
Redukty i reguły decyzyjne tworzone są poprzez polecenia *Reducts/Rules/->Calculate reducts or rules*. Wyświetlane jest okno dialogowe umożliwiające wybór metod i opcji obliczania reguł czy reduktów (rys. 20).



Rys. 20. Wybór metod wyznaczania reduktów i reguł, szczegółowy opis funkcji w instrukcji programu RSES [5]

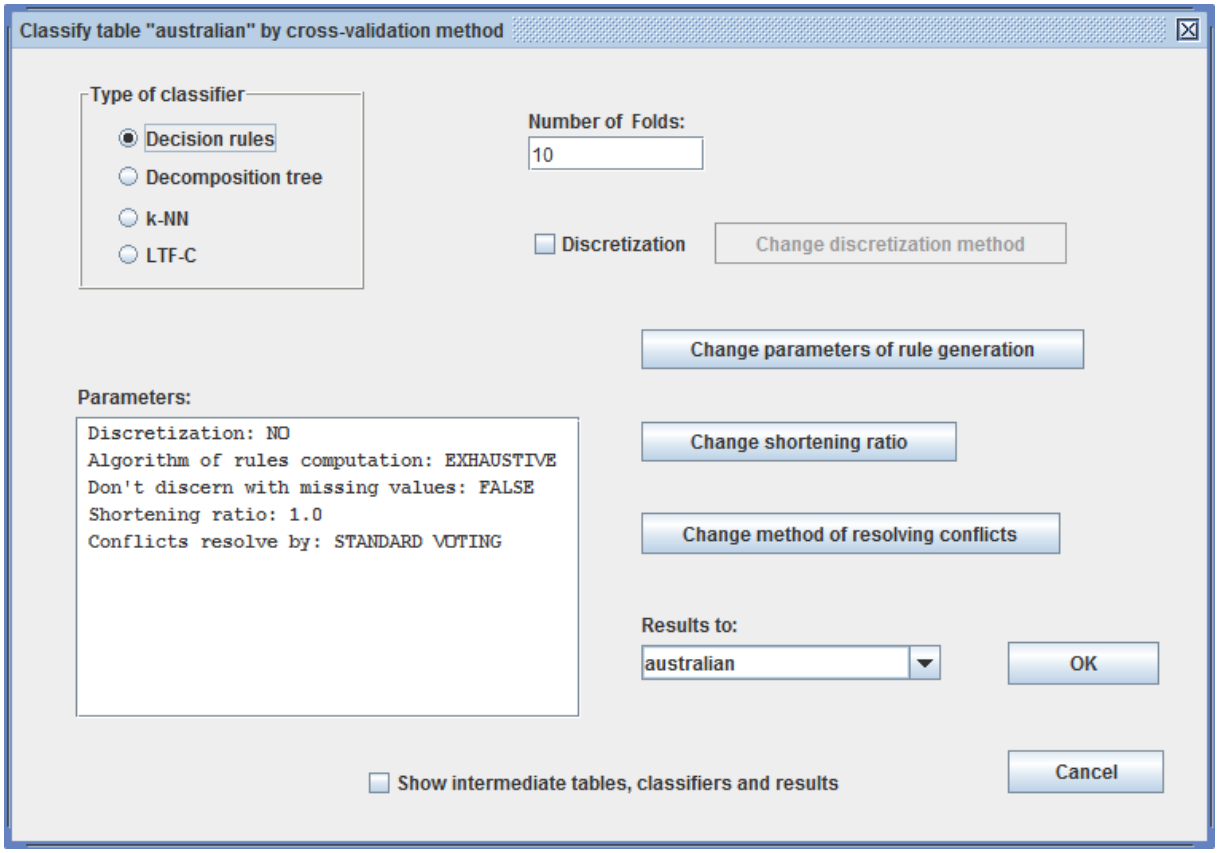
Każdorazowo program oczekuje zdefiniowania nazwy zbioru, do którego zapisane mają zostać reguły (menu *generale to set* na rys. 20), dlatego przed utworzeniem nowego zbioru reguł lub reduktów należy wcześniej utworzyć nowe obiekty (wybierając odpowiednie ikony w pasku bocznym), lub automatycznie utworzyć nowe zbiory poprzez wpisanie we wskazane okno dialogowe nowych nazw. Domyślnie wynik jest zapisywany do zbioru reduktów o nazwie identycznej z nazwą tabeli źródłowej.

Analogicznie, poprzez wybór odpowiednich poleceń z menu kontekstowego tabeli możliwe jest stworzenie sieci neuronowej, drzewa decyzyjnego oraz kombinacji liniowych parametrów.

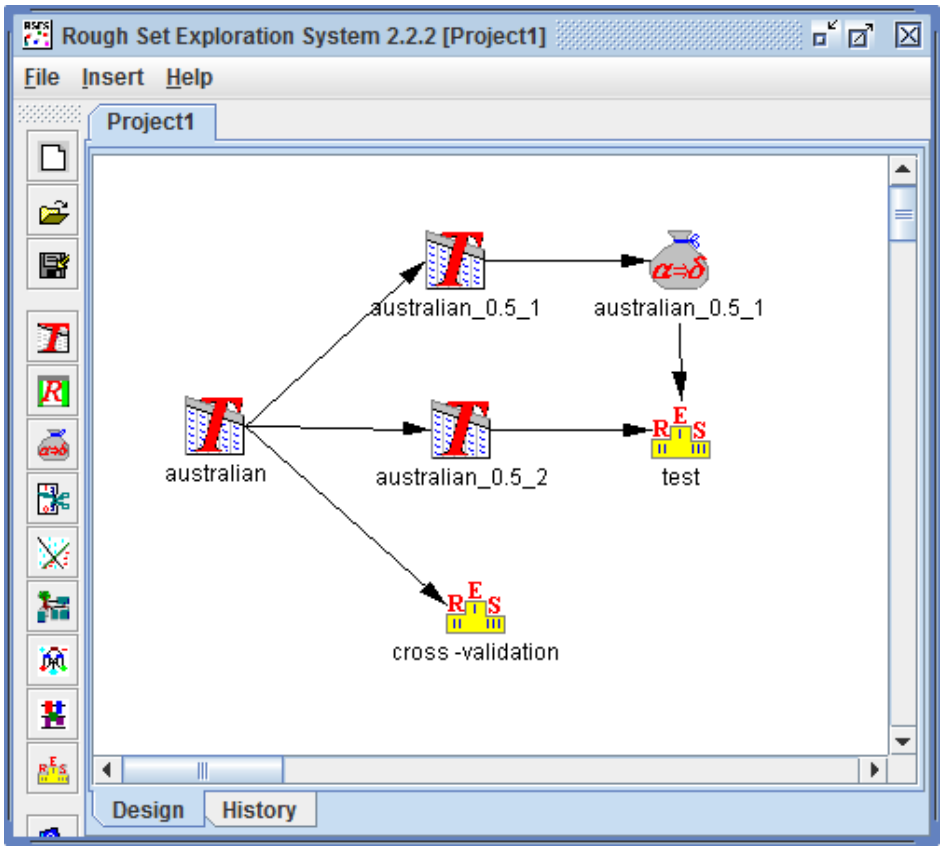


Rys. 21. Wynik tworzenia reguł na podstawie jednej z tabel

Reguły mogą być też generowane na podstawie reduktów, wówczas należy na zbiorze utworzonych reduktów kliknąć prawy przycisk myszy i wybrać opcję *Generate rules*. Uzyskane reguły można skracać, generalizować oraz filtrować. Opcje te dostępne są z menu kontekstowego reguł po kliknięciu na ikonie reguł prawym przyciskiem myszy. Klasyfikacja możliwa jest w dwóch procedurach – w procedurze walidacji krzyżowej (*cross-validation*) oraz w procedurze zbiór testowy – zbiór treningowy. Aby sklasyfikować w procedurze walidacji krzyżowej, należy zaznaczyć tabelę, która ma zostać sklasyfikowana, kliknąć prawym przycisk myszy i wybrać opcję *Classify->Cross-validation*. Pojawi się okno dialogowe, umożliwiające wybór metody klasyfikacji. Jeżeli jedna tabela ma być klasyfikowana kilka razy, należy pamiętać o stworzeniu wcześniej obiektów do przechowywania wyników oraz wyborze odpowiedniego obiektu, do którego wyniki mają zostać zapisane.

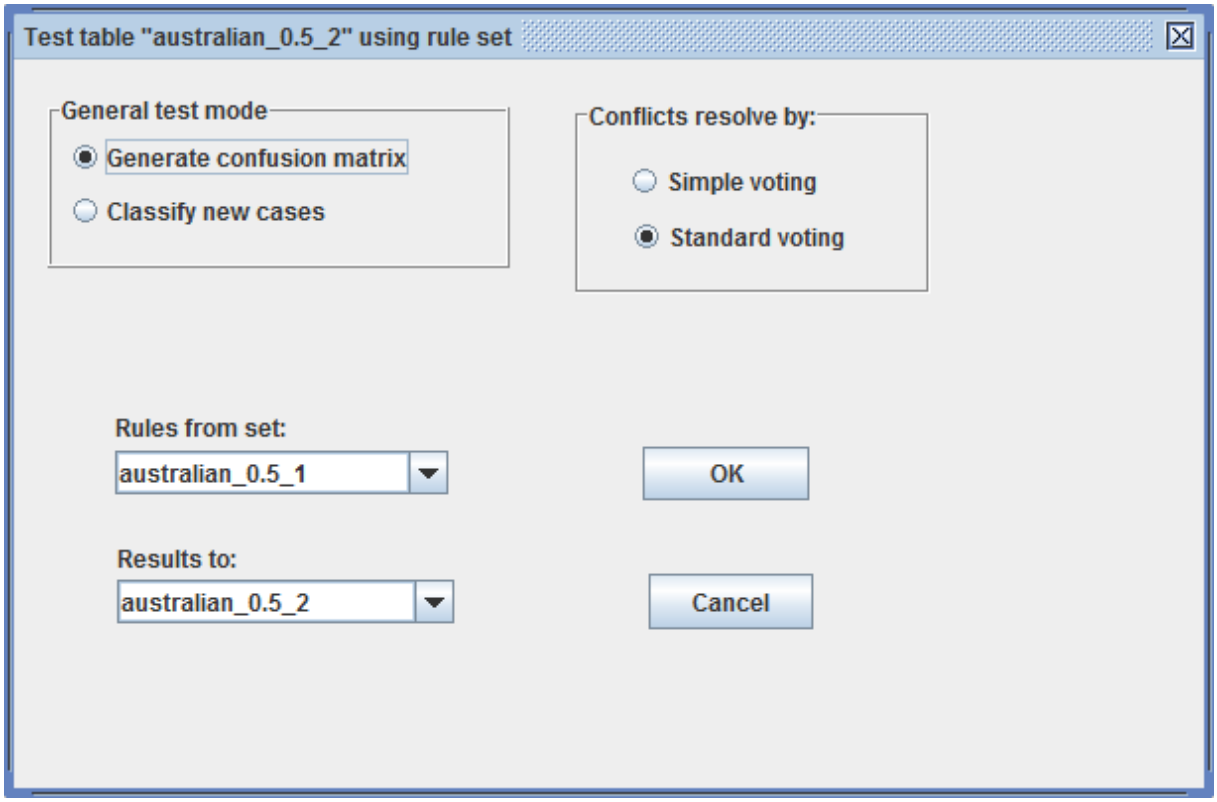


Rys. 22. Wybór parametrów testowania metodą walidacji krzyżowej, szczegółowy opis funkcji w instrukcji programu RSES [5]



Rys. 23. Wynik testowania tabeli *australian_0.5_2* regułami, zapisany w tabeli wyników *test* oraz wynik walidacji krzyżowej *cross-validation*

Klasyfikacja metodą zbioru testowego polega na wybraniu tabeli, która ma zostać sklasyfikowana, kliknięciu prawego przycisku myszy oraz wyborze opcji *Classify->Test Table Using Rule Set*. Pojawi się okno dialogowe, umożliwiające wybór opcji klasyfikacji.



Rys. 24. Wybór opcji klasyfikacji tabeli testowej: *confusion matrix* – wynik w postaci tabeli podsumowującej liczbę poprawnie i niepoprawnie sklasyfikowanych obiektów z tabeli testowej, dla których znane są prawidłowe decyzje, *new cases* – dla tabeli testowej, w której nie są znane prawidłowe decyzje i wszystkie przypadki są nowe i niesklasyfikowane, *rules from set* – źródło reguł, *rules to* – nazwa tablicy z wynikami, *simple voting* – każda pasująca reguła jest jednym równorzędnym głosem, *standard voting* – pasująca reguła jest tym ważniejsza im więcej przypadków ją potwierdza

Podgląd zawartości tabel, zbiorów reguł, reduktów oraz wyników klasyfikacji można uzyskać poprzez podwójne kliknięcie myszką na obiekcie.

Etapy pracy z programem

Poniżej przedstawiono szczegółowo przykładowy proces postępowania w przypadku klasyfikacji nowych danych. Proponowane jest przebadanie skuteczności klasyfikacji w zależności od liczby reduktów, analiza wyników walidacji krzyżowej, wydzielenie z danych tablicy testowej i jej klasyfikacja. Spostrzeżenia i wyniki należy zamiescić w sprawozdaniu projektowym.

a. Utworzyć nowy projekt i nową tabelę, do której załadować należy tabelę z pliku **heart_disease.tab**

b. Dla całej tabeli wyznaczyć reguły decyzyjne przy pomocy algorytmu genetycznego. Liczbę reduktów ustawić na 10.

c. Dla całej tabeli wyznaczyć redukty metodą genetyczną, a następnie na bazie reduktów wygenerować nowy zbiór reguł.

d. Porównać zbiory reguł uzyskane w podpunktach a i b (liczba reguł i przypadków, które pokrywa reguła długości reguł). Z czego wynikają różnice?

e. Powtórzyć punkty b-d dla liczby reduktów 8,6,4,2.

f. Dokonać klasyfikacji zbioru w procedurze walidacji krzyżowej. Użyć algorytmu genetycznego tworzenia reguł. Ilość reduktów ustawić kolejno na 10,8,6,4,2.

g. W sprawozdaniu zamieścić wszystkie 5 macierzy pomyłek. Porównać wyniki uzyskane dla różnych ilości reduktów. Skomentować obserwację.

h. Dokonać podziału zbioru na zbiór testowy i treningowy poprzez jego podział w stosunku 3:7.

i. Dla zbioru treningowego wygenerować reguły. Użyć algorytmu genetycznego. Liczbę reduktów ustawić na 10.

j. Dla zbioru treningowego wygenerować redukty, a następnie na bazie reduktów wygenerować nowy zbiór reguł. Użyć algorytmu genetycznego. Liczbę reduktów ustawić na 10.

k. Dokonać klasyfikacji zbioru testowego dla dwóch uzyskanych zbiorów reguł.

l. Skomentować wyniki i w sprawozdaniu umieścić obie macierze pomyłek.

m. Powtórzyć punkty i-k dla liczby reduktów 8,6,4,2.

n. Porównać wyniki klasyfikacji dla różnych ilości reduktów.

o. Stworzyć nową tabelę i załadować do niej plik z danymi treningowymi satellite_trn.tab

p. Stworzyć nową tabelę i załadować do niej plik z danymi testującymi satellite_tst.tab

q. Dla pliku treningowego stworzyć zbiór reguł przy pomocy algorytmu genetycznego. Liczbę reduktów ustawić na 10.

r. Dla pliku testowego stworzyć zbiór reduktów, a następnie na jego podstawie wygenerować zbiór reguł.

s. Dokonać generalizacji reguł uzyskanych w podpunktach q i r. Zmieniać parametr generalizacji w zakresie 0.9-0.5 z krokiem co 0.1.

t. Dla pliku treningowego stworzyć sieć neuronową z domyślnymi ustawieniami.

u. Dla pliku treningowego stworzyć drzewo decyzyjne z domyślnymi ustawieniami.

v. Dokonać klasyfikacji zbioru testowego wszystkimi uzyskanymi metodami.

w. Porównać i skomentować wyniki.

x. Rozważyć, w jaki sposób generalizacja wpływa na klasyfikację nowych przypadków i dlaczego?

y. Do sprawozdania dołączyć projekt w postaci pliku o rozszerzeniu *.rses.

5 Literatura

[1] Pawlak Z., *Rough sets*. International Journal of Computer and Information Sciences 11, pp. 341–356, 1982

[2] Marek W., Pawlak Z., *Rough Sets and Information Systems*. Fundamenta Informaticae 17, pp. 105–115, 1984

[3] Pawlak Z., *Rough Sets – Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, Dordrecht, 1991

[4] Komorowski J., Polkowski L., Skowron A., *Rough Set: A Tutorial*. Rough fuzzy hybridization: A new trend in decision-making, pp. 3-98, Springer, 1999

[5] RSES 2.1. Rough Set Exploration System. Podręcznik Użytkownika. Publikacja elektroniczna http://logic.mimuw.edu.pl/~rses/RSES_doc.pdf. Warszawa, 2004

[6] Bargiela A., Pedrycz W. (eds.), *Human-Centric Information Processing Through Granular Modelling*, Springer -Verlag, Heidelberg, 2009

[7] Gacek A, Pedrycz W., *A characterization of electrocardiogram signals through optimal allocation of information granularity*, Journal Artificial Intelligence in Medicine, 54, 2, pp. 125-134, 2012

[8] Gomolińska A,, *Zbiory przybliżone w obliczeniach granularnych*, seminarium, Poznań 2011, http://idss.cs.put.poznan.pl/site/fileadmin/seminaria/2011/poznan11.pdf

[9] Lin T.Y., Yao Y.Y., Zadeh L.A. (eds.), *Data Mining, Rough Sets and Granular Computing*, Physica-Verlag, Heidelberg, 2002

[10] Pawlak Z., *Granularity of knowledge, indiscernibility and rough sets*, Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., Anchorage, Alaska, USA, 4-9 May 1998, pp. 106 – 110, vol. 1, 1998

[11] Pedrycz W. (ed.), *Granular Computing: An Emerging Paradigm*, Physica-Verlag, Heidelberg, 2001

[12] Pedrycz W., Gacek A., *Temporal granulation and its application to signal analysis*, Information Sciences, 143, 1-4, 2002, pp. 47-71; Application of information granules to description and processing of temporal, 2002

[13] Pedrycz W., Skowron A., Kreinovich V. (eds), *Handbook of Granular Computing*, Wiley&Sons, Chichester, West Sussex, England, 2008

[14] Polkowski L., Skowron A., *Towards Adaptive Calculus of Granules*, Proc. 1998 IEEE International Conference on Fuzzy Systems, pp. 111-116, 1998

[15] Wang Y., *The Theoretical Framework of Cognitive Informatics*, International Journal of Cognitive Informatics and Natural Intelligence, IGI Publishing, USA, 1(1), Jan., pp. 1-27, 2007

[16] Wang Y., Zadeh L.A., Yao Y.Y., *On the System Algebra*, Foundations for Granular Computing, Int. J. of Software Science and Computational Intelligence, 1(1), pp. 64-86, January-March 2009

[17] Zadeh L.A., *Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic*, Fuzzy Sets and Systems, Volume 90, Issue 2, pp. 111–127, 1997