STUDIA INFORMATICA 2001
Volume 22 Number 4 (46)

Marek SIKORA
Politechnika Śląska, Instytut Informatyki

FILTRACJA ZBIORU REGUŁ DECYZYJNYCH WYKORZYSTUJĄCA FUNKCJE OCENY JAKOŚCI REGUŁ

Streszczenie. W pracy przedstawiono algorytmy pozwalające ograniczyć liczbę reguł decyzyjnych wykorzystywanych w opisie i klasyfikacji. Prezentowane algorytmy wykorzystują funkcje oceniające jakość reguł do wyselekcjonowania reguł najbardziej istotnych z punktu widzenia klasyfikacji i opisu. Omówiono kilka funkcji oceniających jakość reguły. Zaprezentowano wyniki przeprowadzonych eksperymentów. Do wstępnego wyznaczania zbioru reguł decyzyjnych zastosowano tolerancyjny model zbiorów przybliżonych [1].

FILTERING OF DECISION RULES SET USING RULES QUALITY FUNCTION

Summary. In this article the algorithms allowing to decrease the number of decision rules using in description and classification were introduced. Presented algorithms use rules quality functions for selection the most important in classification and description rules. Several rules quality functions were discussed. The results of the experiments were presented. To generation decision rules set the tolerance based rough sets model was used.

1. Wstęp

Gwałtowny wzrost liczby i wielkości baz danych przyczynił się do dynamicznego rozwoju nowych algorytmów poszukujących prawidłowości w danych. Dziedzina informatyki badająca własności takich algorytmów znana jest pod nazwą "pozyskiwania wiedzy z baz danych" (*ang. knowledge discovery in databases*). Pozyskiwanie wiedzy definiowane jest jako

nietrywialny proces poszukiwania prawdziwych, wcześniej nieznanych, zrozumiałych i użytecznych wzorców z danych.

Zbiór danych reprezentowany jest w postaci tablicy decyzyjnej $DT=(U,A\cup\{d\})$, U jest skończonym zbiorem obiektów, A skończonym zbiorem atrybutów warunkowych oraz d jest atrybutem decyzyjnym. Dla każdego z atrybutów $a\in A$ określony jest zbiór wartości przyjmowanych przez atrybut w zbiorze U, zbiór ten oznacza się D_a. Wzorzec występujący w danych opisujący zależności pomiędzy wartościami atrybutów ze zbioru A (atrybutami warunkowymi) a wartością atrybutu decyzyjnego może być reprezentowany jako reguła decyzyjna o postaci: $(a_1,V_{a_1})\wedge..\wedge(a_k,V_{a_k})\rightarrow(d,v)$. Dowolne (a,V_a) nazywane jest deskryptorem, $V_a\subseteq D_a$. Wartość $v\in D_d$ jest pewną wartością atrybutu decyzyjnego. Zapis (a,V_a) oznacza, że atrybut a przyjmuje wartości w V_a (V_a w zależności od typu atrybutu a będzie zbiorem lub przedziałem). W skrócie regułę decyzyjną można zapisać jako $\varphi\rightarrow\psi$, φ nazywamy poprzednikiem reguły, ψ nazywamy następnikiem reguły. O regułach postaci $\varphi\rightarrow(d,w)$ mówi się, że tworzą one opis (opisują) klasy decyzyjnej $X_w=\{u\in U:d(u)=w\}$. Zapis $d(u)=w$ oznacza, że obiekt u dla atrybutu d przyjmuje wartość w. Przyjmuje się również mówić, że klasą decyzyjną reguły $\varphi\rightarrow(d,w)$ jest klasa X_w.

Istnieje kilka metodologii wyznaczania zbioru reguł decyzyjnych z danych [1][2]. O jakości wyznaczonych reguł decyduje osiągana przez nie dokładność klasyfikacji. W celu wyznaczenia dokładności klasyfikacji zbiór danych poddawanych analizie dzieli się na dwie rozłączne podtablice: treningową i testową. Analizując dane z tablicy treningowej, wyznacza się opisy poszczególnych klas decyzyjnych. Na obiektach z tablicy testowej przeprowadzany jest proces klasyfikacji. W niniejszej pracy przyjęto następujący mechanizm klasyfikacji: dla obiektu testowego u wyznaczany jest stopień zaufania tego obiektu do każdej z klas decyzyjnych, zgodnie ze wzorem:

$$conf(X_v,u)=\sum_{r\in RUL_{X_v}(DT),dist(r,u)\leq\varepsilon}(1-dist(r,u))quality(r),$$

gdzie $dist(r,u)$ jest odległością obiektu testowego u od reguły r (np. Euklidesową, Hamminga), w szczególności jeżeli $\varepsilon=0$, to klasyfikacja odbywa się przez reguły pasujące, a o obiekcie u mówimy, że rozpoznaje on regułę r; $quality(r)$ jest miarą odzwierciedlającą jakość (siłę) reguły r. Funkcja decyzyjna f przyporządkowująca dowolnemu obiektowi u wartość atrybutu decyzyjnego określona jest wzorem:

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

Dokładnością klasyfikacji klasy decyzyjnej X_w będzie stosunek liczby obiektów testowych poprawnie zaklasyfikowanych do klasy X_w do liczby obiektów testowych faktycznie

należących do klasy X_w. Średnią dokładnością klasyfikacji będzie średnia arytmetyczna z dokładności klas decyzyjnych.

Proces klasyfikacji pozwala zweryfikować zbiór reguł pod względem ich możliwości klasyfikacji obiektów nieznanych. Z punktu widzenia odkrywania wiedzy w bazach danych bardzo istotne jest również to, aby w zbiorze reguł znajdowały się reguły odzwierciedlające ogólne zależności występujące w danych oraz aby zależności te były łatwe do interpretacji przez użytkownika. Warunek taki może być spełniony, kiedy liczba reguł opisujących daną klasę decyzyjną będzie mała i w zbiorze reguł będą znajdować się reguły uznane za silne względem zadanego kryterium. Najbardziej pożądaną cechą reguły jest to, aby zależność opisywana przez regułę była w jak największej ilości przypadków prawdziwa (reguła jak najdokładniejsza) oraz aby regułę taką rozpoznawało jak najwięcej obiektów (reguła mająca duże pokrycie).

Kiedy liczba reguł jest mała i każda z reguł posiada wyżej wymienione cechy, to zależności opisywane przez reguły mogą być zweryfikowane przez użytkownika oraz istnieje duże prawdopodobieństwo, że reguły odzwierciedlają rzeczywiste zależności występujące w danych i nie są to zależności wynikające z niedokładności lub wręcz błędów zawartych w danych treningowych.

Niezależnie od metodologii wyznaczania reguł decyzyjnych rozważane są różne metody ograniczania liczby reguł wchodzących w skład opisu klasy decyzyjnej. Do metod takich można zaliczyć:

- wyznaczanie reguł, które wystarczają do pokrycia zbioru treningowego [2][3],
- uogólnianie reguł [3][4]:
  - poprzez skracanie reguł,
  - poprzez sklejanie reguł ,
- filtrację zbioru reguł.

W szczególności na pierwszą z przedstawionych powyżej metod można spojrzeć jako na specjalny rodzaj filtracji.

W niniejszej pracy przedstawiono algorytmy filtracji reguł decyzyjnych, wykorzystujące funkcje oceniające jakość reguły do wskazywania reguł najlepszych w wejściowym zbiorze reguł. W drugim rozdziale omówiono kilka z funkcji oceniających jakość reguły, na końcu pracy przedstawiono wykresy omawianych funkcji. Trzeci rozdział zawiera opis stosowanych algorytmów filtracji. W czwartym rozdziale przedstawiono wyniki eksperymentów.

2. Funkcje oceniające jakość reguł

Funkcję określoną na zbiorze reguły *RUL*, przyjmującą wartości w zbiorze liczb rzeczywistych, można nazwać funkcją oceniającą jakość reguły:

$quality: RUL \rightarrow R$,

W zależności od przyjętego kryterium jakości funkcja oceniająca będzie miała różną postać. Wartości rozpatrywanych w artykule funkcji oceniających jakość reguły mogą być wyznaczone na podstawie znajomości tablicy kontyngencji [5], pozwalającej opisać zachowanie reguły w stosunku do treningowego zbioru obiektów.

Jak już wspominano, w zbiorze reguł decyzyjnych *RUL* każda reguła ma postać $\varphi \rightarrow \psi$. Dowolna reguła $\varphi \rightarrow \psi$ ustala dwa podziały zbioru U, każdy wyznaczany odpowiednio przez poprzednik i następnik reguły. Zbiór U można zapisać jako sumę zbiorów $U = U_{\varphi} \cup U_{\neg\varphi}$ oraz $U = U_{\psi} \cup U_{\neg\psi}$.

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:

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

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

$n_{\psi} = n_{\varphi\psi} + n_{\neg\varphi\psi} = card(U_{\psi})$ liczba obiektów należących do klasy decyzyjnej opisywanej przez regułę $\varphi \rightarrow \psi$;

$n_{\neg\psi} = n_{\varphi\neg\psi} + n_{\neg\varphi\neg\psi} = card(U_{\neg\psi})$ liczba obiektów nie należących do klasy decyzyjnej opisywanej przez regułę $\varphi \rightarrow \psi$;

$n_{\varphi\psi} = card(U_{\varphi} \cap U_{\psi})$ liczba obiektów wspierających regułę $\varphi \rightarrow \psi$;

$n_{\varphi\neg\psi} = card(U_{\varphi} \cap U_{\neg\psi})$; $n_{\neg\varphi\psi} = card(U_{\neg\varphi} \cap U_{\psi})$; $n_{\neg\varphi\neg\psi} = card(U_{\neg\varphi} \cap U_{\neg\psi})$.

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

Łatwo zauważyć, iż dla dowolnej reguły prawdziwe są nierówności $1 \leq n_{\varphi\psi} \leq card(U_{\psi})$, $0 \leq n_{\varphi\neg\psi} \leq card(U_{\neg\psi})$.

Dla ustalonej reguły $\varphi \rightarrow \psi \in RUL$ funkcja oceniająca jakość tej reguły może być zdefiniowana jako:

$$q_{\varphi\rightarrow\psi}: \{1,..,card(U_\psi)\} \times \{0,..,card(U_{\neg\psi})\} \rightarrow R,$$

posiadająca własność $q_{\varphi\rightarrow\psi}(<n_{\varphi\psi}, n_{\varphi\rightarrow\psi}>) = quality(\varphi\rightarrow\psi)$.

Można zdefiniować następujące funkcje oceniające jakość reguły:

1. Funkcja oceniająca dokładność reguły:

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

Wartość funkcji odzwierciedla dokładność reguły, czyli stosunek liczby obiektów wspierających regułę do liczby obiektów rozpoznających regułę. Dla reguł dokładnych wartość funkcji zawsze równa jest 1. Ocena reguły proponowana przez funkcję $q^{accuracy}$ nie uwzględnia, jak wiele obiektów z tablicy treningowej rozpoznaje regułę. Funkcja przyjmuje wartości w przedziale [0,1] i jest monotoniczna ze względu na swoje zmienne (rys. 1).

2. Funkcja oceniająca pokrycie reguły:

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

Wartość funkcji mówi o ilości obiektów z klasy decyzyjnej opisywanej przez regułę, które rozpoznają regułę w stosunku do ogólnej liczebności przykładów klasy decyzyjnej. Im więcej obiektów z klasy decyzyjnej reguły rozpoznaje regułę, tym pokrycie reguły jest większe. Ocena proponowana przez $q^{coverage}$ nie uwzględnia dokładności reguły. Funkcja przyjmuje wartości w przedziale [0,1] i jest monotoniczna ze względu na zmienne $n_{\varphi\psi}$ $n_{\varphi\rightarrow\psi}$. (ze względu na zmienną $n_{\varphi\rightarrow\psi}$ funkcja jest stała (rys. 2)).

3. Funkcje o genezie empirycznej:

Przedstawione poniżej dwie funkcje zostały zaproponowane odpowiednio przez R. S. Michalskiego [3] oraz przez autorów programu IREP [6]. Michalski proponuje funkcję o następującej postaci:

$$q^{Michalski}_{\varphi\rightarrow\psi}(<n_{\varphi\psi}, n_{\varphi\rightarrow\psi}>) = \frac{n_{\varphi\psi}}{n_\varphi} * w + \frac{n_{\varphi\psi}}{n_\psi} * (1-w), \quad w \in [0,1],$$

uwzględnia ona zarówno dokładność, jak i pokrycie reguły, parametr w wykorzystywany jest do wyrażania preferencji użytkownika, któremu w jednych sytuacjach bardziej może zależeć na dokładności reguły, w innych na jej pokryciu [3]. Widać, że zwiększając wartość w lepszą ocenę będą miały reguły o większej dokładności. Potwierdzono eksperymentalnie, że pokrycie reguły rośnie wraz ze spadkiem jej dokładności i odwrotnie. Kiedy użytkownik wie, jak wzrost pokrycia reguły powinien rekompensować spadek jej dokładności (lub odwrotnie), to można wyliczyć wartość wagi w, która zapewni pożądane proporcje

pomiędzy pokryciem i dokładnością reguły. Gdy dane są dwie reguły *r1* i *r2*, które mają mieć taką samą wartość oceny, to zapisując $q_{r1}^{Michalski} = q_{r1}^{accuracy} w + q_{r1}^{coverage}(1-w)$, żądaną wartość *w* można obliczyć z równości :

$$q_{r1}^{accuracy} w + q_{r1}^{coverage}(1-w) = q_{r2}^{accuracy} w + q_{r2}^{coverage}(1-w).$$

Funkcja proponowana przez Michalskiego przyjmuje wartości w zbiorze [0,1] i jest monotoniczna ze względu na swoje zmienne (rys. 3).

Programy uczące się reguł IREP [6] i RIPPER wykorzystują podobne postacie funkcji oceniających regułę. Funkcje te oceniając dany zbiór reguł utworzą identyczny ranking reguł. Poniżej przedstawiona jest funkcja wykorzystywana w programie IREP:

$$q_{\varphi\to\psi}^{IREP}(<n_{\varphi\psi}, n_{\varphi\to\psi}>) = \frac{n_{\varphi\psi} + |U_{\neg\psi}| - n_{\varphi\to\psi}}{|U|}$$

funkcja jest monotoniczna ze względu na swoje zmienne $n_{\varphi\psi}$, $n_{\varphi\to\psi}$ (rys. 4). Przyjmuje wartości w zbiorze [0,1], przy czym wartość 1 osiągana jest wtedy, kiedy reguła jest dokładna i pokrywa całą swoją klasę decyzyjną. Wartość funkcji uzależniona jest zarówno od dokładności, jak i pokrycia ocenianej reguły. Kiedy oceniana reguła jest dokładna ($n_{\varphi\to\psi}=0$), wtedy rozpatrywane jest pokrycie reguły. Kiedy reguła jest niedokładna (aproksymacyjna), liczba $n_{\varphi\to\psi}$ zapobiega sytuacji, w której dwie reguły o tym samym pokryciu, a różnej dokładności uzyskują taką samą ocenę.

Za funkcje o genezie empirycznej można również uznać funkcje proponowane przez Brazdila [5] oraz Kaufmana w programie PROMISE [3].

4. Funkcja Pearsona:

$$q_{\varphi\to\psi}^{Pearson}(<n_{\varphi\psi}, n_{\varphi\to\psi}>) = \frac{\left(n_{\varphi\psi} n_{\neg\varphi\to\psi} - n_{\varphi\to\psi} n_{\neg\varphi\psi}\right)^2}{n_{\psi} n_{\neg\psi} n_{\varphi} n_{\neg\varphi}}$$

jest w statystyce matematycznej wykorzystywana do weryfikacji hipotezy o niezależności dwóch jakościowych cech niemierzalnych [7] (test niezależności χ^2). Badana populacja opisana jest w tablicy kontyngencji. Gdy cechy są niezależne, co w rozważanym w pracy przypadku znaczy, że dokładność reguły jest identyczna jak dokładność wynikająca z rozkładu przykładów pomiędzy klasy decyzyjne w zbiorze treningowym, wartość funkcji równa jest zero ($q_{\varphi\to\psi}^{Pearson}=0$). Wartość funkcji $q_{\varphi\to\psi}^{Pearson}$ zależy od siły związku badanych cech, nie jest jednak sygnalizowany kierunek tej zależności, dlatego z punktu widzenia oceny reguły należy tak zmodyfikować obliczanie wartości tej funkcji, że jeżeli spełniony jest warunek $\frac{n_{\varphi\psi}}{n_{\varphi\to\psi}} < \frac{n_{\psi}}{n_{\neg\psi}}$, to $q_{\varphi\to\psi}^{Pearson}(<n_{\varphi\psi}, n_{\varphi\to\psi}>) := -q_{\varphi\to\psi}^{Pearson}(<n_{\varphi\psi}, n_{\varphi\to\psi}>)$. Poprawka taka pozwoli oceniać wszystkie reguły nawet te bardzo słabe, które bez jej wprowadzenia

mogłyby zostać uznane za reguły silne. Modyfikacja taka będzie miała duże znaczenie, gdy funkcję $q_{\varphi\to\psi}^{Pearson}$ stosuje się jako kryterium optymalności w algorytmie budowy lub skracania reguł decyzyjnych oraz do oceny progów tolerancji [4].

Funkcja $q_{\varphi\to\psi}^{Pearson}$ bez poddania jej modyfikacji nie jest monotoniczna ze względu na swoje zmienne $n_{\varphi\psi}$, $n_{\varphi\to\psi}$ (rys. 5), przyjmuje wartości w przedziale [0,1]. Po modyfikacji funkcja przyjmuje wartości w przedziale [-1,1]. Można zauważyć, że wartość funkcji może być nieokreślona jedynie w przypadku, kiedy $n_{\neg\varphi}=0$, dzieje się tak wtedy, gdy regułę rozpoznają wszystkie przykłady treningowe. Taką regułę należy uznać za bardzo słabą i przyporządkować jej wartość -1 lub 0, w zależności od tego czy zmodyfikowano postać funkcji czy nie.

Inne funkcje wywodzące się ze statystyki matematycznej (zwłaszcza z dwuwymiarowej analizy dyskretnej) można znaleźć w [5] [7].

5. Funkcja mająca swoją genezę w teorii informacji

Teoria informacji (teoria sygnałów), a zwłaszcza pojęcie entropii było wykorzystywane w pracach różnych badaczy dotyczących optymalnej konstrukcji drzewa decyzyjnego [8]. Funkcje oceny reguł decyzyjnych mające swoją genezę w teorii informacji można zdefiniować adaptując do tego celu znane z metod indukcji drzew decyzyjnych pojęcie korzyści informacji *Gain* i *LimitedGain* [8]:

$$q_{\varphi\to\psi}^{Gain}(\langle n_{\varphi\psi}, n_{\varphi\to\psi}\rangle) = \mathrm{Info}(U) - \mathrm{Info}_{\varphi\to\psi}(U).$$

$$Info(U) = -\left[\frac{n_\psi}{|U|}log_2\frac{n_\psi}{|U|} + \frac{n_{\neg\psi}}{|U|}log_2\frac{n_{\neg\psi}}{|U|}\right]$$ to entropia zbioru przykładów treningowych.

$$Info_{\varphi\to\psi}(U) = \frac{n_\varphi}{|U|}Info(\varphi\to\psi) + \frac{|U|-n_\varphi}{|U|}Info(\neg(\varphi\to\psi))$$

gdzie $Info(\varphi\to\psi)$ – entropia zbioru przykładów pokrywanego przez regułę $\varphi\to\psi$, $Info(\neg(\varphi\to\psi))$ – entropia zbioru przykładów nie pokrywanego przez regułę $\varphi\to\psi$, (zbiór U- to wtedy zbiór wszystkich przykładów rozpoznających regułę $\varphi\to\psi$).

Podobnie jak funkcja Pearsona, w przypadku kiedy spełniony jest warunek $\frac{n_{\varphi\psi}}{n_{\varphi\to\psi}} < \frac{n_\psi}{n_{\neg\psi}}$, celowa wydaje się zmiana znaku wartości funkcji. Przed wyznaczeniem wartości funkcji należy sprawdzić, czy badana reguła jest dokładna, tzn. $n_{\varphi\to\psi}=0$, jeżeli tak jest, należy przyjąć, że $Info(\varphi\to\psi)=0$ [8]. Prezentowana funkcja nie jest monotoniczna ze względu na każdą swoją zmienną, zakres zmienności funkcji to przedział [0,1].

Za funkcje mające swoją genezę w teorii informacji należy uznać również funkcję proponowaną przez I. Kanonenkę i I. Bratko [5] oraz funkcję stosowaną w programie AQ18 [3].

3. Filtracja zbioru reguł

Zbiór reguł po filtracji powinien charakteryzować się następującymi cechami: mieć zdolność klasyfikacji podobną do nie filtrowanego zbioru reguł, liczba reguł w odfiltrowanym zbiorze powinna być możliwie najmniejsza.

Wyraźnie zarysowują się dwa podejścia do filtracji reguł decyzyjnych:

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

Pierwsze podejście reprezentuje poniższy algorytm wyznaczania pokrycia zbioru treningowego.

Algorytm generujący pokrycie zbioru treningowego (Filtracja 1)
dane: *q – wybrana funkcja oceniająca regułę; RUL – zbiór reguł; U –obiekty treningowe*
deklaracje: *RUL`- przefiltrowny zbiór reguł*
początek
RUL`:=∅
posortuj malejąco reguły według wartości funkcji
oceniającej q, RUL:= $< rule_1,..,rule_{card(RUL)} >$
i=1;
dopóki *card(U)>0*
U:=U\{u : dist(u,rule$_i$)=0 };
RUL`:=RUL`∪{rule$_i$};
RUL=RUL\({rule$_i$}∪{r ∈RUL: r została wyznaczona z obiektu o, dla którego dist(o, rule$_i$)=0, oraz *d(u)=d(o)});*
i:=i+1;
zwróć jak wynik *RUL`*
koniec

W powyższym algorytmie najpierw wyznaczany jest ranking wszystkich reguł, a następnie budowę pokrycia zbioru treningowego rozpoczynamy od najlepszej reguły. Warto zauważyć, iż do prawidłowego działania algorytmu potrzebna jest wiedza o obiektach, z których wyznaczono poszczególne reguły, można to uznać za ograniczenie algorytmu. Algorytmu w przedstawionej postaci nie można zastosować do dostarczonego z zewnątrz (tzn. nie wyznaczanego tym samym programem) zbioru reguł.

Inne podejście do filtracji reprezentują dwa algorytmy prezentowane poniżej. Oba algorytmy, poza rankingiem reguł utworzonym przez wybraną funkcję oceniającą jakość reguły, wykorzystują także informację o uzyskanej przez wszystkie reguły dokładności klasyfikacji. Aby wyniki filtracji były niezależne od danych testowych, ze zbioru treningowego konieczne jest wyznaczenie tzw. zbioru strojącego (tuningowego), na którym przeprowadzana będzie początkowa klasyfikacja, w szczególnym przypadku zbiorem tym może być cały zbiór treningowy.

Algorytm (Filtracja 2)
dane *Reguły[liczba klas decyzyjnych][liczba reguł dla klasy] – wejściowy zbiór reguł podzielonych na opisy klas decyzyjnych, q – funkcja oceny jakości reguły, T- zbiór treningowy lub wydzielony z niego zbiór tuningowy, d – klasa decyzyjna, Dokładność[d] – dokładność klasyfikacji klasy d na zbiorze T dla nie filtrowanego zbioru reguł*
deklaracje *NoweReguły[liczba klas decyzyjnych][] – przefiltrowany zbiór reguł, r - reguła decyzyjna, NieMaRegul, Stop - zmienne boolowskie,*
początek
dla każdej klasy decyzyjnej d
Sortuj Reguły[d] malejąco wg wartości funkcji q
NoweReguły[d][0] := Reguły[d][0]
i := 0
powtarzaj
i :=i + 1; NieMaRegul := Tak;
Stop := Tak
dla każdej klasy decyzyjnej d
jeżeli i < Reguły[d].LiczbaRegul
NieMaRegul := Nie
jeżeli dokładność klasyfikacji klasy d na zbiorze T z użyciem reguł NoweReguły jest mniejsza niż Dokładność[d]
Stop := Nie
r := Reguły[d][i]
dodaj na próbę do NoweReguły[d] regułę r
jeżeli po dodaniu do zbioru NoweReguły reguły r nie wzrosła dokładność klasyfikacji klasy d lub średnia dokładność klasyfikacji, usuń z NoweReguły[d] dodaną na próbę regułę r
dopóki NieMaRegul lub Stop
zwróć jako wynik NoweReguły
koniec

Początkowy opis każdej klasy decyzyjnej składa się z jednej – najlepszej reguły, następnie dodajemy po jednej regule do opisu każdej klasy, jeżeli dokładność klasy się zwiększy, regułę pozostawiamy w opisie, jeżeli nie, rozpatrujemy kolejną regułę. O kolejności rozpatrywania reguł decyduje ranking reguł ustalony przez funkcję 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. Ze względu na kryterium decydujące o dodaniu reguły do nowego opisu klasy decyzyjnej algorytm nie gwarantuje, iż odfiltrowane reguły uzyskają identyczną dokładność klasyfikacji jak zbiór wszystkich reguł.

Można zmienić algorytm podając maksymalny spadek dokładności klasyfikacji, na jaki godzi się użytkownik bądź maksymalną liczbę reguł dla każdej klasy decyzyjnej, wpływając tym samym na warunki zatrzymania pracy algorytmu.

Powyższy algorytm można nazwać *filtracją w przód*, idea tego algorytmu jest przeciwna do przedstawionego poniżej algorytmu filtracji polegającego na usuwaniu ze zbioru reguł, reguł niepotrzebnych do klasyfikacji. Usuwanie reguł zaczyna się od najsłabszych reguł z każdej klasy decyzyjnej, takie podejście do filtracji można nazwać *filtracją w tył*.

Algorytm (Filtracja 3)

dane: *Reguły[][] – tablica reguł decyzyjnych z podziałem na klasy decyzyjne,*
max_d – liczba reguł decyzyjnych opisujących najliczniejszą klasę decyzyjną
q – funkcja oceniająca jakość reguły

początek

dla każdej klasy decyzyjnej d
sortuj malejąco Reguły[d] według wartości funkcji q;
Acc:= średnia dokładność klasyfikacji z użyciem reguł Reguły;
DiffAcc:= różnica między najwyższą a najniższą dokładnością klasyfikacji klas decyzyjnych;
dla i:=0 do max_d
dla każdej klasy decyzyjnej d
jeżeli i<Reguły[d].LiczbaReguł
usuń na próbę regułę Reguły[d][Reguły[d].LiczbaReguł –i]
jeżeli po usunięciu reguły średnia dokładność klasyfikacji zmalała w stosunku do Acc lub po usunięciu reguły zwiększyła się różnica DiffAcc
zwróć usuniętą regułę;
zwróć jako wynik Reguły;

koniec

Utrzymanie różnicy w dokładności pomiędzy najdokładniejszą a najmniej dokładną klasą decyzyjną gwarantuje, że odfiltrowany zbiór reguł decyzyjnych zachowa taką samą czułość [3] jak zbiór nie filtrowany. Można do algorytmu wprowadzić parametry *DeltaAcc i DeltaDiffAcc* pozwalające użytkownikowi wyrażać swoje preferencje dotyczące akceptowanego przez niego spadku dokładności klasyfikacji bądź straty czułości klasyfikatora.

Na koniec warto wspomnieć, iż można w prosty sposób odfiltrować zbiór reguł poprzez ustalenie granicznej wartości funkcji oceniającej regułę, jaką musi przekroczyć reguła, aby wejść w skład opisu klasy decyzyjnej. Wartość taka może zostać wyznaczona arbitralnie przez użytkownika lub może zostać wyznaczona adaptacyjnie, tak aby dokładność klasyfikacji osiągana na zbiorze tuningowym była jak najwyższa.

We wszystkich przedstawionych algorytmach filtracji można używać innego kryterium decydującego o tym, czy dodać (usunąć – w przypadku filtracji w tył) regułę do (z) wyjściowego opisu klasy decyzyjnej. Kryterium takim może być wartość funkcji:

$$evaluate(RUL') = \lambda * accuracy(RUL') + (1-\lambda) * simplicity(RUL'),$$

gdzie: RUL' – reguły wybrane w bieżącym etapie filtracji, $\lambda \in [0,1]$, RUL – zbiór wszystkich reguł, $simplicity(RUL') = \frac{card(RUL) - card(RUL')}{card(RUL)}$.

4. Eksperymenty z danymi

Do przetestowania działania przedstawionych algorytmów wybrano zbiór danych znany w literaturze jako Australian-Credit (dane z projektu Statlog). W zbiorze danych znajduje się blisko 700 obiektów opisanych 15 atrybutami typu nominalnego i ciągłego. Prezentowane wyniki uzyskano wykorzystując metodologię testowania znaną jako 10-fold cross-validation. Klasyfikacja odbywała się przez reguły pasujące, tzn. takie, których odległość od obiektu testowego była równa zero. Obiekty testowe nierozpoznane w procesie klasyfikacji uznawane były za sklasyfikowane błędnie. Filtracji poddawano zbiór reguł wyznaczony metodą prezentowaną w pracy [4], polegającą na sekwencyjnym budowaniu reguły decyzyjnej na podstawie: informacji o zdolności rozróżniania obiektów przez poszczególne atrybuty warunkowe, wartości funkcji oceniającej regułę.

Zbiorem strojącym – tuningowym, wykorzystywanym w algorytmach filtracji był cały zbiór uczący. W procesie klasyfikacji o sile reguły zawsze decydowała dokładność reguły (funkcja *accuracy)*.

Tabela 1

Wyniki algorytmu *Filtracja 1*

Liczba wszystkich reguł 63, dokładność klasyfikacji 0.85, rozpoznanie obiektów testowych 100%.						
	accuracy	*coverage*	*Michalski W=0.5*	*irep*	*pearson*	*gain*
Liczba reguł po filtracji	*23*	*8*	*10*	*12*	*10*	*2*
Dokładność klasyfikacji po filtracji	*0.76*	*0.80*	*0.82*	*0.82*	*0.86*	*0.87*
Rozpoznanie obiektów po filtracji	*100%*	*100%*	*100%*	*100%*	*100%*	*100%*

Tabela 2

Wyniki algorytmu *Filtracja 2*

Liczba wszystkich reguł 63, dokładność klasyfikacji 0.85, rozpoznanie obiektów testowych 100%.						
	accuracy	*coverage*	*Michalski W=0.5*	*irep*	*pearson*	*gain*
Liczba reguł po filtracji	-	*2*	*2*	*2*	*2*	*2*
Dokładność klasyfikacji po filtracji	-	*0.84*	*0.87*	*0.87*	*0.87*	*0.87*
Rozpoznanie obiektów po filtracji	-	*97%*	*100%*	*100%*	*100%*	*100%*

Tabela 3

Wyniki algorytmu *Filtracja 3*

Liczba wszystkich reguł 63, dokładność klasyfikacji 0.85, rozpoznanie obiektów testowych 100%.						
	accuracy	*coverage*	*Michalski W=0.5*	*irep*	*pearson*	*gain*
Liczba reguł po filtracji	*14*	*16*	*15*	*15*	*13*	*14*
Dokładność klasyfikacji po filtracji	*0.76*	*0.80*	*0.82*	*0.83*	*0.84*	*0.85*
Rozpoznanie obiektów po filtracji	*91%*	*98%*	*100%*	*100%*	*100%*	*100%*

Jak wspominano, algorytm *Filtracja 1* można potraktować jako algorytm budowania pokrycia zbioru treningowego za pomocą reguł decyzyjnych. Budując pokrycie zbioru treningowego poczynając od pierwszego (losowego) obiektu, z tablicy treningowej osiągnięto następujące wyniki: dokładność klasyfikacji 0.85, liczba reguł 16, rozpoznanie obiektów testowych 100%. Wyniki zawarte w tabeli 1 sugerują, że lepiej budować pokrycie zbioru treningowego poczynając od obiektu, z którego otrzymano najlepszą regułę niż z losowo wybranego obiektu.

W tabeli 2 nie przytoczono wyników dotyczących filtracji wykorzystującej do oceny reguły dokładność reguły, gdyż były one dramatycznie złe. Ze względu na bardzo małą zdolność rozpoznawania obiektów testowych przez przefiltrowany zbiór reguł, w algorytmie *Filtracja 2* nie należy wykorzystywać funkcji oceniającej dokładność reguły.

Algorytm *Filtracja 3* podobny jest do często stosowanego algorytmu zmniejszającego liczbę reguł, polegającego na kolejnym, losowym usuwaniu reguły i mierzeniu dokładności klasyfikacji tak zmniejszonego zbioru reguł. Jeżeli po usunięciu reguły wartość współczynnika klasyfikacji nie maleje, znaczy to, że reguła jest zbędna, w przeciwnym przypadku nie można usunąć reguły. Z przytoczonych wyników widać, że filtracja realizowana za pomocą trzeciego algorytmu daje najgorsze wyniki.

5. Podsumowanie

Spośród proponowanych algorytmów filtracji zbioru reguł najlepszy w działaniu okazuje się być algorytm *Filtracja 2*. Dla prezentowanego zbioru danych wykorzystanie w algorytmie różnych funkcji oceniających nie miało istotnego wpływu na liczbę reguł tworzących ostateczny opis klas decyzyjnych i uzyskiwaną dokładność klasyfikacji.

Wyniki algorytmu *Filtracja 1* były już uzależnione od użytej funkcji oceniającej regułę. Zastosowanie tego algorytmu pozwoliło uzyskać większą dokładność klasyfikacji i mniejszy opis klas decyzyjnych, niż zapewniał to algorytm wyznaczający reguły wystarczające do pokrycia zbioru przykładów treningowych.

Algorytm *Filtracja 3*, pomimo że pozwalał na zmniejszenie liczby reguł w stosunku do liczby reguł znajdujących się w początkowym opisie klas decyzyjnych, osiągał najgorsze wyniki spośród prezentowanych algorytmów. Przeprowadzone eksperymenty pokazują, że lepszymi metodami filtracji są metody wykorzystywane w algorytmach *Filtracja 1* lub *Filtracja 2*.

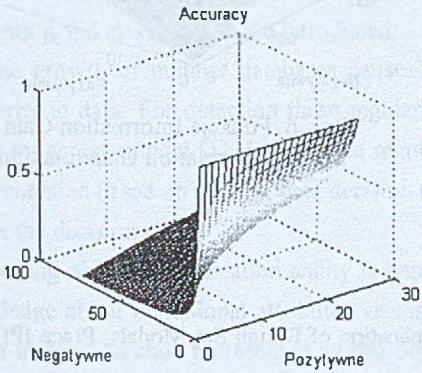
Przedstawione funkcje oceniające regułę potwierdziły swoją użyteczność. Wykorzystanie innych niż dokładność i pokrycie funkcji oceniających regułę pozwala, w algorytmach filtracji

i uogólniania (skracania, sklejania) reguł, uzyskiwać dobre wyniki w zakresie osiąganej dokładności klasyfikacji i zmniejszenia opisu klas decyzyjnych.

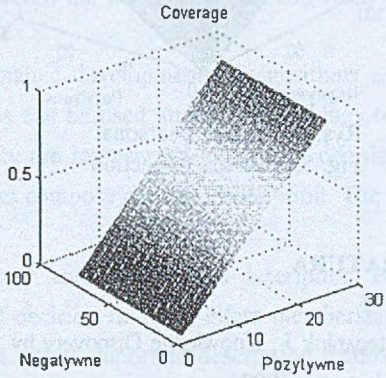
Dalsze prace będą polegać na wykonaniu większej ilości eksperymentów w celu zdobycia materiału porównującego działanie poszczególnych funkcji oceniających. Porównanie takie dotyczyć będzie zarówno osiąganej po filtracji dokładności klasyfikacji, jak i liczby reguł w końcowych opisach klas decyzyjnych.

6. Wykresy funkcji

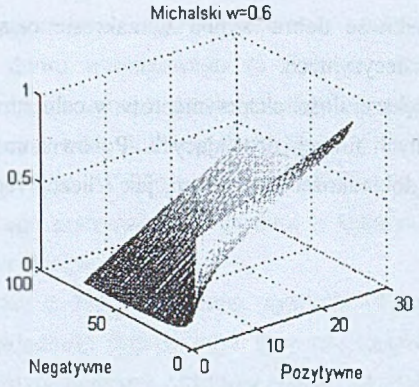
Na rysunkach przyjęto następujące oznaczenia: oś opisana jako *Negatywne* odzwierciedla zakres zmienności argumentu $n_{\varphi\neg\psi}$, oś opisana jako *Pozytywne* odzwierciedla zakres zmienności argumentu $n_{\varphi\psi}$. Wykresy rysowane są dla następujacego rozkładu przykładów pomiędzy klasy decyzyjne: liczba przykładów – 100, liczba przykładów z klasy reguły – 25, liczba pozostałych przykładów 75. W zależności od rozkładu przykładów pomiędzy klasy decyzyjne wykresy funkcji mogą wyglądać inaczej, nie zmienia to jednak własności funkcji opisanych w rozdziale pierwszym. Dla funkcji Pearsona i Information Gain przedstawione są wykresy przed modyfikacją.



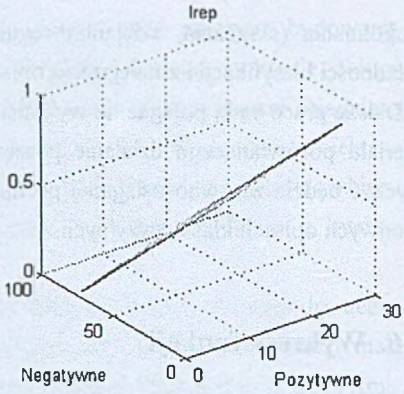
Rys. 1. Funkcja dokładność
Fig. 1. Accuracy function



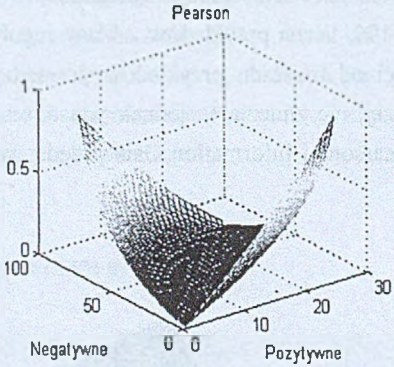
Rys. 2. Funkcja pokrycie
Fig. 2. Coverage function



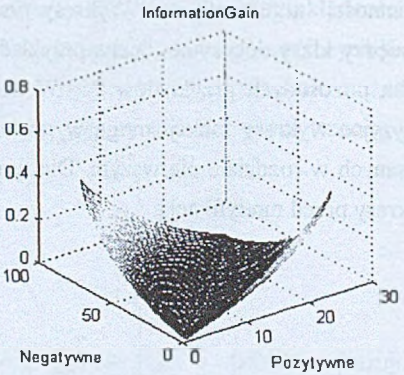
Rys. 3. Funkcja Michalskiego
Fig. 3. Michalski function



Rys. 4. Funkcja IREP
Fig. 4. IREP function



Rys. 5. Funkcja Pearsona
Fig. 5. Pearson`s function



Rys. 6. Funkcja Information Gain
Fig. 6. Information Gain function

LITERATURA

1. Stepaniuk J.: Knowledge Discovery by Application of Rough Set Models, Prace IPI PAN, Warszawa 1999.
2. Grzymała-Buse J.: LERS system for learning from examples based on rough sets, Handbook of App. and Advances of the Rough Sets Theory, R. Słowiński ed., Kluwer Academic Pub., 1992.
3. Kaufman K. A., Michalski R. S.: Learning in Inconsistent World, Rule Selection in STAR/AQ18, Machine Learning and Inference Laboratory, Report, P99-2, 1999.
4. Sikora M., Proksa. P: Wyznaczanie progów tolerancji i aproksymacyjnych reguł decyzyjnych, Studia Informatica, No. 1 (42), vol. 22, Gliwice 2001.

5. Bruha I.: Quality of decision rules: Definitions and classification schemes for multiple rules. Machine Learning and Statistics:The Interface, John Wiley and Sons, 1995.
6. Furnkranz J., Widmer G.: Incremental Reduced Error Pruning, Proceedings of the Eleventh International Conference of Machine Learning, New Brunswick, NJ, 1994.
7. Bishop Y., Fienberg S., Holland P.:Discrete Multivariate Analysis: Theory and Practice, MIT Press, Cambridge, MA, 1991.
8. Quinlan J.R.,C4.5 Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, California, 1993.
9. Sikora M., Proksa P. Wyznaczanie Progów Tolerancji i Aproksymacyjnych Reguł Decyzyjnych, Studia Informatica, No. 3(44), 2001 (w druku).

Recenzent: Dr hab. Jarosław Stepaniuk, prof. Pol. Białostockiej

Wpłynęło do Redakcji 21 lutego 2001 r.

Abstract

In this article the algorithms allowing to decrease the number of decision rules using in description and classification are introduced.

The growth of number databases causes dynamic development of algorithms searching regularity in data. For detection these regularities can be used among other things tolerance based rough sets model [1]. In this case regularity are represented in form of decision rules. The generated based on training data decision rules compose concept description. The concept means the decision class.

Testing objects classification ability is important feature of created description. Based on knowledge about conditional attributes values of decision rule one infers membership of the object in decision class represented by this object. Other important description feature should be its conciseness and user's ability of interpretation. The description composed of minimal number of rule is the conciseness description.

In this article three heuristic algorithms which decrease number of decision class description rules are introduced. The algorithms use rule quality functions to select the best rules from the initial description. Six quality functions are discussed. Charts of these functions are showed in last chapter. Based on them, some properties are indicated and some modifications of presented functions are proposed.

Obtained results of experiments are placed in the chapter five. How the algorithm generating coverage of training set work is showed in Table 1. In Table 2 are presented the results of forward filtration algorithm action. Initially, empty set is description of decision class. The rule is added to the description if it causes increase of classification accuracy. The value of rule quality function decides about the order of adding the rule to the description. In Table 3 are presented the results of backward filtration algorithm action. Beginning from the worst rule, the rules don't caused the decrease of classification accuracy are removed.

Using in filtration process other then accuracy and coverage rule quality functions is postulate as a conclusion. Forward filtration algorithm is the best of presented algorithms.