STUDIA INFORMATICA 2006
Volume 27 Number 1 (66)

Marcin MICHALAK, Marek SIKORA
Politechnika Śląska, Instytut Informatyki

NetTRS – INTERNETOWE NARZĘDZIE DO ANALIZY DANYCH METODĄ ZBIORÓW PRZYBLIŻONYCH

Streszczenie. W artykule opisano serwis internetowy netTRS, pozwalający na realizację zadań eksploracji danych z wykorzystaniem biblioteki Tolerance Rough Sets, za pośrednictwem przeglądarki WWW. Dotychczas biblioteka TRS była dostępna jedynie w postaci standardowego programu wyposażonego w interfejs graficzny. Opisywany serwis pozwala na znacznie szersze niż do tej pory udostępnienie możliwości, które oferuje biblioteka, a także może być rozwijany w przyszłości.

Słowa kluczowe: aplikacja internetowa, reguły decyzyjne, klasyfikacja, zbiory przybliżone

NetTRS – WEB TOOL FOR DATA ANALYSIS BY THE ROUGH SETS THEORY

Summary. This paper presents internet system netTRS, which makes possible to analyse data, using TRS library and Internet browser. Until now, the library was accessible only as a local service with GUI. netTRS system makes it easier to use TRS library as a tool for induction and postprocessing of decision rules and can be developed in future.

Keywords: web application, decision rules, classification, rough sets

1. Wstęp

Na przełomie XX i XXI wieku rozwój Internetu dokonywał się na co najmniej dwóch płaszczyznach. Po pierwsze, stale wzrastała liczba jego użytkowników, po drugie natomiast globalna sieć zaczęła funkcjonować jako platforma umożliwiająca pracę pomiędzy dowolnymi komputerami z prawie każdego miejsca świata. Stąd łatwo zauważyć ogólnoświatową

tendencję polegającą na udostępnianiu istniejących już usług za pośrednictwem Internetu (znakomitym przykładem może być bankowość elektroniczna czy sklepy internetowe).

Niniejszy artykuł przedstawia system informatyczny, który doskonale wpisuje się w drugą grupę rozwiązań internetowych. Przez interfejs graficzny, jakim jest okno przeglądarki internetowej, użytkownik otrzymuje możliwość analizy danych, m.in. za pomocą narzędzi oferowanych przez tolerancyjny model zbiorów przybliżonych.

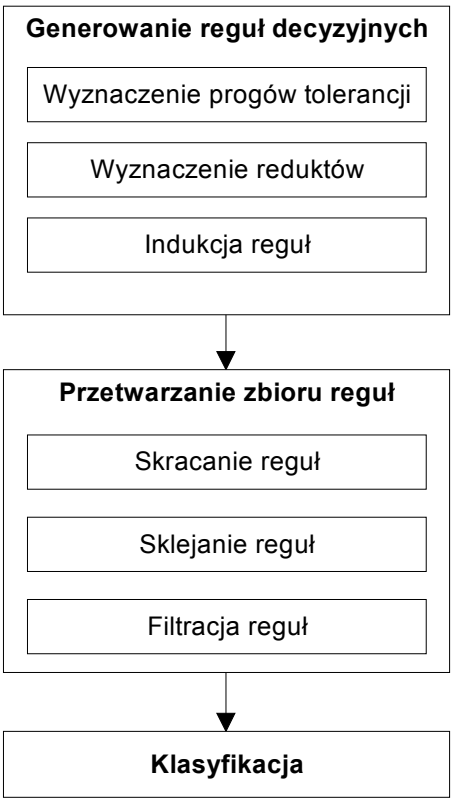
2. Biblioteka TRS

Biblioteka TRS powstała jako narzędzie analizy danych metodą tolerancyjnego modelu zbiorów przybliżonych [15], końcowym wynikiem analizy są zbiory reguł opisujące zależności, jakie udało się odkryć w analizowanym zbiorze danych. Zbiory reguł, uzyskane w czasie analizy nietrywialnych zbiorów danych, są zazwyczaj sporych rozmiarów, sprawia to znaczną trudność w interpretacji uzyskanych modeli danych i ich zastosowaniach praktycznych. Z tego względu biblioteka TRS posiada zaimplementowanych szereg algorytmów, które pozwalają zmniejszyć liczbę otrzymanych reguł. Większość z zawartych w bibliotece algorytmów została opisana szczegółowo w wielu publikacjach, [8, 9, 10] z tego też względu poniżej zostaną przedstawione jedynie zarysy poszczególnych algorytmów.

Typowy schemat analizy danych z wykorzystaniem biblioteki TRS jest przedstawiony na rysunku pierwszym. Przed samym wyznaczeniem reguł użytkownik musi zdefiniować wartości tzw. progów tolerancji [15, 16] w celu „dyskretyzacji" przestrzeni cech opisujących analizowany fragment rzeczywistości. Po zdefiniowaniu tych wartości możliwe jest wygenerowanie reguł lub ograniczenie liczby cech opisujących dane poprzez obliczenie tzw. reduktów [7, 13].

W kolejnym etapie analizy do wyznaczonych reguł można zastosować różne algorytmy, umożliwiające redukcję ich liczby. Biblioteka TRS pozwala na skracanie reguł (usuwanie przesłanek z części warunkowej reguły), sklejanie reguł (łącznie kilku reguł w jedną regułę) oraz filtrację reguł (usuwanie ze zbioru reguł reguł słabych – nieistotnych). Wszystkie z wymienionych algorytmów wykorzystują wartości tzw. miar oceniających atrakcyjność reguł [2, 11]. W bibliotece TRS można wyznaczyć wartości kilkunastu miar oceniających [9].

Końcowym etapem analizy jest weryfikacja jakości uzyskanego zbioru reguł, weryfikację tę przeprowadza się poprzez obliczenie dokładności klasyfikacji, jaką uzyskuje końcowy zbiór reguł na testowym zbiorze obiektów.



Rys. 1. Schemat analizy danych z użyciem biblioteki TRS
Fig. 1. Scheme of data analysis with TRS library

2.1. Wyznaczanie reguł decyzyjnych

Z formalnego punktu widzenia analizy jest zbiór danych zawartych w tablicy decyzyjnej ***DT***=(U, $A\cup\{d\}$), gdzie U jest skończonym zbiorem obiektów, A jest skończonym zbiorem atrybutów warunkowych (cech opisujących obiekty), natomiast d jest wyróżnionym atrybutem decyzyjnym. Każdy atrybut $a\in A\cup\{d\}$ posiada pewną dziedzinę D_a i może być traktowany jako funkcja $a{:}U\rightarrow D_e$.

W tolerancyjnym modelu zbiorów przybliżonych wprowadza się pewną relację podobieństwa $\tau\subseteq U\times U$, pozwalającą wskazać w zbiorze U podzbiory obiektów podobnych, które następnie zostają wykorzystane do tworzenia reguł.

2.1.1. Algorytmy poszukiwania wartości progów tolerancji

Dla każdego atrybutu $a\in A$ określona jest miara odległości $\delta_a{:}D_a\times D_a\rightarrow[0,\infty]$, o własnościach $\forall x,y\in U$ $\delta_a(a(x),a(x))=0$ i $\delta_a(a(x),a(y))=\delta_a(a(y),a(x))$, gdzie $a(x)$ oznacza wartość atrybutu $a\in A$, jaką przyjmuje obiekt $x\in U$. W bibliotece TRS zaimplementowano dwie miary odległości znane jako *diff* i *vdm* [15].

Przy ustalonym podzbiorze atrybutów $B \subseteq A$ dla każdego atrybutu $a_i \in B$ $i=1,2,..,n$ ustalony jest próg tolerancji ε_{a_i}. Relacja $\tau_B(\varepsilon_{a_1}, \varepsilon_{a_2}, .., \varepsilon_{a_n})$ definiowana jest wtedy następująco:

$$\forall x,y \in U \; \langle x,y \rangle \in \tau_B(\varepsilon_{a_1}, \varepsilon_{a_2}, .., \varepsilon_{a_n}) \Leftrightarrow \forall a_i \in B \; [\delta_{a_i}(a_i(x), a_i(y)) \le \varepsilon_{a_i}]$$

Zbiór obiektów podobnych do obiektu $x \in U$ ze względu na zbiór atrybutów $B \subseteq A$ określa wartość funkcji niepewności $I_B: U \to 2^U$ zdefiniowanej następująco:

$$\forall y \in U, y \in I_B(x) \Leftrightarrow \langle x,y \rangle \in \tau_B(\varepsilon_{a_1}, \varepsilon_{a_2}, .., \varepsilon_{a_n}).$$

Innymi słowy, zbiór $I_B(x)$ jest zbiorem tolerancji elementu x. Po wyznaczeniu progów tolerancji możliwe jest wyznaczenie zbioru reguł decyzyjnych [7, 15], tworzących opisy klas decyzyjnych. Klasą decyzyjną nazywamy zbiór $X_v = \{x \in U: d(x) = v\}$, gdzie $v \in D_d$.
Pojedyncza reguła decyzyjna ma postać $(a_1, V_{a_1}) \wedge .. \wedge (a_k, V_{a_k}) \to (d,v)$, dowolne (a, V_a) nazywane jest deskryptorem oraz $V_a \subseteq D_a$ a v jest pewną wartością atrybutu decyzyjnego. Regułę taką można interpretować jako wyrażenie warunkowe typu „IF spełniony jest zbiór warunków THEN, wartość atrybutu decyzyjnego przyjmuje konkretną wartość".

W tolerancyjnym modelu zbiorów przybliżonych, reguły budowane są wokół obiektów generatorów, dla konkretnego obiektu generatora reguły $x \in U$ zakres deskryptora V_a wyznacza się na podstawie znajomości progu tolerancji ε_a, w sposób następujący:

$$V_a = \{a(y) \in D_a: y \in I_a(x)\}.$$

Z powyższego wynika jasno, że wartości progów tolerancji będą miały duży wpływ na postać reguł, a zatem również na ich jakość.

W bibliotece TRS zaimplementowano dwa sposoby wyznaczania wartości progów tolerancji:

- za pomocą algorytmu genetycznego [5, 16];
- wykorzystując strategię polegającą na tym, iż szukany jest wektor o postaci $(\varepsilon,..,\varepsilon)$; jako początkową wartość ε_{min} przyjmuje się 0, końcowa wartość to $\varepsilon_{max}=1$; określana jest liczba k, o jaką po każdej iteracji będzie zwiększać się ε; dla każdego wektora, poczynając od wektora (0,..,0), a kończąc na (1,..,1), obliczana jest wartość funkcji oceniającej wektor; jako wektor najlepszy pamiętany jest ten, który uzyskał najlepszą ocenę.

Standardowo, jako funkcję oceniającą jakość wektora progów tolerancji w obu wymienionych sposobach wykorzystuje się funkcję o postaci [15]:

$$w\,\gamma(d) + (1-w) v_{SRI}(R_d, R_{I_A})$$

gdzie $\gamma_B(d) = \dfrac{card(POS_B(d))}{card(U)}$, $POS_B(d)$ jest B-obszarem pozytywnym [7] tablicy decyzyjnej $\boldsymbol{DT} = (U, A \cup \{d\})$ oraz $R_d = \{\langle x,y \rangle \in U \times U : d(x) = d(y)\}$,

$R_{I_A} = \{< x, y > \in U \times U: y \in I_A(x)\}$ i $0<w\leq 1$ oraz v_{SRI} jest tzw. przybliżoną funkcją przynależności [7]. Przedstawiona funkcja pozwala znajdować takie progi tolerancji dla atrybutów warunkowych, aby jak najwięcej obiektów mających te same wartości atrybutu decyzyjnego było z sobą w relacji przy równoczesnym ograniczeniu do minimum przypadków, kiedy w relacji znajdują się obiekty posiadające różne wartości atrybutu decyzyjnego.

W bibliotece TRS jako miary oceniające wektor progów tolerancji wykorzystano również specjalnie do tego celu zaadaptowane miary oceniające jakość reguł, ze względu na oczywistą analogię pomiędzy jakością reguł i wektorem progów tolerancji. Sposób adaptacji reguł i mechanizmy wykorzystane w algorytmie genetycznym szczegółowo opisano w [11].

2.1.2. Miary oceniające jakość reguł

Reguły opisujące klasy decyzyjne powinny odzwierciedlać ogólne zależności występujące w danych, a jednocześnie być łatwe do interpretacji. Możliwe jest to wtedy, gdy liczba reguł dla danej klasy decyzyjnej jest mała, a reguły ją opisujące uznane zostaną za silne względem dwóch podstawowych kryteriów: dokładności i pokrycia. Najbardziej pożądanymi cechami reguł są dokładność (zależność opisywana przez regułę powinna być prawdziwa w jak największej liczbie przypadków) oraz pokrycie (liczba obiektów rozpoznających regułę powinna być jak największa).

Z punktu widzenia rachunku prawdopodobieństwa zależność prezentowana przez regułę, która jest dokładna i charakteryzuje się dużym pokryciem, opisuje ogólną prawidłowość, jaką udało się odkryć w danych.

Aby możliwe było wskazywanie reguł najbardziej pożądanych w opisie, potrzebne jest kryterium oceniające jakość reguł. Zazwyczaj do tego celu wykorzystywane są miary, bazujące na tablicy kontyngencji tworzonej dla każdej reguły.

Tablicę kontyngencji dla reguły decyzyjnej $\varphi \rightarrow \psi$ przedstawia się w postaci macierzy kwadratowej w sposób następujący:

$n_{\varphi\psi}$	$n_{\varphi\neg\psi}$	n_{φ}
$n_{\neg\varphi\psi}$	$n_{\neg\varphi\neg\psi}$	$n_{\neg\varphi}$
n_{ψ}	$n_{\neg\psi}$	

gdzie:

$n_{\varphi}= n_{\varphi\psi}+ n_{\varphi\neg\psi}=|U_{\varphi}|$ liczba obiektów rozpoznających regułę $\varphi \rightarrow \psi$;

$n_{\neg\varphi}= n_{\neg\varphi\psi}+ n_{\neg\varphi\neg\psi}=|U_{\neg\varphi}|$ liczba obiektów nie rozpoznających reguły $\varphi \rightarrow \psi$;

$n_{\psi}= n_{\varphi\psi}+ n_{\neg\varphi\psi}=|U_{\psi}|$ liczba obiektów z klasy decyzyjnej opisywanej przez $\varphi \rightarrow \psi$;

$n_{\neg\psi}=n_{\varphi\neg\psi}+n_{\neg\varphi\neg\psi}=|U_{\neg\psi}|$ liczba obiektów spoza klasy decyzyjnej opisywanej przez $\varphi \rightarrow \psi$;

$n_{\varphi\psi}=|U_{\varphi} \cap U_{\psi}|$ liczba obiektów wspierających regułę $\varphi \rightarrow \psi$; $n_{\varphi\neg\psi}=|U_{\varphi} \cap U_{\neg\psi}|$; $n_{\neg\varphi\psi}=|U_{\neg\varphi} \cap U_{\psi}|$; $n_{\neg\varphi\neg\psi}=|U_{\neg\varphi} \cap U_{\neg\psi}|$.

Korzystając z informacji zawartej w tablicy kontyngencji oraz faktu, iż dla znanej reguły decyzyjnej $\varphi\rightarrow\psi$, znane są wartości $|U_{\psi}|$ i $|U_{\neg\psi}|$, można wartość każdej wykorzystywanych w bibliotece TRS funkcji wyznaczyć na podstawie wartości $n_{\varphi\psi}$ i $n_{\varphi\neg\psi}$.

Spośród stosowanych miar oceniających do najczęściej używanych należą dokładność oraz pokrycie

$$q^{accuracy}_{\varphi\rightarrow\psi}(<n_{\varphi\psi},n_{\varphi\neg\psi}>)=\frac{n_{\varphi\psi}}{n_{\varphi}} \qquad q^{coverage}_{\varphi\rightarrow\psi}(<n_{\varphi\psi},n_{\varphi\neg\psi}>)=\frac{n_{\varphi\psi}}{n_{\psi}}.$$

Pozostałe miary starają się łączyć ocenę dokładności i pokrycia, w bibliotece TRS zaimplementowano poza dokładnością i pokryciem następujące miary: *Brazdil, Cohen, Coleman, Coverage, Gain, IKIB, IREP, Michalski, Pearson.* Szczegółowy opis własności wymienionych funkcji można znaleźć m.in. w [2, 9].

W bibliotece możliwe jest również obliczenie wartości wyżej wymienionych miar dla obiektów unikalnie pokrywanych przez daną regułę oraz wartości miary mieszanej wyrażonej wzorem:

$$q(r)_{new}=q(r)_{unique}/q(r)_{normal}$$

gdzie: $q(r)_{new}$ nowa wartość miary, $q(r)_{unique}$ wartość miary q, uwzględniająca tylko obiekty unikalnie rozpoznające regułę r, $q(r)_{normal}$ wartość miary q obliczana w sposób typowy.

Istotną własnością reguły, z punktu widzenia możliwości interpretacji zależności, jaką dana reguła reprezentuje, jest liczba deskryptorów warunkowych występujących w regule. Dla reguły decyzyjnej $\varphi\rightarrow\psi$ zbiór $descr(\varphi\rightarrow\psi)$ zawiera wszystkie atrybuty, które tworzą deskryptory warunkowe w tej regule. Zbiór $descr(\varphi\rightarrow\psi)$ zdefiniowany jest następująco:

$$descr(\varphi\rightarrow\psi)=\{a\text{: } (a,V) \textit{ jest deskryptorem warunkowym w regule } \varphi\rightarrow\psi\}.$$

Zaproponowana w bibliotece TRS formuła oceniająca regułę ze względu na liczbę deskryptorów warunkowych q^{length}:$RUL\rightarrow[0,1)$ zdefiniowana jest jako:

$$q^{length}(r)=1-\frac{|descr(r)|}{|A|}$$

Im wartość funkcji q^{length} jest większa, tym reguła jest prostsza, tzn. ma mniej deskryptorów warunkowych. Możliwe jest również dokonanie oceny reguły, uwzględniając zarówno jej jakość, jak i długość:

$$q^{nazwa\_funkcji}_{\varphi\rightarrow\psi}\ q^{length}(\varphi\rightarrow\psi),$$

gdzie $q^{nazwa\_funkcji}_{\varphi\rightarrow\psi}$ jest dowolną z wymienionych wcześniej miar oceniających.

2.1.3. Algorytmy indukcji reguł

Mając ustalony wektor progów tolerancji (w szczególności mogą to być same zera), można przystąpić do wyznaczania reguł decyzyjnych. W bibliotece TRS zaimplementowano

standardowy algorytmy indukcji reguł z tzw. reduktów relatywnych, przy czym dostępne są trzy wersje tego algorytmu: indukcja wszystkich reguł, indukcja zadanej liczby reguł (reduktów), indukcja jednej reguły z quasi-najkrótszego reduktu (tutaj wykorzystano heurystyczny algorytm Johnsona [6]). Algorytm wykorzystuje uogólnioną macierz odróżnialności modulo d, co ogranicza rozmiar analizowanego zbioru danych do kilku tysięcy obiektów. Na macierzy odróżnialności oparto również zaproponowany przez autorów biblioteki algorytm RMatrix [8]. Poniżej omówiono krótko ideę algorytmu RMatrix.

Niech $\boldsymbol{DT}=(U,A\cup\{d\})$, $U=\{x_1,x_2,..,x_n\}$, uogólnioną macierzą rozróżnialności tablicy $\boldsymbol{DT}$ nazywamy macierz kwadratową $M_d(\boldsymbol{DT})=\{c_{i,j}: 1\leq i,j\leq n\}$ o elementach zdefiniowanych następująco:

$$c_{ij}=\{a\in A: (\delta_a(x_i,y_j)>\varepsilon_a)\wedge(d(x_i)\neq d(y_j))\},$$

$$c_{ij}=\{\varnothing: d(x_i)=d(y_j)\}.$$

Gdy obiekty x_i,y_j mają taką samą wartość atrybutu decyzyjnego, w komórce c_{ij} macierzy odróżnialności znajduje się zbiór pusty. W przeciwnym przypadku, znajduje się w niej zbiór atrybutów, których wartości wystarczają do odróżnienia obiektu x_i od x_j w analizowanym zbiorze danych.

Algorytm RMatrix wykorzystuje informacje zawarte w macierzy odróżnialności. Każda reguła budowana jest wokół pewnego obiektu generatora. Każdemu obiektowi odpowiada jeden wiersz (kolumna) w macierzy odróżnialności. Atrybut, występujący najczęściej w tym wierszu, pozwala najlepiej odróżnić od obiektu generatora obiekty z innych niż generator klas decyzyjnych. Tworząc ranking atrybutów ze względu na częstotliwość ich występowania, w odpowiednim wierszu macierzy odróżnialności możemy otrzymać quasi-minimalny redukt relatywny dla obiektu generatora.

Zaimplementowany w bibliotece algorytm RMatrix generuje jedną regułę z każdego obiektu generatora. Dodanie kolejnego deskryptora powoduje to, że reguła staje się coraz dokładniejsza, przy jednoczesnym ograniczaniu wartości pokrycia reguły. Miara oceniająca jakość reguły dba o to, aby reguła wyjściowa nie była zbyt dopasowana do danych treningowych.

Algorytm RMatrix może wykorzystywać jedną ze znanych miar oceniających reguły (*accuracy*, *Michalski*, *Brazdil*, *IREP*, *Pearson*, *Cohen*, *Coleman, IKIB, gain* [2, 9]).

dane: $\boldsymbol{DT}=(U, A\cup\{d\})$, *wektor progów tolerancji, q – miara oceniająca regułę*
x – obiekt-generator reguły, c_x *– wiersz macierzy odróżnialności modulo d odpowiadającej obiektowi x*

początek

utwórz regułę r, która zawiera jedynie deskryptor decyzyjny d=d(x)

$r_{best} := r$

ustal porządek atrybutów warunkowych ($a_{i_1}, a_{i_2}, .., a_{i_N}$), *tak że atrybut występujący najczęściej w* c_x *jest pierwszy (atrybut występujący najrzadziej jest ostatni)*

dla każdego *i := 1, ... ,card(A)*

dodaj deskryptor $a \in V_a$ *do części warunkowej reguły r (gdzie* $V_a=\{a(y) \in D_a: y \in I_a(x)\}$*)*

jeżeli $q(r) > q(r_{best})$, *to* $r_{best} := r$

jako regułę wyjściową zwróć r_{best}

koniec

Poza algorytmem RMatrix w bibliotece zaimplementowano również algorytm MODLEM [14], który może działać na danych numerycznych nie poddanych żadnej wstępnej obróbce. Aby zmniejszyć liczbę wyznaczanych tym algorytmem reguł, zaproponowano jego modyfikację [12] polegającą, podobnie jak w algorytmie RMatrix, na ocenie tworzonej reguły (po każdej iteracji algorytmu) za pomocą miary oceniającej. Jeśli wartość miary zaczyna spadać, proces formowania reguły kończy się. Taka poprawka pozwala na generowanie mniejszej liczby bardziej ogólnych reguł, które charakteryzują się dobrymi własnościami opisowymi i klasyfikacyjnymi.

2.2. Uogólnianie i filtracja reguł

Wyznaczając reguły decyzyjne metodami bazującymi na tolerancyjnym modelu zbiorów przybliżonych, godzimy się na możliwość otrzymania reguł aproksymacyjnych (tzn. niedokładnych). Niezależnie od tego, czy generowane są wszystkie minimalne reguły decyzyjne, czy do budowy reguł wykorzystywany jest algorytm heurystyczny, bardzo często wyznaczonych reguł jest dużo, co znacząco zmniejsza ich moc opisową. W bibliotece TRS implementowano kilka algorytmów, pozwalających na wykonanie tzw. postprocessingu reguł. Postprocessing ma na celu ograniczenie liczby reguł (czyli zwiększenie ich mocy opisowej) przy jednoczesnym zachowaniu dobrych zdolności klasyfikacyjnych.

W bibliotece TRS postprocessing można wykonać na dwa sposoby:

- poprzez uogólnianie reguł (skracanie reguł, sklejanie reguł),
- poprzez filtrację reguł (usuwanie ze zbioru reguł, reguł zbędnych ze względu na pewne kryterium).

Skracanie reguł polega na usuwaniu z części przesłankowej reguły pewnych deskryptorów warunkowych. Każda reguła w postaci nieskróconej posiada przyporządkowaną wartość miar oceniającej. Skracanie następuje tak długo, aż wartość ta nie spadnie poniżej zadanego przez użytkownika progu. Parametr taki ustalany jest dla opisu każdej klasy osobno. Kolejności usuwania deskryptorów ustalana jest zgodnie ze strategią wspinaczki.

Sklejanie reguł polega na otrzymaniu jednej ogólniejszej reguły z dwóch mniej ogólnych. Zaimplementowany w bibliotece algorytm sklejania reguł bazuje na następujących założeniach:

- sklejaniu podlegają reguły opisujące tę samą klasę decyzyjną,
- dla reguł $\varphi 1 \rightarrow \psi$ i $\varphi 2 \rightarrow \psi$ zostanie podjęta próba ich sklejenia, jeśli $descr(\varphi 1 \rightarrow \psi) \subseteq descr(\varphi 2 \rightarrow \psi)$ lub $descr(\varphi 2 \rightarrow \psi) \subset descr(\varphi 1 \rightarrow \psi)$; sklejać będziemy tylko te reguły, których części warunkowe zbudowane są na podstawie podobnego zbioru atrybutów warunkowych,
- sklejanie polega na łączeniu zbioru wartości odpowiadających sobie deskryptorów warunkowych. Jeśli (a, V_a^1), deskryptor występuje w części warunkowej reguły $\varphi 1 \rightarrow \psi$ oraz deskryptor (a, V_a^2) występuje w części warunkowej reguły $\varphi 2 \rightarrow \psi$, to po sklejeniu tych deskryptorów powstanie deskryptor (a, Va) o własności: $V_a^1 \subseteq V_a$ i $V_a^2 \subseteq V_a$.

Sterowanie procesem sklejania reguł przebiega według następujących zasad:

- z dwóch reguł r_1 i r_2 o własności $q_{r_1} > q_{r_2}$ jako „bazową", na podstawie której budowana będzie nowa sklejona reguła, wybierana jest reguła r_1. Zapis q_{r_1} oznacza wartość miary oceniającej q dla reguły r_1;
- deskryptory warunkowe sklejane są sekwencyjnie; o kolejności sklejania deskryptorów decyduje wartość miary oceniającej nowo powstałą regułę r; aby wskazać najlepszy do sklejenia deskryptor, stosowana jest strategia wspinaczki;
- proces sklejania kończy się z chwilą, kiedy nowa reguła r rozpoznaje wszystkie pozytywne przykłady treningowe rozpoznawane przez reguły r_1 i r_2;
- jeśli r jest nową regułą i $q_r \geq \lambda$, to w miejsce reguł r_1 i r_2 wstawiana jest do opisu klasy decyzyjnej reguła r, w przeciwnym przypadku reguł r_1, oraz r_2 nie można skleić; parametr λ określa graniczną wartość, poniżej której jakość reguł według danej funkcji oceniającej nie może spaść (w szczególności dla każdych sklejanych reguł r_1 i r_2, $\lambda = \max\{q_{r_1}, q_{r_2}\}$.

Filtracja zbioru reguł polega na usuwaniu z opisów klas decyzyjnych reguł, które ze względu na ustalone kryterium okazują się zbędne. Zmniejszając opis klas decyzyjnych, filtracja nie ingeruje w budowę wyznaczonych reguł, a jedynie ogranicza ich liczbę.

W bibliotece TRS zaimplementowano dwa rodzaje algorytmów filtracji:

- nie uwzględniające dokładności klasyfikacji nie filtrowanego zbioru reguł,
- uwzględniające dokładności klasyfikacji nie filtrowanego zbioru reguł.

Pierwsze podejście reprezentuje algorytm filtracji „*Z pokrycia*". W algorytmie tym najpierw wyznaczany jest ranking reguł, a następnie budowę pokrycia zbioru treningowego rozpoczyna się od najlepszej reguły. Kolejne reguły dodawane są zgodnie z kolejnością ich

występowania w rankingu. W momencie gdy wszystkie przykłady ze zbioru treningowego zostaną pokryte, pozostałe, nie dodane jeszcze reguły, są odrzucane.

Drugie podejście do filtracji reprezentują algorytmy „*W przód* ” oraz „*Wstecz*”. Oba algorytmy, poza rankingiem reguł utworzonym przez wybraną miarę oceniającą jakość reguły, wykorzystują także informację o uzyskanej przez wszystkie reguły dokładności klasyfikacji. Aby zachować niezależność filtracji od danych testowych, klasyfikacja w czasie filtracji przeprowadzana jest na zbiorze tzw. tuningowym, będącym zbiorem osobnych obiektów niezależnym od zbioru treningowego.

Dla filtracji „*W przód*” początkowy opis każdej klasy decyzyjnej składa się z jednej – najlepszej reguły. Następnie do opisu każdej klasy dodawane są pojedyncze reguły. Jeżeli dokładność klasy zwiększy się, reguła pozostawiana jest w opisie, jeżeli nie, rozpatrywana jest kolejna reguła. O kolejności rozpatrywania reguł decyduje ranking reguł ustalony przez miarę oceniającą jakość reguły. Dodawanie reguł do opisu klasy decyzyjnej kończy się z chwilą osiągnięcia identycznej dokładności klasyfikacji, jaką uzyskuje nie filtrowany zbiór reguł lub kiedy zbiór reguły się skończy.

Algorytm filtracji „*Wstecz*” oparty jest na idei odwrotnej. Z każdej klasy decyzyjnej usuwane są reguły począwszy od najsłabszych. O kolejności usuwania reguł decyduje ranking reguł ustalony przez miarę oceniającą jakość reguły. Utrzymanie różnicy w dokładności pomiędzy najdokładniejszą a najmniej dokładną klasą decyzyjną gwarantuje, że odfiltrowany zbiór reguł decyzyjnych zachowa taką samą czułość jak zbiór nie filtrowany.

Poza tym zaimplementowano również prosty algorytm usuwający reguły o jakości mniejszej od zadanego przez użytkownika progu.

Algorytmy filtracji szczegółowo opisane są m.in. w [9].

2.3. Klasyfikacja

Klasyfikacja rozumiana jest jako proces przyporządkowywania obiektów do odpowiadających im klas decyzyjnych jedynie na podstawie znajomości cech, charakteryzujących te obiekty.

Algorytm klasyfikujący (decyzyjny) jest algorytmem automatycznego podejmowania decyzji, dokonującym w sposób automatyczny przyporządkowania wartości atrybutu decyzyjnego na podstawie wartości atrybutów warunkowych i zbioru wyznaczonych reguł decyzyjnych.

Efektywność algorytmu decyzyjnego mierzy się osiąganą przez niego dokładnością klasyfikacji, którą rozumiemy jako stosunek obiektów poprawnie przyporządkowanych do klas decyzyjnych do wszystkich obiektów testowych.

Obiekty rozpoznane przez reguły jednej klasy decyzyjnej uznajemy za zaklasyfikowane do tej klasy. W przypadku gdy obiekt rozpoznawany jest przez reguły należące do opisów różnych klas decyzyjnych, wartości atrybutu decyzyjnego nie można przypisać w sposób jednoznaczny. Do klasyfikacji takich obiektów biblioteka TRS wykorzystuje tzw. mechanizm głosowania [4]. Każdej wyznaczonej regule przyporządkowany jest pewien stopień zaufania (jest to po prostu wartość miary oceniającej te reguły). W bibliotece TRS klasyfikacja odbywa się poprzez zsumowanie stopni zaufania reguł z każdej klasy decyzyjnej rozpoznających obiekt testowy. Obiekt przyporządkowany jest do tej klasy, dla której otrzymana suma jest maksymalna. Zdarzają się także przypadki, gdy obiektu testowego nie rozpoznaje żadna z reguł wchodzących w skład wyznaczonych opisów klas decyzyjnych. W takim przypadku możliwe jest obliczenie odległości obiektu testowego od dowolnej reguły i uznanie, że reguły leżące dostatecznie blisko obiektu testowego rozpoznają go.

Reasumując, w bibliotece zaimplementowano następujący schemat przyporządkowywania obiektowi testowemu wartości atrybutu decyzyjnego. Dla każdego obiektu testowego u wyznaczany jest stopień zaufania tego obiektu do każdej z klas decyzyjnych, zgodnie ze wzorem:

$$conf(X_v, u) = \sum_{r \in RUL_{X_v}(\mathrm{DT}), dist(r,u) \le \varepsilon} (1\text{-}dist(r,u))q(r),$$

gdzie $dist(r,u)$ jest odległością obiektu testowego u od reguły r (np. Euklidesową, Hamminga), ε jest akceptowalną, maksymalną odległością obiektu od reguły (w szczególności jeżeli $\varepsilon=0$, klasyfikacja odbywa się przez reguły, które dokładnie rozpoznają obiekt testowy), Wartość $q(r)$ jest siłą głosu każdej reguły (czyli wspomnianym wcześniej stopniem zaufania do reguły). Ostatecznie, funkcja decyzyjna f przyporządkowująca dowolnemu obiektowi u wartość atrybutu decyzyjnego określona jest wzorem:

$$f(u)=v_{max} \Leftrightarrow conf(X_{v_{max}}, u) = \max_{v \in D_d}(conf(X_v, u)).$$

Jak wykazano w [4] (gdzie badano na drodze statystycznej różne klasyfikatory regułowe), powyższy schemat klasyfikacji jest najlepszym schematem klasyfikacji opartym na głosowaniu.

3. NetTRS – architektura systemu

System netTRS jest przykładem aplikacji internetowej zrealizowanej w technologii ASP.NET, której podstawowym zadaniem jest udostępnienie końcowemu użytkownikowi możliwości obliczeniowych biblioteki TRS. Jak dotychczas, z systemu netTRS można

korzystać jedynie za pośrednictwem przeglądarki Internet Explorer, ale trwają prace mające na celu udostępnienie go także użytkownikom innych popularnych przeglądarek.

Mając na uwadze możliwość rozbudowy funkcjonalności systemu, już na etapie projektowania został on podzielony na dwie warstwy, które komunikują się ze sobą poprzez bazę danych[1].

Pierwsza warstwa odpowiedzialna jest za pobieranie danych przekazywanych przez użytkownika oraz udostępnienie mu wyników analizy. Składa się ona z zestawu stron internetowych. Część z nich odpowiedzialna jest za definiowanie nowego zadania analizy, a część pozwala użytkownikowi na podgląd wyników analizy, a także na pobranie ich i zapisanie u siebie na dysku lokalnym.

Druga warstwa tworząca system odpowiedzialna jest przede wszystkim za wykonywanie kolejnych zadań analizy, zleconych przez wszystkich użytkowników, którzy zainicjowali jakiekolwiek zadania. Ta część systemu zajmuje się również ułatwieniem użytkownikowi pobrania danych. Uatwienie to polega na spakowaniu wszystkich plików wejściowych i wyjściowych do jednego skompresowanego zbioru, który z kolei udostępniony jest do pobrania przez wspomnianą już pierwszą część systemu.

Dodatkową, niezwiązaną z samym procesem analizy danych, częścią serwisu jest „panel administratora", czyli zestaw stron, pozwalający na dodawanie/usuwanie użytkowników, a także na sterowanie wykonywaniem zadań po stronie obliczeniowej systemu.

4. Interfejs użytkownika

Z serwisu mogą skorzystać użytkownicy, którzy uzyskali od administratora parametry swojego konta (login i hasło), aby stworzone zostało konto nowego użytkownika, konieczne jest zatem skontaktowanie się z administratorem systemu. Serwis www obsługujący system netTRS można podzielić na dwie części: definiowania zadań, jakie mają zostać wykonane w ramach eksperymentu, oraz oglądania i pobierania wyników analizy.

4.1. Tworzenie i parametryzacja zadań analizy

Jeśli użytkownik poprawnie wprowadzi swój login i hasło, uzyska dostęp do stron panelu użytkownika, dzięki którym będzie mógł korzystać z systemu.

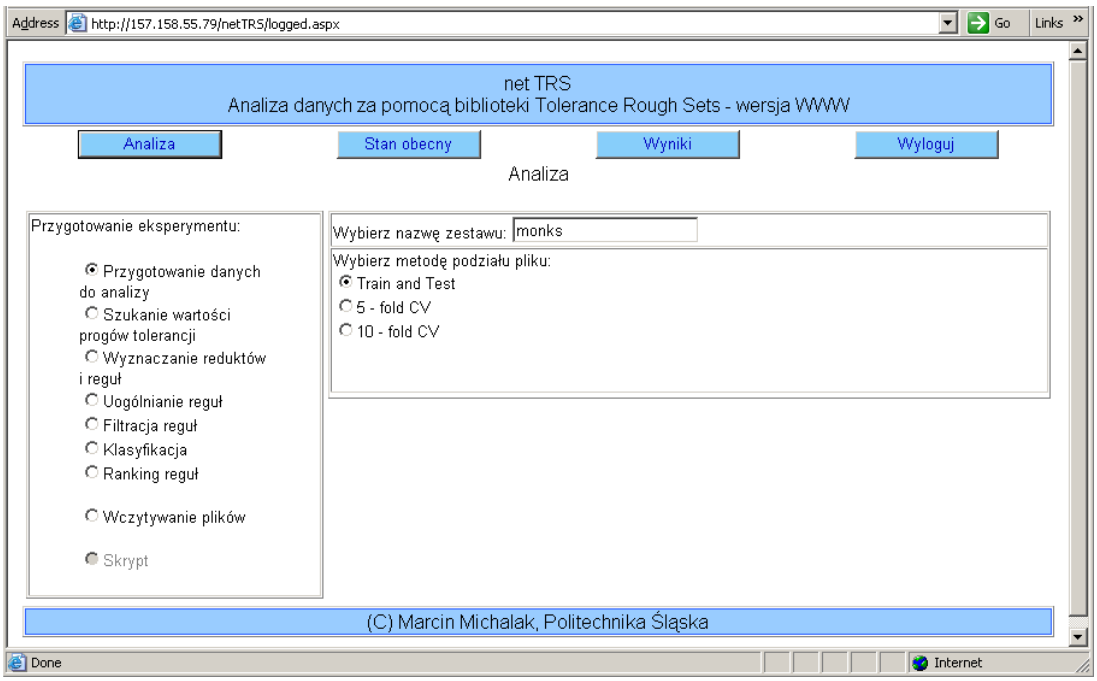
[1] W systemie istnieje również komunikacja poprzez tzw. „kolejki systemowe", nie jest ona jednak wykorzystywania w zadaniach analizy danych, a tylko w celach administracyjnych, przez co jej opis został pominięty.

Zestaw stron ma budowę analogiczną do „zakładkowej”. Na pierwszej zakładce *Analiza* (rys. 2) użytkownik ustala rodzaj nowego eksperymentu (do wyboru są trzy powszechnie znane metodologie testowania: train and test, 5 – fold CV oraz 10 – fold CV). Po wybraniu opcji *Wczytywanie plików* udostępniana jest strona, z której możliwe jest pobranie plików treningowych (poddawanych analizie), testowych (weryfikacyjnych) i plików pozwalających na dostrojenie wykorzystywanych algorytmów (są to tzw. pliki tuningowe).

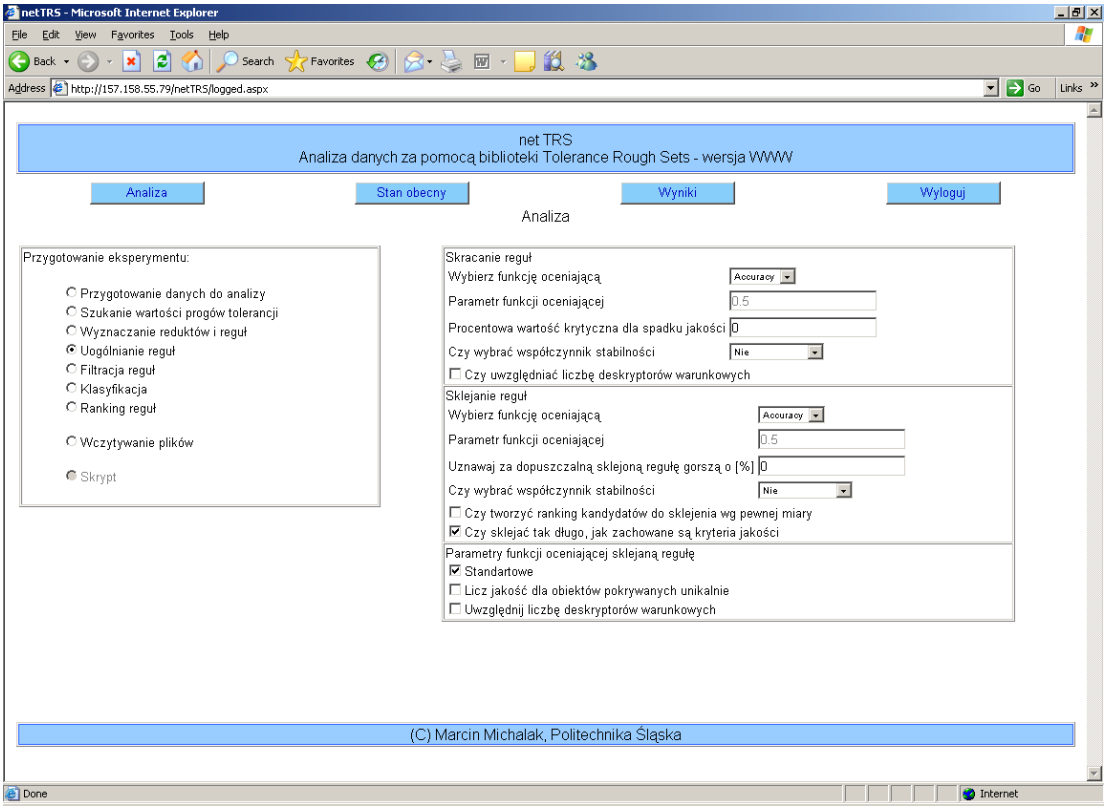
Wybieranie kolejnych opcji na zakładce *Analiza* (*Wyznaczanie reduktów i reguł*; *Uogólnianie reguł* itd.) umożliwia użytkownikowi ustalenie parametrów poszczególnych wybranych przez niego etapów analizy. Na rysunku trzecim zaprezentowano stronę konfiguracyjną algorytmu skracania i sklejania reguł.

Po wczytaniu plików z danymi dostępna staje się zakładka *Skrypt*, na której w postaci tekstowej użytkownik może obejrzeć parametry wszystkich zdefiniowanych przez niego zadań analizy w ramach danego eksperymentu. Wstawienie eksperymentu do kolejki zadań ma miejsce po kliknięciu przycisku *Wczytaj* na zakładce *Skrypt*.

Ponieważ użytkownik może inicjalizować wiele eksperymentów, zakładka *Stan obecny* umożliwia podglądanie, w jakiej fazie realizacji znajdują się eksperymenty zainicjowane przez użytkownika.



Rys. 2. Nadawanie etykiety nowemu eksperymentowi
Fig. 2. New experiment labeling



Rys. 3. Strona algorytmów uogólniania reguł
Fig. 3. Rules generalization algorithms side

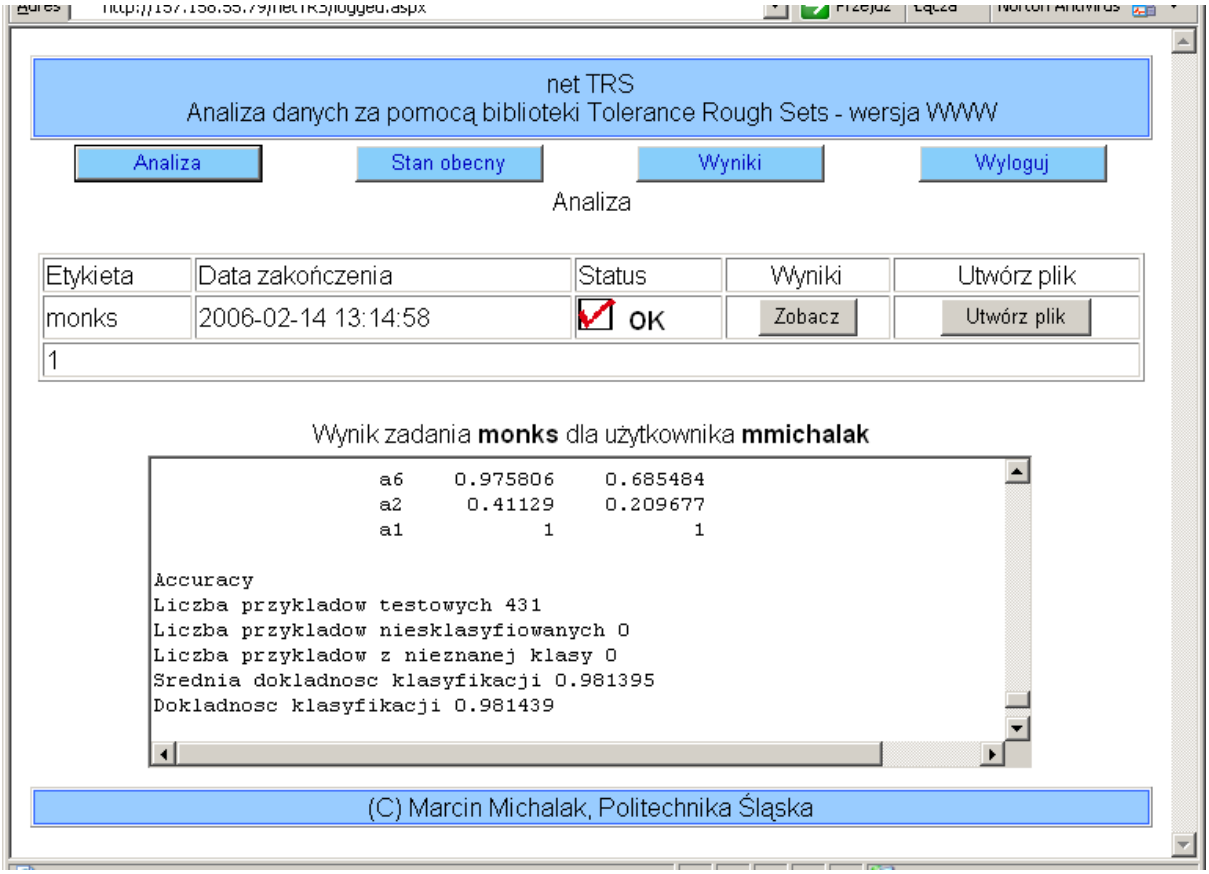
Zdefiniowane przez użytkownika eksperymenty „obsługiwane” są po stronie serwera przez program *TRSExecutor*, który cyklicznie pobiera je z bazy danych, w której prowadzony jest rejestr użytkowników, zdefiniowanych przez nich eksperymentów oraz stopień ich realizacji. Program *TRSExecutor* uruchamia wykonywalną wersję biblioteki TRS i przekazuje do niej plik sterujący przebiegiem kolejnego eksperymentu. Po zakończeniu obliczeń związanych z danym eksperymentem wszystkie pliki wynikowe oraz pliki, na których przeprowadzano eksperyment, kompresowane są w jednym archiwum, które użytkownik może pobrać ze strony *Wyniki*.

4.2. Przeglądanie i pobieranie wyników analizy

W miarę zakończenia realizacji kolejnych eksperymentów zdefiniowanych przez użytkownika wyniki tych eksperymentów pojawiają się na *Wyniki* (rys. 4). Użytkownik może w dowolnej chwili obejrzeć wyniki zakończonego eksperymentu z poziomu serwisu, a także może wyniki te pobrać jako archiwum i oglądać je dopiero na swoim komputerze.

Podstawowym plikiem wynikowym generowanym przez bibliotekę TRS jest plik tekstowy o domyślnej nazwie *out.txt*. W pliku, w zależności od tego, jakie algorytmy uruchomiono w ramach eksperymentu, znajdują się oryginalnie wygenerowane reguły, reguły

po skracania, reguły po sklejaniu, reguły po filtracji oraz wyniki klasyfikacji. Zawartość tego pliku można oglądać po naciśnięciu klawisza *Zobacz* na zakładce *Wyniki*. Biblioteka TRS generuje również pliki pomocnicze, w których znajdują się m.in. reguły wraz z wartościami ich miar oceniających. Jak już wspominano wcześniej, wszystkie pliki będące wynikiem eksperymentu wraz z danymi, na których eksperyment przeprowadzano, zostają spakowane i są dostępne dla użytkownika.



Rys. 4. Podgląd pliku out.txt z wynikami klasyfikacji
Fig. 4. View of the out.txt file, with classification results

5. Podsumowanie

System netTRS przedstawiony w niniejszym artykule umożliwia zdalną realizację zadań analizy danych za pomocą narzędzi udostępnianych przez bibliotekę TRS. Do chwili obecnej bibliotekę można było użytkować jako standardowy program wyposażony w interfejs graficzny lub za pomocą interpretera skryptów sterującego zadaniami do wykonywania. System netTRS pozwala na udostępnienie możliwości, jakie oferuje biblioteka dużo szerszemu gronu użytkowników, którzy za pomocą przeglądarki internetowej mogą korzystać z algorytmów w niej zaimplementowanych.

System został zaprojektowany w taki sposób, by móc jak najłatwiej dostosowywać go do rozwijających się możliwości samej biblioteki. Sposób wykonywania eksperymentów polegający na tym, iż każdy eksperyment realizowany jest jako osobny proces obliczeniowy, sprawia, że system przygotowany jest do rozproszonego wykonywania obliczeń na komputerach działających w sieci komputerowej. W chwili obecnej prowadzone są prace nad takim właśnie rozszerzeniem funkcjonalności serwisu.

W chwili obecnej system w dalszym ciągu poddawany jest testom w Zakładzie Teorii i Projektowania Systemów Komputerowych, Instytutu Informatyki Politechniki Śląskiej. System dostępny jest pod adresem http://157.158.55.79/netTRS .

LITERATURA

1. Bazan J.: Metody wnioskowań aproksymacyjnych dla syntezy algorytmów decyzyjnych. Praca doktorska. Uniwersytet Warszawski. Wydział Matematyki, Informatyki i Mechaniki, 1998. Promotor prof. dr hab. Andrzej Skowron.
2. Bruha I.: Quality of Decision Rules: Definitions and Classification Schemes for Multiple Rules. Nakhaeizadeh G., Taylor C.C. (ed.), Machine Learning and Statistics, The Interface. John Wiley and Sons, 1997.
3. Bishop Y., Fienberg S., Holland P.: Discrete Multivariate Analysis: Theory and Practice. MIT Press, Cambridge, MA, 1991.
4. Grzymała-Busse J., Wang C. P.: Classification Methods in Rule Induction, Intelligent Information Systems. Proceedings of the Workshop held in Dęblin, Poland 2-5 June, 1996, pp. 120-126.
5. Goldberg D. E.: Algorytmy genetyczne i ich zastosowania. Wydawnictwo Naukowo-Techniczne, Warszawa 1998.
6. Nguyen H. S., Nguyen S. H.: Some efficient algorithms for rough set methods. Proceedings of the Sixth International Conference, Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'96) 2, July 1-5, Granada, Spain pp. 1451-1456.
7. Pawlak Z.: Rough sets: Theoretical aspects of reasoning about data. Dordrecht: Kluwer, 1991.
8. Sikora M.: Approximate decision rules induction algorithm using rough sets and rule-related quality measures. Archiwum Informatyki Teoretycznej i Stosowanej Nr 4, 2004.
9. Sikora M.: Filtracja zbioru reguł decyzyjnych wykorzystująca funkcje oceny jakości reguł. Studia Informatica Vol. 46, No.4, Gliwice 2001.

10. Sikora M.: An algorithm for generalization of decision rules by joining. Foundation on Computing and Decision Sciences, Vol. 30, No. 3, 2005.
11. Sikora M., Proksa P.: Algorithms for generation and filtration of approximate decision rules, using rule-related quality measures. Bulletin of International Rough Set Society Vo. 5, No. 1/2. Proceedings of the International Workshop on Rough Set Theory and Granular Computing (RSTGC-2001).
12. Sikora M., Proksa M.: Induction of decision and association rules for knowledge discovery in industrial databases. DM-IEEE, IEEE International Conference of Data Mining, Brighton, 01-04, November 2004.
13. Skowron A., Rauszer C.: The Discernibility Matrices and Functions in Information systems. Słowiński R. (ed.): Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory, Dordrecht, Kluwer, 1992, pp. 331-362.
14. Stefanowski J.: Rough set based rule induction techniques for classification problems. Proc. 6-th European Congress for Intelligent Techniques and Soft Computing, vol.1, Achen, Sept. 7-10, 1998, pp.107-119.
15. Stepaniuk J.: Knowledge Discovery by Application of Rough Set Models. ICS PAS Reports No. 887, Warszawa, 1999.
16. Stepaniuk J., Krętowski M.: Decision System Based on Tolerance Rough Sets. Proceedings IIS 1995, June 5-9, Augustów, Poland, ICS Polish Academy of Sciences.

Recenzent: Dr inż. Grzegorz Drwal

Wpłynęło do Redakcji 8 marca 2006 r.

Abstract

This article describes a Web system netTRS, that makes possible to perform data analysis using tolerance rough set theory and web browser window as user interface.

As distinct from present applications, that also give functionality of TRS library and works in local mode, this system makes TRS library available for much more users, than it was possible before.

Every system user can define new data analysis task (choose data analysis schema, upload data files, point appropriate algorithms and their parameters) via a set of internet pages. The results of experiments are also available via internet; both as a embedded text file and as a compressed archive file with all input and output files.

Another advantage of netTRS system is its module architecture. Each part performs separate tasks: files upload and results presentation (one module) and data analysis (second module). This kind of open architecture makes possible to develop the system in future.

Further development work forecast to adapt this system for collaboration with other web browsers and creation of new module, that distributes separate data analysis tasks between more than one computer.

It is easy to notice, that netTRS system fulfils two basic demands: makes it easier to use TRS library as a tool for induction and postprocessing of decision rules and allows for developing it in brand new directions.

Adresy

Marcin MICHALAK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska, Marcin.Michalak@polsl.pl.
Marek SIKORA: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska, Marek.Sikora@polsl.pl.