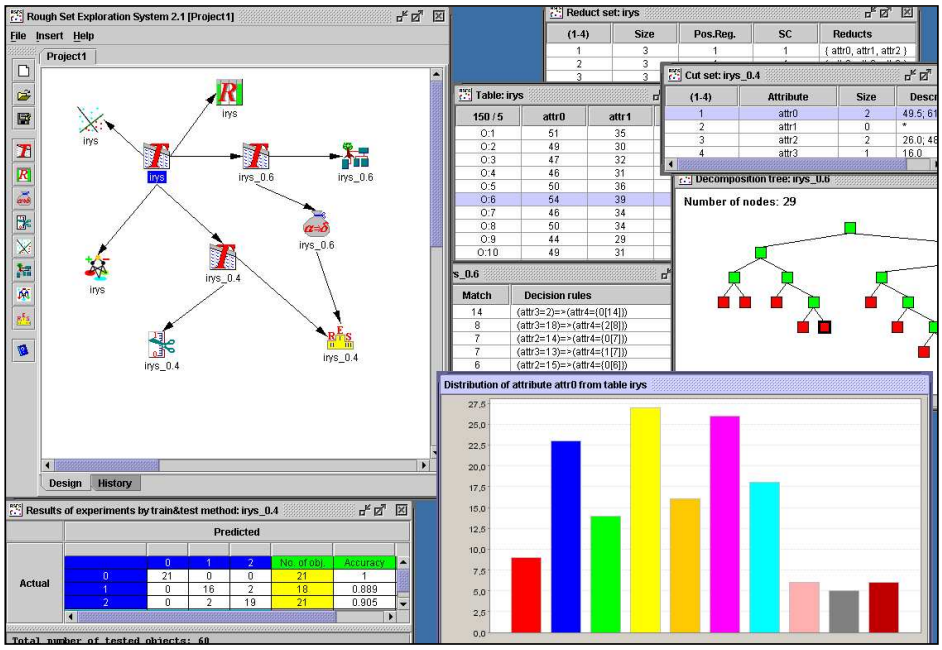
RSES 2.1

Rough Set Exploration System



PODRĘCZNIK UŻYTKOWNIKA

Warszawa, 2004

Spis treści

Rozdział 1

Wprowadzenie do systemu RSES

System RSES 2.1 (ang: *Rough Set Exploration System* 2.1) jest narzędziem komputerowym umożliwiającym analizę danych w postaci tablicowej, z zastosowaniem teorii zbiorów przybliżonych (patrz np. [22]).

System RSES został stworzony przez zespół badawczy kierowany przez prof. dr hab. Andrzeja Skowrona. Obecnie, w skład zespołu rozwijającego system należą: Jan Bazan (Uniwersytet Rzeszowski), Rafał Latkowski (Uniwersytet Warszawski), Michał Mikołajczyk (Uniwersytet Warszawski), Nguyen Hung Son (Uniwersytet Warszawski), Nguyen Sinh Hoa (Polsko-Japońska Wyższa Szkoła Technik Komputerowych), Andrzej Skowron (Uniwersytet Warszawski), Dominik Ślęzak (Uniwersytet w Regina oraz Polsko-Japońska Wyższa Szkoła Technik Komputerowych), Piotr Synak (Polsko-Japońska Wyższa Szkoła Technik Komputerowych), Marcin Szczuka (Uniwersytet Warszawski), Arkadiusz Wojna (Uniwersytet Warszawski), Marcin Wojnarski (Uniwersytet Warszawski), Jakub Wróblewski (Polsko-Japońska Wyższa Szkoła Technik Komputerowych).

System RSES jest dostępny w sieci Internet. Informacje na jego temat, jak i sam system, można pozyskać ze strony:

`http://logic.mimuw.edu.pl/~rses`

1.1 Historia powstania systemu

W 1993 roku, podczas przygotowywania pracy magisterskiej Krzysztofa Przyłuckiego i Joanny Słupek realizowanej pod kierunkiem prof. dr hab. Andrzeja Skowrona w Zakładzie Logiki Instytutu Matematyki Uniwersytetu Warszawskiego, powstał system komputerowy o nazwie *System analizy tablic decyzyjnych*. Był on napisany w języku C++ i pracował pod systemem operacyjnym

Windows 3.11. Oprócz promotora i autorów wspomnianej wyżej pracy magisterskiej, do powstania tego systemu aktywnie przyczynili się także: Jan Bazan, Tadeusz Gąsior i Piotr Synak.

Rok później (1994) powstała pierwsza wersja systemu RSES (wersja 1.0). Napisana w języku C++, była dostępna dla systemu operacyjnego HP-UX (odmiana Unix-a) i mogła być wykorzystywana jedynie na stacjach roboczych Apollo (Hewlett Packard). Nad powstaniem systemu RSES 1.0 pracowali: Jan Bazan, Agnieszka Chądzyńska, Nguyen Hung Son, Nguyen Sinh Hoa, Adam Cykier, Andrzej Skowron, Piotr Synak, Marcin Szczuka i Jakub Wróblewski.

W 1996 roku powstała biblioteka oprogramowania RSES-lib 1.0. Napisana w języku programowania C++, mogła być używana zarówno pod systemem Unix jak i Microsoft Windows. Biblioteka RSES-lib 1.0 została wykorzystana jako jądro obliczeniowe tworzonego w latach 1996–1998, we współpracy pomiędzy w. w. Zakładem Logiki a Politechniką w Trondheim (Norwegia), systemu komputerowego ROSETTA (ang. *Rough Set Toolkit for Analysis of Data*) pracującego pod systemem operacyjnym Microsoft Windows 9x/NT (patrz np. [21]). Nad stworzeniem biblioteki RSES-lib 1.0 pracowali: Jan Bazan, Nguyen Hung Son, Nguyen Sinh Hoa, Adam Cykier, Andrzej Skowron, Piotr Synak, Marcin Szczuka i Jakub Wróblewski.

Kolejna wersja biblioteki RSES-lib, tzn. RSES-lib 2.0 powstała w latach 1998–1999, głównie na potrzeby przeprowadzenia eksperymentów w projekcie badawczym ESPRIT-CRIT2 (finansowany z Unii Europejskiej), którego jeden z podprojektów był realizowany w Zakładzie Logiki Instytutu Matematyki Uniwersytetu Warszawskiego pod kierownictwem prof. dr hab. Andrzeja Skowrona. Nad powstaniem biblioteki oprogramowania RSES-lib 2.0 pracowali: Jan Bazan, Nguyen Hung Son, Nguyen Sinh Hoa, Andrzej Skowron, Piotr Synak, Marcin Szczuka i Jakub Wróblewski.

W roku 2000 powstała kolejna wersja systemu RSES, tym razem mająca graficzny interfejs użytkownika przystosowany do pracy w środowisku Microsoft Windows 9x/NT/2000/Me. Był to system napisany w języku C++ i oparty o bibliotekę oprogramowania RSES-lib 2.0. Ponieważ była to pierwsza wersja systemu RSES z interfejsem dla Microsoft Windows, nazwano ją wersją 1.0. W powstanie tej wersji systemu byli zaangażowani: Jan Bazan, Nguyen Hung Son, Nguyen Sinh Hoa, Andrzej Skowron, Piotr Synak, Marcin Szczuka i Jakub Wróblewski.

Rok 2002 przyniósł kolejną, znacząco różną wersję systemu RSES (wersję 2.0). Tym razem system został napisany w języku Java, przy czym podczas pracy systemu, została wykorzystywana także biblioteka oprogramowania RSES-lib 2.0 (napisana w C++). Do uzyskania wersji binarnych biblioteki RSES-lib 2.0 użyto kompilatora GCC, który jest dostępny dla wszystkich znaczących platform sprzętowo-programowych. Dlatego począwszy od tej wersji

system jest udostępniany zarówno dla systemu operacyjnego Microsoft Windows 9x/NT/2000/Me/XP jak i dla systememu Linux.

Rok później powstała obecna wersja systemu – RSES 2.1. Została ona wyposażona w nowy, poprawiony i bardziej przyjazny interfejs użytkownika. Oprócz tego dodano do systemu wiele ważnych metod obliczeniowych.

1.2 Przeznaczenie i możliwości systemu RSES

Głównym przeznaczeniem systemu jest umożliwienie przeprowadzania eksperymentów na danych tablicowych.

Ogólnie, system RSES oferuje następujące możliwości:

- importowanie danych z plików tekstowych,
- wizualizacja i wstępna obróbka danych pozwalająca między innymi na dyskretyzację i uzupełnienie brakujących wartości w danych,
- konstruowanie oraz stosowanie klasyfikatorów z możliwością oceny ich jakości, zarówno dla małych jak i dużych danych.

System RSES jest narzędziem informatycznym o obsłudze bardzo łatwej do opanowania. Jednak pozwala na przeprowadzenie wielu nietrywialnych eksperymentów obliczeniowych związanych z analizą posiadanych zbiorów danych przy zastosowaniu teorii zbiorów przybliżonych.

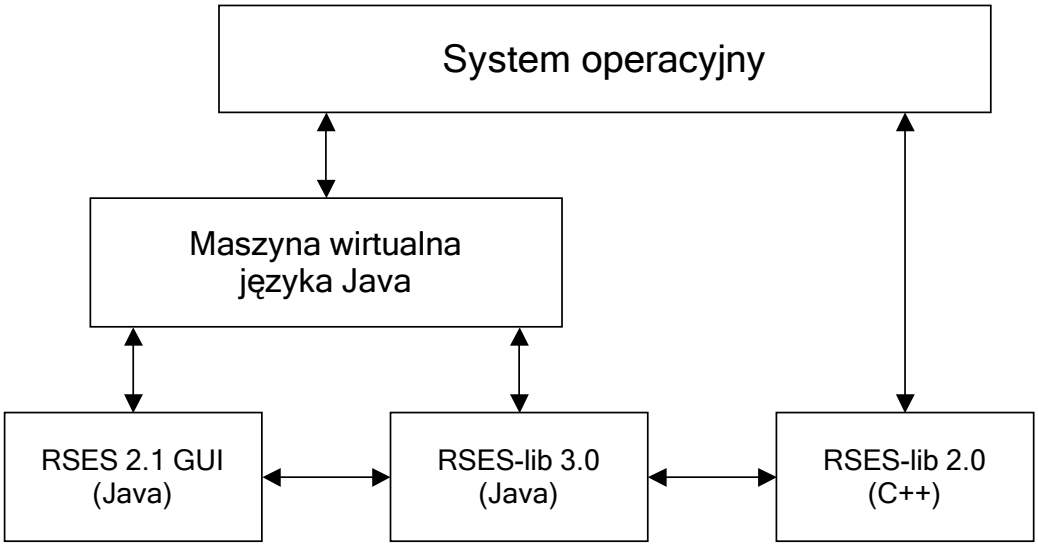
1.3 Informacje techniczne i instalacja systemu

Szacunkowe minimalne wymagania sprzętowe systemu RSES:

- procesor klasy Pentium 200 MHz
- 128 MB pamięci RAM
- maszyna wirtualna języka Java w wersji 1.4 lub wyższej,
- system operacyjny MS Windows 9x/NT/2000/Me/XP lub Linux.

System RSES jest napisany głównie w języku Java. Dlatego do jego uruchomienia potrzebna jest maszyna wirtualna języka Java (JVM). Oprócz tego do poprawnej pracy systemu niezbędna jest odpowiednia wersja biblioteki RSES-lib 2.0, która została napisana w języku C++ (patrz także 1.1).

Biblioteka RSES-lib 2.0 jest rozprowadzana w postaci pliku z kodem wykonywalnym o nazwie `RSES.kernel`, przy czym są dwie wersje tego pliku: jedna dla systemu Microsoft Windows oraz druga dla systemu Linux. Na rysunku przedstawiono omawianą właśnie ogólną architekturę systemu. Warto zauważyć, że ta część RSES-a, która jest napisana w języku Java dzieli się na dwie części. Część pierwsza (tzn. RSES 2.1 GUI) to graficzny interfejs użytkownika całego systemu. Natomiast druga część jest nowym jądrem obliczeniowym systemu i figuruje pod roboczą nazwą RSES-lib 3.0. Biblioteka ta została napisana w prawdzie w języku Java, ale wykorzystuje intensywnie bibliotekę RSES-lib 2.0 napisaną w języku C++.



Rysunek 1.1: Schemat architektury systemu RSES

Na stronie `http://logic.mimuw.edu.pl/~rses` można znaleźć odpowiednie zasoby plikowe umożliwiające instalację RSES-a zarówno dla systemu MS Windows jak i dla systemu Linux. Zasoby te poza samym systemem RSES 2.1 zawierają zestaw przykładowych tabelek z danymi. Dzięki temu użytkownik może przetestować działanie systemu i zapoznać się z nim bez konieczności przygotowywania własnych danych. Mamy nadzieję, że zapoznanie się z systemem ułatwi późniejsze przygotowanie własnych danych i zaprojektowanie interesujących użytkownika eksperymentów.

1.3.1 Instalacja w systemie MS Windows

Użytkownicy systemu MS Windows posiadający już zainstalowaną wirtualną maszynę Javy mogą ściągnąć wersję wykonywalnego pliku instalacyjnego systemu RSES. Jeśli jednak w systemie operacyjnym nie ma jeszcze wirtualnej maszyny Javy lub zainstalowana wersja Javy jest starsza niż wersja 1.4, wówczas należy ściągnąć stosowną wersję SDK lub JRE ze strony firmy SUN i zainstalować ją zanim zainstalujemy system RSES.

Po ściągnięciu instalacji na dysk komputera, należy ją uruchomić. Instalacja odbywa się zgodnie z ogólnie znanymi standardami instalowania programów w systemie MS Windows. Podczas instalacji użytkownik ma możliwość wyboru podstawowych parametrów instalacji. W razie wątpliwości, można pozostawić proponowane przez program instalacyjny wartości domyślne.

1.3.2 Instalacja w systemie Linux

W celu uruchomienia systemu RSES pod systemem Linux, konieczne jest zainstalowanie maszyny wirtualnej języka Java w wersji 1.4 lub wyższej. Następnie do wybranego katalogu rozpakowujemy archiwum TAR ściągnięte ze strony RSES-a. Dla poprawnego działania systemu RSES pod Linuxem, trzeba jeszcze zmodyfikować odpowiednio zmienną systemową `LD_LIBRARY_PATH`. Najwygodniej można zrobić to tak:

```
export LD_LIBRARY_PATH=.:$LD_LIBRARY_PATH
```

Jak wiadomo, dzięki powyższej modyfikacji można uruchamiać za pomocą terminala programy znajdujące się w bieżącym katalogu terminala. Tak właśnie można uruchomić system RSES (patrz początek następnego rozdziału).

Rozdział 2

Zasady użytkowania systemu RSES

Aby uruchomić system RSES 2.1 w środowisku Microsoft Windows należy otworzyć menu Start/Programy/Rses2 i wybrać opcję Rough Set Exploration System ver. 2.1.

Natomiast dla uruchomienia systemu RSES 2.1 w systemie Linux, należy za pomocą terminala, którego bieżącym katalogiem jest katalog, do którego podczas instalacji skopiowaliśmy plik `Rses.jar`, wykonać polecenie:

```
java -jar Rses.jar
```

Aby zakończyć pracę z aplikacją należy wybrać z menu /File/Exit (Alt+fe) a następnie potwierdzić swoją decyzję. Dzięki temu nie jest możliwe przypadkowe zamknięcie aplikacji i związana z tym utrata danych.

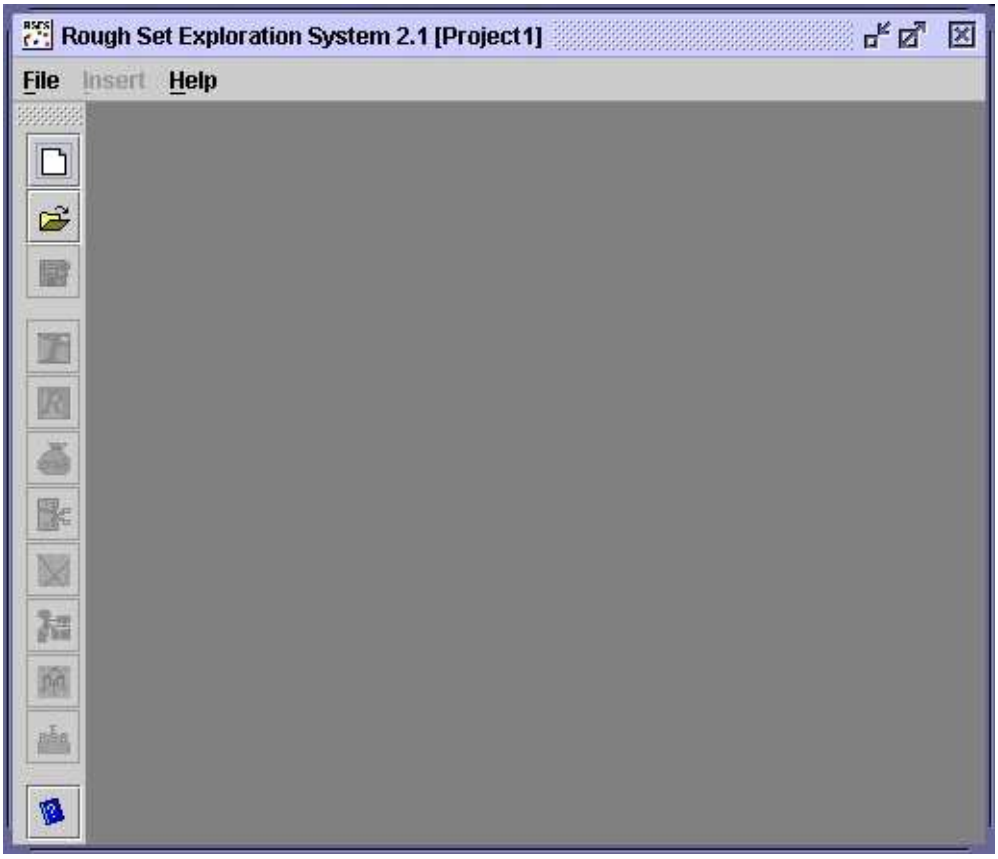
Po uruchomieniu aplikacji pojawi się główne okno systemu RSES 2.1 zawierające menu główne, pasek narzędziowy oraz obszar roboczy przeznaczony na projekty użytkownika.

Menu główne zawiera ogólne opcje i funkcjonalności jakie oferuje system RSES. Część z tych funkcjonalności dostępna jest również za pośrednictwem paska narzędziowego, oraz menu kontekstowego.

2.1 Zarządzanie projektami

Użytkownik ma możliwość równoległego projektowania wielu scenariuszy eksperymentów. Pozwala na to system zarządzania projektami. Jednak jednoczenie może być uruchomiony tylko jeden eksperyment.

W tym miejscu warto zauważyć, że do wykonywania wielu eksperymentów jednocześnie (w środowisku rozproszonym) służy dodatkowy moduł systemu



Rysunek 2.1: Główne okno systemu RSES 2.1 po uruchomieniu

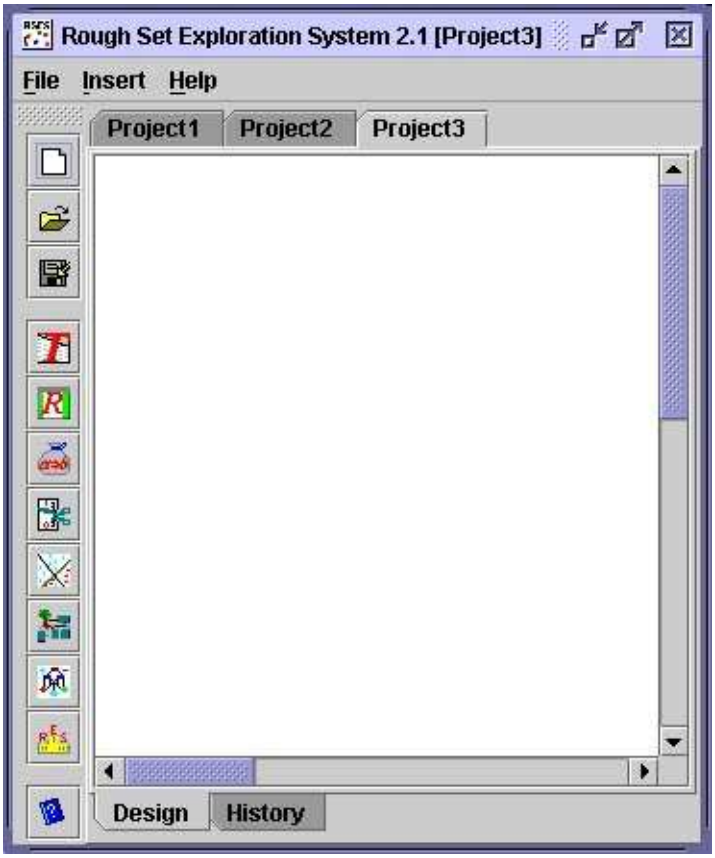
Rysunek 2.2: Menu główne

Rysunek 2.3: Pasek narzędziowy

RSES o nazwie *Dixer*. Dla wygody wykonywania eksperymentów w środowisku rozproszonym, Dixer ma osobny interfejs graficzny pozwalający na łatwe zaprogramowanie wielu równoległych eksperymentów w środowisku rozproszonym. (patrz na stronie `http://logic.mimuw.edu.pl/~rses`).

Nowy projekt w systemie RSES możemy utworzyć na jeden z trzech poniższych sposobów:

- wybierając odpowiednią pozycję menu /File/New project
- za pomocą skrótu klawiszowego: Alt+fn
- klikając na pierwszą ikonkę (licząc od góry) na pasku narzędziowym



Rysunek 2.4: Nowy projekt w systemie RSES

Po utworzeniu projektu możemy umieszczać w jego wnętrzu obiekty takie jak tabele z danymi, zbiory reguł decyzyjnych, zestawienia wyników klasyfikacji itp. Więcej na temat samych obiektów jak i ich obsługi można przeczytać w rozdziale 2.2.

W górnej części projektu mamy zakładki z nazwą aktualnie edytowanego projektu (jaśniejsza) oraz nazwami pozostałych projektów (ciemniejsze). Klikając na nie lewym klawiszem myszki możemy się przełączać pomiędzy nimi co umożliwia nam efektywną pracę nad kilkoma projektami jednocześnie.

Na dole każdego projektu znajdują się zakładki umożliwiające przełączanie się pomiędzy dwoma następującymi widokami projektu :

- widok Design – standardowy widok do pracy nad projektem
- widok History – widok umożliwiający prześledzenie historii projektu, w której rejestrowane są wszystkie ważne operacje wykonane przez użytkownika systemu nad bieżącym projektem.

Zarówno w dolnej części widoku Design projektu jak i po lewej jego stronie znajdują się suwaki umożliwiające poruszanie się po projekcie. Są one niezwykle przydatne w sytuacji gdy widoczny fragment projektu nie zawiera wszystkich interesujących nas szczegółów.

W menu /File znajdziemy opcje umożliwiające zarówno zapisanie jak i odczytanie stanu projektu co pozwoli nam na powrócenie do pracy nad projektem nawet po pewnym czasie. Dokładniejszy opis funkcji dostępnych w menu głównym jest przedstawiony w rozdziale 2.3.

Po kliknięciu lewym klawiszem myszki w obszarze roboczym projektu (na białym polu), uzyskamy dostęp do menu kontekstowego, które pozwoli nam na utworzenie nowego obiektu w projekcie (patrz podrozdział 2.5).

2.2 Obiekty

W projekcie możemy umieszczać rozmaite obiekty, które dzielimy na następujące kategorie:

- Tablice danych
- Zbiory reduktów
- Zbiory reguł
- Zbiory cięć
- Kombinacje liniowe
- Drzewa dekompozycji
- Klasyfikator LTF-C (Local Transfer Function Classifier)

- Wyniki eksperymentów klasyfikacji

Aby utworzyć nowy obiekt możemy wybrać odpowiednią pozycję z menu /Insert (przy pomocy myszki lub skrótu klawiszowego), możemy skorzystać z menu kontekstowego projektu lub kliknąć lewym klawiszem myszki na odpowiednią ikonkę w pasku narzędziowym, przedstawiającą obiekt, który chcemy wstawić. Jeśli tworzymy obiekt poprzez menu kontekstowe projektu[1], to zostaje on wstawiony w miejscu kursora myszy, w przeciwnym przypadku jest on tworzony na środku widocznego obszaru projektu, lub w miejscu przesuniętym o kilka pikseli w losowo wybranym kierunku. Losowe przesunięcie jest wykonywane w tym celu, aby wielokrotnie wstawiane obiekty metodą klikania na ikonie paska narzędziowego lub opcji w menu systemu (czyli bez podania dokładnej lokalizacji wstawianego obiektu za pomocą menu kontekstowego) nie pokrywały się całkowicie, gdyż wtedy niewygodne byłoby ich uchwycenie myszką w celu przesunięcia w inne miejsce projektu.

Jeśli obiekty są ze sobą logicznie powiązane to w projekcie będą reprezentowane jako dwie ikonki połączone strzałką. Na przykład jeśli z pewnej tabeli policzymy reguły to powstanie strzałka prowadząca od tabeli do zbioru reguł.

Jeśli chcemy przesunąć jakiś obiekt w inne miejsce projektu, to musimy w tym celu go zaznaczyć, następnie przycisnąć na nim lewy klawisz myszki i nie puszczając go przesunąć przy pomocy ruchu myszką. Zwolnienie przycisku myszki spowoduje postawienie obiektu w nowym miejscu.

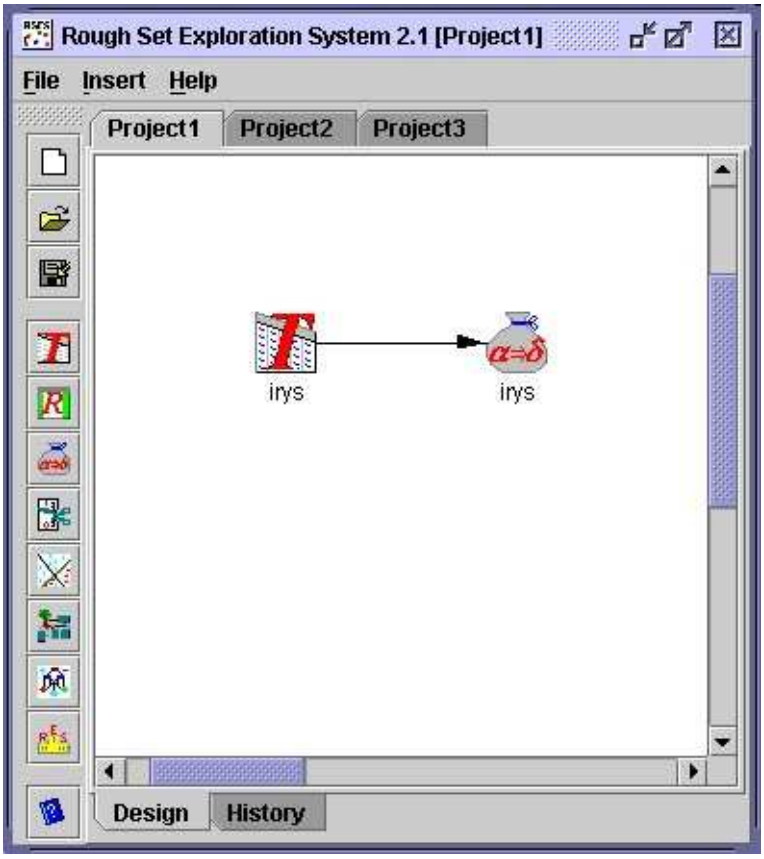
Możemy też przesunąć kilka obiektów jednocześnie. W tym celu musimy zaznaczyć wybrane obiekty, przycisnąć na jednym z wybranych obiektów lewy klawisz myszki i nie puszczając go przesunąć przy pomocy ruchu myszką. Podobnie jak poprzednio, zwolnienie przycisku myszki spowoduje pozostawienie wybranych obiektów we wskazanym miejscu.

Aby zaznaczyć kilka obiektów, należy przycisnąć lewy przycisk myszki i trzymając go przeciągnąć kursor myszki tworząc prostokątny obszar, wewnątrz, którego wszystkie obiekty zostaną zaznaczone. Obszar ten zaznaczamy zaczynając od wyznaczenia jego lewego górnego wierzchołka.

Inną funkcjonalnością ułatwiającą zanaczanie grup obiektów, jest możliwość dodania do grupy lub usunięcia, z zaznaczonej grupy obiektów jednego wskazanego obiektu. W tym celu trzymając przycisk `Ctrl` wskazujemy wybrany obiekt lewym przyciskiem myszki. Operacja ta nie zmienia zaznaczenia pozostałych obiektów.

Każdy obiekt ma swoje specyficzne menu kontekstowe umożliwiające między innymi zmianę nazwy, usunięcie, zapisanie i odczytanie, duplikowanie

[1]Menu kontekstowe projektu będziemy dalej nazywali menu ogólnym



Rysunek 2.5: Dwa obiekty powiązane ze sobą (tabelka i reguły)

obiektu, oglądanie jego zawartości i inne operacje specyficzne dla danego rodzaju obiektu. Do menu kontekstowego obiektu uzyskujemy dostęp poprzez kliknięcie prawym przyciskiem myszy na wybranym obiekcie. Szczegółowy opis wszystkich dostępnych funkcji z poziomu menu kontekstowego został umieszczony w rozdziale 3.

Naciskając prawym przyciskiem myszki na obiekcie należącym do zaznaczonej grupy obiektów uzyskujemy dostęp do kontekstowego menu grupy obiektów. Szczegółowy opis tego menu został przedstawiony w punkcie 2.5.

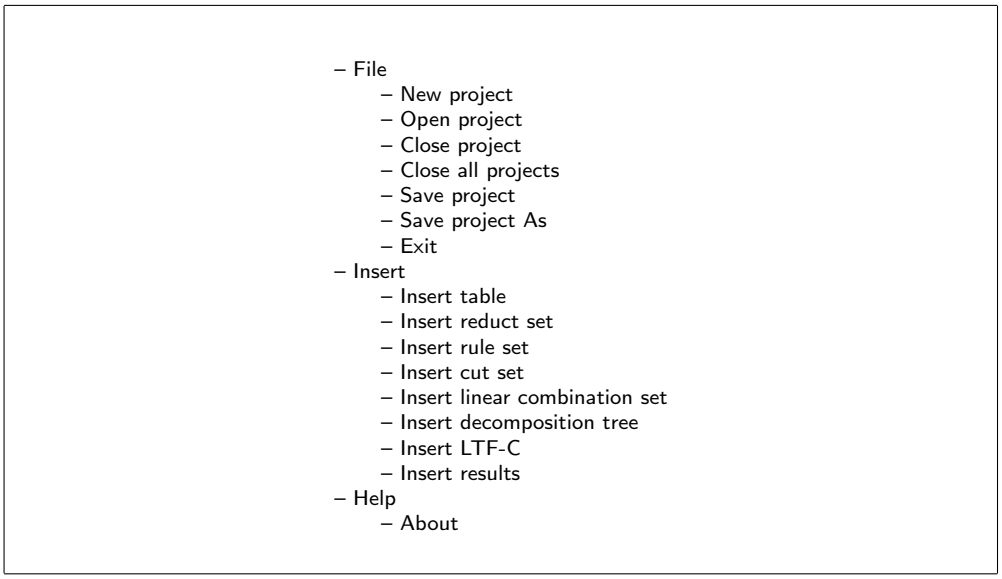
Zarówno obiekty jak i grupy obiektów mogą być usuwane z projektu za pomocą odpowiedniego menu kontekstowego lub poprzez naciśnięcie przycisku `Delete`. Każdorazowe usunięcie obiektu lub obiektów wymaga dodatkowego potwierdzenia w okienku dialogowym.

2.3 Menu główne systemu

Pozycje z menu głównego można wybierać przy pomocy kliknięcia lewym przyciskiem myszki na interesującej nas pozycji lub poprzez skróty klawiszowe. Skróty klawiszowe budowane są zgodnie z zasadą: Alt+<litera>, gdzie <litera> to podkreślony znak w nazwie opcji menu jaką chcemy wybrać. Przytrzymując klawisz Alt i wciskając kolejne znaki możemy uzyskać szybki dostęp do wszystkich funkcji. Na przykład naciskając Alt+fn utworzymy nowy projekt (/File/New project).

Przy niektórych opcjach w menu znajdują się ikonki odpowiadające ikonom mającym takie same znaczenie na pasku narzędziowym. Dotyczy to również ogólnego menu kontekstowego.

Poniżej przedstawiamy krótki opis wszystkich pozycji dostępnych w menu głównym:



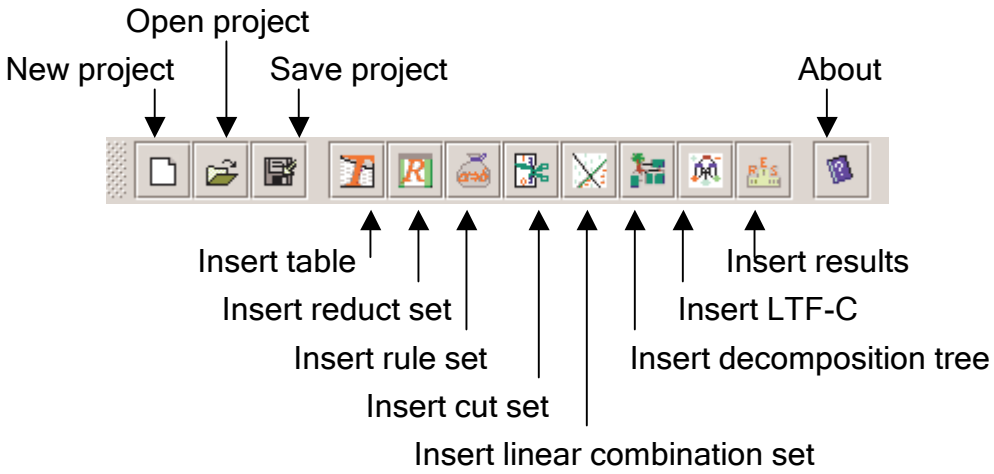
Rysunek 2.6: Schemat menu głównego

- File – zarządzanie projektami
  - New project – utworzenie nowego projektu
  - Open project – odczytanie projektu zapisanego uprzednio na dysku
  - Close project – zamknięcie aktywnego projektu
  - Close all projects – zamknięcie wszystkich otwartych projektów
  - Save project – zapisanie aktywnego projektu na dysku
  - Save project As – zapisanie aktywnego projektu w pliku o podanej nazwie
  - Exit – zamknięcie systemu RSES
- Insert – wstawianie nowych obiektów do aktywnego projektu
  - Insert table – wstawienie tablicy z danymi
  - Insert reduct set – wstawienie zbioru reduktów
  - Insert rule set – wstawienie zbioru reguł
  - Insert cut set – wstawienie zbioru cięć i/lub podziałów zbioru wartości atrybutów
  - Insert linear combination set – wstawienie zbioru kombinacji liniowych
  - Insert decomposition tree – wstawienie drzewa dekompozycji
  - Insert LTF-C – wstawienie klasyfikatora LTF-C (Local Transfer Function Classifier)
  - Insert results – wstawienie obiektu do przetwarzania wyników klasyfikacji
- Help – pomoc i informacje
  - About – podstawowe informacje o systemie RSES

Uwaga! Wszystkie obiekty wstawiane za pomocą menu głównego pojawiają się na środku widocznego fragmentu projektu lub w miejscu przesuniętym o kilka pikseli w losowo wybranym kierunku. Losowe przesunięcie jest wykonywane w tym celu, aby wielokrotnie wstawiane obiekty za pomocą opcji z menu (czyli bez podania dokładnej lokalizacji wstawianego obiektu) nie pokrywały się całkowicie, gdyż wtedy niewygodne byłoby ich uchwycenie myszką w celu przesunięcia w inne miejsce projektu.

2.4 Pasek narzędziowy

Pasek narzędziowy zawiera wybrane polecenia z menu głównego i menu kontekstowego projektu. Dzięki temu zapewnia użytkownikowi szybki dostęp do najważniejszych funkcji. Opis tych funkcji i ich znaczenie przedstawiamy poniżej.



Rysunek 2.7: Pasek narzędziowy, oraz odpowiadające ikonom polecenia z menu głównego

- New project – utworzenie nowego projektu
- Open project – odczytanie projektu zapisanego uprzednio na dysku
- Save project – zapisanie aktywnego projektu na dysku
- Insert table – wstawienie tablicy z danymi
- Insert reduct set – wstawienie zbioru reduktów
- Insert rule set – wstawienie zbioru reguł
- Insert cut set – wstawienie zbioru cięć i/lub podziałów zbioru wartości atrybutów
- Insert linear combination set – wstawienie zbioru kombinacji liniowych
- Insert decomposition tree – wstawienie drzewa dekompozycji

- Insert LTF-C – wstawienie klasyfikatora LTF-C (*Local Transfer Function Classifier*)
- Insert results – wstawienie obiektu do przetwarzania wyników
- About – podstawowe informacje o systemie RSES

Uwaga! Wszystkie obiekty wstawiane za pomocą paska narzędziowego pojawiają się na środku widocznego fragmentu projektu lub w miejscu przesuniętym o kilka pikseli w losowo wybranym kierunku. Losowe przesunięcie jest wykonywane w tym celu, aby wielokrotnie wstawiane obiekty za pomocą paska narzędziowego (czyli bez podania dokładnej lokalizacji wstawianego obiektu) nie pokrywały się całkowicie, gdyż wtedy niewygodne byłoby ich uchwycenie myszką w celu przesunięcia w inne miejsce projektu.

2.5 Menu kontekstowe

Menu kontekstowe jest związane z konkretnym obiektem. Możemy je otworzyć klikając na obiekt prawym przyciskiem myszy.

Dla różnego typu obiektów różna jest budowa menu kontekstowego. W dalszej części podręcznika przedstawiamy przegląd dostępnych opcji we wszystkich menu kontekstowych.

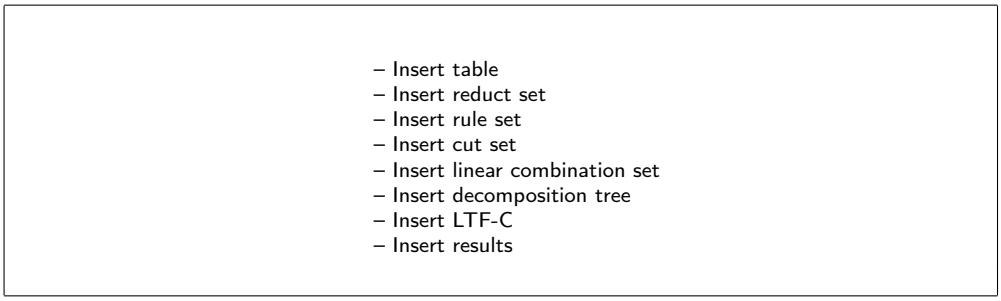
Poza menu kontekstowymi związanymi z obiektami dostępne jest również menu ogólne związane z samym projektem oraz menu kontekstowe dla grup obiektów.

Menu ogólne to menu kontekstowe dla projektu. Możemy je wywołać poprzez kliknięcie prawym przyciskiem myszki na pustym (białym) obszarze projektu.

Polecenia w menu ogólnym są również dostępne poprzez menu główne i ikony na pasku narzędziowym.

Wykaz poleceń:

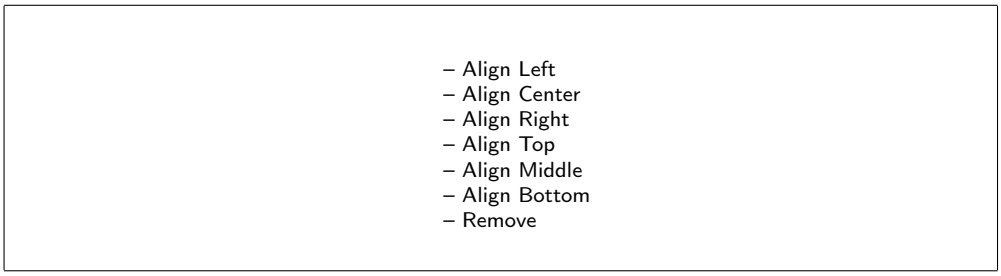
- Insert table – wstawienie tablicy z danymi
- Insert reduct set – wstawienie zbioru reduktów
- Insert rule set – wstawienie zbioru reguł
- Insert cut set – wstawienie zbioru cięć
- Insert linear combination set – wstawienie zbioru kombinacji liniowych
- Insert decomposition tree – wstawienie drzewa dekompozycji



Rysunek 2.8: Schemat menu ogólnego (menu kontekstowego dla projektu)

- Insert LTF-C – wstawienie klasyfikatora LTF (*Local Transfer Function Classifier*)
- Insert results – wstawienie obiektu przechowującego wyniki klasyfikacji

Menu kontekstowe dla grup obiektów jest dostępne po kliknięciu prawym przyciskiem myszy na jednym z wybranych obiektów.



Rysunek 2.9: Schemat menu kontekstowego dla grupy obiektów

Wykaz poleceń:

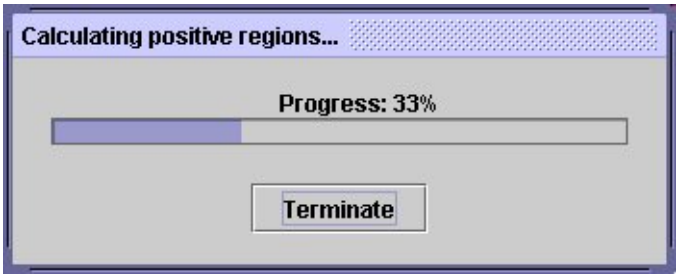
- Align Left – wyrównanie zaznaczonych obiektów do obiektu najbardziej wysuniętego na lewo
- Align Center – wyśrodkowanie w poziomie zaznaczonych obiektów
- Align Right – wyrównanie zaznaczonych obiektów do obiektu najbardziej wysuniętego na prawo
- Align Top – wyrównanie zaznaczonych obiektów do obiektu znajdującego się najwyżej

- Align Middle – wyśrodkowanie w pionie zaznaczonych obiektów
- Align Bottom – wyrównanie zaznaczonych obiektów do obiektu znajdującego się najniżej
- Remove – usunięcie zaznaczonych obiktów

Polecenia Allign Left, Allign Center, Allign Right, Allign Top, Allign Middle i Allign Bottom ustawiają wszystkie zaznaczone obiekty w jednej linni. Może to spowodować wzajemne nachodzenie obiektów na siebie.

2.6 Raportowanie statusu i postępu obliczeń

Użytkownik w każdej chwili może przełączyć się na widok historii projektu (przy pomocy zakładki na dole okna projektu). Widok ten zawiera informacje o obiektach, o wykonanych operacjach na tych obiektach, a także informacje o ewentualnych błędach lub przerwaniu obliczeń na życzenie użytkownika.



Rysunek 2.10: Wizualizacja postępów obliczeń

Użytkownik ma możliwość przerwania obliczeń. Jest to możliwe dzięki przyciskowi Terminate który towarzyszy wizualizacji postępów obliczeń. Po kliknięciu na przycisku Terminate obliczenia zostają natychmiast przerwane. Po przerwaniu obliczeń pojawia się stosowny komunikat informacyjny.

Rozdział 3

Przegląd obiektów w projektach

W niniejszym rozdziale został przedstawiony przegląd obiektów dostępnych w projektach systemu RSES oraz dostępnych na tych obiektach operacji. Szczegóły dotyczące algorytmów i znaczenia związanych z nimi opcji zostały przedstawione w rozdziale 4.

3.1 Tablice

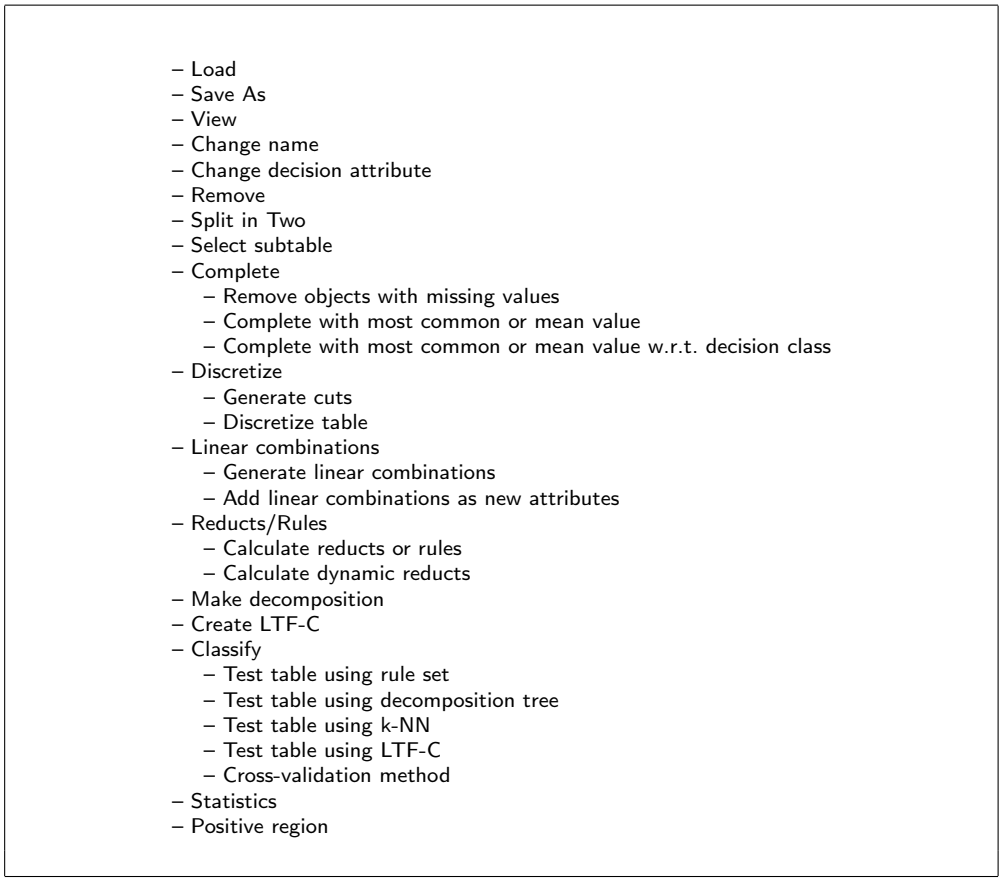
Tablice są najważniejszymi obiektami projektu. Reprezentują one tablice z danymi i umożliwiają ich podgląd, edycję, a także wykonywanie z ich pomocą obliczeń.

Rysunek 3.1: Ikonka reprezentująca tablicę decyzyjną

Dwukrotne kliknięcie lewym przyciskiem myszki na ikonce reprezentującej tablicę jest równoważne poleceniu View z menu kontekstowego i umożliwia podgląd danych.

Wykaz poleceń menu kontekstowego dla tablicy:

- Load – wczytanie danych w formacie RSES, RSES 1.0, Rosetta, Weka, XML
- Save As – zapisanie tablicy z danymi w podanym przez użytkownika pliku w formacie RSES lub XML



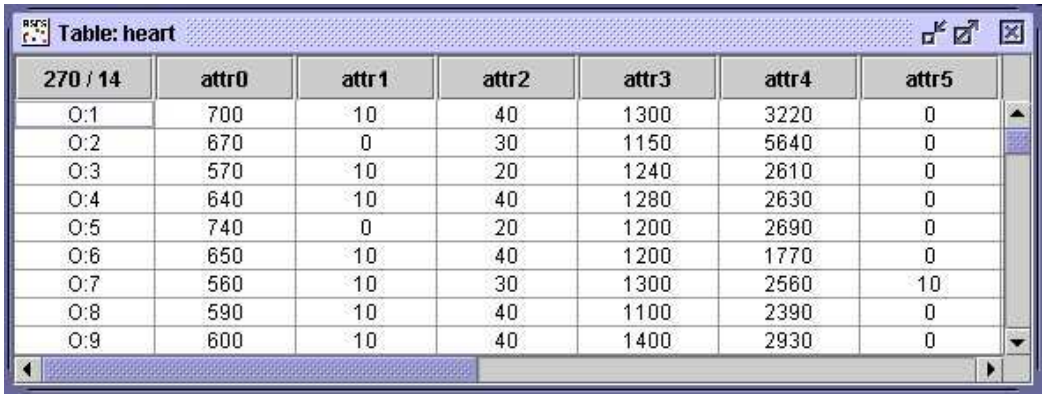
Rysunek 3.2: Schemat menu kontekstowego dla tablicy

- View – podgląd zawartości tablicy (patrz rysunek 3.3), użytkownik może dowolnie przewijać okno, a także zmieniać jego rozmiar, zależnie od potrzeb

- Change name – zmiana nazwy tablicy (patrz rysunek 3.4), nazwa ta jest zapisywana wraz z tablicą do pliku jako nazwa tablicy (a nie nazwa pliku z tablicą). Nazwę tablicy można również zmienić poprzez dwukrotne kliknięcie na nazwę znajdującą się pod ikonką tablicy.

- Change decision attribute – wskazanie atrybutu, który będzie nowym atrybutem decyzyjnym. Wskazany atrybut staje się ostatnim atrybutem (zostaje przeniesiony na koniec tablicy).

- Remove – usunięcie tablicy (konieczne jest potwierdzenie)

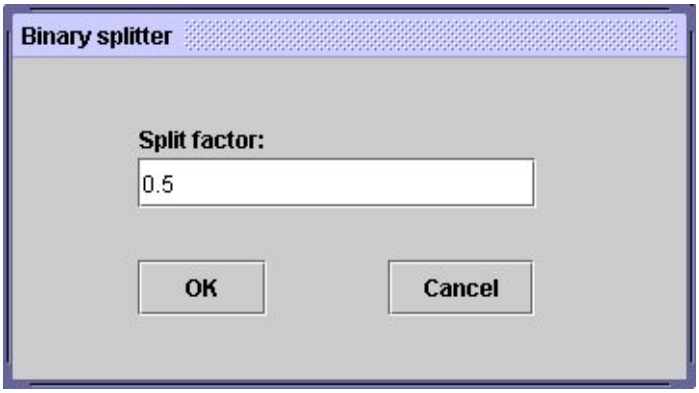


Rysunek 3.3: Podgląd zawartości tablicy danych

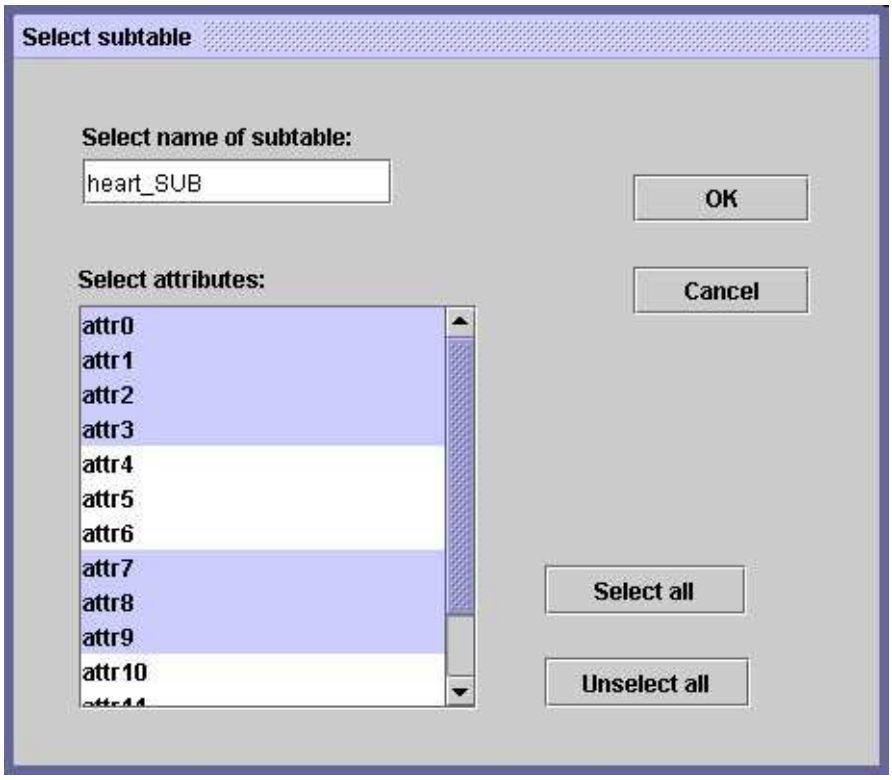
Rysunek 3.4: Zmiana nazwy tablicy

- Split in Two – losowy podział tablicy na dwie części (patrz rysunek 3.5), tak że ich suma daje tabelę wyjściową. Współczynnik Split factor podawany przez użytkownika z przedziału od 0.0 do 1.0, determinuje rozmiar pierwszej tabeli, druga jest jej dopełnieniem. Na przykład wartość współczynnika Split factor równa 0.45 oznacza, że rozmiar pierwszej tabeli po podziale będzie wynosił 45 procent obiektów tabeli wejściowej, natomiast do drugiej tabeli zostanie wstawione pozostałe 55 procent obiektów. Obiekty po podziale otrzymują nazwy związane z nazwą tablicy wejściowej oraz wartością współczynnika Split factor.

- Select subtable – wybór całej lub części tablicy (patrz rysunek 3.6) poprzez wskazanie podzbioru interesujących nas atrybutów. W przypadku wybrania wszystkich atrybutów system wykona kopię tablicy.



Rysunek 3.5: Podział tablicy na dwie części



Rysunek 3.6: Wybór podtablicy

- Complete – uzupełnianie brakujących wartości (patrz podrozdział 4.1). Użytkownik podaje nazwę nowej tablicy, która zostanie utworzona na podstawie bieżącej tablicy i która nie będzie już zawierała brakujących

wartości.

- Discretize – dyskretyzacja danych [5] i generowanie cięć dla atrybutów (patrz podrozdział 4.2)
- Linear combinations – generowanie kombinacji liniowych atrybutów, stanowiących nowe atrybuty w tablicach (patrz podrozdział 4.3)
- Reducts/Rules – opcja ta pozwala na wygenerowanie reduktów, reduktów dynamicznych lub reguł decyzyjnych (patrz podrozdział 4.4)
- Make decomposition – opcja ta pozwala na wykonanie dekompozycji tablicy decyzyjnej (patrz podrozdział 4.5)
- Create LTF-C – stworzenie klasyfikatora LTF-C (*Local Transfer Function Classifier*) bazującego na sieciach neuronowych (patrz podrozdział 4.7)
- Classify – uruchomienie procesu klasyfikacji przy pomocy wskazanego klasyfikatora. Klasyfikator ten musi już istnieć i być wyuczony, czyli musi być uformowany i gotowy do użycia. Niezależnie od wyboru rodzaju klasyfikatora, użytkownik będzie miał możliwość wyboru jednego z dwóch typów klasyfikacji:
  - Generate confusion matrix – wyliczenie macierzy błędu klasyfikacji
  - Classify new cases – klasyfikacja nowych przypadków i dopisanie do testowanych danych kolumny z wyliczoną decyzją

  Wyboru tego można dokonać w sekcji General test mode jaką posiada okienko dialogowe każdego z klasyfikatorów.

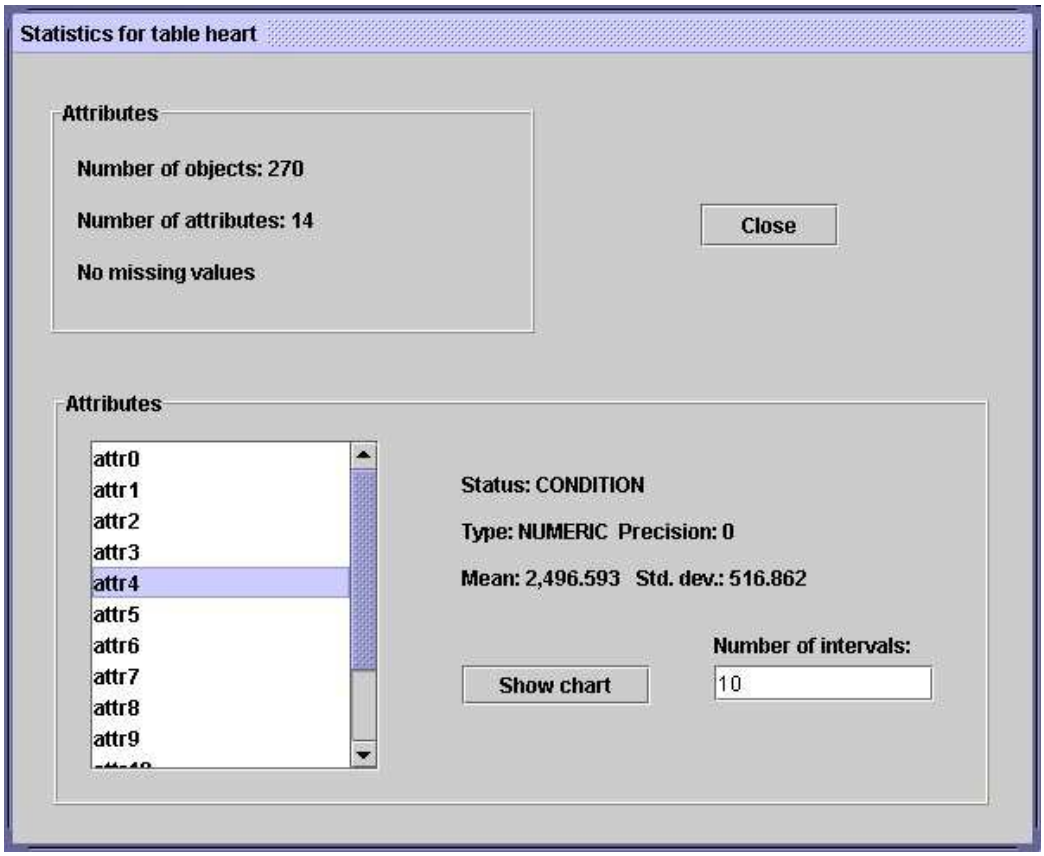
  Użytkownik ma do wyboru następujące metody klasyfikacji:

  - Test table using rule set – klasyfikacja przy pomocy zbioru reguł (patrz podrozdział 4.4)
  - Test table using decomposition tree – klasyfikacja przy pomocy drzewa dekompozycji (patrz podrozdział 4.5)
  - Test table using k-NN – klasyfikacja wybranej tabeli przy pomocy algorytmu k najbliższych sąsiadów (patrz podrozdział 4.6)
  - Test table using LTF-C – klasyfikacja przy pomocy klasyfikatora LTF-C (patrz podrozdział 4.7)

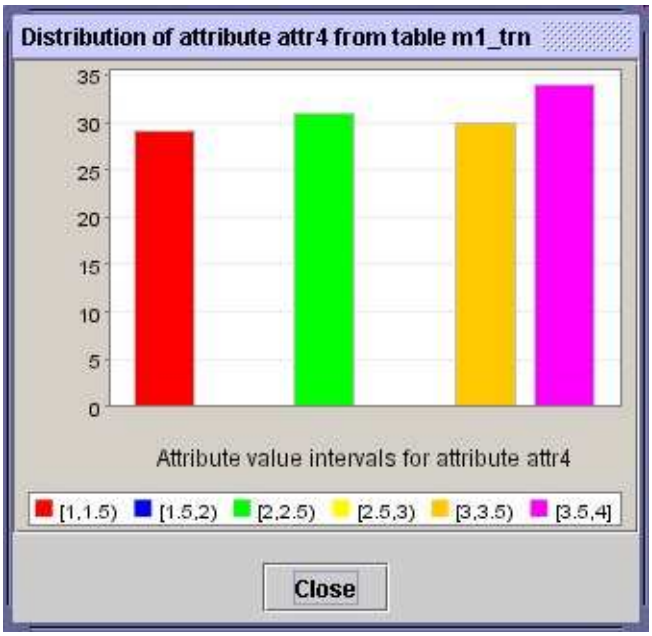
– Cross-validation method – klasyfikacja przy pomocy metody cross-validation (patrz podrozdział 4.8)

- Statistics – wypisanie podstawowych informacji o tablicy i atrybutach (patrz rysunek 3.7).

  Użycie przycisku Show chart umożliwia obejrzenie rozkładu wartości dla wybranego atrybutu (patrz rysunek 3.8). Dla przypadku atrybutów symbolicznych, zliczane są wystąpienia poszczególnych wartości i prezentowany jest histogram dla tych wartości. Natomiast dla atrybutów o wartościach numerycznych (ciągłych), dziedzina atrybutu jest dzielona na wskazaną liczbę równych przedziałów (opcja Number of intervals), zaś prezentowany wykres jest histogramem dla tych właśnie przedziałów wartości.



Rysunek 3.7: Informacje o tablicy z danymi



Rysunek 3.8: Wykres obrazujący rozkład wartości atrybutu

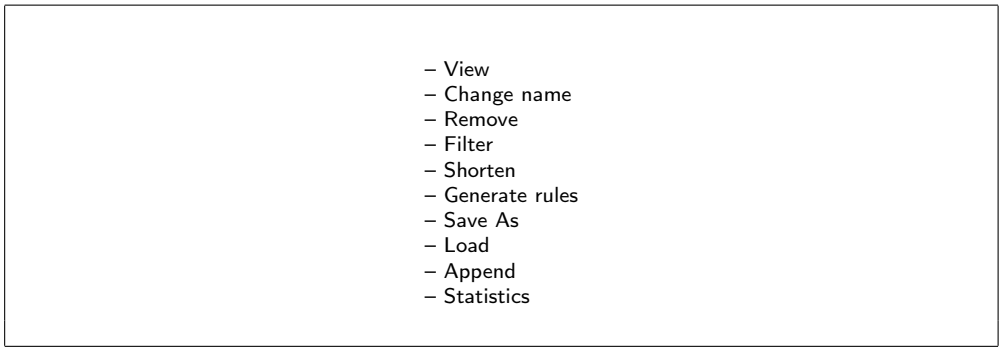
- Positive region – zliczanie obszaru pozytywnego dla danej tablicy (*Uwaga: ostatnia kolumna w tablicy jest traktowana jako kolumna decyzyjna. Kolejności kolumn nie można zmienić w widoku tablicy. Możliwe jest jednak przeniesienie wybranej kolumny na koniec tablicy poprzez wybranie opcji* Change decision attribute *z menu kontekstowego tablicy.*)

3.2 Zbiory reduktów

Reduktem danego systemu informacyjnego nazywamy taki zbiór atrybutów, który pozwala na rozróżnienie par obiektów w systemie informacyjnych, a jednocześnie żaden jego podzbiór właściwy nie ma tej własności.

Rysunek 3.9: Ikonka reprezentująca zbiór reduktów

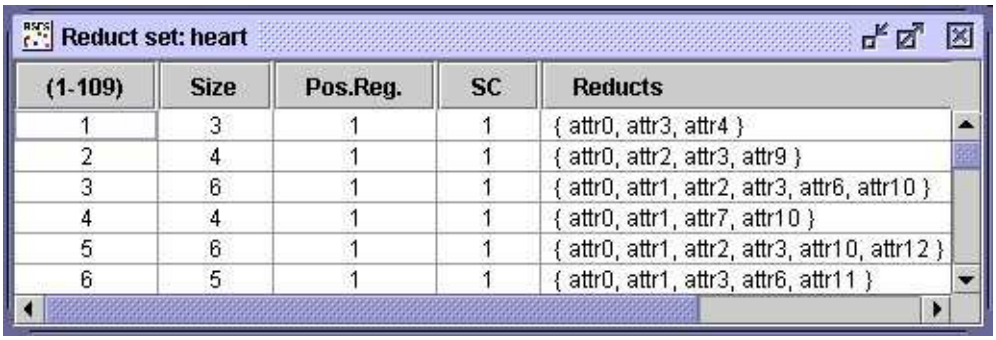
Dwukrotne kliknięcie lewym przyciskiem myszki na ikonce jest równoważne poleceniu View z menu kontekstowego i umożliwia podgląd zawartości obiektu.



Rysunek 3.10: Schemat menu kontekstowego dla zbioru reduktów

Wykaz poleceń menu kontekstowego dla zbioru reduktów:

- View – podgląd zawartości obiektu (patrz rysunek 3.11), użytkownik może dowolnie przewijać okno, a także zmieniać jego rozmiar, zależnie od potrzeb



Rysunek 3.11: Podgląd zawartości zbioru reduktów

Okno podglądu reduktów składa się z pięciu kolumn. Pierwsza z nich to kolumna z numerami porządkowymi, a pozostałe to odpowiednio:

  - Size – rozmiar reduktu
  - Pos.Reg. – obszar pozytywny przy obcięciu tablicy do tego reduktu
  - SC – wartość współczynnika stabilności (*ang. stability coefficient*) reduktu, który określa stabilność reduktu w sensie pojęcia *reduktu dynamicznego* (patrz [2]).
  - Reducts – redukt wypisany w formie listy atrybutów wchodzących w jego skład

- Change name – zmiana nazwy obiektu (patrz rysunek 3.4), nazwa ta jest zapisywana wraz z obiektem do pliku jako nazwa obiektu (a nie nazwa pliku z obiektem). Nazwę obiektu można również zmienić poprzez dwukrotne kliknięcie na nazwę znajdującą się pod ikonką.
- Remove – usunięcie zbioru reduktów (konieczne jest potwierdzenie)
- Filter – filtrowanie zbioru reduktów, użytkownik pozbyć się reduktów o określonym współczynniku stabilności. Przed użyciem tej opcji bardzo pomocne jest obejrzenie statystyk dla filtrowanego zbioru reduktów.
- Shorten – skracanie reduktów, użytkownik podaje współczynnik określający „agresywność" skracania reduktów, czyli współczynnik o nazwie Shortening ratio (1.0 oznacza brak skracania, zaś wartości bliższe 0.0 oznaczają bardziej agresywne skracanie, gdzie współczynnik dokładności skracanych reduktów może spaść właśnie do poziomu podanego współczynnika)
- Generate rules – generowanie reguł decyzyjnych w oparciu o zbiór reduktów i podaną tablice z danymi (patrz także podrozdział 4.4)
- Save As – zapisanie zbioru reduktów do pliku
- Load – wczytanie zbioru reduktów z podanego pliku
- Append – doklejenie z podanego pliku reduktów do istniejącego już zbioru reduktów. Powtarzające się redukty są pomijane.
- Statistics – wypisanie podstawowych statystyk dotyczących zbioru reduktów (patrz rysunek 3.12).

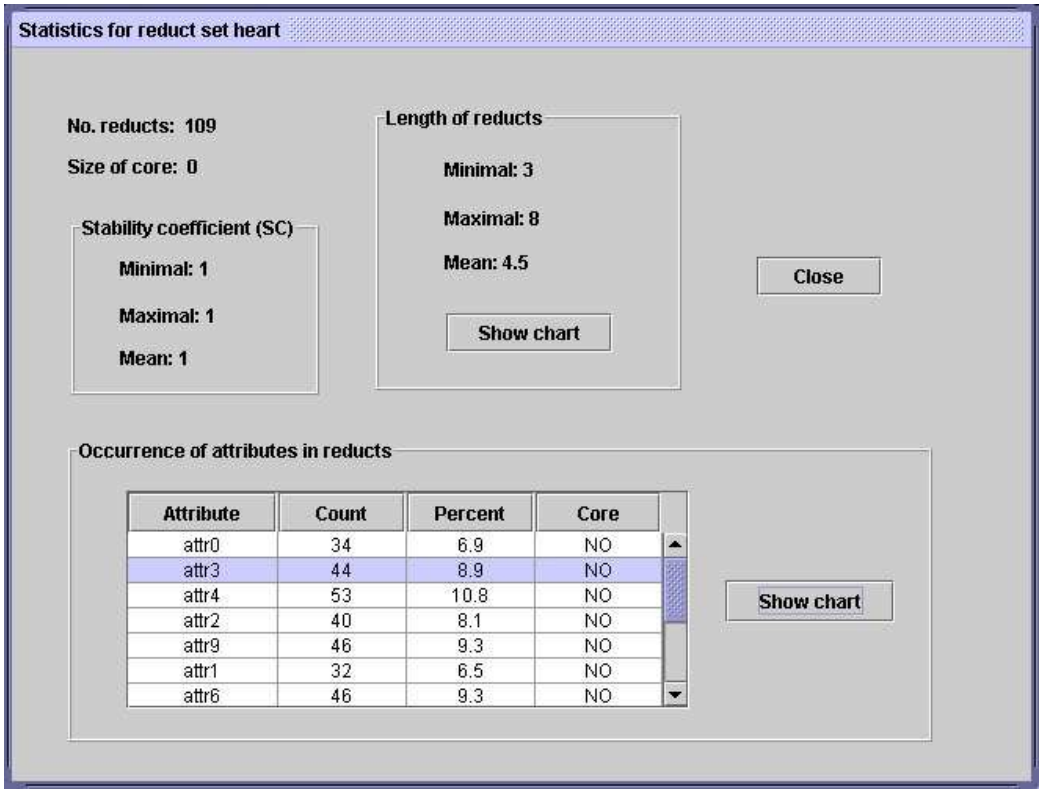
  Użytkownik ma również możliwość obejrzenia wykresów obrazujących rozkład długości reduktów oraz udziału poszczególnych atrybutów w budowie reduktów.

3.3 Zbiory reguł

Reguły decyzyjne umożliwiają klasyfikowanie obiektów, czyli przypisywanie obiektom pewnych decyzji. Gdy mamy zbiór reguł klasyfikujących obiekty do różnych klas decyzyjnych, możemy zorganizować głosowanie miedzy tymi regułami – w ten sposób powstanie prosty system regułowy.

Dwukrotne kliknięcie lewym przyciskiem myszki na ikonce jest równoważne poleceniu View z menu kontekstowego i umożliwia podgląd obiektu.

Wykaz poleceń menu kontekstowego dla zbioru reguł:



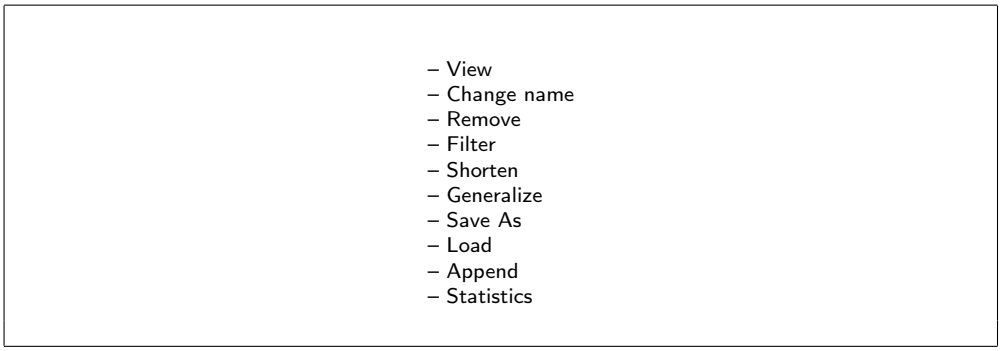
Rysunek 3.12: Informacje o zbiorze reduktów

Rysunek 3.13: Ikonka reprezentująca zbiór reguł

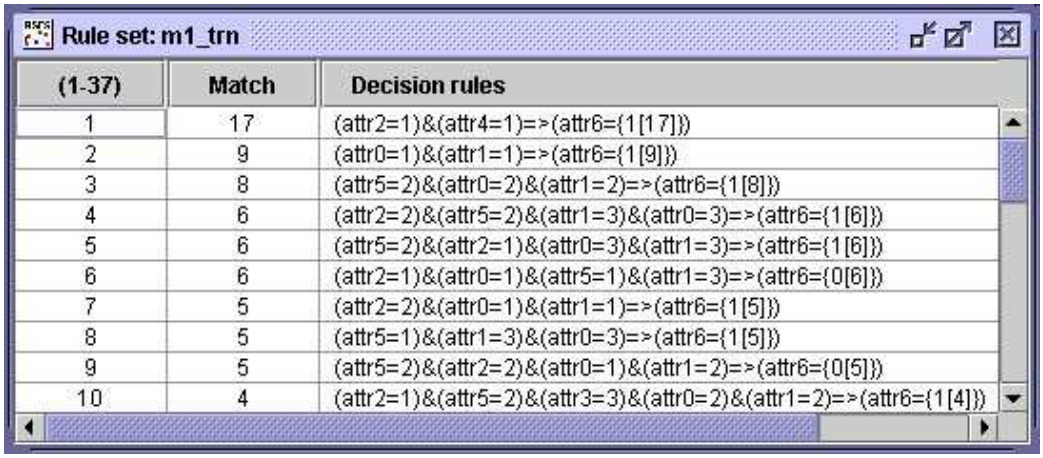
- View – podgląd zawartości obiektu (patrz rysunek 3.15), użytkownik może dowolnie przewijać okno, a także zmieniać jego rozmiar, zależnie od potrzeb

  Podgląd reguł również składa się z trzech kolumn. Pierwsza kolumna pełni funkcję informacyjną i porządkową, znaczenie kolejnych kolumn to:

  - Match – liczba obiektów ze zbioru treningowego pasujących do poprzednika reguły (wsparcie reguły)
  - Decision rules – reguła wypisana w postaci formuły logicznej



Rysunek 3.14: Schemat menu kontekstowego dla zbioru reguł



Rysunek 3.15: Podgląd zawartości zbioru reguł

- Change name – zmiana nazwy obiektu (patrz rysunek 3.4), nazwa ta jest zapisywana wraz z obiektem do pliku jako nazwa obiektu (a nie nazwa pliku z obiektem). Nazwę obiektu można również zmienić poprzez dwukrotne kliknięcie na nazwę znajdującą się pod ikonką.

- Remove – usunięcie zbioru reguł (konieczne jest potwierdzenie)

- Filter – filtrowanie zbioru reguł, użytkownik może usunąć ze zbioru reguł reguły dla wybranej klasy decyzyjnej lub też pozbyć się reguł o określonym wsparciu. Przed użyciem tej opcji bardzo pomocne jest obejrzenie statystyk dla filtrowanego zbioru reguł.

- Shorten – skracanie reguł, użytkownik podaje współczynnik określający

„agresywność" skracania reguł, czyli współczynnik o nazwie Shortening ratio (1.0 oznacza brak skracania, zaś wartości bliższe 0.0 oznaczają bardziej agresywne skracanie reguł, gdzie współczynnik dokładności skracanych reguł może spaść właśnie do poziomu podanego współczynnika – patrz [2], [5])

- Generalize – uogólnianie reguł , użytkownik podaje współczynnik określający „agresywność" generalizacji reguł (1.0 oznacza generalizację bez zmniejszenia dokładności otrzymanych reguł, zaś wartości bliższe 0.0 oznaczają bardziej agresywną generalizację reguł, przy której współczynnik dokładności powstałych reguł może spaść do poziomu wyspecyfikowanego współczynnika)
- Save As – zapisanie zbioru reguł do pliku
- Load – wczytanie zbioru reguł z podanego pliku
- Append – doklejenie z podanego pliku reguł do istniejącego już zbioru reguł. Powtarzające się reguły są pomijane.
- Statistics – wypisanie podstawowych statystyk dotyczących zbioru reguł (patrz rysunek 3.16).

  Użytkownik ma również możliwość obejrzenia wykresów obrazujących rozkład długości reguł oraz rozkład liczby reguł w poszczególnych klasach decyzyjnych.

3.4 Zbiory cięć

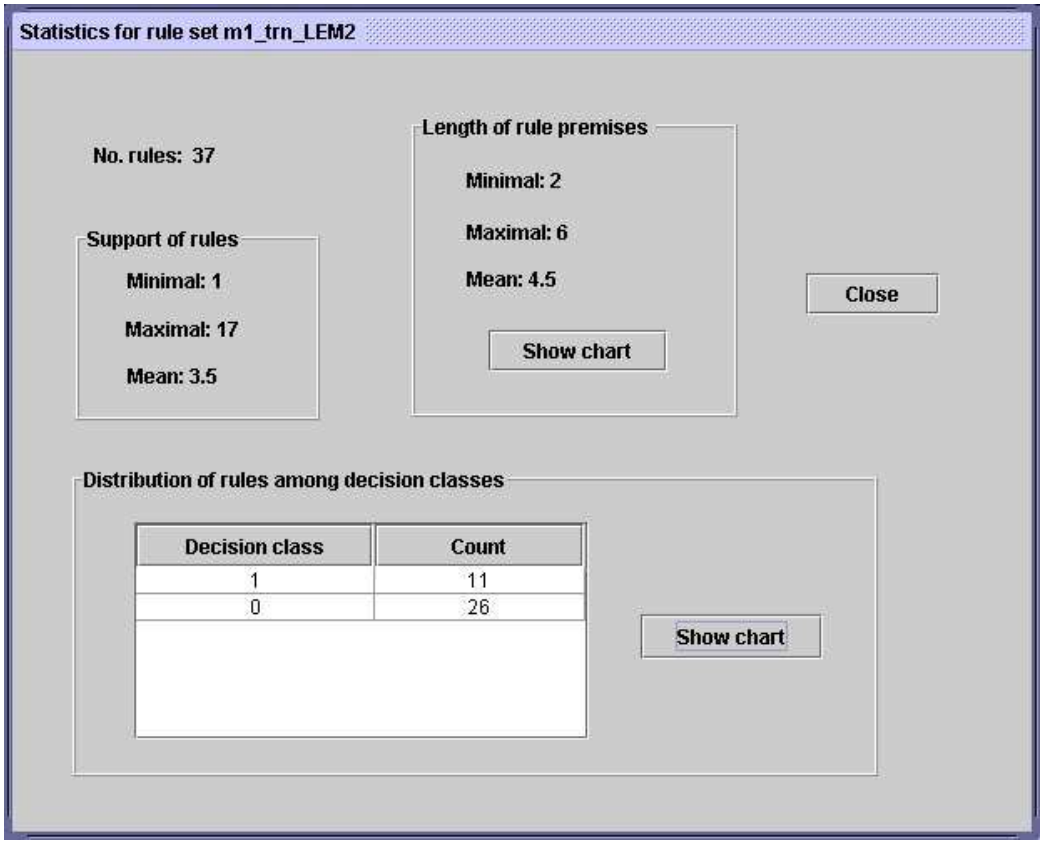
Poprzez cięcia rozumiemy podział wartości atrybutu, przy czym może to być podział wartości atrybutu numerycznego na przedziały liczbowe za pomocą pojedynczych wartości tego atrybutu, albo podział zbioru wartości atrybutu symbolicznego na rozłączne podzbiory. Cięcia mogą posłużyć na przykład do dyskretyzacji.

Dwukrotne kliknięcie lewym przyciskiem myszki na ikonce jest równoważne poleceniu View z menu kontekstowego i umożliwia podgląd obiektu.

Wykaz poleceń menu kontekstowego dla zbioru cięć:

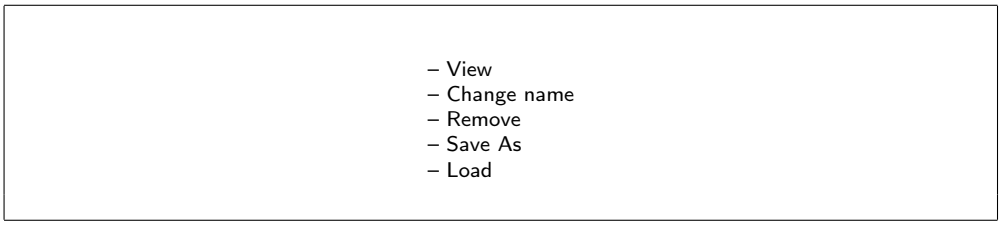
- View – podgląd zawartości obiektu (patrz rysunek 3.19), użytkownik może dowolnie przewijać okno, a także zmieniać jego rozmiar, zależnie od potrzeb

  Podgląd zbioru cięć składa się z czterech kolumn. Pierwsza spełnia funkcję porządkującą, znaczenie pozostałych kolumn jest następujące:

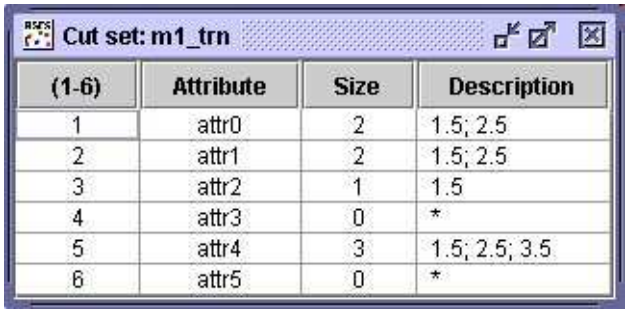


Rysunek 3.16: Informacje o zbiorze reguł

Rysunek 3.17: Ikonka reprezentująca zbiór cięć



Rysunek 3.18: Schemat menu kontekstowego dla zbioru cięć


Cut set: m1_trn

(1-6)	Attribute	Size	Description
1	attr0	2	1.5; 2.5
2	attr1	2	1.5; 2.5
3	attr2	1	1.5
4	attr3	0	*
5	attr4	3	1.5; 2.5; 3.5
6	attr5	0	*

Rysunek 3.19: Podgląd zawartości zbioru cięć

  - Attribute – nazwa atrybutu dla którego wyliczono cięcia
  - Size – liczba cięć na atrybucie
  - Description – lista wartości reprezentująca cięcia na atrybucie, znak $*$ oznacza brak cięć.

- Change name – zmiana nazwy obiektu (patrz rysunek 3.4), nazwa ta jest zapisywana wraz z obiektem do pliku jako nazwa obiektu (a nie nazwa pliku z obiektem). Nazwę obiektu można również zmienić poprzez dwukrotne kliknięcie na nazwę znajdującą się pod ikonką.
- Remove – usunięcie zbioru cięć (konieczne jest potwierdzenie)
- Save As – zapisanie zbioru cięć do pliku
- Load – odczytanie zbioru cięć z podanego pliku

3.5 Kombinacje liniowe

Kombinacją liniową nazywamy sumę ważoną wartości niektórych dostępnych atrybutów. Dla różnych zestawów wag otrzymujemy różne kombinacje liniowe, które mogą być następnie traktowane jako nowe atrybuty.

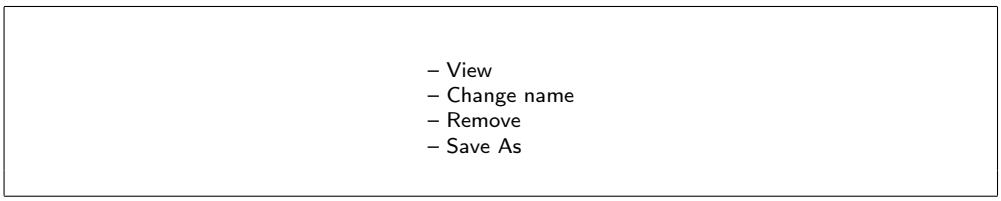
Kombinacje liniowe tworzone są z k-elementowych podzbiorów atrybutów. Podzbiory te, jak również parametry kombinacji liniowych, dobierane są automatycznie za pomocą adaptacyjnego algorytmu optymalizacyjnego. Miara jakości używana do optymalizacji jest jedną z trzech miar opisanych w [23], uwzględniających potencjalną jakość reguł decyzyjnych tworzonych z danej kombinacji liniowej. Użytkownik może wybrać liczbę nowych atrybutów, jak również liczbę składników każdej kombinacji liniowej, podając

Pattern of linear combinations. Jest to ciąg cyfr oznaczających liczbę źródłowych atrybutów dla każdej kombinacji liniowej. Np. ciąg „223344" oznacza wygenerowanie 6 nowych atrybutów (kombinacji liniowych) zbudowanych z par (dwa pierwsze), trójek (dwa środkowe) i czwórek (dwa ostatnie) atrybutów oryginalnych.

Rysunek 3.20: Ikonka reprezentująca zbiór kombinacji liniowych

Podgląd kombinacji liniowych składa się z dwóch kolumn: kolumny porządkującej i kolumny z wypisanymi kombinacjami liniowymi.

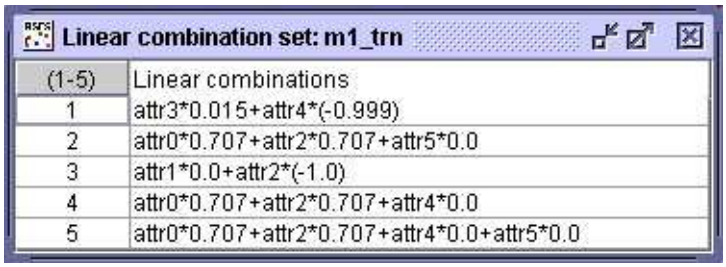
Dwukrotne kliknięcie lewym przyciskiem myszki na ikonce jest równoważne poleceniu View z menu kontekstowego i umożliwia podgląd obiektu.



Rysunek 3.21: Schemat menu kontekstowego dla zbioru kombinacji liniowych

Wykaz poleceń menu kontekstowego dla zbioru liniowych kombinacji:

- View – podgląd zawartości obiektu (patrz rysunek 3.22), użytkownik może dowolnie przewijać okno, a także zmieniać jego rozmiar, zależnie od potrzeb



(1-5)	Linear combinations
1	attr3*0.015+attr4*(-0.999)
2	attr0*0.707+attr2*0.707+attr5*0.0
3	attr1*0.0+attr2*(-1.0)
4	attr0*0.707+attr2*0.707+attr4*0.0
5	attr0*0.707+attr2*0.707+attr4*0.0+attr5*0.0

Rysunek 3.22: Podgląd zawartości zbioru kombinacji liniowych

Podgląd zbioru kombinacji liniowych składa się z dwóch kolumn. Pierwsza z nich to kolumna porządkowa, druga zaś przedstawia kombinację liniową wypisaną jako formułę arytmetyczną na atrybutach.

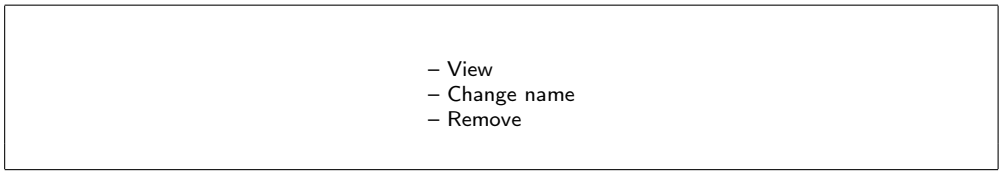
- Change name – zmiana nazwy obiektu (patrz rysunek 3.4), nazwa ta jest zapisywana wraz z obiektem do pliku jako nazwa obiektu (a nie nazwa pliku z obiektem). Nazwę obiektu można również zmienić poprzez dwukrotne kliknięcie na nazwę znajdującą się pod ikonką.
- Remove – usunięcie zbioru kombinacji liniowych (konieczne jest potwierdzenie)
- Save As – zapisanie zbioru kombinacji liniowych do pliku

3.6 Drzewa dekompozycji

Drzewo dekompozycji, dekomponuje dane na fragmenty o zadanym rozmiarze (patrz [20], [4]). Fragmenty te (liście drzewa dekompozycji) są następnie używane do poszukiwania reguł decyzyjnych (patrz np. [5]).

Rysunek 3.23: Ikonka reprezentująca drzewo dekompozycji

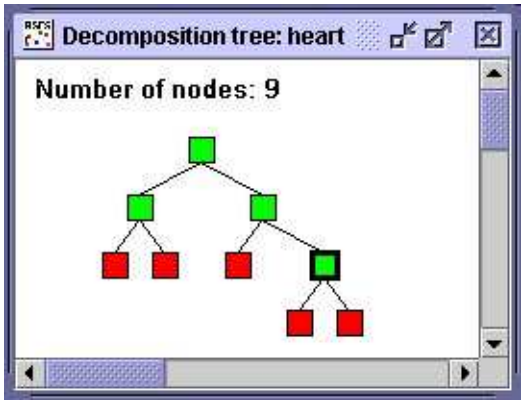
Dwukrotne kliknięcie lewym przyciskiem myszki na ikonce jest równoważne poleceniu View z menu kontekstowego i umożliwia podgląd obiektu.



Rysunek 3.24: Schemat menu kontekstowego dla drzewa dekompozycji

Wykaz poleceń menu kontekstowego dla drzewa dekompozycji:

- View – podgląd zawartości obiektu (patrz rysunek 3.25), użytkownik może dowolnie przewijać okno, a także zmieniać jego rozmiar, zależnie od potrzeb



Rysunek 3.25: Podgląd drzewa dekompozycji

Podgląd drzewa dekompozycji składa się z informacji o liczbie węzłów, oraz graficznej prezentacji drzewa dekompozycji. Zatrzymując kursor myszki na wybranym węźle otrzymamy informację o tym węźle w postaci spełnianego przez niego wzorca wynikającego z konstrukcji drzewa. Każdy węzeł posiada własne menu kontekstowe, zawierające opcję podglądu węzła (dostępną również poprzez dwukrotne kliknięcie). Dodatkowo menu kontekstowe dla liści drzewa dekompozycji, zawiera opcje umożliwiające podgląd reguł oraz ich zapisanie na dysk.

- Change name – zmiana nazwy obiektu (patrz rysunek 3.4), nazwa ta jest zapisywana wraz z obiektem do pliku jako nazwa obiektu (a nie nazwa pliku z obiektem). Nazwę obiektu można również zmienić poprzez dwukrotne kliknięcie na nazwę znajdującą się pod ikonką.

- Remove – usunięcie drzewa dekompozycji (konieczne jest potwierdzenie)

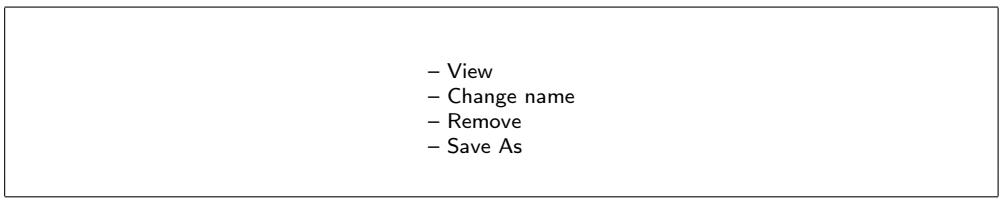
3.7 Klasyfikatory LTF-C

LTF-C (*Local Transfer Function Classifier*) (patrz [24]) to sieć neuronowa do zadań klasyfikacyjnych, o architekturze zbliżonej do sieci radialnych (RBF). Składa się z dwóch warstw neuronów. Pierwsza warstwa (tzw. ukryta) zawiera neurony o gaussowskiej funkcji transferu, które wykrywają w danych treningowych skupiska wzorców z tej samej klasy. Każdy neuron tej warstwy ma przypisaną klasę, której skupisko stara się wykryć. Drugą warstwę tworzą neurony liniowe, które segregują odpowiedzi neuronów ukrytych według

przypisanych klas i sumują je, formułując ostateczną odpowiedź sieci.

Rysunek 3.26: Ikonka reprezentująca klasyfikator LTF-C

Dwukrotne kliknięcie lewym przyciskiem myszki na ikonce jest równoważne poleceniu View z menu kontekstowego i umożliwia podgląd obiektu.



Rysunek 3.27: Schemat menu kontekstowego dla klasyfikatora LTF-C

Wykaz poleceń menu kontekstowego dla klasyfikatora LTF-C:

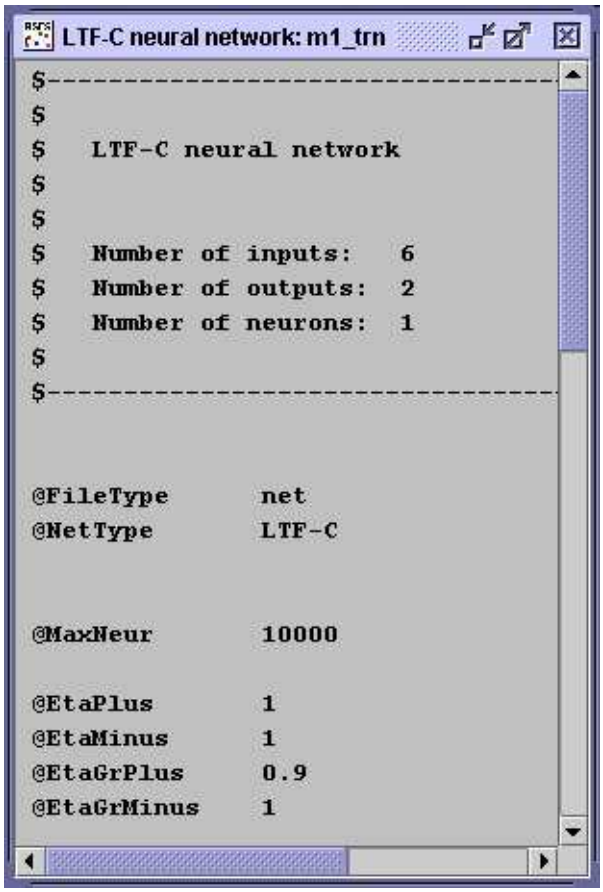
- View – podgląd zawartości obiektu (patrz rysunek 3.28), użytkownik może dowolnie przewijać okno, a także zmieniać jego rozmiar, zależnie od potrzeb

  W podglądzie sieci LTF-C pokazywana jest zawartość pliku, jaki jest tworzony po zapisaniu sieci na dysk. Plik ten składa się z dwóch części: wartości użytych parametrów uczenia, oraz opisu struktury sieci i każdego jej neuronu.

  Wartość każdego parametru poprzedzona jest znakiem `@` i nazwą parametru. Spośród kilkunastu parametrów, wypisanych w pliku (od `@MaxNeur` do `@DeltaTr`), istotne dla użytkownika są tylko `@EtaUse` i `@UseTr`, których wartością jest liczba z pola Threshold for neuron removal w oknie tworzenia sieci LTF-C. Wartość ta steruje procesem usuwania niepotrzebnych neuronów podczas nauki – im jest większa, tym łatwiej neurony są usuwane i tym mniejsza jest sieć wynikowa.

  Od linii `@Inputs` rozpoczyna się opis sieci:

  – `@Inputs` – liczba atrybutów wejściowych
  – `@Outputs` – liczba atrybutów wyjściowych (czyli liczba klas)
  – `@Neurons` – liczba neuronów



Rysunek 3.28: Podgląd klasyfikatora LTF-C

– `@SumCycles` – liczba przeprowadzonych cykli uczenia (każdy cykl to pojedyncza prezentacja wzorca uczącego). Jest ona równa wartości w polu Number of training cycles w oknie tworzenia sieci.

Dalej następuje opis kolejnych neuronów (patrz rys. 3.29).

Najważniesze parametry definiujące neuron to:

– `@Class` – numer klasy, za którą odpowiada dany neuron, jest to liczba od 0 do (@Outputs-1)

– `@Life` – długość życia danego neuronu wyrażona w liczbie cykli uczenia. Prawie zawsze jest ona krótsza niż `@SumCycles`, ponieważ neurony są tworzone w trakcie nauki, a na początku jest tylko jeden neuron.

```
@<Neuron0
  @Class        1
  @Life         2999
  @AvgUsefuln   0.04410261
  @AvgActiv     0.00400000
  @EtaEta       0.000
  @Out          0.000
  @Height       1.0000

  @Weights
    0.8408  0.3130  0.3921  0.3070

  @RecipOfRadii
    0.3955  0.7920  1.9009  1.5605
 @>
```

Rysunek 3.29: Przykład opisu neuronu w widoku klasyfikatora LTF-C

– `@Weights` – wagi neuronu, czyli współrzędne środka jego pola recepcyjnego – odpowiadają wektorowi wejściowemu, wywołującemu maksymalną odpowiedź neuronu. Kolejne wartości odpowiadają kolejnym atrybutom wejściowym.

  Uwaga! Jeśli przy tworzeniu sieci była zaznaczona opcja Normalize each numeric attribute, to wartości te są znormalizowane tak samo jak odpowiednie atrybuty (atrybuty są normalizowane do zerowej wartości oczekiwanej i jednostkowej wariancji).

– `@RecipOfRadii` – odwrotności promieni neuronu, czyli szerokości jego pola recepcyjnego w każdym wymiarze. Im dana wartość jest mniejsza, tym szersze pole recepcyjne w danym wymiarze, więc tym mniejsze znaczenie danego atrybutu dla odpowiedzi tego neuronu.

  Uwaga! Podobnie jak w przypadku wag, wartości te zależą od normalizacji atrybutów.

Uwaga! Neurony, które zostały usunięte w trakcie nauki, nie występują w opisie sieci. Opisane są tylko neurony, które przetrwały do końca nauki.

- Change name – zmiana nazwy obiektu (patrz rysunek 3.4), nazwa ta jest zapisywana wraz z obiektem do pliku jako nazwa obiektu (a nie nazwa pliku z obiektem). Nazwę obiektu można również zmienić poprzez dwukrotne kliknięcie na nazwę znajdującą się pod ikonką.

- Remove – usunięcie obiektu (konieczne jest potwierdzenie)

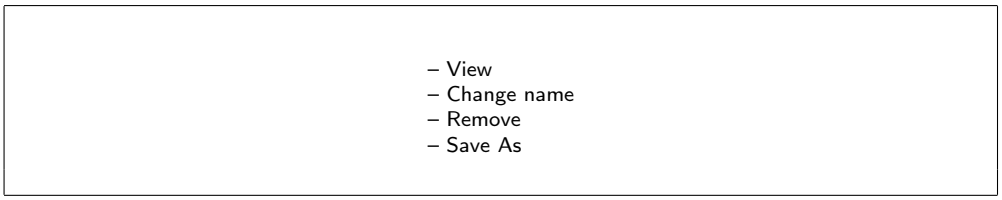
- Save As – zapisanie obiektu do pliku

3.8 Wyniki

Obiekt wyniki służy do przeglądania wyników klasyfikacji

Rysunek 3.30: Ikonka reprezentująca wyniki

Dwukrotne kliknięcie lewym przyciskiem myszki na ikonce jest równoważne poleceniu View z menu kontekstowego i umożliwia podgląd obiektu.

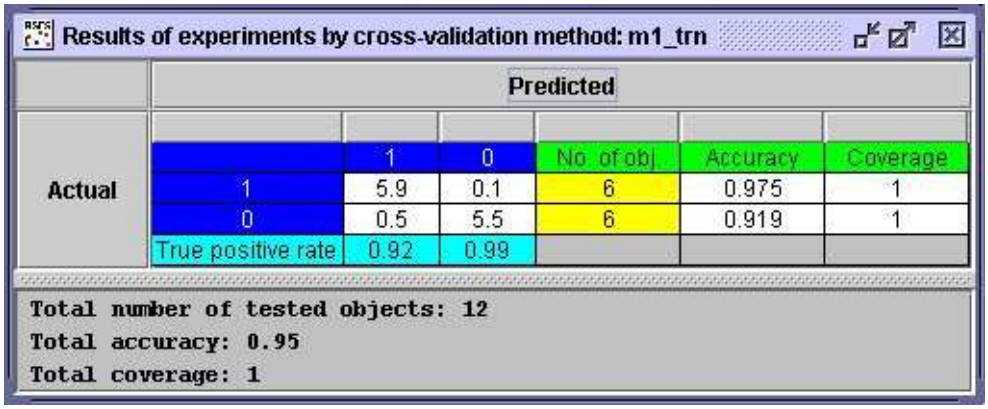


Rysunek 3.31: Schemat menu kontekstowego dla wyników

Wykaz poleceń menu kontekstowego dla wyników:

- View – podgląd zawartości obiektu (patrz rysunek 3.32), użytkownik może dowolnie przewijać okno, a także zmieniać jego rozmiar, zależnie od potrzeb

  Podgląd wyników klasyfikacji zawiera wiele różnych informacji. Jedną z nich jest macierz błędów (*confusion matrix*), która w naszym przykładzie ma postać:



Rysunek 3.32: Podgląd wyników klasyfikacji

	Predicted	
Actual	1	0
1	5.9	0.1
0	0.5	5.5

Wiersze odpowiadają klasom decyzyjnym, zaś kolumny odpowiedziom jakich udzielił system decyzyjny. Na powyższym przykładzie widzimy, że system czasem błędnie klasyfikuje obiekt z klasy 1 przypisując go do klasy 0 (średnio 0.1). Widzimy również, że badany system pięciokrotnie częściej myli klasę 1 z klasą 0. Jeśli wszystkie wartości poza przekątną mają wartość 0 to należy z tego wnioskować, że system nie popełnił ani jednego błędu podczas klasyfikacji.

Na prawo od tablicy błędów znajdują się trzy kolumny z dodatkowymi statystykami:

- No. of obj. – liczba obiektów z danej klasy decyzyjnej
- Acuraccy – stosunek poprawnie sklasyfikowanych obiektów do wszystkich obiektów sklasyfikowanych z tej klasy
- Coverage – stosunek obiektów sklasyfikowanych do wszystkich obiektów z danej klasy decyzyjnej

Ostatni wiersz tabelki zawiera informację o wartości True positive rate dla każdej z klas – parametr informujący o ilości właściwie sklasyfikowanych obiektów do poszczególnych klas.

Pod tabelką znajduje się lista z dodatkowymi informacjami:

  - Total number of tested objects: – liczba wszystkich przetestowanych obiektów
  - Total acuraccy – stosunek poprawnie sklasyfikowanych obiektów (suma wartości na przekątnej macierzy błędów) do wszystkich przetestowanych obiektów
  - Total coverage – stosunek obiektów sklasyfikowanych do wszystkich testowanych obiektów

  W naszym przypadku Total coverage ma wartość 1 co oznacza, że system decyzyjny sklasyfikował wszystkie obiekty. Jednak może się zdarzyć tak, że pewnych obiektów system nie będzie potrafił przydzielić do żadnej z klas decyzyjnych. Wówczas wartość tej statystyki będzie mniejsza od 1.

- Change name – zmiana nazwy obiektu (patrz rysunek 3.4), nazwa ta jest zapisywana wraz z obiektem do pliku jako nazwa obiektu (a nie nazwa pliku z obiektem). Nazwę obiektu można również zmienić poprzez dwukrotne kliknięcie na nazwę znajdującą się pod ikonką.
- Remove – usunięcie wyników (konieczne jest potwierdzenie)
- Save As – zapisanie wyników do pliku

Rozdział 4

Przegląd głównych metod analizy danych

Rozdział ten poświęcony jest przeglądowi głównych metod algorytmicznych analizy danych, jakie zostały zaimplementowane w RSES 2.1. Zawiera on opis znaczenia poszczególnych opcji dostępnych w systemie dla opisywanych metod, a także zarys związanych z nimi podstaw teoretycznych wraz z odwołaniami do literatury.

Szczegóły dotyczące znaczenia i budowy obiektów występujących w projektach RSES-a, o których mowa w tym rozdziale, zostały przedstawione w rozdziale 3.

4.1 Analiza brakujących wartości

Brakujące wartości oznaczane są w tabelkach z danymi jako `MISSING`, `NULL` lub znakiem `'?'` (wielkość liter nie ma znaczenia).

System RSES oferuje cztery następujące podejścia do zagadnienia problemu brakujących wartości:

- usuwanie obiektów z brakującymi wartościami (opcja menu Complete/Remove objects with missing values z menu kontekstowego dla tablicy),
- wypełnianie pustych miejsc, które odbywa się dwiema następującymi metodami (patrz [11]):
  - wypełnianie pustych miejsc najczęściej występującą wartością – dla atrybutów nominalnych lub wartością średnią – dla atrybutów numerycznych (opcja: Complete/Complete with most common or mean value w menu kontekstowym dla tablicy),

  - wypełnianie pustych miejsc najczęściej występującą wartością (dla atrybutów nominalnych) lub wartością średnią (dla atrybutów numerycznych) w obrębie klasy decyzyjnej do której przynależy uzupełniany obiekt (opcja: Complete/Complete with most common or mean value w.r.t. decision class w menu kontekstowym dla tablicy),

- analiza danych bez uwzględniania pustych miejsc (nie uwzględnia się rozróżnialności pomiędzy parami obiektów dla tych atrybutów, gdzie występują puste miejsca, co osiąga się poprzez stosowne ustawienie opcji w okienku dialogowym dotyczącym obliczania reduktów i reguł),

- traktowanie pustych miejsc jako istotne informacje (wartości `NULL` są traktowane jako normalne wartości).

4.2 Cięcia i podziały wartości atrybutów

Za pomocą polecenia Discretize/Generate cuts dostępnego w menu kontekstowym dla tablicy, możemy wygenerować podziały wartości atrybutów, których można następnie użyć na przykład do dyskretyzacji danych.

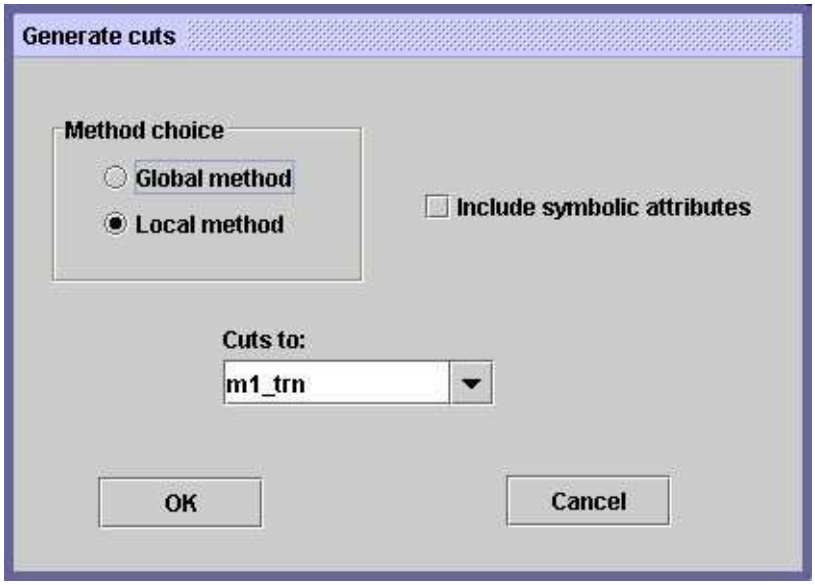
Użytkownik ma do dyspozycji kilka parametrów dla tej metody:

- Method choice – wybór metody dyskretyzacji:
  - Global method – metoda globalna (patrz np. [5]),
  - Local method – metoda lokalna, która jest nieco szybsza od metody globalnej lecz może generować większą liczbę cięć (patrz np. [5]),

- Include symbolic attributes – opcja dostępna tylko dla metody lokalnej, powoduje że algorytm dyskretyzuje również atrybuty nominalne (poprzez grupowanie wartości),

- Cuts to – wybór nazwy obiektu w projekcie do którego będą wstawione obliczone cięcia lub podziały wartości atrybutów.

Polecenie Discretize/Discretize table (dostępne w menu kontekstowym tablicy) umożliwia dyskretyzację tablicy przy pomocy istniejących już cięć. Użytkownik podaje cięcia, którymi chce się posłużyć do dyskretyzacji tablicy (*cięcia muszą istnieć, co oznacza, że należy je wcześniej policzyć lub wczytać z dysku*).



Rysunek 4.1: Opcje generowania cięć i/lub podziałów wartości dla atrybutów

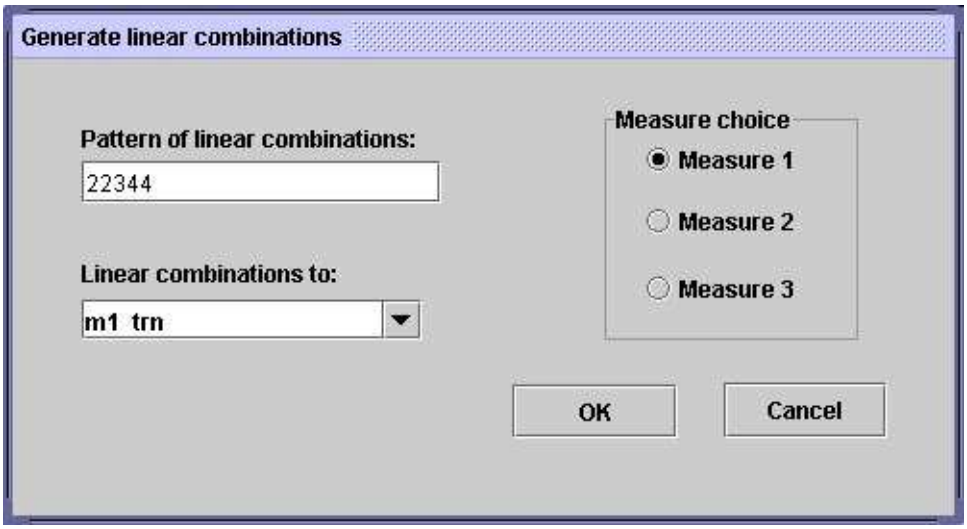
4.3 Kombinacje liniowe

Możliwe jest tworzenie kombinacji liniowych atrybutów, które mogą być następnie użyte do stworzenia nowych atrybutów (patrz [23]). W tym celu należy użyć polecenia Linear combinations/Generate linear combinations z menu kontekstowego dla tablicy.

Przy tworzeniu kombinacji liniowych użytkownik ma do dyspozycji następujące opcje (patrz rysunek 4.2):

- Pattern of linear combinations – schemat tworzenia kombinacji liniowych. Na przykład ciąg 22344 oznacza, że chcemy stworzyć 5 różnych kombinacji liniowych, z czego dwie składają się z par atrybutów, jedna składa się z trzech atrybutów a dwie pozostałe z czterech atrybutów,
- Measure choice – wybór jednej z trzech miar jakości kombinacji liniowej (patrz [23]),
- Linear combinations to – wybór nazwy obiektu w projekcie do którego będą wstawione policzone kombinacje liniowe.

Aby w oparciu o kombinację liniową stworzyć nowy atrybut należy użyć polecenia Linear combinations/Add linear combinations as new attributes z menu kontekstowego dla tablicy. Otrzymamy wówczas nową tablicę zawierającą



Rysunek 4.2: Opcje generowania kombinacji liniowych atrybutów

dodatkowe atrybuty stanowiące kombinacje liniowe oryginalnych atrybutów, użytkownik podaje jakiego zbioru kombinacji liniowych chce użyć do rozbudowania tablicy (*kombinacje liniowe muszą istnieć, co oznacza, że użytkownik musi je wcześniej policzyć*).

Zauważmy, że dzięki opcji Select subtable z menu kontekstowego dla tablicy można dokonywać ręcznej selekcji cech w tablicy z dodanymi atrybutami. Na przykład można pozostawić jedynie nowe cechy i atrybut decyzyjny.

4.4 Redukty i reguły decyzyjne

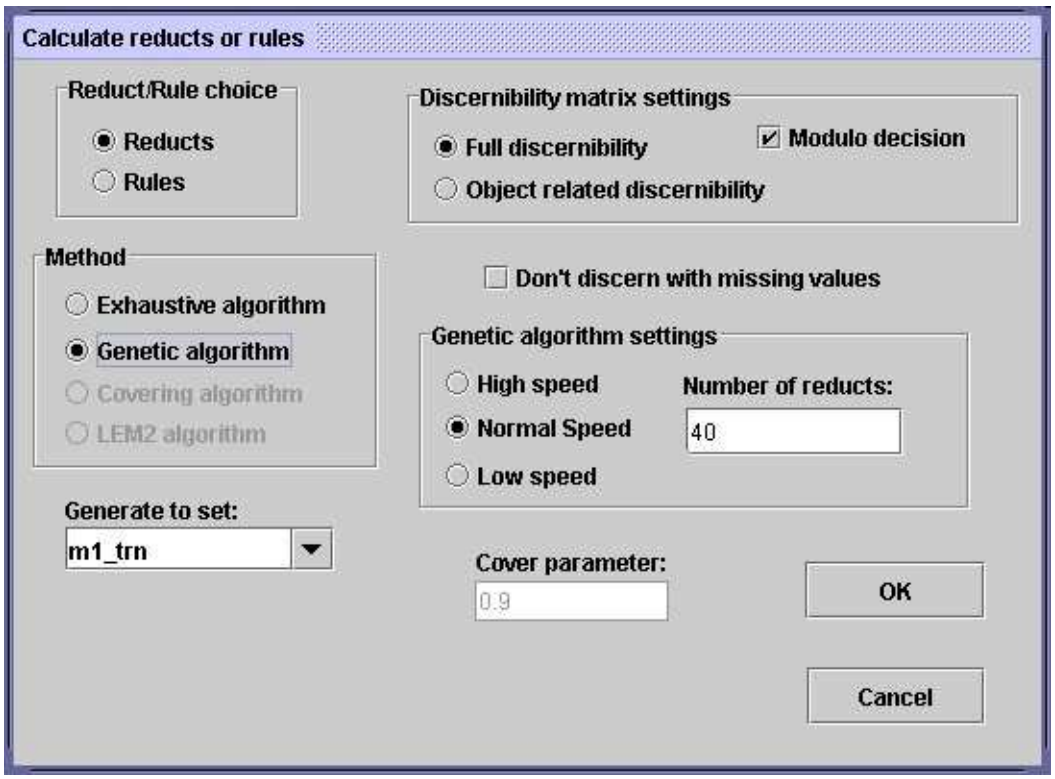
Dla tabeli z danymi możemy policzyć redukty, oraz reguły decyzyjne. W tym celu należy użyć polecenia Reducts/Rules / Calculate reducts or rules z menu kontekstowego dla tablicy.

Użytkownik ma możliwość wyboru, czy chce liczyć redukty czy reguły, oraz ma do dyspozycji wiele opcji (patrz rysunek 4.3) sterujących tym procesem:

- Reduct/Rule choice – wybór trybu pracy:
  - Reducts – generowanie reduktów,
  - Rules – generowanie reguł,

- Discernibility matrix settings – właściwości tworzonej macierzy odróżnialności (*opcja dostępna tylko przy liczeniu reduktów*),
  - Full discernibility – pełna rozróżnialność wszystkich par obiektów (rozróżnialność globalna),
  - Object related discernibility – relatywna rozróżnialność (względem ustalonego obiektu),
  - Modulo decision – rozróżnialność względem decyzji,
- Method – wybór metody obliczeń:
  - Exhaustive algorithm – wyczerpujący algorytm liczenia wszystkich reduktów lub wszystkich reguł z minimalną liczbą deskryptorów w poprzedniku (w zależności od stanu opcji Reduct/Rule choice wyznaczane są wszystkie redukty lub wszystkie reguły lokalne) (patrz [5]),
  - Genetic algorithm – algorytm genetyczny liczenia reduktów i reguł (patrz np. [5]),
  - Covering algorithm – algorytm pokryciowy (*tylko dla reguł* – patrz [5]),
  - LEM2 algorithm – algorytm LEM2 (*tylko dla reguł* – (patrz [10])),
- Genetic algorithm settings – parametry dla algorytmu genetycznego (*opcja dostępna tylko przy wyborze jako metody algorytmu genetycznego*):
  - High speed – tryb szybki,
  - Medium speed – tryb standardowy,
  - Low speed – tryb dokładny (wolny),
  - Number of reducts – maksymalna liczba reduktów jaką chcemy uzyskać (rozmiar populacji),
- Cover parameter – wymagany współczynnik pokrycia zbioru treningowego (*opcja dostępna tylko przy wyborze jako metody algorytmu pokryciowego lub LEM2*),
- Don't discern with missing values – ignorowanie brakujących wartości podczas tworzenia macierzy odróżnialności (chodzi o to, że podczas wyznaczania zbioru atrybutów na których różnią się dwa wybrane obiekty, można nie uwzględniać tych atrybutów, na których różnica dotyczy wartości `NULL` od innej wartości lub wartości `NULL` od wartości `NULL` – patrz także podrozdział 4.1),

- Generate to set – wskazanie nazwy obiektu do którego będą generowane redukty/reguły, może to być nowy lub też istniejący już w projekcie obiekt.



Rysunek 4.3: Opcje generowania reduktów i reguł decyzyjnych

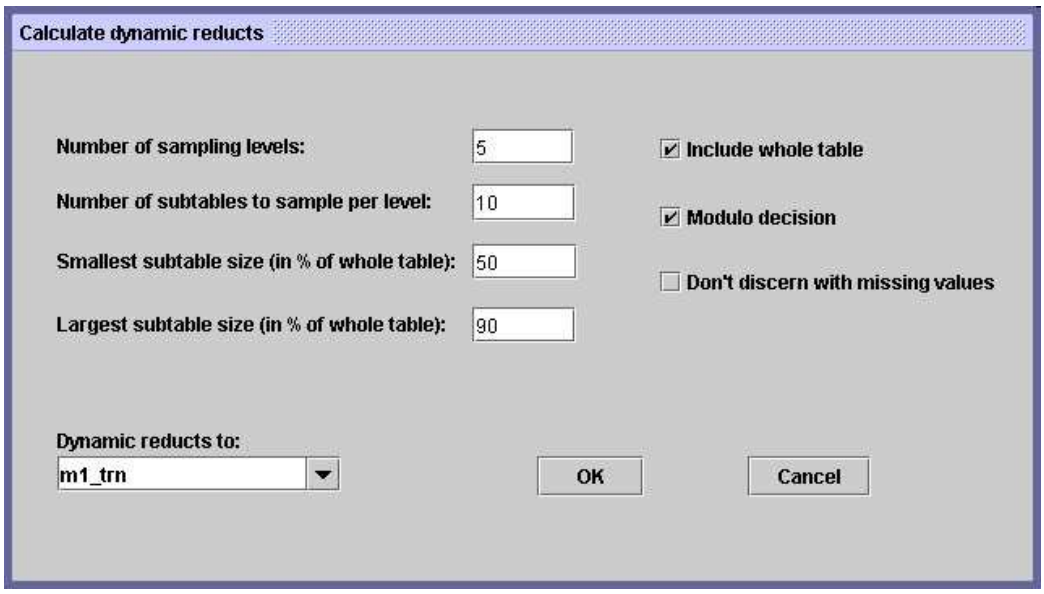
Możliwe jest również policzenie reduktów dynamicznych, czyli reduktów, które pozostają reduktami dla wielu podtablic danej tablicy decyzyjnej (patrz [2], [5]). W tym celu należy użyć polecenia Reducts/Rules / Calculate dynamic reducts z menu kontekstowego dla tablicy.

Do sterowania tym procesem dostępne są następujące opcje (patrz rysunek 4.4):

- Number of sampling levels – liczba poziomów losowań podtablic (próbek),

- Number of subtables to sample per level – liczba podtablic (próbek) losowanych w ramach jednego poziomu,

- Smallest subtable size (in % of whole table) – rozmiar najmniejszej podtablicy (próbki) podany w procentach względem całej tablicy z danymi,
- Largest subtable size (in % of whole table) – rozmiar największej podtablicy podany w procentach względem całej tablicy z danymi,
- Include whole table – umożliwienie wyboru całej tablicy z danymi jako jednej z próbek,
- Modulo decision – rozróżnialność względem decyzji,
- Don't discern with missing values – ignorowanie brakujących wartości (chodzi o to, że podczas wyznaczania zbioru atrybutów na których różnią się dwa wybrane obiekty, można nie uwzględniać tych atrybutów, na których różnica dotyczy wartości `NULL` od innej wartości lub wartości `NULL` od wartości `NULL` – patrz także podrozdział 4.1),
- Dynamic reducts to – wskazanie nazwy obiektu do którego będą generowane redukty, może to być nowy lub też istniejący już obiekt.

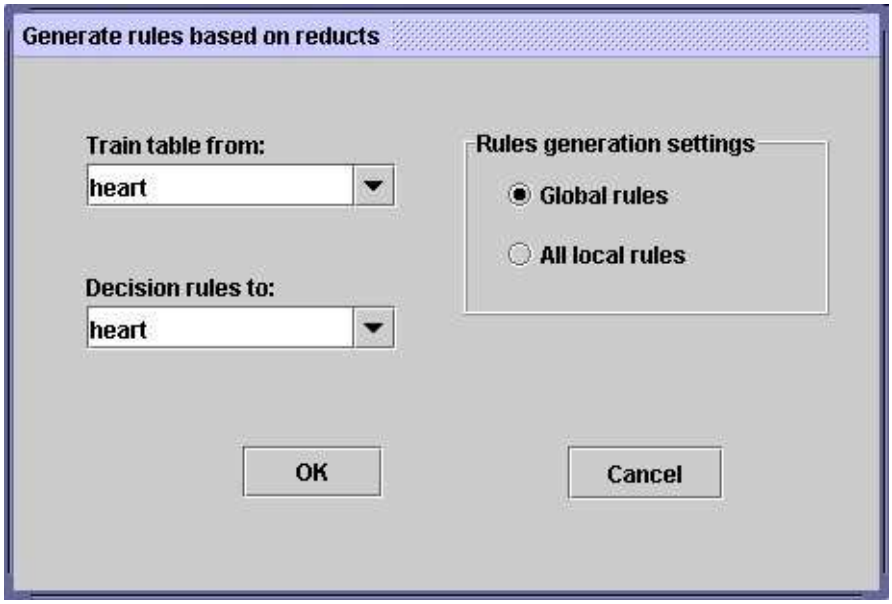


Rysunek 4.4: Opcje generowania reduktów dynamicznych

Po wyliczeniu reduktów możemy ich użyć do liczenia reguł. W tym celu należy użyć polecenia Generate rules z menu kontekstowego dla zbioru reduktów.

Mamy wówczas dostępne następujące opcje (patrz rysunek 4.5):

- Train table from – tablica z danymi (*tablica taka musi istnieć!*),
- Decision rules to – nazwa obiektu, który będzie przechowywał wyliczone reguły,
- Rules generation setting – wybór metody generowania reguł:
  - Global rules – liczenie reguł globalnych, tzn. algorytm przegląda wszystkie obiekty z tablicy treningowej i generuje reguły poprzez mechaniczne przyłożenie reduktów do obiektów, przy czym atrybuty z poprzednika reguł brane są z reduktów, a wartości tych atrybutów oraz wartość atrybutu decyzyjnego brane są z obiektów tablicy treningowej,
  - All local rules – liczenie reguł metodą lokalną, tzn. na podstawie każdego reduktu wyodrębniona jest podtablica zawierająca atrybuty warunkowe występujące w redukcie, po czym liczone są dla niej tzw. *reguły z minimalną liczbą deskryptorów*, czyli reguły względem ustalonego obiektu i decyzji (patrz np. [5]); na koniec wszystkie otrzymane zbiory reguł dla poszczególnych reduktów są sumowane do jednego zbioru reguł.



Rysunek 4.5: Opcje dostępne przy budowie reguł w oparciu o zbiór reduktów

Aby użyć zbioru reguł do klasyfikacji obiektów należy wybrać polecenie Classify/Test table using rule set z menu kontekstowego dla tablicy. W projekcie musi istnieć obiekt przechowujący zbiór reguł, oznacza to konieczność jego wcześniejszego wyliczenia lub wczytania z pliku.

Dostępne parametry (patrz rysunek 4.6) klasyfikacji przy pomocy reguł to:

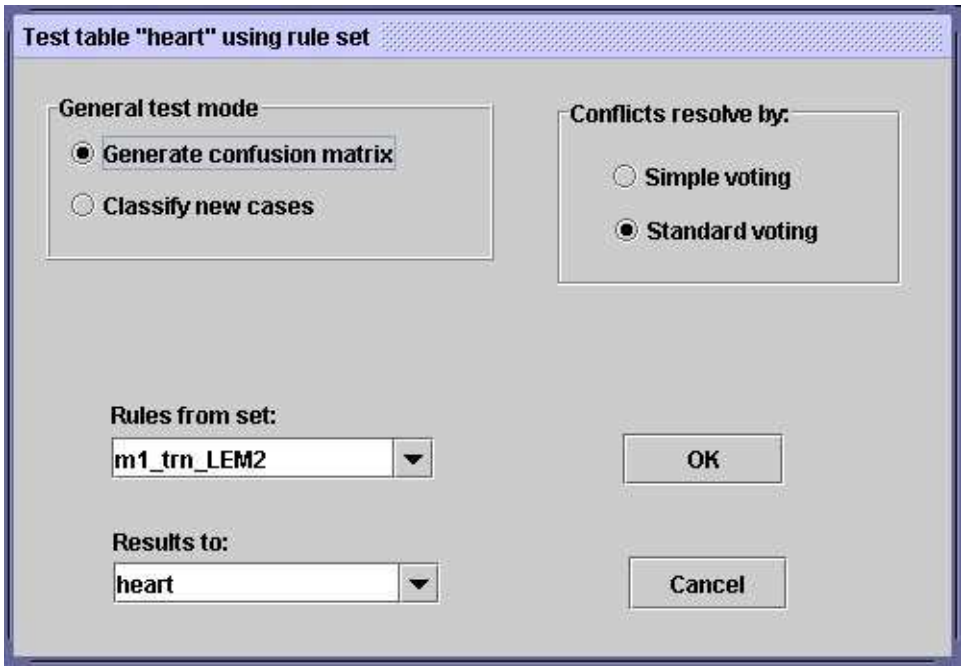
- General test mode – wybór rodzaju testu (co ma być jego wynikiem)
  - Generate confusion matrix – wyliczenie macierzy błędu klasyfikacji (patrz podrozdział 3.8)
  - Classify new cases – klasyfikacja nowych przypadków i dopisanie do testowanych danych kolumny z wyliczoną decyzją
- Conflict resolve by – sposób rozwiązywania konfliktów:
  - Simple voting – proste głosowanie (jedna reguła - jeden głos),
  - Standard voting – standardowy system głosowania (każda reguła głosuje z siłą równą liczbie obiektów ją wspierających),
- Rules from set – zbiór z regułami,
- Results to – wskazanie obiektu w którym umieszczone zostaną wyniki klasyfikacji.

4.5 Dekompozycja danych

Aby wykonać dekompozycję danych, czyli zbudować drzewo dekompozycji, które później można użyć jako klasyfikatora (patrz [20], [4]) należy użyć polecenia Make decomposition z menu kontekstowego dla tablicy.

Do sterowania tym algorytmem dostępne są następujące opcje (patrz rysunek 4.7):

- Maximal size of leaf – maksymalna wielkość liścia drzewa dekompozycji poniżej której nie jest on już dalej dzielony (czyli maksymalna wielkość tabelek w liściach drzewa dekompozycji),
- Discretization in leafs – umożliwienie wstępnej dyskretyzacji atrybutów w tabelkach znajdujących się w liściach po dekompozycji; można ustawić także rodzaj dyskretyzacji (patrz podrozdział 4.2),



Rysunek 4.6: Opcje klasyfikacji przy pomocy reguł lub drzewa dekompozycji

- Shortening ratio – współczynnik skracania reguł decyzyjnych policzonych dla tabelek znajdujących się w poszczególnych liściach drzewa dekompozycji, dla wartości 1.0 reguły nie są skracane, im bliżej zera tym większe dopuszczamy skracanie reguł decyzyjnych (patrz także podrozdział 3.3),

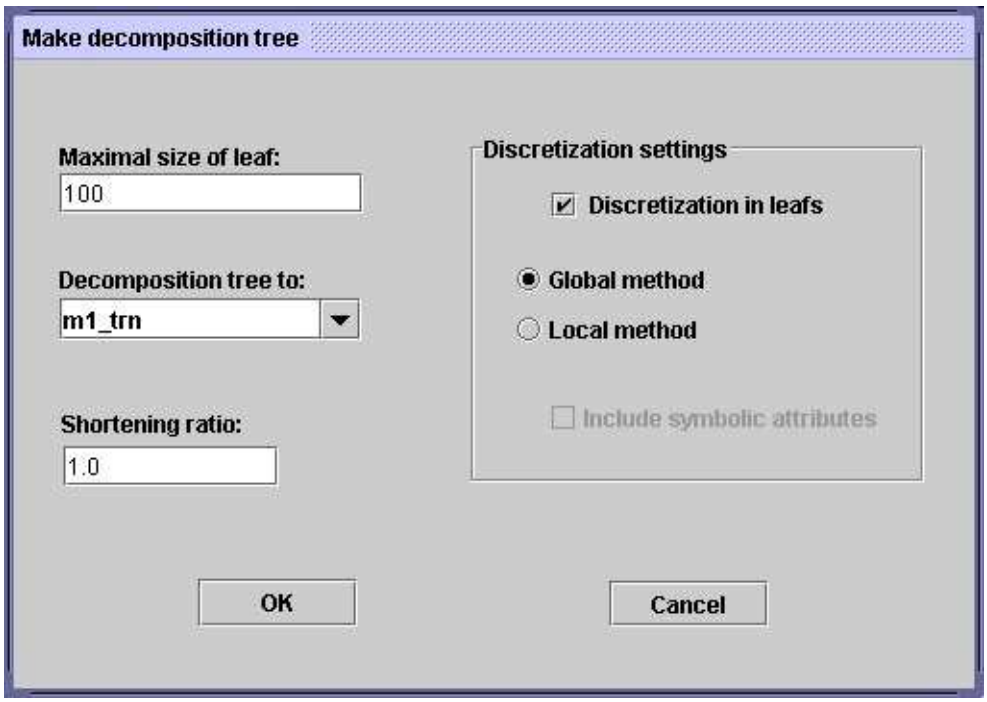
- Decomposition tree to – wybór nazwy obiektu w projekcie do którego będzie wstawione policzone drzewo dekompozycji.

Aby użyć drzewa dekompozycji do klasyfikacji obiektów należy wybrać polecenie Classify/Test table using decomposition tree z menu kontekstowego dla tablicy. W projekcie musi istnieć obiekt przechowujący drzewo dekompozycji, oznacza to konieczność jego wcześniejszego wyliczenia.

Zarówno okno dialogowe (patrz rysunek 4.6), jak i dostępne opcje są analogiczne jak w przypadku klasyfikacji przy użyciu reguł:

- General test mode – wybór rodzaju testu (co ma być jego wynikiem)

  - Generate confusion matrix – wyliczenie macierzy błędu klasyfikacji (patrz podrozdział 3.8)



Rysunek 4.7: Opcje dekompozycji tablic decyzyjnych

  - Classify new cases – klasyfikacja nowych przypadków i dopisanie do testowanych danych kolumny z wyliczoną decyzją

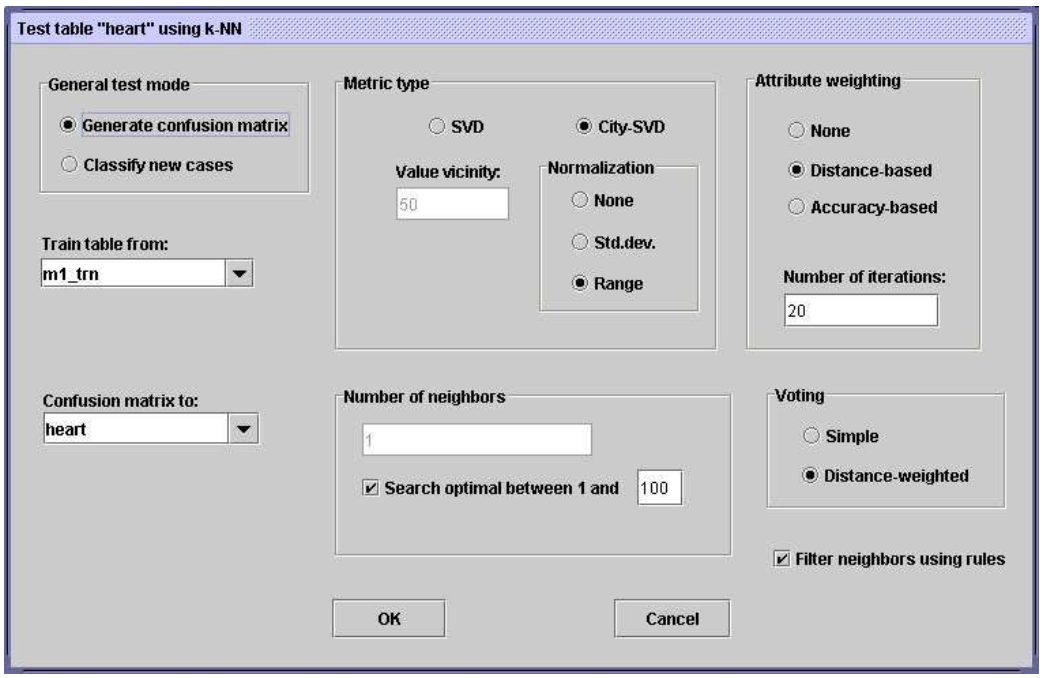
- Conflict resolve by – sposób rozwiązywania konfliktów:
  - Simple voting – proste głosowanie (jedna reguła - jeden głos),
  - Standard voting – standardowy system głosowania (każda reguła głosuje z siłą równą liczbie obiektów ją wspierających),
- Decomposition tree from – drzewo dekompozycji,
- Results to – wskazanie obiektu w którym umieszczone zostaną wyniki klasyfikacji.

4.6 Klasyfikatory typu k-NN

Metoda tworzenia klasyfikatorów typu k-NN (k najbliższych sąsiadów) nie wymaga jawnego uczenia, gdyż w systemie RSES wszelkie operacje inicju-

jące ten rodzaj klasyfikacji wykonywane są automatycznie przed każdym testem. Dlatego aby użyć metody k-NN do klasyfikacji obiektów należy tylko wybrać polecenie Classify/Test table using k-NN z menu kontekstowego dla tablicy, przy czym w projekcie musi już istnieć niepusta tablica z danymi treningowymi.

Algorytm konstruuje miarę odległości między obiektami na podstawie tablicy treningowej i dla każdego testowanego obiektu wybiera decyzje na podstawie decyzji k najbliższych do niego obiektów ze zbioru treningowego (patrz [8, 9]).



Rysunek 4.8: Opcje klasyfikacji przy pomocy k-NN

Metoda ta ma następujące parametry (patrz rysunek 4.8):

- General test mode – wybór rodzaju testu (co ma być jego wynikiem)
  - Generate confusion matrix – wyliczenie macierzy błędu klasyfikacji (patrz podrozdział 3.8)
  - Classify new cases – klasyfikacja nowych przypadków i dopisanie do testowanych danych kolumny z wyliczoną decyzją

- Train table from – określenia, która z tablic w projekcie jest tablicą treningową,
- Results to – wskazanie obiektu w którym umieszczone zostaną wyniki klasyfikacji,
- Metric type – wybór rodzaju metryki mierzącej odległość miedzy obiektami:
  - SVD – metryka *Simple Value Difference* – każdej wartości atrybutu przypisuje wektor rozkładu decyzji dla tej wartości w zbiorze treningowym, odległość między obiektami jest sumą odległości miedzy wartościami poszczególnych atrybutów, a odległość miedzy wartościami atrybutu jest określona jako różnica miedzy rozkładami decyzji przypisanymi do tych wartości. ,

    Parametrem metryki SVD jest Value vicinity, mający znaczenie w przypisywaniu rozkładu decyzyjnego do wartości atrybutów numerycznych, dla danej wartości numerycznej w określa, ile wartości ze zbioru treningowego na danym atrybucie sąsiadujących z wartością w jest branych pod uwagę do wyliczania rozkładu decyzyjnego odpowiadającego wartości w.
  - City-SVD – metryka łączy odległość miejska dla atrybutów numerycznych z odległością *Simple Value Difference* dla atrybutów symbolicznych; odległość między obiektami jest sumą odległości miedzy wartościami poszczególnych atrybutów, dla atrybutów numerycznych jest to różnica bezwzględna między wartościami atrybutu, a dla atrybutów symbolicznych różnica między rozkładami decyzji w zbiorze treningowym dla danych wartości atrybutu.

    Metryka ma jeden parametr Normalization mający znaczenie dla atrybutów numerycznych. Parametr określa wartość, przez jaką jest normalizowana różnica bezwzględna wartości numerycznych. Dostępne wartości tego parametru to:

    None – różnica bezwzględna wartości numerycznych nie jest normalizowana,

    Std.dev. – różnica bezwzględna wartości numerycznych jest normalizowana przez odchylenie standardowe wartości danego atrybutu w zbiorze treningowym,

    Range – różnica bezwzględna wartości numerycznych jest normalizowana przez różnice między maksymalną i minimalną wartością danego atrybutu w zbiorze treningowym,

- Attribute weighting – wybór metody skalowania odległości dla poszczególnych atrybutów w metryce:
  - None – odległości dla poszczególnych atrybutów są sumowane bez skalowania,
  - Distance-based – iteracyjna metoda doboru wag dla poszczególnych atrybutów nakierowana na optymalizacje odległości do poprawnie klasyfikowanych obiektów treningowych,
  - Accuracy-based – iteracyjna metoda doboru wag dla poszczególnych atrybutów nakierowana na optymalizacje klasyfikacji obiektów treningowych,
  - Number of iterations – liczba wykonywanych iteracji, ma znaczenia tylko dla iteracyjnych metod doboru wag,
- Number of neighbours – liczba najbliższych sąsiadów, których decyzje są uwzględniane przy wyliczaniu decyzji testowanego obiektu:
  - Search optimal between 1 and ... – jeśli ta opcja jest zaznaczona, algorytm sam wylicza optymalna liczbę sąsiadów na podstawie zbioru treningowego,
- Voting – wybór metody głosowania najbliższych sąsiadów przy wyborze decyzji dla obiektu testowego:
  - Simple – algorytm przypisuje obiektowi testowemu decyzje najczęściej występująca wśród najbliższych sąsiadów,
  - Distance-weighted – każdy sąsiad głosuje na swoją decyzję z wagą równą odległości między danym sąsiadem i testowanym obiektem. Decyzja o największej sumie poszczególnych wag jest przypisywana do testowanego obiektu,
- Filter neighbours using rules – jeśli ta opcja jest zaznaczona, to przy każdym testowanym obiekcie algorytm wyklucza z głosowania tych sąsiadów, którzy dają lokalną regułę sprzeczną z innym sąsiadem.

4.7 Klasyfikator LTF-C

Aby zbudować klasyfikator LTF-C (*Local Transfer Function Classifier*) bazujący na sieci neuronowej należy użyć polecenia Create LTF-C z menu kontekstowego dla tablicy.

Uczenie sieci LTF-C odbywa się poprzez nieznaczne modyfikowanie jej parametrów i struktury po każdej prezentacji wzorca uczącego. Istnieją cztery rodzaje modyfikacji, przeprowadzanych niezależnie:

1. korekta położenia centrum funkcji gaussowskich,
2. korekta szerokości funkcji gaussowskich (niezależnie dla każdego neuronu ukrytego i każdej składowej wektora wejściowego),
3. dodanie nowego neuronu do warstwy ukrytej,
4. usunięcie niepotrzebnych neuronów z warstwy ukrytej.

Przed rozpoczęciem nauki sieć nie zawiera żadnego neuronu ukrytego. Są one dodawane w trakcie nauki, gdy jest to potrzebne. Dzięki temu użytkownik nie musi sam ustalać rozmiaru warstwy ukrytej, lecz jest on wyznaczany automatycznie.

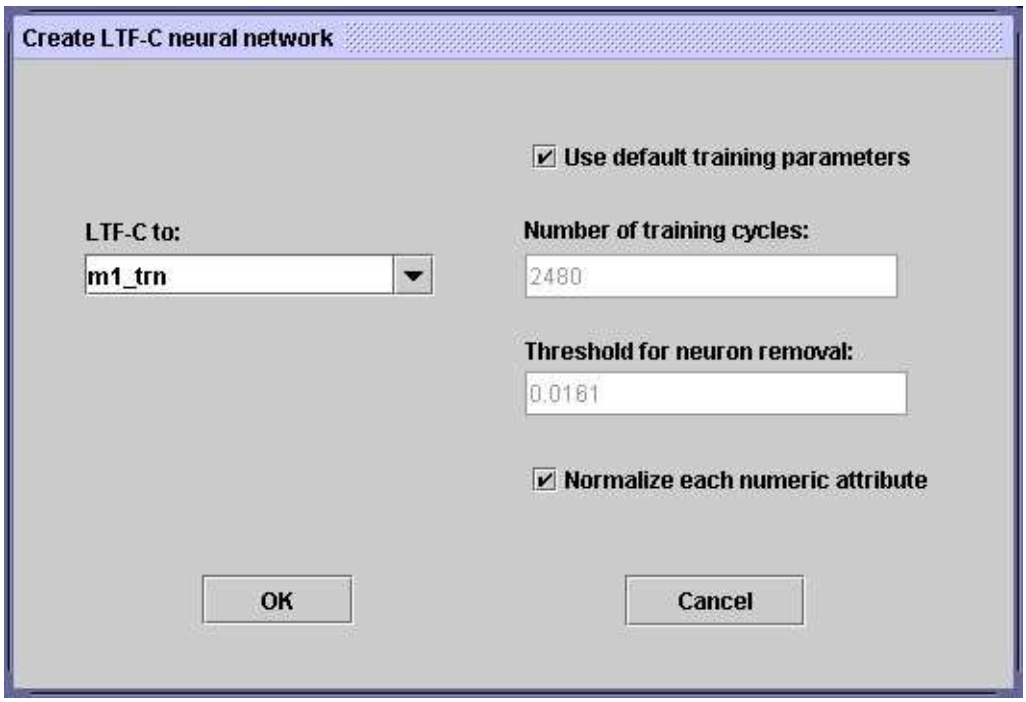
Do sterowania procesem tworzenia klasyfikatora dostępne są następujące opcje (patrz rysunek 4.9):

- LTF-C to – wskazanie nazwy obiektu, w którym zostanie umieszczony klasyfikator LTF-C,
- Use default training parameters – wybór między ręcznym i domyślnym doborem parametrów:
  - Number of training cycles – liczba cykli uczenia klasyfikatora,
  - Threshold for neurons removal – poziom błędu powyżej, którego kasowane są neurony.

Aby użyć klasyfikatora LTF-C do klasyfikacji obiektów należy wybrać polecenie Classify/Test table using LTF-C z menu kontekstowego dla tablicy. W projekcie musi istnieć obiekt przechowujący klasyfikator LTF-C, oznacza to konieczność jego wcześniejszego wyliczenia.

Użytkownik ma do dyspozycji następujące opcje (patrz rysunek 4.10):

- General test mode – wybór rodzaju testu (co ma być jego wynikiem)
  - Generate confusion matrix – wyliczenie macierzy błędu klasyfikacji (patrz podrozdział 3.8)
  - Classify new cases – klasyfikacja nowych przypadków i dopisanie do testowanych danych kolumny z wyliczoną decyzją



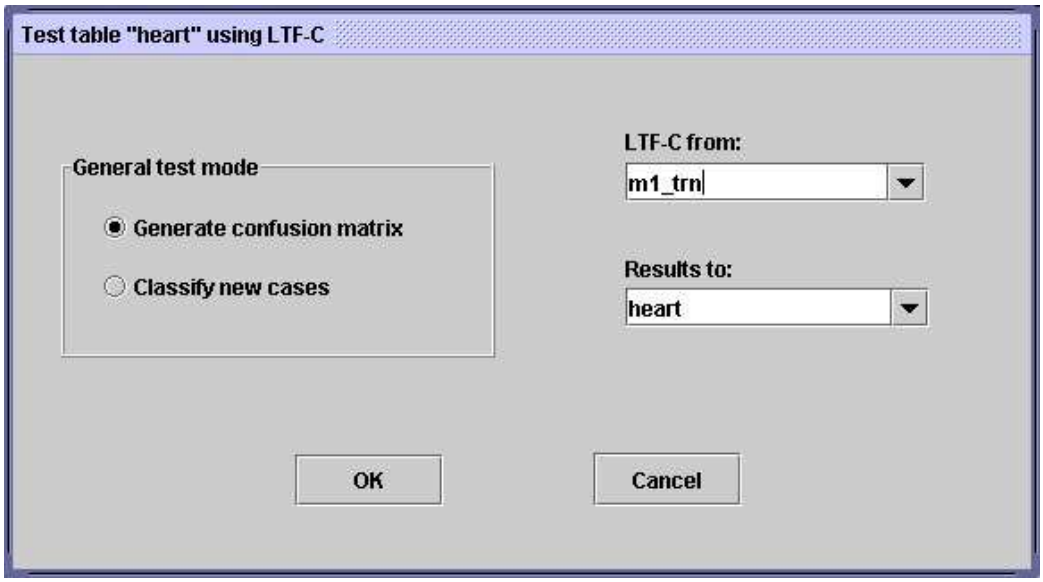
Rysunek 4.9: Opcje budowy klasyfikatora LTF-C

- LTF-C from – klasyfikator LTF-C
- Results to – wybór danych do których stosujemy klasyfikator LTF-C i dla których obliczamy wyniki

4.8 Metoda cross-validation

Metoda cross-validation (patrz np. [12]) polega na wykonaniu wielu testów składających się z uczenia i testowania klasyfikatora. Zbiór danych zostaje podzielony na kilka równych i rozłącznych części. Podczas pojedynczego testu jedną z wyznaczonych części używamy do testowania klasyfikatora, pozostałe zaś do jego budowy. Wykonujemy tyle testów na ile części podzieliliśmy zbiór danych. Końcowy wynik klasyfikacji jest średnią arytmetyczną wyników ze wszystkich wykonanych testów.

Aby użyć tej metody testowania należy wybrać polecenie Classify/Cross-validation method z menu kontekstowego dla tablicy.



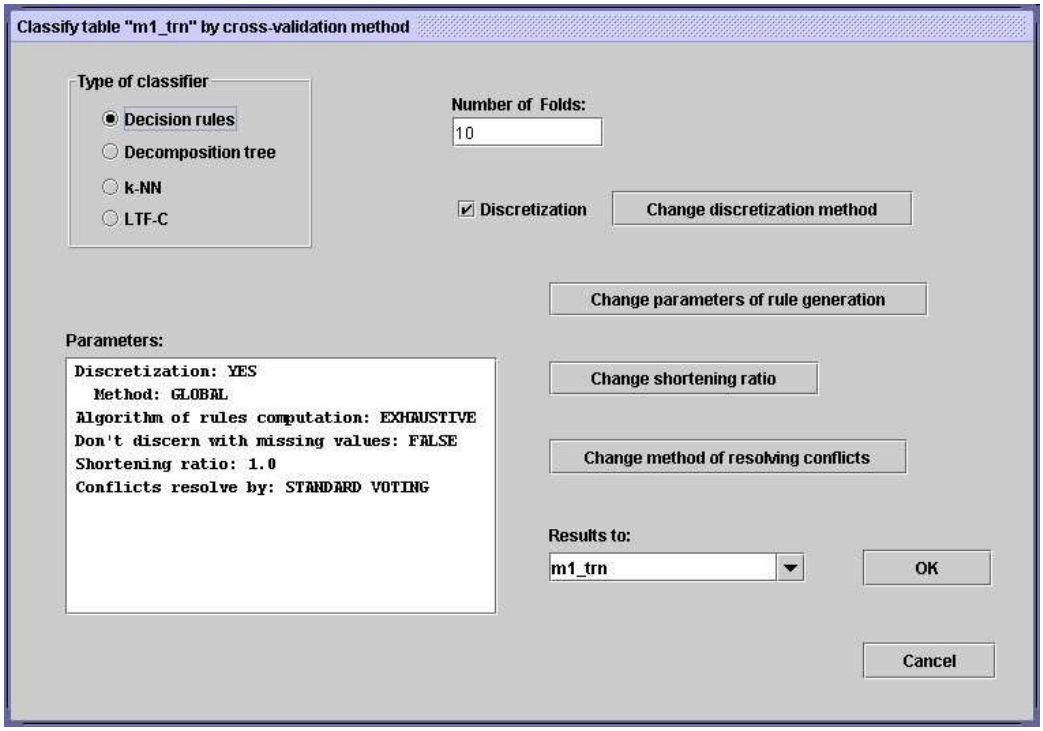
Rysunek 4.10: Opcje klasyfikacji przy pomocy klasyfikatora LTF-C

Przy definiowaniu parametrów tej metody użytkownik ma do dyspozycji następujące opcje (patrz rysunek 4.11):

- Type of classifier – wybór testowanego klasyfikatora:
  - Decision rules – klasyfikacja z użyciem reguł decyzyjnych,
  - Decomposition tree – klasyfikacja przy pomocy drzewa dekompozycji,
  - k-NN – klasyfikacja tabeli przy pomocy algorytmu k najbliższych sąsiadów,
  - LTF-C – klasyfikacja przy pomocy klasyfikatora LTF-C,
- Number of Folds – zdefiniowanie liczby wykonywanych testów w ramach metody cross-validation (liczba ta pokrywa się z liczbą zbiorów na jakie zostanie podzielony zbiór danych)
- Discretization – zaznaczenie tej opcji umożliwi wykonanie wstępnej dyskretyzacji danych (*opcja niedostępna dla klasyfikatora LTF-C*)
- Change discretization method – zmiana metody dyskretyzacji danych, dostępne opcje są analogiczne do opcji dla operacji Generate cuts (*opcja dostępna tylko wtedy, gdy aktywna jest opcja Discretization*)

- Change parameters of ... – zmiana parametrów wybranego klasyfikatora, okienka dialogowe i dostępne w nich opcje są analogiczne do tych jakie zostały opisane dla poszczególnych klasyfikatorów.
- Change shortening ratio – zmiana współczynnika skracania reguł lub drzewa dekompozycji (*opcja dostępna tylko po wybraniu klasyfikacji przy pomocy reguł lub drzewa dekompozycji*)
- Change method of resolving conflicts – wybór metody rozwiązywania konfliktów, analogicznie jak dla reguł i drzew dekompozycji (*opcja dostępna tylko po wybraniu klasyfikacji przy pomocy reguł lub drzewa dekompozycji*)
- Results to – nazwa obiektu w którym umieszczone zostaną uśrednione wyniki testu cross-validation

Dla wygody użytkownika w okienku Parameters znajduje się wykaz aktualnie ustawionych opcji wybranego klasyfikatora (np. rodzaju dyskretyzacji, współczynnika skracania reguł itd.).



Rysunek 4.11: Opcje klasyfikacji przy pomocy metody cross-validation

Rozdział 5

Przykładowe scenariusze pracy z systemem RSES

W tym rozdziale przedstawiamy scenariusze pracy z systemem RSES, które dotyczą kilku standardowych podejść do analizy danych metodami zaimplementowanymi w systemie RSES.

5.1 Scenariusze testowania metodą train-and-test

Jednym z najprostszych scenariuszy postępowania przy analizie danych metodami teorii zbiorów przybliżonych jest przypadek, gdy mamy zbiór danych i chcemy na nim sprawdzić działanie wybranego klasyfikatora metodą *train-and-test*. Metoda ta polega na tym, że wejściowy zbiór danych jest dzielony na dwie rozłączne części. Na jednej z nich, zwanej częścią treningową, uczony jest klasyfikator, po czym następuje testowanie klasyfikatora na drugiej części, zwanej częścią testową. Wynikiem całego procesu są wyniki klasyfikacji na części testowej oraz ewentualnie struktura klasyfikatora otrzymanego dla części treningowej.

Opisany wyżej proces może być parametryzowany rodzajem stosowanego preprocesingu (dyskretyzacja, wypełnianie pustych miejsc itd.) lub rodzajem użytego klasyfikatora (reguły decyzyjne, drzewo dekompozycji itd.).

5.1.1 Klasyfikator regułowy

W podrozdziale rozpatrujemy sytuację, gdy stosujemy metodę train-and-test bez preprocesingu i z użyciem klasyfikatora regułowego. Aby zrealizować opisywany scenariusz, należy z użyciem systemu RSES wykonać następujące

operacje.

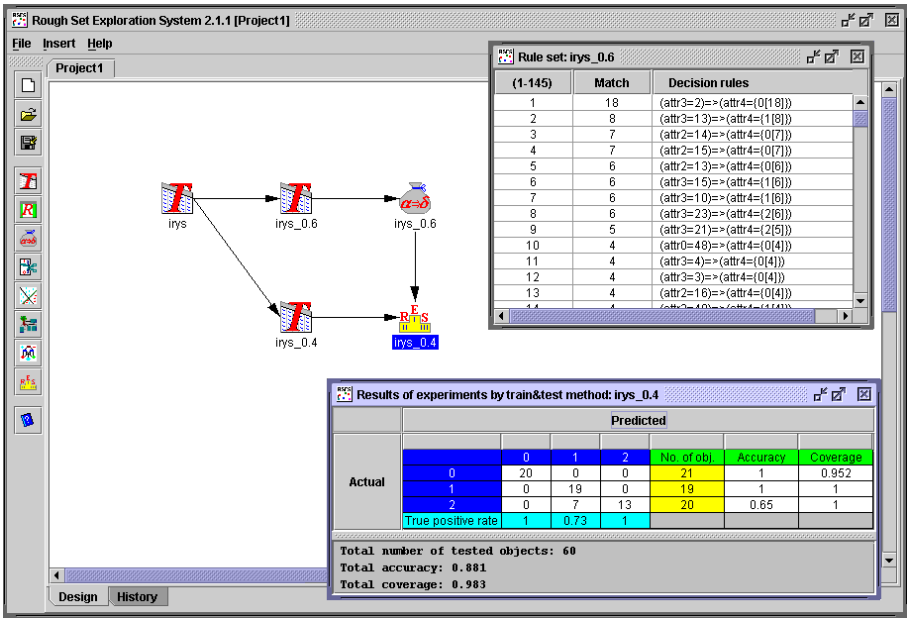
1. Uruchamiamy system RSES i tworzymy nowy projekt
2. Wstawiamy nową tablicę do projektu.
3. Wczytujemy do niej dane (opcja Load z menu kontekstowego); my wybraliśmy plik irys z danych przykładowych w katalogu DATA).
4. Dzielimy tablice na dwie części 60% i 40% (wybieramy z menu kontekstowego tablicy opcję Split in Two, a jako split factor podajemy wartość 0.6). Otrzymujemy w ten sposób dwie nowe tablice: irys_0.6 i irys_0.4. Pierwszej użyjemy do nauki, a drugiej do testowania.
5. Liczymy reguły z tablicy irys_0.6 (otwieramy jej menu kontekstowe, wybieramy opcję Reducts/Rules / Calculate reducts or rules, następnie wybieramy w okienku dialogowym liczenie reguł i na przykład Exhaustive algorithm). Powstaje nowy obiekt – zbiór reguł o nazwie takiej jak tabelka czyli irys_0.6.
6. Testujemy wyliczone reguły na tablicy irys_0.4 (otwieramy menu kontekstowe tej tablicy i wybieramy opcję Classify/Test table using rule set, jako Rules from set ustawiamy irys_0.6, zaś results to na irys_0.4, opcje rozwiązywania konfliktów ustawiliśmy dla naszego przykładu na Standard voting; dodatkowo opcja General test mode ma być ustawiona na Generate confusion matrix). Powstaje nowy obiekt z wynikami o nazwie irys_0.4.
7. Oglądamy wyniki klasyfikacji oraz reguły decyzyjne (dwukrotnie klikamy na obiekcie z wynikami irys_0.4 oraz na obiekcie z regułami irys_0.6)

Na rysunku 5.1 pokazujemy wygląd projektu systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem regułowym.

5.1.2 Klasyfikator regułowy i skalowanie

Poprzedni scenariusz może zostać wzbogacony o możliwość skalowania (dyskretyzacji) tablicy treningowej. Jest to bardzo ważne w przypadku wielu tablic z atrybutami numerycznymi, ponieważ operacja skalowania może znacząco polepszyć wyniki klasyfikacji. Aby zrealizować poprzedni scenariusz wraz ze skalowaniem danych, należy wykonać następujące operacje.

1. Uruchamiamy system RSES i tworzymy nowy projekt.



Rysunek 5.1: Wygląd systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem regułowym

2. Wstawiamy nową tablicę do projektu.

3. Wczytujemy do niej dane (opcja Load z menu kontekstowego); ponownie wybieramy plik irys z danych przykładowych w katalogu DATA).

4. Dzielimy tablice na dwie części 60% i 40% (wybieramy z menu kontekstowego tablicy opcję Split in Two, a jako split factor podajemy wartość 0.6). Otrzymujemy w ten sposób dwie nowe tablice: irys_0.6 i irys_0.4. Pierwszej użyjemy do nauki, a drugiej do testowania.

5. Liczymy zbiór cięć dla tablicy irys_0.6 (z menu kontekstowego tej tablicy wybieramy opcję Discretize/Generate cuts, w okienku dialogowym wybieramy rodzaj cięć, na przykład globalne, oraz podajemy plik z którego mają być liczone cięcia, czyli irys_0.6 – opcja Cuts to). Powstaje nowy obiekt zawierający cięcia o nazwie irys_0.6.

6. Dyskretyzujemy tablicę irys_0.6 (z menu kontekstowego tej tablicy wybieramy opcję Discretize/Discretize table i potwierdzamy, że chcemy użyć zbioru cięć o nazwie irys_0.6). Powstaje nowa tablica o nazwie irys_0.6D.

7. Dyskretyzujemy tablicę irys_0.4 (z menu kontekstowego tej tablicy wybieramy opcję Discretize/Discretize table i potwierdzamy, że chcemy

użyć zbioru cięć o nazwie irys_0.6). Powstaje nowa tablica o nazwie irys_0.4D.

8. Liczymy reguły z tablicy irys_0.6D (otwieramy jej menu kontekstowe, wybieramy opcję Reducts/Rules, następnie wybieramy w okienku dialogowym liczenie reguł, tym razem LEM2 algorithm, w wyniku czego powstaje nowy obiekt – zbiór reguł o nazwie takiej jak tabelka czyli irys_0.6D).

9. Testujemy wyliczone reguły na tablicy irys_0.4D (otwieramy menu kontekstowe tej tablicy i wybieramy opcję Classify/Test table using rule set, jako Rules from set ustawiamy irys_0.6D, zaś results to na irys_0.4D, opcje rozwiązywania konfliktów ustawiliśmy dla naszego przykładu na Standard voting; dodatkowo opcja General test mode ma być ustawiona na Generate confusion matrix). Powstaje nowy obiekt z wynikami o nazwie irys_0.4D.

10. Oglądamy wyniki klasyfikacji oraz cięcia (dwukrotnie klikamy na obiekcie z wynikami irys_0.4D oraz na obiekcie z cięciami irys_0.6).

Na rysunku 5.2 pokazujemy wygląd projektu systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem regułowym i ze skalowaniem.

5.1.3 Drzewo dekompozycji

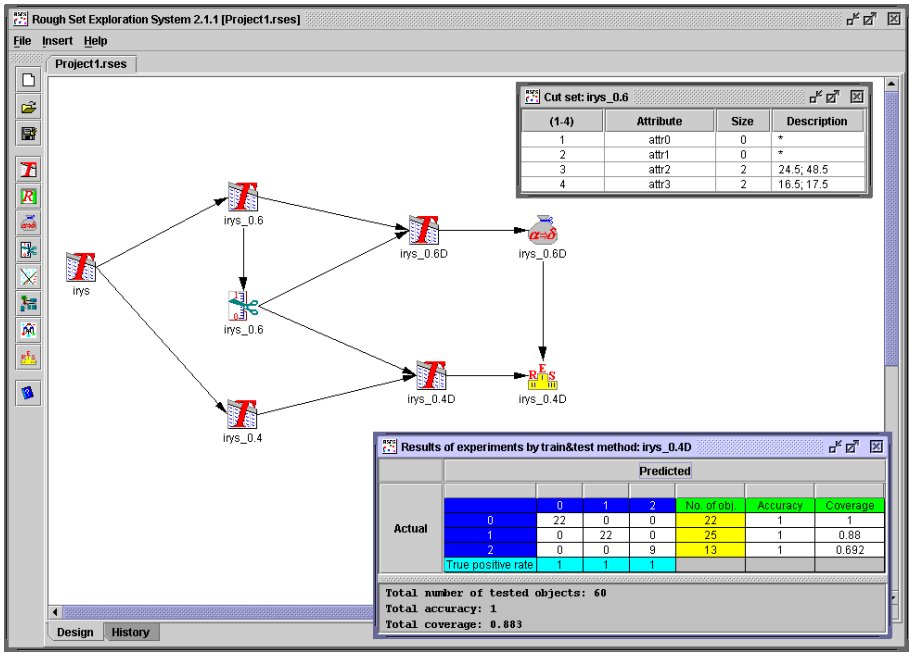
Często w przypadku większych zbiorów danych (powyżej 1000 obiektów), stosowanie tradycyjnych klasyfikatorów regułowych jest utrudnione lub nawet niemożliwe, gdyż złożoność obliczeniowa algorytmów liczących reguły wymaga czasochłonnych obliczeń. Dlatego w systemie RSES zaimplementowano metodę dekompozycji danych, która potrafi tworzyć klasyfikatory nawet dla danych liczących setki tysięcy obiektów.

Poniżej prezentujemy standardowy scenariusz tworzenia i testowania klasyfikatora opartego na drzewie dekompozycji.

1. Uruchamiamy system RSES i tworzymy nowy projekt.

2. Wstawiamy nową tablicę do projektu, która będzie tablicę treningową.

3. Wczytujemy do niej dane (opcja Load z menu kontekstowego) wybierając plik sat_trn z danych przykładowych w katalogu DATA).



Rysunek 5.2: Wygląd systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem regułowym i ze skalowaniem

4. Tworzymy drzewo dekompozycji dla tablicy sat_trn (z menu kontekstowego tablicy sat_trn wybieramy opcję Make decomposition, ustawiamy maksymalny rozmiar liścia na przykład na wartość 500, jako Decomposition tree to ustawiamy tablicę sat_trn włączamy dyskretyzację i ustawiamy parametr Shortening ratio na wartość 1.0). Powstaje nowy obiekt przechowujący drzewo dekompozycji o nazwie sat_trn.

5. Wstawiamy nową tablicę do projektu, która będzie tablicę testową.

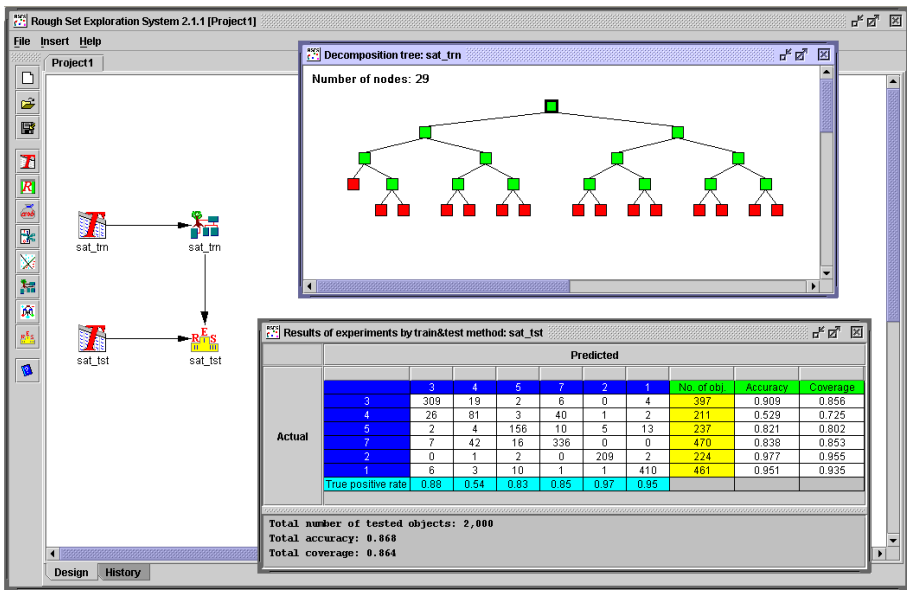
6. Wczytujemy do niej dane (opcja Load z menu kontekstowego) wybierając plik sat_tst z danych przykładowych w katalogu DATA).

7. Testujemy wyliczone drzewo dekompozycji na tablicy sat_tst (otwieramy menu kontekstowe tej tablicy i wybieramy opcję Classify/Test table using decomposition tree, jako Decomposition tree from set ustawiamy sat_trn, zaś results to na sat_tst, opcje rozwiązywania konfliktów ustawiliśmy dla naszego przykładu na Standard voting; dodatkowo opcja General test mode ma być ustawiona na Generate confusion matrix). Powstaje nowy obiekt z wynikami o nazwie sat_tst.

8. Oglądamy wyniki klasyfikacji oraz drzewo dekompozycji (dwukrotnie

klikamy na obiekcie z wynikami sat_tst oraz na obiekcie z drzewem dekompozycji sat_trn)

Na rysunku 5.3 pokazujemy wygląd projektu systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem opartym na drzewie dekompozycji.



Rysunek 5.3: Wygląd systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem opartym na drzewie dekompozycji

5.1.4 Klasyfikator k-NN

Dla wielu zbiorów danych eksperymentalnych bardzo użyteczne mogą być metody konstruowania klasyfikatorów typu k-NN. Dlatego system RSES dostarcza wielu narzędzi do konstruowania tego typu klasyfikatorów. Opierają się one nie tylko na klasycznym podejściu statystycznym, ale są wzbogacone wieloma dodatkowymi metodami i technikami teorii zbiorów przybliżonych (patrz podrozdział 4.6). Oprócz tego, dzięki zastosowaniu wyspecjalizowanych metod dekompozycyjnych, wspomniane metody mogą być stosowane dla stosunkowo dużych zbiorów danych.

Poniżej prezentujemy przykładowy scenariusz testowania klasyfikatora opartego na metodach typu k-NN.

1. Uruchamiamy system RSES i tworzymy nowy projekt.

2. Wstawiamy nową tablicę do projektu, która będzie tablicę treningową.
3. Wczytujemy do niej dane (opcja Load z menu kontekstowego) wybierając plik sat_trn z danych przykładowych w katalogu DATA).
4. Wstawiamy nową tablicę do projektu, która będzie tablicę testową.
5. Wczytujemy do niej dane (opcja Load z menu kontekstowego) wybierając plik sat_tst z danych przykładowych w katalogu DATA).
6. Testujemy tablicę sat_tst używając tablicy sat_trn. W tym celu otwieramy menu kontekstowe tablicy sat_tst i wybieramy opcję Classify/Test table using k-NN, jako Train table from ustawiamy sat_trn, zaś Confusion matrix to na sat_tst; oprócz tego jako aktywne ustawiamy opcje Generate confusion matrix, Metric type->City SVD i Normalization->Range. Natomiast dla przyspieszenia obliczeń (kosztem być może utraty jakości klasyfikatora) aktywizujemy opcje: Attribute weighting->None, Voting->Simple, Numbers of neighbours->1 oraz odznaczamy opcję Search optimal between 1 and ... (mają być obliczenia bez wyszukiwania optymalnej liczby sąsiadów). Powstaje nowy obiekt z wynikami o nazwie sat_tst.
7. Oglądamy wyniki klasyfikacji (dwukrotnie klikamy na obiekcie z wynikami sat_tst)

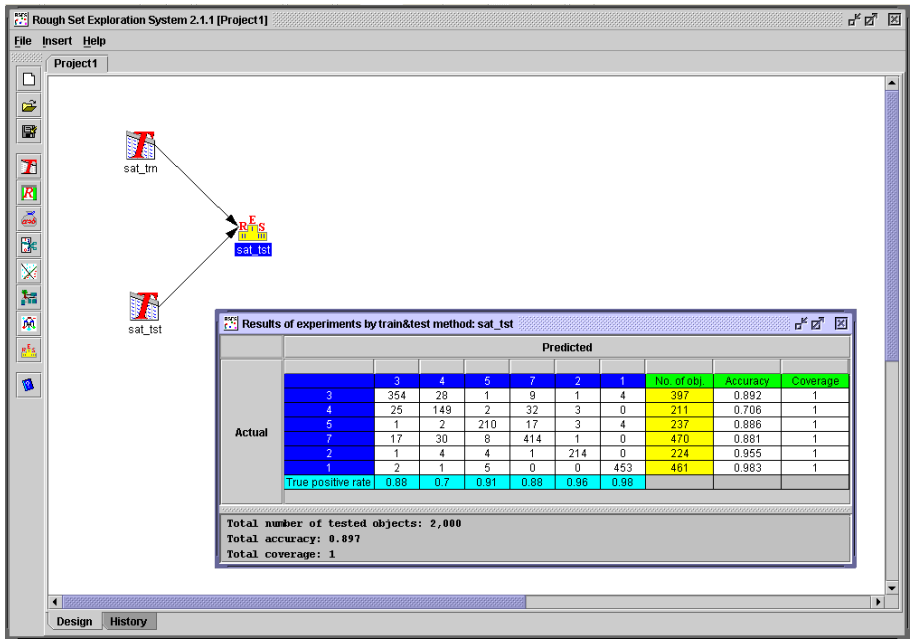
Na rysunku 5.4 pokazujemy wygląd projektu systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem opartym na wybranej metodzie typu k-NN.

5.1.5 Klasyfikator neuronowy LTF-C

Jak pokazują badania eksperymentalne (patrz [24]), klasyfikatory LTF-C (*Local Transfer Function Classifier*) bazujące na sieci neuronowej, pozwalają uzyskać bardzo dobre wyniki dla numerycznych zbiorów danych. Jednocześnie można je traktować, jako alternatywne podejście w stosunku do klasyfikatorów opartych na klasycznych metodach teorii zbiorów przybliżonych. Dlatego też, na gruncie teorii zbiorów przybliżonych klasyfikatorów LTF-C używa się w celach porównawczych oraz do analizy takich zbiorów danych, dla których klasyczne metody z teorii zbiorów przybliżonych nie dają zadawalających rezultatów.

Poniżej prezentujemy przykładowy scenariusz tworzenia i testowania klasyfikatora opartego na metodach typu LTF-C.

1. Uruchamiamy system RSES i tworzymy nowy projekt.



Rysunek 5.4: Wygląd systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem opartym na metodzie typu k-NN

2. Wstawiamy nową tablicę do projektu, która będzie tablicę treningową.

3. Wczytujemy do niej dane (opcja Load z menu kontekstowego) wybierając plik sat_trn z danych przykładowych w katalogu DATA).

4. Tworzymy klasyfikator LTF-C dla tablicy sat_trn (z menu kontekstowego tablicy sat_trn wybieramy opcję Create LTF-C, jako LTF-C too ustawiamy tablicę sat_trn, zaznaczamy opcje Use default training parameters oraz Normalize each numeric attribute. Powstaje nowy obiekt przechowujący klasyfikator LTF-C o nazwie sat_trn.

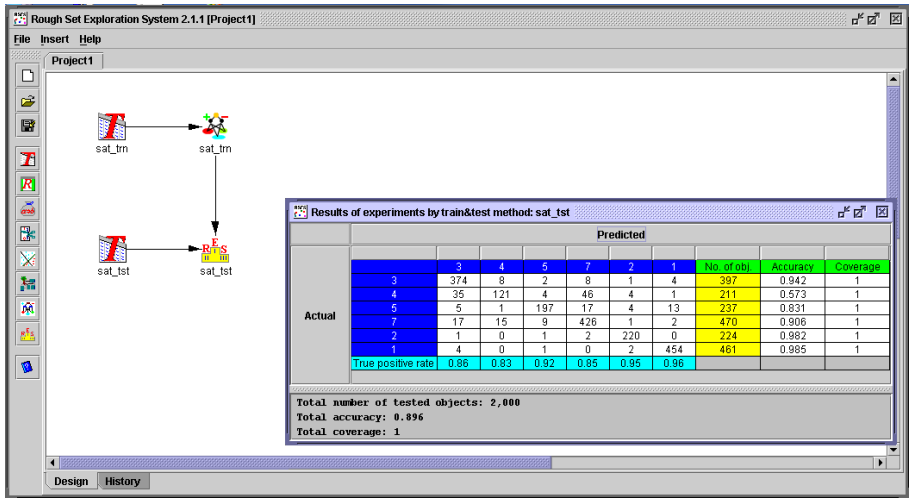
5. Wstawiamy nową tablicę do projektu, która będzie tablicą testową.

6. Wczytujemy do niej dane (opcja Load z menu kontekstowego) wybierając plik sat_tst z danych przykładowych w katalogu DATA).

7. Testujemy klasyfikator LTF-C na tablicy sat_tst (otwieramy menu kontekstowe tej tablicy i wybieramy opcję Classify/Test table using LTF-C, jako LTF-C from ustawiamy sat_trn, zaś Results to na sat_tst. Powstaje nowy obiekt z wynikami o nazwie sat_tst.

8. Oglądamy wyniki klasyfikacji (dwukrotnie klikamy na obiekcie z wynikami sat_tst)

Na rysunku 5.5 pokazujemy wygląd projektu systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem opartym na metodzie typu LTF-C.



Rysunek 5.5: Wygląd systemu RSES po zakończeniu wykonywania scenariusza train-and-test z klasyfikatorem opartym na metodzie typu LTF-C

5.2 Scenariusze testowania metodą cross-validation

Dla zbiorów danych liczących poniżej 1000 obiektów, najczęściej stosowaną metodą testowania klasyfikatorów jest metoda *cross-validation* (patrz podrozdział 4.8). Metoda ta polega na tym, że wejściowy zbiór danych jest dzielony na kilka równych i rozłącznych części. Podczas pojedynczego testu jedną z wyznaczonych części używamy do testowania klasyfikatora, pozostałe zaś do jego budowy. Wykonujemy tyle testów na ile części podzieliliśmy zbiór danych. Końcowy wynik klasyfikacji jest średnią arytmetyczną wyników ze wszystkich wykonanych testów.

Poniżej prezentujemy przykładowy scenariusz tworzenia i testowania klasyfikatora opartego na regułach decyzyjnych metodą cross-validation, przy czym jako klasyfikator można tutaj użyć dowolnego typu klasyfikatora dostępnego w systemie RSES.

1. Uruchamiamy system RSES i tworzymy nowy projekt.

2. Wstawiamy nową tablicę do projektu.

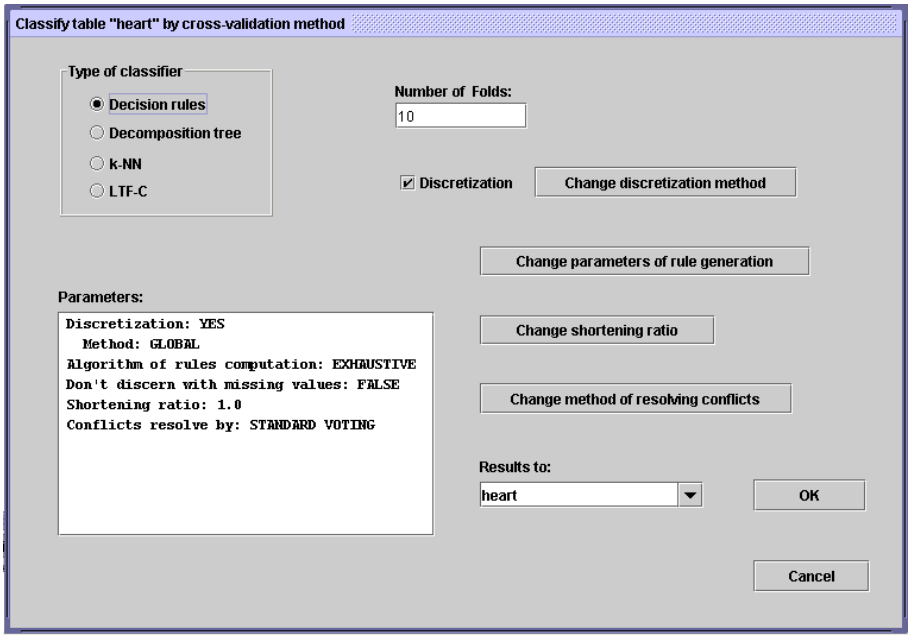
3. Wczytujemy do niej dane (opcja Load z menu kontekstowego) wybierając plik heart z danych przykładowych w katalogu DATA).

4. Testujemy tablicę heart metodą cross-validation. W tym celu otwieramy menu kontekstowe tablicy heart i wybieramy opcję Classify/Cross validation method, jako Type of classifier ustawiamy Decision rules, wartość Number of folds ustawiamy na 10, jako metodę dyskretyzacji ustawiamy metodę globalną (opcja Discretization + przycisk Change discretization method), za pomocą przycisku Change parameters of rule generation ustawiamy metodę wyczerpującą liczenia reguł (wszystkie reguły), za pomocą przycisku Change shortening ration współczynnik skracania ustawiamy na 1.0, za pomocą przycisku Change method of resolving conflicts ustawiamy standardową metodę rozstrzygania konfliktów pomiędzy regułami i wreszcie wartość pola Results to ustawiamy na heart. Ustawione parametry można sprawdzić w polu Parameters. Wygląd okienka dialogowego z parametrami ustawionymi według powyższych wskazówek można zobaczyć na rysunku 5.6. Po pewnym czasie obliczeń powstaje nowy obiekt z wynikami o nazwie heart.

5. Teraz można już obejrzeć wyniki klasyfikacji (dwukrotnie klikamy na obiekcie z wynikami heart)

5.3 Scenariusze ekspertowego generowania decyzji

Wszystkie opisane wcześniej scenariusze analizy danych, zarówno do celów tworzenia jak i testowania klasyfikatorów wykorzystywały tablice danych zawierające atrybut decyzyjny. Tymczasem w wielu praktycznych przypadkach analizy danych spotykamy się z taką sytuacją, że w prawdzie mamy pewien zbiór danych zawierający atrybut decyzyjny, co jest niezbędne dla wyuczenia rozpoznawania wartości tego atrybutu przez klasyfikator, jednak istnieje także potrzeba klasyfikowania obiektów dla których wartość atrybutu decyzyjnego nie jest znana. Innymi słowy, trzeba klasyfikować obiekty z takich tablic, w których brak jest atrybutu decyzyjnego. System RSES może być stosowany w takiej sytuacji i jego działanie polega na tym, że na podstawie wygenerowanego wcześniej klasyfikatora, można wygenerować wartość



Rysunek 5.6: Przykładowy wygląd okienka dialogowego z parametrami testowania klasyfikatora metodą cross-validation.

decyzji dla każdego obiektu (sklasyfikować każdy obiekt), a otrzymane wartości decyzji są dopisywane do tablicy z obiektami testowymi jako dodatkowy atrybut umieszczony na końcu wszystkich atrybutów.

Aby móc zaprezentować powyższe możliwości systemu RSES, potrzebne są dwie tablice danych: jedna z atrybutem decyzyjnym i druga bez atrybutu decyzyjnego. Wtedy będzie można utworzyć klasyfikator na podstawie pierwszej tablicy i wygenerować wartości decyzji dla drugiej tablicy, dodając je jako dodatkowy atrybut w drugiej tablicy. Aby to osiągnąć, w poniższym scenariuszu wykorzystujemy fakt, że w systemie RSES można wybierać podtablice z danej tablicy danych poprzez wyselekcjonowanie atrybutów z tablicy wejściowej.

1. Uruchamiamy system RSES i tworzymy nowy projekt.
2. Wstawiamy nową tablicę do projektu, która będzie tablicę treningową.
3. Wczytujemy do niej dane (opcja Load z menu kontekstowego) wybierając plik heart z danych przykładowych w katalogu DATA). W ten sposób otrzymujemy tablicę dla której będzie tworzony klasyfikator.
4. Z tablicy heart wybieramy podtablicę składającą się tylko z atrybutów warunkowych. W tym celu otwieramy menu kontekstowe tablicy heart

i wybieramy opcję Select subtable a w pojawiającym się okienku dialogowym wybieramy wszystkie atrybuty oprócz atrybutu attr13, który jest atrybutem decyzyjnym. Powstaje nowy obiekt z tablicą o nazwie heart_SUB, który przechowuje tablicę o zawartości identyczne z tablicą heart, ale bez kolumny decyzyjnej.

5. Liczymy reguły z tablicy heart (otwieramy jej menu kontekstowe, wybieramy opcję Reducts/Rules / Calculate reducts or rules, następnie wybieramy w okienku dialogowym liczenie reguł i na przykład Exhaustive algorithm). Powstaje nowy obiekt – zbiór reguł o nazwie takiej jak tabelka czyli heart.

6. Testujemy wyliczone reguły na tablicy heart_SUB, ale w taki sposób, że nie generujemy macierzy błędów, ale generujemy dla każdego obiektu wartość decyzji i umieszczamy ją w nowym, wstawionym na końcu tabeli atrybucie. W tym celu otwieramy menu kontekstowe tablicy heart_SUB i wybieramy opcję Classify/Test table using rule set, jako Rules from set ustawiamy heart, zaś opcje rozwiązywania konfliktów ustawiamy na Standard voting. Dodatkowo opcja General test mode musi być ustawiona na Classify new cases, dzięki czemu nie trzeba ustawiać zawartości pola Results to, bo staje się ono nieaktywne. W wyniku naciśnięcia przycisku OK obiekty z tablicy heart_SUB zostaną sklasyfikowane, a wygenerowane dla nich wartości decyzji będą dopisane w dodatkowym atrybucie, którego nazwa będzie taka sama jak atrybutu decyzyjnego w tablicy heart.

Na koniec zauważmy, że po zakończeniu wykonywania powyższego scenariusza, kolumna decyzyjna dopisana do tablicy heart_SUB powinna być identyczna z tą, która znajduje się w tablicy heart. Wynika to z faktu, że tablica heart jest niesprzeczna i wygenerowano dla niej wszystkie reguły, dające pokrycie wszystkich obiektów z tablicy heart a zatem i z tablicy heart_SUB.

Dodatek A

Wybrane formaty plików systemu RSES 2.1

Poniższy dodatek zawiera opisy najważniejszych formatów plików tekstowych jakich system RSES 2.1 używa do zapisywania i odczytywania informacji.

Aby prezentowane listingi plików były czytelniejsze dodano w nich dodatkowe spacje. Spacje te nie mają wpływu na odczyt danych przez system RSES 2.1.

A.1 Zbiory danych

Zbiory danych przeznaczone do analizy przez system RSES 2.1 powinny być w postaci takiej jak na rysunku A.1.

W nazwach tabel, atrybutów oraz wartości atrybutów akceptowalne są spacje; jednak nazwa ze spacją musi być umieszczona w cudzysłowie lub pomiędzy znakami apostrofu. Na przykład atrybut medyczny o dwusłowowej nazwie *serum cholestoral* w systemie RSES powinien być nazwany jako *"serum cholestoral"* (w cudzysłowie) lub *'serum cholestoral'* (pomiędzy znakami apostrofu). Aby były dobrze wyświetlane w interfejsie, polskie znaki w nazwach tabel, atrybutów i wartości atrybutów powinny być kodowane w standardzie ISO-8859-2.

Przeanalizujmy przykład z rysunku A.1. Pierwszy wiersz o treści:

```
TABLE therapy
```

zawiera informację o nazwie tabeli. Pozwala to na wygodne zarządzanie wieloma zbiorami danych w ramach jednego projektu. Kolejny wiersz o treści:

```
ATTRIBUTES 5
```

```
TABLE therapy
ATTRIBUTES 5
 temperature numeric 1
 headache numeric 0
 cough symbolic
 catarrh symbolic
 disease symbolic
OBJECTS 4
 38.7    7       no  no  angina
 38.3    MISSING yes yes influenza
 MISSING 3       no  no  cold
 36.7    1       no  no  healthy
```

Rysunek A.1: Przykładowy zbiór danych w formacie systemu RSES 2.1

zawiera informację o liczbie wszystkich kolumn (atrybutów) w danych. Po tym wierszu następuje opis atrybutów:

```
temperature numeric 1
headache numeric 0
cough symbolic
catarrh symbolic
disease symbolic
```

Każdy wiersz opisuje jeden atrybut. Opis ten składa się z nazwy atrybutu i informacji o jego typie. Typ `symbolic` oznacza atrybut nominalny o wartościach symbolicznych. Typ `numeric` oznacza atrybut numeryczny. W przypadku atrybutu numerycznego podawana jest także precyzja, która determinuje liczbę miejsc znaczących po przecinku. Na przykład `numeric 1` opisuje atrybut numeryczny z jednym miejscem po przecinku, zaś `numeric 0` – atrybut numeryczny o wartościach całkowitych.

Po opisie atrybutów znajdujemy informację o liczbie obiektów w tabelce.

```
OBJECTS 4
```

Po tej informacji znajdują się dane[1] czyli kolejne wiersze wartości atrybutów. Kolejność atrybutów odpowiada kolejności w jakiej zostały wcześniej

[1]W podanym przykładzie dodano dodatkowe spacje dla czytelności prezentowanej tabelki. Spacje te jednak nie są konieczne i nie mają one wpływu na poprawność odczytania i przetwarzania tabelki.

opisane.

```
38.7     7        no   no   angina
38.3     MISSING  yes  yes  influenza
MISSING  3        no   no   cold
36.7     1        no   no   healthy
```

Symbol `MISSING` oznacza brakującą wartość, czyli miejsce w tabeli, które jest puste, którego wartości nie znamy. Takie miejsca mogą być później uzupełniane. Brakujące wartości mogą być również oznaczane jako `NULL` lub `'?'`.

A.2 Zbiory reduktów

Obecnie przyjrzymy się w jaki sposób system RSES 2.1 zapisuje do pliku tekstowego zbiory reduktów. Na rysunku A.2 jest pokazany przykładowy zbiór reduktów w formacie systemu RSES 2.1. Przeanalizujmy ten przykład.

```
REDUCTS (5)
{ temperature, headache } 1.0
{ headache, cough } 1.0
{ headache, catarrh } 1.0
{ cough } 0.25
{ catarrh } 0.25
```

Rysunek A.2: Przykładowy zbiór reduktów w formacie systemu RSES 2.1

Pierwszy wiersz o treści:

```
REDUCTS (5)
```

zawiera informację o liczbie reduktów. W tym przykładzie mamy pięć reduktów.

Kolejne wiersze zawierają wypisane redukty (po jednym redukcie w każdym wierszu).

```
{ temperature, headache } 1.0
{ headache, cough } 1.0
{ headache, catarrh } 1.0
{ cough } 0.25
{ catarrh } 0.25
```

Opis każdego reduktu składa się z listy nazw atrybutów oddzielonych przecinkami i zawartych w nawiasach `{` i `}` oraz z liczby określającej obszar pozytywny przy obcięciu tablicy danych do tego reduktu.

Na powyższym przykładzie pierwszy redukt składa się z dwóch atrybutów: `temperature` i `headache`. Wartość współczynnika określającego obszar pozytywny dla tego atrybutu wynosi 1.0 co oznacza, że tablica obcięta do atrybutów z tego reduktu + oryginalny atrybut decyzyjny jest niesprzeczna. Podobnie interpretujemy kolejne wiersze.

A.3 Zbiory reguł

Obecnie przyjrzymy się w jaki sposób system RSES 2.1 zapisuje do pliku tekstowego zbiory reguł decyzyjnych. Na rysunku A.3 jest pokazany przykładowy zbiór reguł w formacie systemu RSES 2.1. Przeanalizujmy ten przykład.

Pierwszy wiersz o treści:

```
RULE_SET Demo
```

zawiera nazwę zbioru reguł. Umożliwia to zarządzanie wieloma różnymi zbiorami reguł w ramach jednego projektu.

Kolejne wiersze zawierają informacje o atrybutach i ich typie. Informacje te są niezbędne by system mógł sprawdzić zgodność danych ze stosowanymi na nich regułami. Znaczenie poszczególnych elementów tego opisu jest takie samo jak w przypadku tablicy z danymi (patrz A.1).

```
ATTRIBUTES 5
 temperature numeric 1
 headache numeric 0
 cough symbolic
 catarrh symbolic
 disease symbolic
```

Po opisie atrybutów znajduje się lista możliwych wartości atrybutu decyzyjnego (ostatni atrybut w tabeli danych).

```
DECISION_VALUES 4
angina
influenza
cold
healthy
```

```
RULE_SET Demo
ATTRIBUTES 5
 temperature numeric 1
 headache numeric 0
 cough symbolic
 catarrh symbolic
 disease symbolic
DECISION_VALUES 4
angina
influenza
cold
healthy
RULES 16
(temperature=38.7)&(headache=7)=>(disease=angina[1]) 1
(temperature=38.3)&(headache=MISSING)=>
(disease=influenza[1]) 1
(temperature=MISSING)&(headache=3)=>(disease=cold[1]) 1
(temperature=36.7)&(headache=1)=>(disease=healthy[1]) 1
(headache=7)&(cough=no)=>(disease=angina[1]) 1
(headache=MISSING)&(cough=yes)=>(disease=influenza[1]) 1
(headache=3)&(cough=no)=>(disease=cold[1]) 1
(headache=1)&(cough=no)=>(disease=healthy[1]) 1
(headache=7)&(catarrh=no)=>(disease=angina[1]) 1
(headache=MISSING)&(catarrh=yes)=>(disease=influenza[1]) 1
(headache=3)&(catarrh=no)=>(disease=cold[1]) 1
(headache=1)&(catarrh=no)=>(disease=healthy[1]) 1
(cough=no)=>(disease={angina[1],cold[1],healthy[1]}) 3
(cough=yes)=>(disease=influenza[1]) 1
(catarrh=no)=>(disease={angina[1],cold[1],healthy[1]}) 3
(catarrh=yes)=>(disease=influenza[1]) 1
```

Rysunek A.3: Przykładowy zbiór reguł w formacie systemu RSES 2.1

Pierwszy wiersz zawiera informację o liczbie możliwych wartości atrybutu decyzyjnego, zaś w kolejnych wierszach wypisane są wszystkie możliwe wartości.

Pozostała część pliku poświęcona jest opisowi reguł, przy czym pierwszy

wiersz opisu reguł o treści:

```
RULES 16
```

zawiera informację o ich liczbie. W tym przykładzie mamy 16 reguł.

Kolejne wiersze zawierają wypisane reguły (po jednej regule w każdym wierszu).

```
(temperature=38.7)&(headache=7)=>(disease=angina[1]) 1
(temperature=38.3)&(headache=MISSING)=>
(disease=influenza[1]) 1
(temperature=MISSING)&(headache=3)=>(disease=cold[1]) 1
(temperature=36.7)&(headache=1)=>(disease=healthy[1]) 1
...
(catarrh=no)=>(disease={angina[1],cold[1],healthy[1]}) 3
...
```

W naszym przykładzie pierwsza reguła mówi, że jeśli pacjent ma temperaturę 38.7C, oraz wartość atrybutu `headache` wynosi 7, to choroba powinna być rozpoznana jako `angina` i wspiera ją jeden przypadek. Wsparcie reguły, czyli liczba przykładów ze zbioru treningowego, które ją spełniają (pasują do poprzednika reguły), zapisane jest na końcu wiersza opisującego regułę.

W przypadku reguł z uogólnioną decyzją składającą się z kilku wartości decyzji, tak jak poniżej:

```
...
(catarrh=no)=>(disease={angina[1],cold[2],healthy[1]}) 4
...
```

podawane jest wsparcie łączne, zaś wsparcie rozbite na poszczególne wartości decyzji podawane jest w nawiasach kwadratowych, na przykład: `angina[1]` oznacza, że w zbiorze treningowym był dokładnie jeden przypadek spełniający lewą część formuły i posiadający decyzję `angina`.

A.4 Zbiory cięć

Teraz opiszemy w jaki sposób system RSES 2.1 zapisuje do pliku tekstowego zbiory cięć. Na rysunku A.4 jest pokazany przykładowy zbiór cięć w formacie systemu RSES 2.1. Przeanalizujmy ten przykład.

Pierwszy wiersz o treści:

```
CUT_SET demo_global
```

```
CUT_SET demo_global
ATTRIBUTES 4
INCLUDED_SYMBOLIC false
temperature numeric 1
[ 38.5 ]
headache numeric 0
[ 5.0 ]
cough symbolic
[  ]
catarrh symbolic
[  ]
```

Rysunek A.4: Przykładowy zbiór cięć w formacie systemu RSES 2.1

zawiera informację o nazwie zbioru cięć.

Drugi wiersz:

```
ATTRIBUTES 4
```

mówi o liczbie atrybutów w zbiorze danych, na którym zostały wyznaczone cięcia.

Trzeci wiersz:

```
INCLUDED_SYMBOLIC false
```

zawiera informację o tym, czy grupowano atrybuty symboliczne. Opcja ta jest dostępna w przypadku wyboru metody lokalnej poszukiwania cięć.

Kolejne wiersze zawierają informacje o nazwie atrybutu, jego typie i cięcia na tym atrybucie. Dla każdego atrybutu są przeznaczone dwa wiersze. Pierwszy opisuje atrybut, drugi cięcia.

```
temperature numeric 1
[ 38.5 ]
headache numeric 0
[ 5.0 ]
cough symbolic
[  ]
catarrh symbolic
[  ]
```

Pusta para nawiasów `[ ]` oznacza, że na danym atrybucie nie ma żadnego cięcia.

W przypadku atrybutów numerycznych jako cięcie rozumiemy wartość, która dzieli wartości tego atrybutu na dwa zbiory. Przy pomocy kilku takich cięć (wartości atrybutu) możemy podzielić zbiór wartości atrybutu na kilka podzbiorów.

Na przykład atrybut `temperature` został podzielony na dwa zbiory. Jeden z nich to wartości mniejsze niż 38.5, zaś drugi – większe niż 38.5.

W przypadku atrybutów symbolicznych, możemy rozpatrywać podziały tylko dla metody lokalnego poszukiwania cięć przy zaznaczonej opcji Include symbolic attributes. W takim przypadku może nastąpić podział wartości atrybutu na dwa zbiory. Każdy z tych zbiorów wartości jednoznacznie reprezentuje to cięcie. Dlatego aby opisać cięcie na atrybucie symbolicznym wystarczy podać podzbiór wartości tego atrybutu. Na rysunku A.5 prezentujemy zbiór cięć wygenerowany w oparciu o tę samą tablicę co uprzednio, lecz tym razem metodą lokalną z dopuszczeniem podziałów na atrybutach symbolicznych.

```
CUT_SET demo_local
ATTRIBUTES 4
INCLUDED_SYMBOLIC true
temperature numeric 1
[ 37.5 ]
headache numeric 0
[ 2.0 5.0 ]
cough symbolic
[ { MISSING,no } ]
catarrh symbolic
[ ]
```

Rysunek A.5: Przykładowy zbiór cięć dla metody lokalnej z cięciami na atrybutach symbolicznych w formacie systemu RSES 2.1

Jak widzimy atrybut `cough` został podzielony na dwa zbiory wartości. Pierwszy to wartości `MISSING` i `no`, drugi zaś zawiera wszystkie pozostałe wartości tego atrybutu atrybutu.

Powyższy przykład pokazuje również, że metoda lokalna może dać nam inne cięcia niż metoda globalna. W przypadku atrybutu `temperature` zmienił

się punkt podziału, zaś atrybut `headache` został podzielony nie na dwa lecz na trzy zbiory wartości.

A.5 Kombinacje liniowe

Przyjrzyjmy się w jaki sposób system RSES 2.1 zapisuje do pliku tekstowego zbiory kombinacji liniowych. Na rysunku A.6 jest pokazany przykładowy zbiór kombinacji liniowych formacie systemu RSES 2.1. Przeanalizujmy ten przykład.

```
DIRECTIONS (4)
temperature*0.707+headache*0.707
temperature*0.705+headache*0.705+disease*(-0.062)
temperature*0.707+headache*0.707
temperature*0.696+headache*0.696+disease*0.174
```

Rysunek A.6: Przykład pliku z kombinacjami liniowymi

Pierwszy wiersz o treści:

```
DIRECTIONS (4)
```

zawiera informację o tym, że ten zbiór opisuje kombinacje liniowe (inaczej mówiąc: kierunki główne), oraz informację o ich liczbie. W tym przykładzie mamy 4 kombinacje liniowe.

Kolejne wiersze zawierają wypisane kombinacje liniowe (po jednej kombinacji w każdym wierszu).

```
temperature*0.707+headache*0.707
temperature*0.705+headache*0.705+disease*(-0.062)
temperature*0.707+headache*0.707
temperature*0.696+headache*0.696+disease*0.174
```

Opis każdej z kombinacji liniowej składa się ze wzoru opisującego nowy kierunek. Taki wzór może być użyty do stworzenia nowego atrybutu. System RSES 2.1 umożliwia użycie kombinacji liniowych do tworzenia nowych atrybutów w zbiorze danych.

A.6 Klasyfikator LTF-C

Plik zawierający opis klasyfikatora LTF-C jest dosyć obszerny dlatego przedstawimy go w kilku fragmentach. Wiele szczegółów dotyczących klasyfikatora LTF-C i formatu jego zapisu można znaleźć w punkcie 3.7.

Cały opis klasyfikatora LTF-C możemy podzielić na części: nagłówek, parametry sieci, opis neuronów oraz dodatkowy opis atrybutów analizowanej tablicy.

Nagłówek składa się z dwóch wierszy. Pierwszy zawiera nazwę klasyfikatora, tutaj nosi on nazwę `demo`. Drugi wiersz zawiera informację o liczbie wierszy opisujących parametry sieci (w tym przykładzie jest to liczba 62).

```
LTF_CLASSIFIER demo
62
```

Dalej znajdujemy podstawowe informacje o sieci umieszczone na początku pliku dla wygody użytkownika. Informacje te są wypisane jako komentarz i w dalszej części pojawią się ponownie.

```
$-------------------------------------------------
$   LTF-C neural network
$   Number of inputs:   2
$   Number of outputs:  4
$   Number of neurons:  1
$-------------------------------------------------
```

Wartość każdego parametru poprzedzona jest znakiem `@` i nazwą parametru. Spośród kilkunastu parametrów, wypisanych w pliku (od `@FileType` do `@SumCycles`), istotne dla użytkownika są tylko `@EtaUse` i `@UseTr`

```
@FileType  net
...
@EtaUse    0.013
@UseTr     0.013
...
@SumCycles 120
```

Dalej następuje opis kolejnych neuronów. Szczegółowo opisano to w podrozdziale 3.7.

```
@<Neuron0
 @Class    1
 @Life     0
 ...
@>
```

Na końcu pliku znajdują się następujące informacje o danych: informacja o atrybucie decyzyjnym (nazwa i typ), informacja o wartościach atrybutu decyzyjnego (liczba wartości i ich nazwy), informacja o normalizacji atrybutów (tak->true albo nie->false), wartości średnie atrybutów numerycznych (z liczbą tych atrybutów na początku) oraz odchylenia standardowe wartości atrybutów numerycznych (z liczbą tych atrybutów na początku).

```
disease symbolic
DECISION_VALUES 4
angina
influenza
cold
healthy
true
4
37.73333333333333
3.0
0.0
0.0
4
0.8617811013631386
2.1908902300206643
0.0
0.0
```

A.7 Wyniki klasyfikacji

Obecnie przyjrzymy się w jaki sposób system RSES 2.1 zapisuje do pliku tekstowego wyniki klasyfikacji. Na rysunku A.7 są pokazane przykładowe wyniki klasyfikacji zapisane w formacie systemu RSES 2.1. Przeanalizujmy ten przykład.

Pierwszy wiersz zawiera informację o zawartości zbioru.

```
TEST RESULTS:
```

```
TEST RESULTS:
 Global coverage=0.8333333333333334
 Global accuracy=0.8333333333333334
Decision classes:
 angina influenza cold healthy
Confusion matrix:
 0.0 0.0 0.0 0.0
 0.0 0.6666666666666666 0.0 0.0
 0.0 0.3333333333333333 0.0 0.0
 0.0 0.0 0.0 0.6666666666666666
True positive rates for decision classes:
 0.0 0.5 0.0 0.6666666666666666
Accuracy for decision classes:
 0.0 0.6666666666666666 0.0 0.6666666666666666
Coverage for decision classes:
 0.0 0.66666666666666 0.33333333333333 0.66666666666666
```

Rysunek A.7: Przykładowe wyniki klasyfikacji zapisane w formacie systemu RSES 2.1

Dwa kolejne wiersze zawierają globalne wartości statystyk `coverage` i `accuracy`.

```
Global coverage=0.8333333333333334
Global accuracy=0.8333333333333334
```

Następnie znajdujemy wypisane wszystkie możliwe wartości kolumny decyzyjnej.

```
Decision classes:
 angina influenza cold healthy
```

Kolejne wiersze zawierają pełną confusion matrix.

```
Confusion matrix:
 0.0 0.0 0.0 0.0
 0.0 0.6666666666666666 0.0 0.0
 0.0 0.3333333333333333 0.0 0.0
 0.0 0.0 0.0 0.6666666666666666
```

Na końcu są podawane statystyki z podziałem na klasy decyzyjne. Każda ze statystyk jest opisana przez dwa wiersze. Pierwszy opisuje nazwę i przeznaczenie statystyki, natomiast drugi – wartości dla kolejnych klas decyzyjnych (w kolejności takiej jak zostały wymienione w opisywanym pliku).

```
True positive rates for decision classes:
 0.0 0.5 0.0 0.6666666666666666
Accuracy for decision classes:
 0.0 0.6666666666666666 0.0 0.6666666666666666
Coverage for decision classes:
 0.0 0.66666666666666 0.33333333333333 0.66666666666666
```

Bibliografia

[1] J.G. Bazan, Son H. Nguyen, Trung T. Nguyen, A. Skowron and J. Stepaniuk (1998): Decision rules synthesis for object classification. In: E. Orowska (ed.), Incomplete Information: Rough Set Analysis, Physica – Verlag, Heidelberg, pp. 23–57.

[2] J. G. Bazan (1998): A Comparison of Dynamic and non-Dynamic Rough Set Methods for Extracting Laws from Decision Table. In: L. Polkowski, A. Skowron (eds.), Rough Sets in Knowledge Discovery, Physica – Verlag, Heidelberg, pp. 321–365.

[3] J. G. Bazan (1998): Metody wnioskowań aproksymacyjnych dla syntezy algorytmów decyzyjnych. Ph. D. thesis, supervisor A. Skowron, Warsaw University, pp. 1–179. (In Polish only)

[4] J. Bazan, M. Szczuka (2000): RSES and RSESlib – A Collection of Tools for Rough Set Computations (Postscript). Extended version of paper submitted to RSCTC'2000

[5] J. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, and J. Wróblewski(2000): Rough set algorithms in classification problem. In L. Polkowski, S. Tsumoto, and T. Lin, editors, Rough Set Methods and Applications, Physica-Verlag, Heidelberg New York, pp. 49–88.

[6] J. Bazan, M. Szczuka, J. Wróblewski (2002): A New Version of Rough Set Exploration System, Lecture Notes in Artificial Intelligence 2475, 397–404, Berlin, Heidelberg: Springer-Verlag.

[7] J. Bazan, M. Szczuka, A. Wojna, M. Wojnarski (2004): On Evolution of Rough Set Exploration System, Lecture Notes in Artificial Intelligence 3066, 592–601, Berlin, Heidelberg: Springer-Verlag.

[8] G. Gora, A. Wojna (2002): RIONA: A Classifier Combining Rule Induction and k-NN Method with Automated Selection of Optimal Neighbourhood, Proceedings of the Thirteenth European Conference on

Machine Learning, ECML 2002, Helsinki, Finland, Lecture Notes in Artificial Intelligence, 2430, Springer-Verlag, pp. 111–123

[9] G. Gora, A. Wojna (2002): RIONA: A New Classification System Combining Rule Induction and Instance-Based Learning, Fundamenta Informaticae, 51(4), pp. 369–390

[10] J. Grzymała-Busse (1997): A New Version of the Rule Induction System LERS Fundamenta Informaticae, Vol. 31(1), pp. 27–39

[11] J. Grzymala-Busse and M. Hu (2000): A comparison of several approaches to missing attribute values in data mining. Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing RSCTC'2000, October 16–19, 2000, Banff, Canada, 340–347.

[12] D. Michie, D. J. Spiegelhalter, C. C. Taylor (1994): Machine learning, neural and statistical classification. Ellis Horwood, New York.

[13] Son H. Nguyen and A. Skowron (1997). Quantization of real value attributes: Rough set and boolean reasoning approach. Bulletin of International Rough Set Society 1/1, pp. 5–16.

[14] Son H. Nguyen (1997). Discretization of real value attributes. Boolean reasoning approach. Ph. D. thesis, supervisor A. Skowron, Warsaw University

[15] Son H. Nguyen, Hoa S. Nguyen (1998). Discretization Methods in Data Mining. In: L. Polkowski, A. Skowron (eds.): Rough Sets in Knowledge Discovery. Physica-Verlag, Heidelberg, pp. 451–482.

[16] Hoa S. Nguyen, H. Son Nguyen (1998). Pattern extraction from data. Fundamenta Informaticae 34/1-2, pp. 129–144.

[17] Son H. Nguyen (1998). From Optimal Hyperplanes to Optimal Decision Trees. Fundamenta Informaticae 34/1–2, pp. 145–174.

[18] Hoa S. Nguyen, A. Skowron and P. Synak (1998). Discovery of data patterns with applications to decomposition and classfification problems. In: L. Polkowski and A. Skowron (eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Physica-Verlag, Heidelberg, pp. 55–97.

[19] Hoa S. Nguyen (1999). Discovery of generalized patterns. Proceedings of the Eleventh International Symposium on Methodologies for Intelligent Systems, Foundations of Intelligent Systems (ISMIS'99), June 8–11,

Warsaw, Lecture Notes in Artificial Intelligence 1609, Springer-Verlag, Berlin, pp. 574–582.

[20] Hoa S. Nguyen (1999). Data regularity analysis and applications in data mining. Ph. D. thesis, supervisor B. Chlebus, Warsaw University.

[21] A. Øhrn, J. Komorowski (1997): ROSETTA – A rough set tool kit for analysis of data, *Proceedings of the Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97) at the Third Joint Conference on Information Sciences (JCIS'97),* Research Triangle Park, NC, March 2–5 (1997) 403–407.

[22] Z. Pawlak (1991): *Rough sets: Theoretical aspects of reasoning about data*. Dordrecht: Kluwer.

[23] D. Ślęzak, J. Wróblewski (1999). Classification Algorithms Based on Linear Combinations of Features. Proc. of PKDD'99, Prague, Czech Republic, Springer-Verlag (LNAI 1704), Berlin Heidelberg 1999, pp. 548–553.

[24] M. Wojnarski (2003): LTF-C: Architecture, Training Algorithm and Applications of New Neural Classifier. Fundamenta Informaticae, Vol. 54(1), pp. 89–105. IOS Press, 2003

[25] J. Wróblewski (1998). Genetic algorithms in decomposition and classification problem. In: L. Polkowski and A. Skowron (eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems, Physica-Verlag, Heidelberg, pp. 471–487

Indeks