POLITECHNIKA OPOLSKA
WYDZIAŁ ELEKTROTECHNIKI AUTOMATYKI I INFORMATYKI
INSTYTUT AUTOMATYKI I INFORMATYKI

Ewelina Piotrowska

Analiza parametrów morfometrycznych komórek dla komputerowego wspomagania diagnostyki medycznej

Rozprawa doktorska
przygotowana pod kierunkiem
dr hab. inż. Włodzimierza Stanisławskiego, prof. Politechniki Opolskiej

OPOLE 2011

Pragnę złożyć serdeczne podziękowania
Panu dr hab. inż. Włodzimierzowi Stanisławskiemu,
prof. Politechniki Opolskiej
za merytoryczne ukierunkowanie
niniejszej rozprawy doktorskiej,
cenne rady i dyskusje, które w sposób znaczący
przyczyniły się do jej powstania,
a także za cierpliwość, życzliwość i wsparcie
w trakcie realizacji i redagowania pracy.

Dziękuję firmie MetaSystems GmbH za udostępnienie
zbioru danych komórek pęcherza moczowego.

Niniejszą pracę dedykuję mojej rodzinie,
a w szczególności mężowi,
w podziękowaniu za pomoc, troskę i wyrozumiałość
oraz wszelkie słowa otuchy w czasie jej pisania.

Spis treści

ROZDZIAŁ 1

Wstęp

1.1. Wprowadzenie

„Diagnostyka jest dziedziną, która zajmuje się rozpoznawaniem badanego stanu rzeczy przez zaliczenie go do znanego typu lub gatunku, poprzez przyczynowe i całościowe wyjaśnienie tego stnu rzeczy, określenie jego fazy obecnej oraz przewidywanego dalszego rozwoju" [ChoKos02]. Pojęcie diagnostyki kojarzone jest często z medycyną, jako działaniem zajmującym się rozpoznawaniem chorób na podstawie objawów. Jednak wzrost zapotrzebowania na automatyczne metody oceny stanów obiektów sterowania sprawił, że pojęcie diagnostyki stało się przedmiotem badań w dyscyplinie automatyka i robotyka.

O ważności problemów diagnostyki może świadczyć ciesząca się dużym zainteresowaniem cykliczna konferencja „Diagnostyka Procesów i Systemów", realizowana od 1996 pod kierunkiem profesorów J. Korbicza, J.M. Kościelnego oraz Z. Kowalczuka. Poruszane tematy dotyczą prowadzonych prac badawczych i naukowych w zakresie diagnostyki leżącej na pograniczu automatyki, informatyki, medycyny oraz innych dyscyplin, które charakteryzują się systemowym podejściem do analizowanych problemów. W ostatnich latach można zaobserwować rosnące zainteresowanie rozwojem technik sztucznej inteligencji. Pozwalają one na powszechne stosowanie systemów doradczych, a także sformalizowanych metod pozyskiwania, gromadzenia, uzgadniania i uogólniania wiedzy [ChoKos02,Kos02,Pie03]. Nieustannie rozwijane są modele neuronowe, które oprócz możliwości uczenia, cechuje przydatność do modelowania nieliniowości, odporność na zakłócenia oraz zdolność do uogólniania wiedzy zawartej w sieci [DraSwi08,PatKor02,Rut07]. W przypadku, gdy wiedza o diagnozowanym obiekcie jest nieprecyzyjna to znajdują zastosowanie modele rozmyte bazujące na teorii zbiorów rozmytych zapoczątkowanej przez L.Zadeha [BrzSwi08,Kos02c,WalBla08,Wal07,Kos01,KosSyf02]. Szerokim zainteresowaniem w problemach diagnostyki cieszą się także algorytmy genetyczne lub ich kombinacje [ObuKor02,WitKor02,KowBia02,Tim07].

Nieustanny rozwój technik pomiarowych prowadzi do ciągłego gromadzenia licznych danych związanych z przebiegiem procesów. Gromadzone dane wykorzystywane są w układach monitorujących i diagnozujących, co wymaga poszukiwania narzędzi umożliwiających interpretację takich wyników [Cho02, Bub01, MocTom09]. Znajduje tutaj zastosowanie teoria rozpoznawania wzorców (ang. pattern recognition), w której odzwierciedla się zdolność ludzkiego umysłu do poznawania świata zewnętrznego i klasyfikacji „podobnych" zjawisk. Zaletą tej teorii jest możliwość budowy narzędzi diagnostycznych opartych na numerycznych danych obserwacji lub pomiarach diagnozowanego procesu [Roz79, TadOgi07, Dud07].

1.2. Teoria rozpoznawania wzorców w diagnostyce

Początek rozwoju teorii rozpoznawania wzorców związany jest z badaniami nad metodami sztucznej inteligencji przypadającymi na lata 50-te XX wieku [OgiTad09,Ros58,Grabo03]. Pierwsze techniki rozpoznawania bazowały na systemach sieci neuronowych znanych pod nazwą Perceptron, opracowanych przez F. Rosenbalta w 1958r. Perceptron zbudowany przez niego wraz z C. Wightmanem był częściowo elektromechanicznym, częściowo elektronicznym urządzeniem, którego przeznaczeniem było rozpoznawanie znaków alfanumerycznych. Osiągniecie to stało się przyczyną dalszego rozwoju metod rozpoznawania obrazów takich jak metody minimalno-odległościowe, metody aproksymacyjne czy metody probabilistyczne.

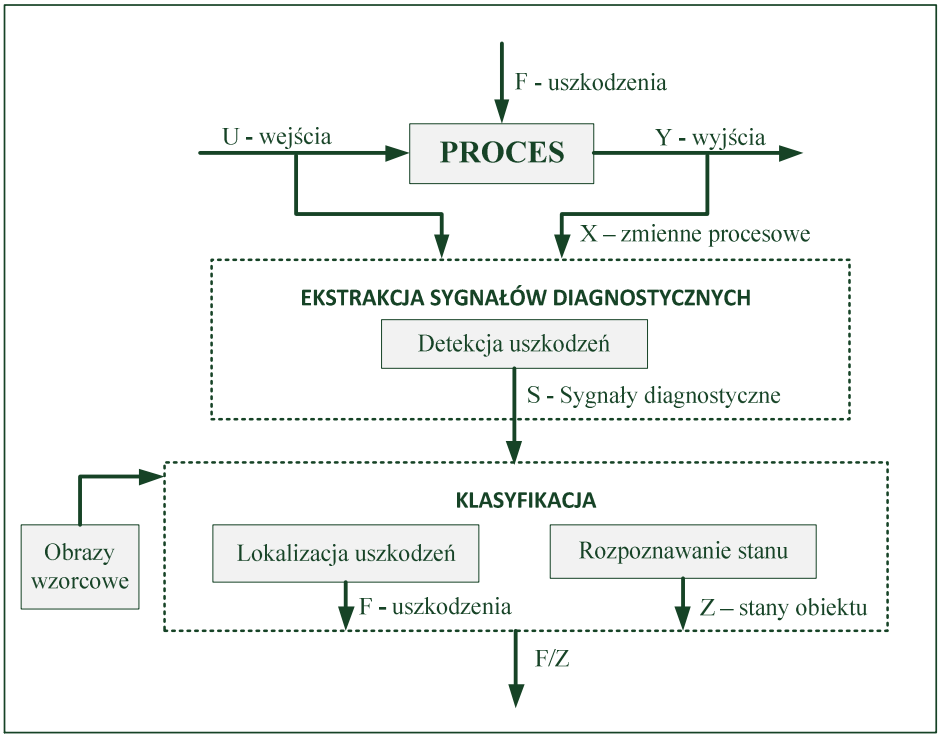
Dyscyplinami, w których zapoczątkowywano stosowanie teorii rozpoznawania są: medycyna, lingwistyka, kryminalistyka, itp. Przykładem analizowanych obiektów mogą być: sygnał kardiotokografii (KTG) [JeWro02,Tad91], sygnał mowy [Sho99,Tad88, Wsz09, Tra09], odcisk palca [WajWoj09], stany awaryjne urządzeń technicznych [KulŁuk09], itp. Każdy z wymienionych obiektów charakteryzuje się pewnym zestawem właściwości - cech, które odróżniają go od innych obiektów w zbiorze. Niektóre z cech mogą zawierać większą ilość informacji niż pozostałe. Dlatego jednym z zadań w teorii rozpoznawania jest minimalizacja opisu polegająca na poszukiwaniu takiego przekształcenia pierwotnej przestrzeni cech w pewną inną przestrzeń o mniejszym wymiarze, które nie doprowadzi do istotnego zwiększenia wartości funkcji straty. Nowy zestaw cech nazywa się podsystemem o największej wartości informacyjnej.

Przykładowy schemat diagnostyki bazującej na rozpoznawaniu wzorców przedstawiono na rys. 1.1. Rolę cech pełnią sygnały diagnostyczne. Na schemacie wyróżniono dwie fazy: ekstrakcja cech oraz klasyfikacja uszkodzeń lub stanów obiektu. Faza ekstrakcji cech polega na odwzorowaniu przestrzeni zmiennych procesowych X w przestrzeń sygnałów diagnostycznych S. W fazie klasyfikacji realizowane jest odwzorowanie przestrzeni sygnałów diagnostycznych S w przestrzeń uszkodzeń F lub stanów obiektu Z. Klasyfikacja przeprowadzana jest w oparciu o próbę uczącą. Na jej podstawie określane są obrazy wzorcowe dla wszystkich możliwych stanów obiektu lub klas [Tad91, Kos02b,ChoKos02,MarKor02,SobMal78].

Zwiększanie liczby cech pozwala na dokładniejsze opisanie zjawiska. Niestety wpływa negatywnie na możliwości interpretacji. Im więcej cech jest wykorzystywanych, tym większy jest rozmiar przestrzeni rozważań. Stanowi to także poważny problem obliczeniowy, który wynika ze złożoności pamięciowej i czasowej implementowanych algorytmów.

W analizie danych wielowymiarowych dąży się więc do redukcji przestrzeni cech, pozwalającej na zastąpienie danych pierwotnych zbiorami odpowiednio zagregowanymi i uporządkowanymi. Uogólniając, redukcję danych można przeprowadzić poprzez [SobMal78]:

- selekcję informacji polegającą na redukcji liczby cech opisujących obiekty;
- zastąpienie ciągłego zakresu zmienności wartości cech, wartościami w postaci dyskretnej;
- zmniejszenie liczby obiektów reprezentujących poszczególne klasy.



Rys. 1.1. Schemat diagnostyki jako procesu rozpoznawania wzorców, obejmujący fazę detekcji uszkodzeń oraz fazę lokalizacji uszkodzeń lub rozpoznawania stanu obiektu

1.3. Analiza problemu

Tematyka podjęta w rozprawie dotyczy problemu rozpoznawania wzorców. Liczne udoskonalenia istniejących metod, a także poszukiwania nowych algorytmów świadczą o tym, że żadne z opracowanych rozwiązań nie jest całkowicie wolne od wad. Można wyodrębnić dwa kierunki prowadzonych badań:

- budowanie specjalistycznych systemów rozpoznających; prace prowadzone są przez specjalistów z zakresu dziedzin technicznych [Sob08,GloPat07,HreKor07,WnuSyf09];
- budowa modeli matematycznych i algorytmów rozpoznawania obrazów, które mogą pełnić rolę narzędzi w badaniach naukowych [CiuUrb07,TadOgi02].

Główną motywację do realizacji badań prowadzonych przez autorkę stanowi problem analizy danych wielowymiarowych oraz potrzeba opracowania algorytmów rozpoznawania obiektów należących do klas o małej częstości występowania. Zagadnieniom rozpoznawania poświęca się wiele miejsca na konferencjach w kraju i zagranicą. Najczęściej jednak weryfikację prezentowanych metod przeprowadza się na zbiorach o małej liczebności obiektów lub o małej liczebności cech, co wpływa na znaczne skrócenie czasu obliczeń. Istotnym aspektem poprawiającym efektywność rozpoznawania jest także wykorzystywanie zbiorów danych, które charakteryzują się dobrymi uwarunkowaniami do separacji obiektów względem klas.

W pracy podjęto problem analizy parametrów morfometrycznych komórek, uzyskanych z systemu skaningowego Metafer firmy Metasystems, we wstępnej diagnostyce nowotworu pęcherza moczowego. Badania prowadzone są w oparciu o cechy ilościowe określone na podstawie analizy obrazów mikroskopowych. Wykorzystywany zbiór danych charakteryzuje się wysoką liczbą cech (212) oraz wysoką liczbą obiektów (ok. 23000). Zbiór danych charakteryzuje się dodatkowo niezrównoważonym rozkładem względem klas obiektów. W zbiorze wyodrębniono dwie

klasy: komórki zdrowe - liczące ok. 97% obiektów, oraz komórki nowotworowe liczące niecałe 3% obiektów. Klasa mniejszościowa (ang. minority class), odpowiadająca komórkom nowotworowym, zawiera zdecydowanie mniej obiektów niż klasa większościowa (ang. majority class). Ponieważ większość algorytmów uczących zakłada w przybliżeniu zrównoważenie klas, to opisane powyżej zachowanie powoduje trudności w fazie uczenia i obniża zdolność predykcyjną [Cha10, FeGa11, GaSa10,StWi05].

Podstawowym kryterium oceny metod klasyfikacji są pojęcia przedstawione w rozdziale 2.6: dokładność, czułość i swoistość. W optymalizacji algorytmów rozpoznawania wzorców szczególną uwagę zwrócono na czułość, która określa na ile klasyfikator jest zdolny do wykrywania przypadków z danej klasy oraz na swoistość, która określa na ile decyzja klasyfikatora o przynależności do wybranej klasy jest charakterystyczna wyłącznie dla tej klasy. Opracowane algorytmy optymalizowano w kierunku maksymalizacji czułości wykrywania komórek nowotworowych.

Drugi podjęty w pracy problem związany jest z poszukiwaniem zbioru cech umożliwiających wiarygodną identyfikację klas obiektów. We wstępnej diagnostyce nowotworu pęcherza moczowego klasyfikację komórek można przeprowadzić uwzględniając wszystkie wyznaczone cechy, których liczba wynosi 212. Dla większości klasyfikatorów jest to niecelowe ze względu na silne skorelowanie analizowanych cech, czy wysoką złożoność obliczeniową algorytmów.

O aktualności kierunków badań podjętych w pracy świadczy propozycja grupy roboczej Komitetu Automatyki i Robotyki Polskiej Akademii Nauk w sprawie opracowania wniosku o powołanie Strategicznego Programu Badawczego pod nazwą: „Rozszerzenie Internetu - Zrobotyzowane inteligentne systemy usługowe wspomagające człowieka”. Poruszane w pracy zagadnienia, związane z odpowiednim doborem zestawu cech, są elementem algorytmów przetwarzania i rozpoznawania obrazów (pkt 4.1). Innym ważnym problem badawczym są metody uczenia (pkt. 4.10). Zastosowane w pracy metody są przykładem algorytmów uczenia nadzorowanego.

W ramach prowadzonych prac autorka zrealizowała szereg analiz, które były motywacją do udoskonalania opracowywanych algorytmów. Wyniki prowadzonych prac naukowo-badawczych prezentowane były na konferencjach i seminariach. Dorobek naukowy autorki stanowi 15 publikacji, w tym 9 autorskich i 6 współautorskich. Wśród najważniejszych należy wymienić [StaSzy06,Szy07a,Szy07b,SzySta08,Szy08, Szy09a, PioSta11].

1.4. Sformułowanie problemu

Przedmiotem prowadzonych badań jest analiza danych wielowymiarowych z zastosowaniem metod rozpoznawania wzorców w celu diagnostyki. Szczególną uwagę zwrócono na zastosowanie zbiorów przybliżonych (ang. Rough Sets, RS) do redukcji liczebności zbioru cech oraz do klasyfikacji przypadków na podstawie dobranych cech.

1.5. Cel, teza i zakres pracy

1.5.1. Cel pracy

Celem pracy jest:

1. Opracowanie metody klasyfikacji nadzorowanej z zastosowaniem zbiorów przybliżonych i metody k-najbliższych sąsiadów z maksymalizacją czułości klasy mniejszościowej.
2. Opracowanie metody selekcji cech z zastosowaniem teorii zbiorów przybliżonych i analizy korelacyjnej.
3. Opracowanie metody wstępnej diagnostyki nowotworu pęcherza moczowego na podstawie analizy parametrów morfometrycznych komórek z systemu skaningowego.

Aby osiągnąć postawiony cel należy rozwiązać następujące zadania szczegółowe:

1. Opracowanie implementacji algorytmów teorii zbiorów przybliżonych w środowisku obliczeniowym MATLAB z zastosowaniem operacji wektorowych w zadaniach:
    a. Wyznaczanie zbiorów elementarnych.
    b. Wyznaczanie aproksymacji zbiorów, rodziny zbiorów i współczynników aproksymacji.
    c. Wyznaczanie rdzenia i rdzenia względnego.
    d. Wyznaczanie tablicy rozróżnialności systemu informacyjnego i decyzyjnego.
    e. Wyznaczanie funkcji rozróżnialności systemu informacyjnego i decyzyjnego.
    f. Wyznaczanie tablicy i funkcji rozróżnialności dla zadania klasyfikacji.
2. Opracowanie algorytmów równoległych i zastosowanie obliczeń rozproszonych w zadaniach:
    a. Wyznaczanie funkcji rozróżnialności systemu informacyjnego i decyzyjnego.
    b. Dyskretyzacja danych metodą CAIM.
    c. Dyskretyzacja danych metodą CACC.
3. Analiza skuteczności zaproponowanej metody klasyfikacji danych niezrównoważonych na zbiorze komórek pęcherza moczowego.
4. Porównanie skuteczności opracowanej metody klasyfikacji nadzorowanej zbioru komórek pęcherza moczowego z metodami dostępnymi w środowisku MATLAB jak:
    a. Liniowa analiza dyskryminacyjna,
    b. Kwadratowa analiza dyskryminacyjna,
    c. Naiwny klasyfikator Bayesa,
    d. Drzewa decyzyjne.

5. Zbadanie skuteczności zaproponowanej metody selekcji cech na zbiorze komórek pęcherza moczowego.
6. Analiza metod dyskretyzacji w zastosowaniu do zaproponowanej metody selekcji opartej na teorii zbiorów przybliżonych.
7. Analiza możliwości zwiększenia efektywności klasyfikacji zbioru komórek pęcherza moczowego z zastosowaniem próbkowania losowego.

1.5.2. Teza

Selekcja cech oraz dobór algorytmu klasyfikacji do analizy parametrów morfometrycznych komórek prowadzi do efektywnej wstępnej diagnostyki nowotworu pęcherza moczowego.

Zastosowanie zbiorów przybliżonych w zadaniu selekcji cech oraz klasyfikacji, przy zastosowaniu obliczeń równoległych z wykorzystaniem pakietu MATLAB, prowadzi do skutecznej diagnostyki.

1.5.3. Zakres pracy

Rozdział 1 zawiera wprowadzenie do zagadnienia rozpoznawania wzorców w zastosowaniach diagnostycznych. Przedstawiono problem naukowy podjęty w pracy oraz zadania mające na celu potwierdzenie tezy.

W rozdziale 2 przedstawiono podstawy teoretyczne metod wykorzystanych w części analitycznej pracy. Omówiono zagadnienia dotyczące problemów dyskretyzacji danych, selekcji cech, a także klasyfikacji. Zwrócono uwagę na problem danych niezrównoważonych. Przedstawiono miary oceny jakości klasyfikatorów, będące podstawą do wyboru optymalnego zbioru cech.

Rozdział 3 jest prezentacją narzędzia RSAToolbox dla środowiska obliczeniowego MATLAB. Opracowane przez autorkę narzędzie wykorzystano do przeprowadzenia omawianych w pracy analiz. W rozdziale 3 zawarto także podstawy teoretyczne zbiorów przybliżonych. Zakres omówionych definicji ograniczono do systemów decyzyjnych.

W rozdziale 4 omówiono wyniki analiz poszczególnych zadań. W pierwszej części przedstawiono problem komputerowego wspomagania diagnostyki medycznej nowotworu pęcherza moczowego. Zaprezentowano system skaningowy firmy MetaSystems z którego uzyskano dane wykorzystywane w pracy. W dalszej części zawarto analizę parametrów morfometrycznych komórek w celu optymalizacji procesu diagnostyki nowotworu pęcherza moczowego.

Rozdział 5 stanowi podsumowanie otrzymanych wyników. W rozdziale zamieszczono wnioski, a także przedstawiono możliwe kierunki dalszych badań.

Praca zawiera dodatkowo dwa załączniki. Pierwszy, stanowi uzupełnienie rozdziału 3. Zamieszczono w nim przykłady programów oraz ich interpretacje. Drugi załącznik jest opisem parametrów morfometrycznych wykorzystanych w przeprowadzonej pracy badawczej.

ROZDZIAŁ 2

Metodologia rozpoznawania wzorców

2.1. Definicja problemu rozpoznawania wzorców

Zadanie rozpoznawania polega na określaniu przynależności obiektów lub zjawisk do pewnych klas. O każdej z klas można powiedzieć, że należące do niej obiekty charakteryzują się pewnym podobieństwem. Rozpoznawanie jest więc wykrywaniem wzorców (ang. pattern recognition), czyli tego co wspólne, co łączy obiekty w grupy. Wykorzystuje się w tym celu różne kryteria, jak na przykład: miary odległości lub podobieństwa do wzorca, stopnie przynależności do obszaru wzorcowego. Do najczęściej stosowanych ilościowych miar podobieństwa możemy zaliczyć: współczynniki asocjacji (skojarzenia), współczynniki korelacji oraz wskaźniki odległości. Dwie pierwsze miary są przykładem miar zbliżeniowych: im wartości są większe, tym obiekty są sobie bliższe. Wskaźniki odległości są przykładem miar zróżnicowania: im wartość wskaźnika jest większa tym większa jest różnica pomiędzy obiektami [Roz79,Mal02,Gat98,Bub90,Ogi04,Kul00, MarKor02,ChoKos02].

W celu uogólnienia definicji procesu rozpoznania wzorców zastosowano zapis formalny, szczegółowo przedstawiony w pracach [Tad91,OgiTad09,Paw81a].

Niech $U = \{x_1, x_2, .., x_n\}$ oznacza zbiór obiektów lub zjawisk podlegających rozpoznawaniu. Na przestrzeni $U \times U$ zostaje określona relacja równoważności $\widetilde{D}$ nazywaną klasyfikacją. Relacja $\widetilde{D}$ określa rozbicie zbioru U na kolekcję klas równoważności $D_i = [x_i]_{\widetilde{D}}$, odpowiadających poszczególnym rozpoznawanym klasom obiektów. Niech δ oznacza liczbę klas określonych relacją $\widetilde{D}$, natomiast Δ niech będzie zbiorem indeksów klas. Relacja klasyfikacji posiada następujące własności:

$$U = \bigcup_{i \in \Delta} D_i, \tag{2.1}$$

$$\bigwedge_{i,j \in \Delta} D_i \cap D_j = \emptyset, \tag{2.2}$$

$$\bigwedge_{x_u, x_v \in U} x_u \widetilde{D} x_v \Rightarrow \bigvee_{i \in \Delta} (x_u \in D_i) \wedge (x_v \in D_i). \tag{2.3}$$

Elementy $x \in U$, które należą do tego samego zbioru D_i, charakteryzują się podobnymi wartościami pewnych wybranych (ustalonych) cech. Dla każdego $x \in U$ istnieje dokładnie jeden taki zbiór $[x_i]_{\widetilde{D}}$ dany zależnością:

$$[x_i]_{\widetilde{D}} = \{x \in U : x_i \widetilde{D} x\}. \tag{2.4}$$

Z opisu relacji $\widetilde{D}$ oraz ze zbioru Δ wynika istnienie odwzorowania:

$$S : U \to \Delta, \tag{2.5}$$

o następującej własności:

$$\bigvee_{x \in U} S(x) = i \equiv x \in D_i, \tag{2.6}$$

Zadanie rozpoznawania wzorców polega na konstrukcji algorytmu $\hat{S}$, pozwalającego na klasyfikację wzorców w oparciu o odpowiednio dobrany zestaw cech. Algorytm $\hat{S}$ określony jest wyrażeniem:

$$\hat{S}: U \to \Delta \cup \{\varphi\}. \tag{2.7}$$

W powyższym wzorze φ jest jednoelementowym zbiorem symbolizującym rozpoznanie neutralne, czyli sytuację w której algorytm nie potrafi dokonać klasyfikacji analizowanego obiektu do zdefiniowanych wzorców. Odwzorowanie $\hat{S}$ określa cały proces analizy wzorca: od momentu jego rejestracji do ostatecznej klasyfikacji. Analiza przeprowadzana jest w kolejnych etapach, wśród których wyróżnia się następujące zadania [OgiTad09]:

1. Przeprowadzenie recepcji rejestrowanych wzorców, mającej na celu wyznaczenie cech umożliwiających identyfikację obrazu.
2. Określenie przynależności obiektu do poszczególnych klas w oparciu o wyznaczone cechy, mające na celu poszukiwanie takiej klasy, której obiekty najbardziej przypominają analizowany obiekt.
3. Podjęcie decyzji o rozpoznawaniu poprzez jednoznaczne przypisanie analizowanego obiektu do jednej z klas. Wybierana jest klasa, dla której stopień przynależności jest maksymalny.

Każdy z przedstawionych etapów algorytmu $\hat{S}$ można zapisać formalnie jako odwzorowanie, a algorytm $\hat{S}$ jako ich złożenie:

$$\hat{S} = K \circ W \circ F. \tag{2.8}$$

Odwzorowanie F, dane zależnością:

$$F: U \to A, \tag{2.9}$$

określa recepcję rejestrowanych wzorców, polegającą na wyznaczeniu istotnych cech. Symbol A oznacza przestrzeń cech, której elementami są wektor $\underline{a} =< v_1, v_2, \dots, v_j, \dots, v_p >$, gdzie v_j oznacza wartość j-tej cechy, a p jest liczbą wyznaczonych cech. Przyjmując, że składowe wektora $\underline{a}$ mają charakter ilościowy, przestrzeń A traktuje się jako p-wymiarową przestrzeń euklidesową ($A \subseteq R^p$).

Odwzorowanie W przyjmujące postać:

$$W: A \to R^{\delta}, \tag{2.10}$$

polega na określeniu wartości funkcji przynależności $W^i(\underline{a})$ będącej miarą podobieństwa nieznanego obiektu $x \in U$ do poszczególnych klas D_i. Przy założeniu liczby klas równej δ, odwzorowanie prowadzi do wyznaczania c liczb rzeczywistych (R^{δ}). Poszczególne metody rozpoznawania różnią się sposobem odwzorowania F.

Odwzorowanie K wyrażone zależnością:

$$K: R^{\delta} \to \Delta \cup \{\varphi\}, \tag{2.11}$$

jest opisem procesu podejmowania decyzji, czyli wyborem najlepiej dopasowanej klasy. Stosuje się tutaj regułę w postaci [Tad91]:

$$\wedge_{\underline{a} \in A}\left[\left[K\left(W^1(\underline{a}), W^2(\underline{a}), \dots, W^{\delta}(\underline{a})\right) = i\right] \equiv \wedge_{n \in \Delta, n \neq i}\left[W^n(\underline{a}) < W^i(\underline{a})\right]\right]. \tag{2.12}$$

Powyższa reguła definiuje przynależność obiektu do klasy $i \in \Delta$, dla której wartość funkcji przynależności $W^i(\underline{a})$ jest maksymalna. Obiekt zostanie przypisany do klasy φ na przykład w sytuacjach: gdy stopień dominacji funkcji przynależności nad kolejną funkcją jest zbyt mały, wartość dominującej funkcji przynależności jest za mała,

lub gdy stosunek wartości dominującej funkcji przynależności nie wskazuje na jej zdecydowany charakter.

2.2. Struktury danych

Dane są najistotniejszym elementem procesu zdobywania wiedzy dotyczącej złożonych zjawisk. W uproszczonej postaci, zbiór danych można zapisać w postaci macierzy, której przykład zaprezentowany został w tabeli 2.1. Zbiór U będący zbiorem obiektów x_1,x_i, x_n określa się mianem dziedziny zadania. Każdy obiekt opisany jest za pomocą atrybutów. Atrybutem $a \in A$ nazywa się dowolną funkcję określoną na dziedzinie zadania. Zbiór wszystkich atrybutów $A = \{a_1, a_2, \dots, a_p\}$ określa się mianem przestrzeni atrybutów. Wartości atrybutów oznaczono przez v_j^i, gdzie $i = 1,2,..,n$ wskazuje na numer obiektu, a $j = 1,2,..,p$ wskazuje na numer atrybutu. Zbiór wartości jakie może przyjmować atrybut a_j oznaczono przez V_{a_j}.

Tabela 2.1. Macierzowa reprezentacja zbioru danych

U	a_1	…	a_j	…	a_p
x_1	v_1^1	…	v_j^1	…	v_p^1
…	…	…	…		…
x_i	v_1^i	…	v_j^i	…	v_p^i
…	…	…	…		…
x_n	v_1^n	…	v_j^n	…	v_p^n

Macierz w przedstawionej postaci nosi nazwę tablicy informacyjnej, tablicy typu atrybut-wartość lub systemu informacyjnego [MroPlo99]. Szczegółowa definicja systemu informacyjnego, będącego podstawowym pojęciem zbiorów przybliżonych, została przestawiona w rozdziale 3. Wiersze macierzy, będące podstawową jednostką analizy, w zależności od kontekstu nazywane są: jednostkami, instancjami, encjami, przypadkami, obiektami lub rekordami. Kolumny macierzy, stanowiące element charakterystyki obiektu opisujące badane zjawisko, określane są mianem zmiennych, cech, atrybutów, pól, wymiarów, parametrów, własności [Ja93]. Atrybuty mogą być ilościowe lub jakościowe.

Atrybuty ilościowe posiadają wartości liczbowe. Można je podzielić na atrybuty dyskretne oraz atrybuty ciągłe. Pierwszy przyjmuje przeliczalną (w szczególności skończoną) liczbę wartości. Drugi może przyjmować nieprzeliczalne wartości liczbowe, które często pochodzą z pewnego przedziału liczbowego.

Atrybut jakościowy posiada wartości kategoryczne. Atrybuty jakościowe można podzielić na nominalne (symboliczne) i porządkowe. W przypadku atrybutów nominalnych dokonuje się tylko porównania kategorii pod względem tego czy są one takie same czy różne. Szczególnym przypadkiem atrybutu nominalnego jest zmienna binarna posiadająca wyłącznie dwie kategorie: zera i jedynki. Zmienne porządkowe charakteryzują się tym, że można je szeregować. Atrybuty jakościowe można przedstawić za pomocą liczb, najczęściej naturalnych. Na takich liczbach nie dokonuje się operacji arytmetycznych. Nie można tu także zastosować regresji liniowej, polegającej na przewidywaniu jednej zmiennej jako funkcji innych, którą stosuje się analizach wartości numerycznych [Szy07c].

Niektóre algorytmy wstępnie narzucają typ danych. Dotyczy to szczególnie algorytmów operujących na danych dyskretnych. W takim przypadku, gdy zbiór danych

opisany jest atrybutami ciągłymi, należy przeprowadzić ich dyskretyzację (ang. discretization, binning). Dyskretyzacja jest jednym z czynników wpływających na wyniki algorytmów uczenia maszynowego, ich efektywność i dokładność. W ostatnich latach coraz większą uwagę skupia się na problemie dyskretyzacji, szukając możliwości optymalizacji algorytmów [Ciu05,KuCi04,TsLe08,Ngu97,ZhHu04,NgSk95,AmSa03, Be04,ChWo95].

2.3. Dyskretyzacja danych

Problem dyskretyzacji sprowadza się do wyznaczenia sposobu podziału przeciwdziedziny na przedziały [CiPe98,DoKo95,CiPe07,Ci00,Szy09b,Szy09a]. Niech a będzie dowolnym atrybutem ciągłym $a: U \to V_a$, gdzie V_a jest zbiorem możliwych wartości atrybutu a. Dyskretyzacja polega na podziale zbioru $V_a = \left[\theta_1^{V_a}, \theta_2^{V_a}\right]$ na skończoną liczbę, parami rozłącznych przedziałów $I_a^k = \left(\theta_1^{I_a^k}, \theta_2^{I_a^k}\right]$ należących do zbioru V_a, gdzie

$$\cup_{I \in I_a} I = V_a, \tag{2.13}$$

$$\wedge_{I_1, I_2 \in I_a} I_1 \cap I_2 = \emptyset \tag{2.14}$$

Uwzględniając sposób wyznaczania granic przedziałów można wyróżnić trzy grupy metod dyskretyzacji:

- Metody dyskretyzujące bez nadzoru (ang. unsupervised) lub z nadzorem (ang. supervised). Metoda dyskretyzacji z nadzorem przy wyznaczaniu granic wykorzystuje się informację o klasie obserwacji. Pozwala na optymalizację podziału zakresu wartości zmiennych poprzez jak najlepsze ich dopasowanie do klasy obserwacji. Metodę dyskretyzacji z nadzorem wykorzystuje się najczęściej w uczeniu z nadzorem, a metodę dyskretyzacji bez nadzoru w zadaniach grupowania danych.
- Metody globalne (ang. global) lub lokalne (ang. local). W metodach globalnych każda cecha jest dzielona na przedziały w sposób niezależny od innych cech. W metodach lokalnych podziału dokonuje się w określonych obszarach, wyznaczonych przez wartości innych atrybutów. Dyskretyzacja lokalna ma zazwyczaj charakter dyskretyzacji z nadzorem i jest zależna od stosowanego algorytmu uczenia.
- Metody statyczne (ang. static) lub dynamiczne (ang. dynamic). Algorytmy statyczne dokonują dyskretyzacji każdej cechy w osobnej iteracji w sposób niezależny od innych cech, aż do momentu uzyskania zadanej liczby przedziałów. Algorytmy dynamiczne poszukują liczby wszystkich możliwych przedziałów dyskretyzacji równocześnie dla wszystkich cech.

Właściwa dyskretyzacja powinna się charakteryzować możliwie małą liczbą przedziałów, przy jednocześnie wysokim rozróżnianiu obiektów w procesie uczenia. W przypadku dyskretyzacji z nadzorem ważna jest także rozróżnialność przykładów względem kategorii. W doborze odpowiedniej liczby przedziałów dyskretyzacji stosuje się różne heurystyki, jak:

- zaprzestanie podziału na kolejne przedziały, jeśli brak jest poprawy informacyjnej zawartości przedziałów;
- ograniczenie maksymalnej liczby tworzonych przedziałów (lub maksymalnej głębokości rekurencyjnych wywołań algorytmu dyskretyzacji);
- określenie minimalnej liczby obiektów przypadających na przedział dyskretyzacji;
- określenie liczby przedziałów, która nie powinna być mniejsza niż liczba klas;
- w oparciu o heurystykę, która sugeruje, aby liczba przedziałów nie była mniejsza niż liczba klas obiektu (w zadaniu klasyfikacji);

- wyznaczenie liczby przedziałów dla każdego atrybutu, korzystając z zależności

$$\eta_{I_j} = n/(3 \times \delta), \tag{2.15}$$

gdzie n jest liczbą obiektów zbioru uczącego, a δ liczbą klas atrybutu decyzyjnego.

2.3.1. Dyskretyzacja według równej szerokości i liczności

Najczęściej stosowanymi metodami dyskretyzacji są dyskretyzacja według równej szerokości i liczności [Ci00,StaSzy06,Ciu05]. Dyskretyzacja według równej szerokości (ang. Equal Interval Width Discretization - EWD) polega na podziale całego zakresu wartości $V_a = \left[\theta_1^{V_a}, \theta_2^{V_a}\right]$ na η_a podprzedziałów $I_a^k = \left(\theta_1^{I_a^k}, \theta_2^{I_a^k}\right]$ o równej szerokości, gdzie dla każdego $k = 1,2,\dots,\eta_a$, wartości graniczne wynoszą:

$$\theta_1^{I_a^k} = \theta_1^{Va} + (k-1)\frac{\theta_2^{Va} - \theta_1^{Va}}{k}, \tag{2.16}$$

$$\theta_2^{I_a^k} = \theta_1^{Va} + k\frac{\theta_2^{Va} - \theta_1^{Va}}{\eta_a} \tag{2.17}$$

Dyskretyzacja równej liczności (ang. Quantille discretization, Equal Frequency Intervals Discretization -EFD) to metoda w której przedział $V_a = \left[\theta_1^{V_a}, \theta_2^{V_a}\right]$ jest dzielony na η_a zbiorów $I_a^1, I_a^2, \dots, I_a^\eta$ w taki sposób, aby każdy podzbiór zawierał możliwie równą liczbę wartości przykładów trenujących. Przy podziale należy zachować uporządkowanie wartości atrybutu a, według zależności:

$$\max_{v_a \in \mathrm{I_a^k}} V_a \ \leq \min_{v_a \in \mathrm{I_a^{k+1}}} V_a \tag{2.18}$$

W celu skutecznej dyskretyzacji danych walidacyjnych granicę dolną pierwszego przedziału zastępuje się znakiem $-\infty$, a granicę górną ostatniego przedziału znakiem $+\infty$. Metodę dyskretyzacji równej szerokości można rozbudować o wykorzystanie wiedzy eksperckiej. Wtedy granice przedziałów są modyfikowane, bazując na znajomości specyfiki badanego zjawiska.

2.3.2. Dyskretyzacja metodą CAIM

Dyskretyzacja Class-Attribute Interdependence Maximization (CAIM) jest przykładem dyskretyzacji z nadzorem [KuCi04,CiPe07]. Do wyznaczenia granic przedziałów wykorzystuje się macierz kwantyzacji (ang. quanta matrix) przedstawioną w tabeli 2.2. Jest to macierz, w której dla każdego przedziału dyskretyzacji, określa się liczbę obiektów zbioru uczącego należących do możliwych klas.

Możliwymi wartościami granicznymi przedziałów dyskretyzacji są wartości pośrednie pomiędzy każdymi sąsiednimi wartościami ciągłymi atrybutu. Do oceny każdej granicy dyskretyzacji wykorzystuje się współczynnik *caim* będący kryterium dyskretyzacji:

$$caim(d, I_a|a) = \frac{\sum_{k=1}^{\eta} \frac{max_k^2}{q_{+k}}}{\eta}, \tag{2.19}$$

gdzie:

η - liczba przedziałów dyskretyzacji,
k –numer przedziału dyskretyzacji $k = 1,2,\dots,\eta$,
max_k– największa wartość w k-tym przedziale dyskretyzacji,
q_{+k} - całkowita liczba obiektów w k-tym przedziale.

Tabela 2.2. Macierz kwantyzacji

		Przedziały dyskretyzacji					
		I_a^1		I_a^k		I_a^η	Całkowita liczba obiektów danej klasy
		$\theta_1^{I_a^1}, \theta_2^{I_a^1}$	…	$\theta_1^{I_a^k}, \theta_2^{I_a^k}$	…	$\theta_1^{I_a^\eta}, \theta_2^{I_a^\eta}$	
Klasy decyzyjne	v_1	q_{11}	…	q_{1k}	…	$q_{1\eta}$	$\boldsymbol{q_{1+}}$
	…	…	…	…		…	
	v_i	q_{i1}	…	q_{ik}	…	$q_{i\eta}$	$\boldsymbol{q_{i+}}$
	…	…	…	…		…	
	v_δ	$q_{\delta 1}$	…	$q_{\delta k}$	…	$q_{\delta\eta}$	$\boldsymbol{q_{\delta+}}$
	Całkowita liczba obiektów w przedziale dyskretyzacji	$\boldsymbol{q_{+1}}$		$\boldsymbol{q_{+k}}$		$\boldsymbol{q_{+\eta}}$	n

Współczynnik *caim* przyjmuje wartości z przedziału $[0, n]$, gdzie n jest liczbą obiektów zbioru uczącego. Im większa wartość *caim* tym większa zależność pomiędzy klasą a przedziałami. Wartość współczynnika *caim* będzie największa, gdy w każdym z przedziałów obiekty będą należały tylko do jednej z klas. Wtedy $max_k = q_{+k}$, a $caim = {}^{n}/_{\eta}$.

Dyskretyzację rozpoczyna się od podziału przedziału inicjalnego $V_a = \left[\theta_1^{V_a}, \theta_2^{V_a}\right]$ na dwa podprzedziały. Przedział dzielony jest na dwie części w taki sposób, aby zmaksymalizować współzależność pomiędzy klasą a dyskretyzowanym atrybutem. Algorytm dodawania nowych punktów granicznych powtarzany jest tak długo, aż nie będzie możliwa poprawa współczynnika *caim*.

Odmianą algorytmu dyskretyzacji CAIM jest Fast CAIM [KurCio03]. Algorytmy różnią się sposobem generowania możliwych punktów granicznych. W metodzie CAIM uwzględnia się punkty pośrednie pomiędzy dwoma sąsiednimi wartościami. W metodzie Fast CAIM punkty pośrednie wyznacza się tylko pomiędzy wartościami należącymi do różnych klas.

Oba algorytmy są progresywne i nie wymagają od użytkownika zadawania żadnych parametrów. Wadą algorytmu CAIM jest fakt iż przy wyliczaniu współczynnika *caim* pod uwagę brana jest jedynie klasa o największej liczbie wartości.

2.3.3. Dyskretyzacja metodą CACC

Algorytm dyskretyzacji Class-Attribute Contingency Coefficient (CACC) jest modyfikacją algorytmu CAIM [TsLe08]. W algorytmie CACC, jako kryterium wyznaczania granic przedziałów dyskretyzacji, zastosowano współczynnik kontyngencji (ang. contingency coefficient), który mierzy siłę związku pomiędzy zmiennymi. Współczynnik *cacc* określony jest zależnością:

$$cacc = \sqrt{\frac{y}{y+n}}, \tag{2.20}$$

gdzie:

$$y = {}^{n\left[\sum_{i=1}^{\delta}\sum_{k=1}^{\eta}\frac{q_{ik}^2}{q_{i+}q_{+k}} - 1\right]}\Big/_{\log n}. \tag{2.21}$$

Zastosowanie współczynnika *cacc* do wyznaczania granic przedziałów zapobiega wpływowi klasy charakteryzującej się największą liczbą przypadków w danym przedziale.

2.4. Redukcja przestrzeni atrybutów

Celem redukcji przestrzeni atrybutów jest znalezienie zestawu atrybutów najlepszych do realizacji rozważanego zadania rozpoznawania. Kluczowym elementem procesu redukcji przestrzeni atrybutów jest dokonanie oceny przydatności rozważanych zbiorów pod kątem poprawnej klasyfikacji. Do podstawowych zagadnień redukcji przestrzeni atrybutów należy: określenie metody generacji podzbiorów atrybutów oraz określenie kryterium stosowanego do oceny przydatności wyznaczonych podzbiorów atrybutów [Slo10,Szy08].

Modyfikacja przestrzeni atrybutów pozwala na dokładniejsze poznanie dziedziny problemu oraz pozwala lepiej reprezentować wiedzę. Wyróżnia się trzy typy przekształceń: usuwanie istniejących atrybutów, dodawanie nowych atrybutów lub zastępowanie istniejących atrybutów nowymi atrybutami [Ci00]. Usuwanie atrybutów może prowadzić do ograniczenia przestrzeni hipotez. Metoda ta najczęściej wykorzystywana jest przy atrybutach nieistotnych, które nie mają wpływu na stawiane hipotezy, a operacja pozwala na sprawniejsze przeszukiwanie przestrzeni. Dodawanie nowych atrybutów nie może prowadzić do zwiększenia przestrzeni hipotez, jednak może upraszczać ich reprezentację. Nowe atrybuty są zależne funkcyjnie od atrybutów oryginalnych, a w związku z tym informacyjna zawartość przestrzeni atrybutów nie ulega zmianie. W przypadku zastępowania atrybutów mamy do czynienia zarówno z metodą usuwania jak i dodawania atrybutów, przy czym dodawane atrybuty są funkcyjnie zależne od usuwanych. Może to prowadzić do zawężenia lub pozostawienia bez zmian przestrzeni hipotez, ale w każdym przypadku ma na celu ułatwienie skutecznego przeszukiwania. W każdym z wymienionych przekształceń należy pamiętać, aby wybrany wektor atrybutów nie tylko zmniejszał rozmiar przestrzeni atrybutów, ale także umożliwiał podział obiektów na klasy określone wartościami atrybutu decyzyjnego.

Metody poszukiwania zbiorów atrybutów, zachowujących właściwości dyskryminacyjne pełnego zbioru danych, można podzielić na trzy grupy:

a) Wyznaczanie najmniej licznego zbioru atrybutów,
b) Wyznaczanie zbioru atrybutów o zadanej liczbie elementów,
c) Wyznaczanie maksymalnej liczby zbiorów atrybutów.

Każda z wymienionych metod charakteryzuje się wysoką złożonością obliczeniową. Dlatego często poszukuje się metod heurystycznych [Dom04]. Przy zastosowaniu algorytmów heurystycznych otrzymane zbiory atrybutów nie muszą być wszystkimi możliwymi zbiorami, a wyznaczony najmniej liczny zbiór, nie musi być minimalnym.

Przestrzeń potencjalnych rozwiązań można określić na dwa sposoby:

1. Określając zbiór zawierający kombinację wszystkich atrybutów. Istnieje wtedy $2^C - 1$ możliwych podzbiorów atrybutów, gdzie C jest liczbą atrybutów.
2. Określając zbiór zawierający wszystkie możliwe permutacje atrybutów. Wtedy liczba możliwych podzbiorów wynosi $C!$. Każda permutacja określa kolejność dodawania atrybutów do zbioru pustego lub odejmowania atrybutów ze zbioru pełnego.

Przykładem metod związanych z usuwaniem atrybutów są metody selekcji (ang. feature selection) [HaMa05,GuEl03,Grabc03,DasLiu97]. Polegają one na wykorzystaniu podzbioru istotnych atrybutów, czyli znalezieniu podzbioru $A' = \{a_1, a_2, \dots, a_{p'}\}$ dla p' zmiennych objaśniających, gdzie $p' \ll p$. W ten sposób poszukiwany jest najlepszy zbiór atrybutów, który dostarczy podobnych informacji

o obiektach jak oryginalny zbiór. Można wyróżnić dwie kategorie metod selekcji: filtry (ang. filters) - szacują ważność atrybutów i pozycjonują je według otrzymanej wartości [Szy07c,LeiHua03,HuaSet97] oraz powłoki (ang. wrappers) - optymalny zestaw atrybutów jest wybierany spośród wszystkich możliwych kombinacji testowanych w procesie adaptacyjnym [KoJo97,Ciu05].

Przykładem metod związanych z zastępowaniem atrybutów są algorytmy transformacji atrybutów (ang. feature transformation) [HuMo98]. W algorytmach transformacji atrybutów wyróżnia się dwa warianty: konstrukcja cech (ang. feature construction) oraz ekstrakcja cech (ang. feature extraction) [HuMo98,St07,Slo08]. Algorytmy konstrukcji cech odkrywają nowe związki pomiędzy cechami i obiektami, zwiększając tym samym zbiór cech. Dla zbioru p atrybutów $A = \{a_1, a_2, a_3, \dots, a_p\}$ proces konstrukcji polega na utworzeniu m nowych atrybutów $A' = \{a_{p+1}, a_{p+2}, \dots, a_{p+m}\}$, na przykład poprzez operacje logiczne na atrybutach zbioru A. Algorytmy ekstrakcji cech tworzą nowy zbiór atrybutów wykorzystując funkcje odwzorowujące na oryginalnych atrybutach. W wyniku tworzony jest nowy zbiór atrybutów $B = \{b_1, b_2, \dots, b_i, \dots, b_m\}$, gdzie $b_i = F(a_1, a_2, a_3, \dots, a_p)$, a F jest funkcją odwzorowującą.

2.4.1. Analiza głównych składowych

Analiza głównych składowych (ang. Principal Component Analysis, PCA) polega na transformacji początkowych atrybutów we wzajemne ortogonalne nowe zmienne [St07,HaMa05,Lar08]. Nowe atrybutów nazywane składowymi głównymi mają postać:

$$Z_{\mathrm{i}} = e_{\mathrm{i}1}a_{s1} + e_{\mathrm{i}2}a_{s2} + \cdots + e_{\mathrm{ij}}a_{sj} + \cdots + e_{\mathrm{ip}}a_{sp}, \tag{2.22}$$

gdzie $a_{si} = (a_i - \mu_i)/\sigma_{ii}$ oznacza atrybut standaryzowany, μ_{i} jest wartością średnią atrybutu a_i, a σ_{ii} - odchyleniem standardowym.

Celem analizy składowych głównych jest wyznaczenie takich wartości współczynników $\mathrm{e_{i1}}, \mathrm{e_{i2}}, \dots, \mathrm{e_{ip}}$, aby wariancja zmiennej $\mathrm{Z_i}$ była jak największa. Współczynniki te są elementami wektora własnego, odpowiadającego i-tej co do wielkości wartości własnej (λ) macierzy kowariancji atrybutów $\mathrm{a_1}, \mathrm{a_2}, \dots, \mathrm{a_p}$. Znaki i wartości tych współczynników wskazują na sposób i wielkość wpływu j-tego atrybutu na i-tą składową.

Analiza głównych składowych jest cennym narzędziem ponieważ bazuje na rzutowaniach liniowych i minimalizowaniu wariancji (czyli sumarycznego błędu kwadratowego) [HaMa05]. Analizę głównych składowych można wykorzystać w zadaniu redukcji przestrzeni atrybutów pomijając składowe, które wyjaśniają znikomą część zmienności. W praktyce stosowane są trzy kryteria oceny liczby składowych:

- Kryterium części wariancji wyjaśnionej przez składowe główne. Kryterium bazuje na procencie wariancji. Według tego kryterium do rozważań powinno się wziąć taką liczbę składowych, aby skumulowany procent zmienności był możliwie największy. Na przykład, jeżeli pierwsze dwie lub trzy składowe stanowią znaczną część wariancji wszystkich zmiennych (na przykład 80%) to można ograniczyć badania do tych zmiennych.
- Kryterium wartości własnej. Kryterium zdefiniowane przez H.Kaisera [Kai60] mówi, że ponieważ standaryzowane cechy wejściowe mają wariancję równą jeden, to nowe cechy również powinny mieć wariancję przynajmniej równą jeden. Wartość własna wynosząca 1 oznacza, że składowa wyjaśnia zmienność „równoważną" jednemu oryginalnemu atrybutowi. Kryterium to można stosować wyłącznie przy zmiennych standaryzowanych.
- Kryterium wykresu osypiskowego. Kryterium jest wyznaczane w oparciu o wykres liniowy na którym zaznacza się kolejne wartości własne. Wykres zaczyna się wysoko

po lewej stronie, następnie stosunkowo szybko opada i od pewnego punktu staje się płaski. Punkty charakteryzujące się łagodnym spadkiem tworzą tzw. osypisko czynnikowe [Cat66]. Według [Lar08], do dalszych rozważań należy uwzględnić tylko te składowe, które znajdują się powyżej punktu rozpoczynającego osypisko. Według [St07] to czy punkt załamania należy uwzględnić do dalszych analiz jest decyzją badacza.

2.4.2. Analiza korelacyjna

Analiza korelacyjna jest przykładem prostego algorytmu selekcji atrybutów. Jej zaletą jest szybkość działania umożliwiająca znaczne zmniejszenie liczby atrybutów [Ha99,Kwi07, MiKwa06].

Do wyznaczenia istotnych atrybutów wykorzystuje się macierz korelacji. Elementy macierzy korelacji są współczynnikami korelacji próby uczącej, wyznaczonymi zgodnie z zależnością:

$$\bigwedge_{a_k,a_j \in A} r_{kj} = \frac{\sum_{i=1}^{n}\left(v_k^i - \overline{v}_k\right)\left(v_j^i - \overline{v}_j\right)}{\sqrt{\sum_{i=1}^{n}\left(v_k^i - \overline{v}_k\right)^2 \sum_{i=1}^{n}\left(v_j^i - \overline{v}_j\right)^2}}, \tag{2.23}$$

gdzie:

$\overline{v}_k = \frac{1}{n}\sum_{i=1}^{n} v_k^i$, $\overline{v}_j = \frac{1}{n}\sum_{i=1}^{n} v_k^i$ - wartości średnie atrybutów,

j,k –indeks wektorów atrybutów,

$i = 1,2,\dots,n$ – indeks obiektu.

Współczynnik r_{kj} przyjmuje wartości z zakresu $< -1, 1 >$ przy czym wartości bliskie 1 lub -1 oznaczają bardzo wysoką korelację. W przypadku, gdy dwa atrybuty są niezależne współczynnik wynosi 0.

Algorytm selekcji polega na grupowaniu atrybutów skorelowanych powyżej zadanej wartości progowej współczynnika korelacji, a następnie wyborze jednego, który będzie reprezentował tą grupę.

Metodę analizy korelacyjnej zastosowano w dwóch odmianach:

- corr-AA (ang. Correlation - Attribute Attribute) – selekcja atrybutów bez uwzględniania zależności pomiędzy atrybutów, a klasą decyzyjną.
- corr-AC (ang. Correlation - Attribute Class) – selekcja ma charakter nadzorowany; atrybuty wybiera się uwzględniając ich korelację z atrybutem decyzyjnym.

Algorytmy corr-AC oraz corr-AA są przykładem metody zstępującej. Poszukiwanie zestawu atrybutów rozpoczyna się od pełnego zbioru, a następnie w każdym kroku usuwa się atrybuty skorelowane.

2.5. Wybrane metody rozpoznawania

Problem rozpoznawania wzorców charakteryzuje się różnorodnością metod rozwiązywania zadań. Wynikiem wielu prac nad metodami klasyfikacji jest niezliczona liczba opracowanych algorytmów [Grabo03,St07,Ci00,Lar08,Paw81a, KorCwi05]. Różnią się one między innymi złożonością, jakością klasyfikacji, szybkością działania, szybkością uczenia, ograniczeniami pamięci komputerów.

Skuteczne modele klasyfikujące to takie, które potrafią udzielać poprawnych odpowiedzi także dla danych, które nie były dostępne w czasie uczenia, a pochodzą z tej samej dziedziny. Taka własność nazywana jest zdolnością generalizacji, czyli umiejętnością uogólniania treści zawartych w analizowanych danych.

Reguły klasyfikacyjne mogą być narzucone przez eksperta, którego wiedza pozwala na odpowiednią interpretację danych. Jednak w wielu sytuacjach rozpoznawania wzorców nie ma dostępnej apriorycznej informacji na temat reguł przynależności rozpoznawanych obiektów do klas. Jedyna wiedza jaka jest dostępna wynika z analizy zbioru uczącego, dla którego znana jest prawidłowa klasyfikacja obiektów.

W zakresie rozpoznawania wzorców można wyróżnić trzy nurty określania przynależności nieznanych obiektów do zdefiniowanych klas [Tad91, OgiTad09]:
- metody całościowe, w których pod uwagę brane są wszystkie atrybuty rozpoznawanego obiektu. Można przy tym wyróżnić metody odległościowe, aproksymacji funkcji przynależności oraz probabilistyczne;
- metody strukturalne, polegające na określeniu wzajemnych relacji pomiędzy elementami i przeprowadzenia rozpoznania w oparciu o opis strukturalny. Do tej grupy zaliczamy metody oparte na ciągach, drzewach, grafach.
- metody inteligencji obliczeniowej wykorzystujące sieci neuronowe, zbiory rozmyte oraz zbiory przybliżone, itp.

Dla analizowanego wzorca, informację wyjściową klasyfikatora można przypisać do jednego z trzech poziomów [MarKor02,XuKrz92]:
- Poziom abstrakcji: klasyfikator określa klasę lub zbiór możliwych klas,
- Poziom rangi: klasyfikator przypisuje rangi klasom w kolejności zależącej od stopnia przynależności,
- Poziom miar: klasyfikator przypisuje każdej klasie pewną miarę, będącą stopniem przynależności do tej klasy.

Najwięcej informacji o efekcie procesu klasyfikacji zawiera poziom miar. Przykładem takiego klasyfikatora jest klasyfikator Bayesa, który dostarcza informacji o prawdopodobieństwie warunkowym. Poziom miar prezentują także klasyfikatory minimalno-odległościowe, które dostarczają informacji o odległości obrazu od każdej z możliwych klas.

2.5.1. Naiwny klasyfikator Bayesa

Naiwny klasyfikatora Bayesa jest klasyfikatorem statystycznym opartym na twierdzeniu Bayesa. W metodzie tej wykorzystuje się informację o częstości występowania klas obiektów w zbiorze uczącym. Naiwność klasyfikatora Bayesa polega na założeniu niezależności atrybutów [WStat10b,Kwi07,KrzWol08,Krz90].

W analizie Bayesa wykorzystuje się prawdopodobieństwo „a priori", które jest wyznaczane dla każdej z klas v_i na podstawie obserwacji zbioru uczącego:

$$P_{a\ priori}(v_i) = \frac{q_{i+}}{n}. \tag{2.25}$$

Przy klasyfikacji nowego obiektu wyznacza się prawdopodobieństwo wystąpienia każdej z klas spośród obiektów znajdujących się w pobliżu klasyfikowanego. Prawdopodobieństwo nazywa się szansą przynależności do klasy v_i:

$$P_{szansa}(v_i) = \frac{ilość\ obiektów\ klasy\ v_i\ w\ sądzidztwie}{q_{i+}}. \tag{2.26}$$

Na uwagę zasługuje fakt, iż pomimo znaczących różnic w ilości przypadków w klasach, nie jest przesądzone, że nowy obiekt będzie należał do klasy o większym prawdopodobieństwie. Decydujące znaczenie ma prawdopodobieństwo „a posteriori" definiowane regułą Bayesa:

$$P_{a\ posteriori}(v_i) = P_{a\ priori}(v_i) * P_{szansa}(v_i). \tag{2.27}$$

Dla zbioru zmiennych $C = \{c_1, c_1, \dots, c_j \dots, c_p)$, przy założeniu niezależności atrybutów, szansę przynależności obiektu dla klasy v_i można zapisać jako iloczyn prawdopodobieństw:

$$P(C|v_i) = \prod_{j=1}^{p} P(c_j|v_i) \, . \tag{2.28}$$

Natomiast prawdopodobieństwo „a posteriori" będzie miało postać:

$$P(v_i|C) = p(v_i) \prod_{j=1}^{p} P(c_j|v_i) \, . \tag{2.29}$$

Ostatecznie obiekt zostanie przypisany do klasy o wyższym prawdopodobieństwie „a posteriori". Prawdopodobieństwa „a priori" wpływają na trafność klasyfikacji, dlatego można je stosować do poprawienia dokładności lub minimalizowania błędów.

2.5.2. Drzewa klasyfikacyjne

Rozpoznawanie wzorców za pomocą drzew klasyfikacyjnych polega na budowaniu przestrzeni reguł decyzyjnych. Utworzone reguły odpowiadają gałęziom drzewa, w którym węzły pełnią funkcję warunków decyzyjnych. Końcowe elementy drzewa nazywane są liśćmi i odpowiadają poszukiwanym klasom decyzji. Zaletą tego modelu jest przejrzystość, pozwalająca na zrozumienie podstaw jego działania [StaSzy06].

Przykładem algorytmu realizującego klasyfikację według idei drzew decyzyjnych jest metoda CART [Lar06]. Budowane drzewa mają postać binarną – dla każdego węzła tworzone są dwie gałęzie. Przypadki, wykorzystane do uczenia algorytmu, dzielone są rekurencyjnie wykorzystując podobieństwo względem zmiennej celu na etapie uczenia. Dla każdego węzła wyznaczone są wszystkie możliwe podziały atrybutów. Jako ostateczny warunek decyzyjny wybiera się ten podział, który maksymalizuje wartość kryterium:

$$\Phi(s|t) = 2P_L P_P \sum_{j=1}^{\delta} \left| P(j|t_l) - P(j|t_p) \right| , \tag{2.30}$$

gdzie:

t - węzeł

s - możliwy podział

t_l (t_p) – lewy (prawy) potomek węzła

$P_L(P_P)$ – stosunek liczby rekordów w t_l (t_p) do liczby rekordów w zbiorze uczącym

$P(j|t_l)$ ($P(j|t_p)$) – stosunek liczby rekordów należących do klasy j w t_l (t_p) do całkowitej liczby rekordów w węźle

Algorytm zaprzestaje tworzenia węzłów w gałęzi gdy nie jest już możliwe przeprowadzenie nowych podziałów, Wielkość $\Phi(s|t)$ będzie duża gdy czynniki $2P_L P_P$ i $Q(s|t) = \sum_{j=1}^{\delta} \left| P(j|t_l) - P(j|t_p) \right|$ będą duże. Czynnik $Q(s|t)$ będzie tym większy im większy będzie stosunek liczby przypadków w każdej gałęzi. Wartość maksymalna $\Phi(s|t) = \delta$ zostanie osiągnięta, gdy dla każdej klasy wierzchołki poddrzew będą całkowicie jednorodne. Czynnik $2P_L P_P$ osiągnie wartość maksymalną, gdy liczby przypadków w każdej z gałęzi poddrzewa będą równe. Teoretyczna wartość maksymalna będzie równa ½.

Innym algorytmem konstrukcji drzewa klasyfikacyjnego jest C4.5 [Lar06]. Nie jest on ograniczony do przedziałów binarnych co umożliwia tworzenie w węźle osobnych gałęzi dla każdej klasy. Do oceny optymalnego podziału węzła wykorzystuje się miarę zysku informacji (ang. information gain) nazywaną także redukcją entropii (ang. entropy reduction). Wskaźnik zysku można zdefiniować jako [Lar06]:

$$zysk(s) = H(t) - H_s(t) \, , \tag{2.31}$$

gdzie:

$$H(t) = -\sum_{j=1}^{\delta} P_j \, log_2(P_j) \,, \tag{2.32}$$

$$H_s(t) = \sum_{i=1}^{k} P_i H_s(t_i) \,. \tag{2.33}$$

Zmienna $H(t)$ jest entropią węzła decyzyjnego przed podziałem, gdzie P_j ozn. prawdopodobieństwo wystąpienia j-tej klasy w badanym zbiorze. Dla danego podziału s dzielącego zbiór uczący na t podzbiorów, wartość $H_s(t)$ jest średnim zapotrzebowaniem na informację wyznaczonym jako suma ważona entropii dla każdego z podzbiorów podziału. Wartość P_i określa procent rekordów i-tego podziału w zbiorze.

Im wartość $H_s(t)$ jest mniejsza tym bardziej wskaźnik $zysk(s)$ jest większy. Oznacza to, że podział s o wartościach etropii $H_s(t)$ bliższych zeru zawiera niej szumu informacyjnego i powinien zostać wybrany jako reguła decyzyjna węzła.

Przy dobieraniu algorytmu drzewa decyzyjnego należy zwrócić uwagę na typ zmiennych. W przypadku zmiennych jakościowych algorytm C4.5 poprzez tworzenie gałęzi dla każdej z kategorii analizowanej zmiennej może utworzyć drzewa nadmiernie rozgałęzione. W zależności od wielkości analizowanego zbioru, liście mogą zawierać nawet po kilka obiektów (rekordów). Problemu tego nie ma w algorytmie CART, który jest ograniczony do przedziałów binarnych.

2.5.3. Analiza dyskryminacyjna

Analiza dyskryminacyjna jest zespołem metod dyskryminacyjnych i klasyfikacyjnych. W analizie dyskryminacyjnej bada się różnice pomiędzy grupami, analizując kilka zmiennych jednocześnie. Zmienne użyte do rozróżnienia grup nazywa się zmiennymi dyskryminacyjnymi.

Zadanie analizy dyskryminacyjnej można podzielić na dwa etapy: opis i interpretacja różnic między grupowych oraz opis funkcji klasyfikacyjnych. W pierwszym etapie wyznacza się funkcje dyskryminacyjne. Najczęściej stosuję się funkcje liniowe [St07]:

$$f_{d_i} = e_{i0} + e_{i1}A_1 + e_{i2}A_2 + \cdots + e_{ij}A_j + \cdots + e_{ip}A_p, \tag{2.34}$$

gdzie:

f_{d_i} – oznacza i-tą funkcję dyskryminacyjną dla i = 1,2, … , g,

e_{ij} – współczynniki funkcji dyskryminacyjnej wyznaczone na podstawie jej własności,

A_j – zmienna dyskryminacyjna.

Współczynniki e_{ij} określa się w taki sposób, aby średnie klas (centroidy) były jak najbardziej zróżnicowane. Liczba funkcji dyskryminacyjnych nie powinna przekraczać liczby zmiennych dyskryminacyjnych lub liczby klas pomniejszonej o jeden.

Równanie 2.34 określa przekształcenie p-wymiarowej przestrzeni zmiennych dyskryminacyjnych do l-wymiarowej przestrzeni, gdzie nowe współrzędne dla l-tego przypadku w k-tej grupie określone są zależnością:

$$f_{d_{i\,lk}} = e_0 + e_{i1}x_{1\,lk} + e_{i2}x_{2\,lk} + \cdots + e_{ij}x_{j\,lk} + \cdots + e_{ip}x_{p\,lk} \tag{2.35}$$

gdzie:

$i = 1,2, \dots, g$ - i-ta współrzędna nowego układu,

$x_{j\,lk}$ - wartość p-tej zmiennej dyskryminacyjnej dla l-tego przypadku w k-tej klasie określonej atrybutem decyzyjnym.

Problem określenia współczynników e_{ij} sprawdza się do rozwiązania układu p równań:

$$(M - \lambda W)e = 0, \tag{2.36}$$

gdzie:

W – wewnątrzgrupowa macierz kwadratów i iloczynów mieszanych,
M – międzygrupowa macierz kwadratów i iloczynów mieszanych,
e – wektor nieznanych współczynników funkcji dyskryminacyjnych,
λ – wartość własna.

Im mniejsza zmienność wewnątrzgrupowa (punkty skupione wokół centroid klas) i im większa zmienność międzygrupowa (centroidy poszczególnych klas oddalone są od siebie) tym dyskryminacja będzie lepsza. W związku tym znalezienie najlepiej dyskryminujących współczynników wymaga maksymalizacji ilorazu M/W.

Drugim etapem analizy dyskryminacyjnej jest klasyfikacja obiektów. Zadanie polega na porównaniu położenia obiektu względem każdej z centroid i wyborze klasy odpowiadającej najbliższemu obiektowi. W tym celu tworzone są funkcje klasyfikacyjne, umożliwiające wybór odpowiedniej klasy.

Jedną z możliwych jest funkcja zaproponowana przez R.Fishera bazująca na liniowej kombinacji zmiennych. Funkcję klasyfikacyjną wyznacza się oddzielnie dla każdej klasy korzystając z zależności:

$$\mathrm{f_{k_i}} = c_{\mathrm{i}0} + c_{\mathrm{i}1}a_1 + c_{\mathrm{i}2}a_2 + \cdots + c_{\mathrm{ij}}a_j + \cdots + c_{\mathrm{ip}}a_p \tag{2.37}$$

gdzie:

c_{ij} - współczynniki zmiennych dyskryminujących,
a_j – zmienna dyskryminująca.

Funkcji klasyfikacyjnych jest tyle ile klas decyzyjnych. Obiekt przypisuje się do tej klasy, dla której funkcja $\mathrm{f_{k_i}}$ przyjmuje wartość największą.

Innym przykładem funkcji klasyfikującej jest funkcja oparta na odległości Mahalonobisa. Jest to uogólniona miara odległości indywidualnego przypadku od centroidy grupy. Obiekt klasyfikowany jest do tej grupy dla której odległość jest najmniejsza.

Jeśli w zadaniu rozpatrywane są rozkłady normalne o różnych macierzach kowariancji to jako funkcję dyskryminacji stosuje się kwadratową analizę dyskryminacyjną (ang. Quadratic Discriminant Analysis, QDA).

2.6. Miary jakości klasyfikatorów

Miarą jakości klasyfikatora (ang. performance measure) jest jego zdolność do prawidłowego przewidywania lub rozdzielania klas. Podstawowym narzędziem przy ocenie klasyfikatorów jest tabela kontyngencji (ang. contingency table), nazywana także macierzą pomyłek (ang. confusion matrix). Wyznaczone przy jej użyciu współczynniki wykorzystuje się także w innych technikach oceny klasyfikatorów, jak krzywe ROC (ang. Receivier Operating Characteristic) [Faw06,Szy07a,Sta06,KorCwi05] czy wykresy przyrostowe (ang. lift chart) [Ora03, VukCur06,Byr02,Szy07a,wStat10a].

Dla zadania klasyfikacji wielowartościowej o klasach $v_1, v_2, \ldots, v_\delta$ macierz będzie miała postać przedstawioną w tabeli 2.3. W tabeli kontyngencji wiersze odpowiadają rzeczywistej przynależności obiektu do klasy, a kolumny przynależności wyznaczonej przez klasyfikator. Każda komórka tabeli opisuje liczbę obiektów jaka, w wyniku zadania klasyfikacji, została przypisana do danej klasy w odniesieniu do rzeczywistej klasy obiektu. Liczby leżące na głównej przekątnej opisują liczbę prawidłowych klasyfikacji. Pozostałe wartości wskazują na błąd danego klasyfikatora.

Tabela 2.3. Struktura tabeli kontyngencji dla klasyfikacji wielowartościowej

		Klasa obiektu wg klasyfikatora					
		v_1	v_2	…	v_i	…	v_δ
Rzeczywista klasa obiektu	v_1	TP_1	Err_{12}	…	Err_{1i}	…	$Err_{1\delta}$
	v_2	Err_{21}	TP_2	…	Err_{2i}	…	$Err_{2\delta}$
	…	…	…	…	…		…
	v_i	Err_{i1}	…	…	TP_i	…	$Err_{i\delta}$
	…	…	…	…	…		…
	v_δ	$Err_{\delta 1}$	$Err_{\delta 2}$	⋯	$Err_{\delta i}$	…	TP_δ

Tablicę kontyngencji można przekształcić na δ tablic kontyngencji klasyfikatorów binarnych. Przykład binarnej tablicy kontyngencji przedstawiono w tabeli 2.4.

Tabela 2.4. Binarna tablica kontyngencji dla i-tej klasy

		Klasa obiektu wg klasyfikatora	
		N_i	P_i
Rzeczywista klasa obiektu	N_i	TN_i	FP_i
	P_i	FN_i	TP_i

W definiowaniu miar jakości klasyfikatora przyjęto następujące oznaczenia:

- Klasa wyróżniona (P, ang. Positive target) – i-ta klasa, charakteryzująca się szczególnym znaczeniem w danym zjawisku np.: wystąpienie choroby. Liczba obiektów i-tej klasy odpowiada ich rzeczywistej przynależności do klas i określona jest zależnością:

$$P_i = FN_i + TP_i \; ; \tag{2.38}$$

- Klasa negatywna (N, ang. Negative taget) – rzeczywisty zbiór obiektów klas nienależących do klasy wyróżnionej. Liczbę obiektów można wyznaczyć na podstawie zależności:

$$N_i = TN_i + FP_i \; ; \tag{2.39}$$

- Prawidłowe wskazanie i-tej klasy (TP, ang. True Positive, Hit) – określa liczbę obiektów, które klasyfikator poprawnie przypisał do klasy wyróżnionej;
- Błędne wskazanie i-tej klasy (FP, ang. False Positive, False alarm) – określa liczbę obiektów błędnie przypisanych przez klasyfikator do i-tej klasy. W rzeczywistości obiekty nie należą do wyróżnionej klasy. Wartość FP wyrażona jest zależnością:

$$FP_i = \sum_{j=1, j\neq i}^{\delta} Err_{ji} \; ; \tag{2.40}$$

- Prawidłowe odrzucenie i-tej klasy (TN, ang. True Negative, Correct rejection) - określa liczbę obiektów, które zostały prawidłowo przypisane do innych klas. Wartość TN określona jest wzorem:

$$TN_i = \sum_{j=1, j\neq i}^{\delta} TP_j + \sum_{j=1; j\neq i}^{\delta} \sum_{k=1, k\neq i}^{\delta} Err_{jk} \; ; \tag{2.41}$$

- Błędne odrzucenie i-tej klasy (FN, ang. False Negative, Miss) – określa liczbę obiektów, które zostały błędnie sklasyfikowane jako nienależące do wyróżnionej klasy. Wartość FN wyrażą się zależnością:

$$FN_i = \sum_{j=1, j\neq i}^{\delta} Err_{ij} \; . \tag{2.42}$$

Podstawową miarą jakości klasyfikacji jest dokładność (ang. accuracy). Opisuje ona procent obiektów prawidłowo zaklasyfikowanych:

$$acc = \frac{\sum_{i=1}^{\delta} TP_i}{n}. \tag{2.43}$$

Uzupełnieniem współczynnika dokładności są współczynniki czułości oraz swoistości [Arm78]. Czułość (ang. sensitivity, recall, hit rate, true positive rate) jest miarą zdolności klasyfikatora do prawidłowego przewidywania i-tej klasy.

$$sen_i = \frac{TP_i}{TP_i + FN_i}. \tag{2.44}$$

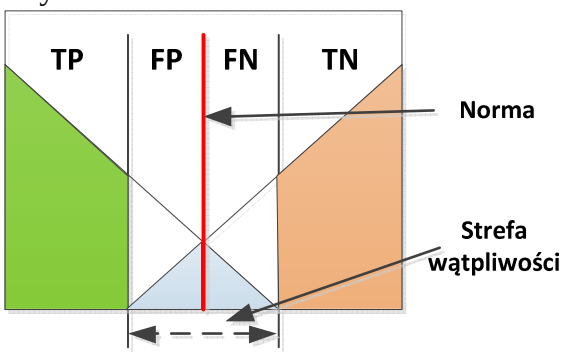
Wartość występująca w mianowniku współczynnika czułości ($TP_i + FN_i$) odpowiada liczbie obiektów należących do i-tej klasy. Czułość jest oszacowaniem prawdopodobieństwa przypisania do i-tej klasy, pod warunkiem, że obiekt należał do i-tej klasy. Czuły klasyfikator powinien dawać małą liczbę wyników fałszywie ujemnych (FN_i), czyli błędnie odrzucanych obiektów.

Swoistość (ang. specificity, true negative rate) jest oszacowaniem prawdopodobieństwa nie przynależności obiektu do i-tej klasy pod warunkiem, że obiekt faktycznie nie należał do i-tej klasy.

$$spe_i = \frac{TN_i}{TN_i + FP_i}. \tag{2.45}$$

Klasyfikator swoisty powinien posiadać jak najmniejszą liczbę wyników fałszywie dodatnich (FP_i). W przypadku klasyfikacji binarnej współczynnik swoistości dla jednej klasy jest współczynnikiem czułości dla innej. Takie rozumowanie nie ma jednak przełożenia na klasyfikację wielowartościową.

Zwiększenie czułości klasyfikatora powoduje zmniejszenie jego swoistości. Dlatego ważnym elementem jest wybór pomiędzy klasyfikatorem czułym, a swoistym. Klasyfikator o największej czułości ma wartość współczynnika równą 1 i w wyniku klasyfikacji zawsze wskaże przynależność do zadanej klasy. Jednocześnie specyficzność takiego klasyfikatora będzie równa 0. Z drugiej strony swoisty klasyfikator wskaże obiekty, które z pewnością nie będą należały do i-tej klasy. Zależność pomiędzy czułością, a swoistością zobrazowano na rys. 2.1.



Rys. 2.1. Zależność czułości i swoistości [Sta06]

Zwiększenie wartości progowej klasyfikacji (przesunięcie normy w prawo) powoduje zmniejszenie błędów odrzucenia, a tym samym wzrost czułości klasyfikatora. Związane jest z tym także powiększenie się obszaru fałszywych wskazań klasy wyróżnionej i jednocześnie spadek swoistości.

Kolejną miarą oceny klasyfikatora jest wartość predykcyjna dodatnia (ang. positive predictive value, ppv) nazywana także precyzją (ang. precision):

$$ppv_i = \frac{TP_i}{TP_i + FP_i}. \tag{2.46}$$

Wartość predykcyjna dodatnia określa prawdopodobieństwo przynależności obiektu do i-tej klasy, gdy wskazywał na to klasyfikator. Uzupełnieniem wartości ppv jest wartość predykcyjna ujemna (ang. negative predictive value, npv) będąca prawdopodobieństwem, tego że obiekt nie należał do i-tej klasy, gdy klasyfikator go odrzucił.

$$npv_i = \frac{TN_i}{TN_i + FN_i}. \tag{2.47}$$

Współczynniki czułości i swoistości pomagają w doborze odpowiedniego klasyfikatora. Natomiast wartości predykcyjne pomagają ocenić możliwą klasę analizowanego obiektu. Dają odpowiedzi na to jakie jest prawdopodobieństwo przynależności do i-tej klasy w zależności od wyniku klasyfikacji. Wartości te zależą od częstości występowania danej klasy w zbiorze.

2.7. Rozpoznawanie wzorców w danych niezrównoważonych

Wiele problemów charakteryzuje się niezrównoważonym rozkładem danych (ang. imbalanced data). W klasyfikacji takich danych można wyróżnić dwa typy klas: mniejszościowa (ang. minority class) oraz większościowa (ang. majority class). Klasa mniejszościowa charakteryzuje się zdecydowanie mniejszą liczbą przypadków, najczęściej nieprzekraczającą 10% liczebności zbioru.

Przykładem niezrównoważenia danych mogą być problemy wykrywania ropy rozlanej na morzu przy użyciu zdjęć satelitarnych [KuHo98], wykrywania nieuczciwych rozmów telefonicznych [FaPr97] czy monitorowania uszkodzeń skrzyni biegów helikoptera [JapMy95]. Problem ten występuję także w medycynie w analizie danych pochodzących z badań przesiewowych [Bat08].

Większość algorytmów uczących zakłada zrównoważenie klas. Powoduje to trudności w fazie uczenia i obniża zdolność predykcyjną Niska jakość klasyfikacji może także wynikać ze złego uwarunkowania danych klasy mniejszościowej, jak: zbyt mała liczba obiektów, nakładanie się obiektów klasy większościowej na mniejszościową czy niejednoznaczność obiektów brzegowych [Cha10, FeGa11, GaSa10, Jap00,StWi05].

Niezrównoważony rozkład klas powoduje też problemy w interpretacji wskaźników jakości klasyfikacji. Szczególną uwagę należy zwrócić na współczynnik dokładności, który odnosi się do wszystkich prawidłowych klasyfikacji. Przykład takiej sytuacji obrazuje tabela 2.5. W zbiorze zawierającym 100 obiektów, 95 jest przypisanych do klasy negatywnej, a 5 do klasy pozytywnej. Pomimo, iż klasyfikator nie wytypował prawidłowo żadnych obiektów klasy pozytywnej to jego dokładność wynosi aż 95%.

Tabela 2.5. Przykładowa macierz pomyłek

		Klasa obiektu wg klasyfikatora	
		0	1
Rzeczywista klasa obiektu	0	95	0
	1	5	0

W analizie danych niezrównoważonych miara klasyfikacji powinna uwzględniać istotność klasy mniejszościowej poprzez maksymalizację liczby poprawnych wskazań w klasie mniejszościowej (TP) i minimalizację liczbę błędnych wskazań klasy większościowej (FP).

Jedną z najczęściej proponowanych w literaturze metryk do oceny jakości klasyfikacji jest miara G_{mean} [GaSa10], opisana zależnością:

$$G_{mean} = \sqrt{sen \cdot spe}\ . \tag{2.48}$$

Wielkość G_{mean} jest średnią geometryczną, która może być rozpatrywana jako korelacja pomiędzy dwoma wskaźnikami. Duża wartość miary G_{mean} wystąpi, gdy oba wskaźniki sen i spe będą wysokie. Mała wartość miary G_{mean} pojawi się, gdy chociażby jeden ze wskaźników będzie posiadał małą wartość.

Inną miarą oceny jakości klasyfikacji w analizie danych niezróżnicowanych jest miara F_{value} [Cha10], opisana wyrażeniem:

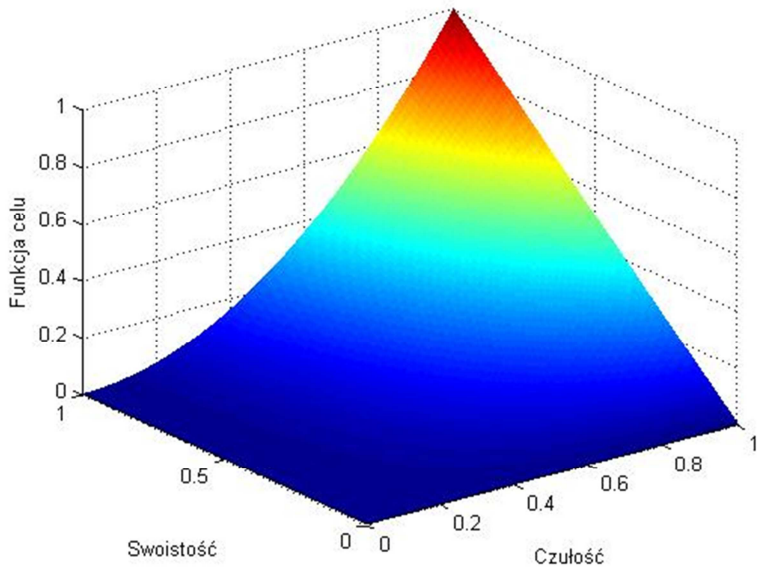
$$F_{value} = \frac{(1+\beta^2)\cdot sen\cdot ppv}{\beta^2\cdot sen+ppv}\ . \tag{2.49}$$

Powyższa miara opisuje zależność pomiędzy trzema wartościami: TP, FP, FN. Współczynnik β odpowiada relatywnej ważności wskaźnika ppv względem sen i najczęściej przyjmuje wartość $\beta = 1$.

W pracy do oceny klasyfikatorów zaproponowano funkcję celu $FMaxSen$ będącą iloczynem współczynników czułości i swoistości, jako:

$$FMaxSen = sen^2 * spe\ . \tag{2.50}$$

Charakterystykę funkcji celu zamieszczono na rysunku 2.2. Druga potęga współczynnika czułości podkreśla jego ważność i umożliwia poszukiwanie klasyfikatora charakteryzującego się maksymalną swoistością przy maksymalnej czułości.



Rys. 2.2. Charakterystyka funkcji celu dla systemu decyzyjnego

W celu poprawienia jakości klasyfikatorów możliwe są dwa kierunki postępowania [FeGa11]:

- modyfikacja algorytmów uczących,
- modyfikacja zbioru danych .

Modyfikacja algorytmów wymaga wprowadzenia zmian, które mają wpływ na etap uczenia klasyfikatora. Zmiany mogą dotyczyć na przykład dopasowania prawdopodobieństw apriori, czy wprowadzenia macierzy kosztów.

Modyfikacja zbioru danych polega na zmianie poziomu niezrównoważenia, zmianie liczebności zbioru lub dekompozycji klas na podobszary. Zmianę poziomu

niezrównoważenia realizuję się poprzez zastosowanie metod: nadpróbkowanie (ang. oversampling), podpróbkowanie (ang. undersampling) lub ich połączenia. Metoda nadpróbkowania polega na zwiększaniu w zbiorze uczącym liczby przypadków klasy mniejszościowej. Metoda podpróbkowania polega na zmniejszaniu przypadków klasy większościowej. Połączenie tych metod pozwala na zachowanie liczebności przypadków zbioru inicjalnego.

2.8. Podsumowanie

Algorytmy rozpoznawania wzorców, związane z metodami sztucznej inteligencji, stanowią istotny element systemów diagnostycznych. Rozpoznawanie wzorców odbywa się na podstawie analizy zbioru cech charakteryzujących poszczególne obiekty.

Do klasycznych metod rozpoznawania wzorców należy zaliczyć klasyfikację minimalno-odległościową, klasyfikację z wykorzystaniem teorii Bayesa, analizę dyskryminacyjną czy analizę drzew decyzyjnych. W pracy zastosowano także rozpoznawanie wzorców oparte na teorii zbiorów przybliżonych. Ideę algorytmu zaprezentowano w rozdziale 3, gdzie omówiono opracowane przez autorkę narzędzie Rough Sests Analysis Toolbox dla środowiska MATLAB.

Podczas rozpoznawania wzorców przy znacznej liczbie atrybutów charakteryzujących poszczególne obiekty, konieczna jest ich redukcja. Zastosowanie redukcji wynika z silnego skorelowania analizowanych atrybutów oraz wysokiej złożoności obliczeniowej niektórych algorytmów klasyfikacji.

Metody redukcji przestrzeni atrybutów, przytoczone na podstawie literatury w rozdziale 2, są jednymi z najczęściej stosowanych. W pracy wykorzystano także selekcję atrybutów z zastosowaniem teorii zbiorów przybliżonych. Algorytmy poszukiwania reduktów dokładnych zaimplementowano w narzędziu Rough Sests Analysis Toolbox.

Dla cech posiadających wartości rzeczywiste możliwa jest ich dyskretyzacja. Zastosowanie dyskretyzacji pozwala na zmniejszenie przestrzeni danych, a także wpływa na przyspieszenie realizacji algorytmów złożonych obliczeniowo. Omówiony algorytm dyskretyzacji metodą równej szerokości (EWD) jest algorytmem powszechnie stosowanym, charakteryzującym się prostą implementacją i szybkością działania. Algorytmy CAIM i CACC są przykładem algorytmów dyskretyzacji, które przy tworzeniu granic przedziałów uwzględniają rozkład klas w zbiorze danych.

Przedstawiono problem danych niezrównoważonych. Nierównomierny rozkład obiektów względem klas wymaga zastosowania dodatkowych kryteriów do oceny jakości klasyfikacji. Zagadnienie jest istotne szczególnie w zadaniach, gdzie priorytetowe jest wykrywanie obiektów należących do klasy mniejszościowej. Jako kryterium zaproponowano funkcję $FMaxSen$, która podkreśla ważność współczynnika czułości przy jednocześnie możliwe wysokiej swoistości.

ROZDZIAŁ 3

Zbiory przybliżone w analizie systemów decyzyjnych

3.1. Wprowadzenie

Teoria zbiorów przybliżonych została wprowadzona przez Zdzisława Pawlaka w 1981 roku [Paw81a,Paw81b,Paw81c]. Zasadniczym elementem, który odróżnia metodę zbiorów przybliżonych od metod statystycznych czy teorii zbiorów rozmytych jest fakt, że dane źródłowe nie ulegają modyfikacjom czy przekształceniom i można je wykorzystać do interpretacji zjawiska. Zagadnienia leżące u podstaw teorii zbiorów przybliżonych zostały bardzo szczegółowo przestawione w [WaMa99,Dom04,Paw83,Rut05,MroPlo99, Baz98,PawSlo07,PaSk07].

Rozwój prac nad zastosowaniem zbiorów przybliżonych można obserwować w nieustannie powiększającej się bazie publikacji Rough Sets Database System[1]. Wśród ośrodków rozwijających teorię zbiorów przybliżonych warto wymienić:

- Zakład Inteligencji Systemów Wspomagania Decyzji w Instytucie Informatyki Politechniki Poznańskiej [2](ROSE) [PreSlo98];
- Instytut Matematyki Politechniki Warszawskiej [3](RSES)[BazSzc00];
- Knowledge Systems Group, Dept. of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway [4](ROSSETTA) [Ohr99];
- Instytut Informatyki Politechniki Warszawskiej (ARES) [Dom04].

Prezentowane w pracy, opracowane przez autorkę, narzędzie Rough Sets Analysis Toolbox (RSA Toolbox, rys.4.1) jest pakietem programów dla środowiska obliczeniowego MATLAB, realizujących podstawowe problemy teorii zbiorów przybliżonych. Zastosowanie środowiska Matlab umożliwia wektorową realizację programów.

[1] http://rsds.univ.rzeszow.pl/
[2] http://idss.cs.put.poznan.pl/site/rose.html
[3] http://logic.mimuw.edu.pl/~rses/
[4] http://www.lcb.uu.se/tools/rosetta/index.php

3.2. Zagadnienia teorii zbiorów przybliżonych

3.2.1. System informacyjny i decyzyjny

Podstawowym pojęciem w teorii zbiorów przybliżonych jest pojęcie systemu informacyjnego. Systemem informacyjnym (ang. information system) określa się uporządkowaną czwórkę:

$$SI = (U, A, V, f), \tag{3.1}$$

gdzie:

U - przestrzeń rozważań będąca niepustym, skończonym zbiorem nazywanym także uniwersum; elementy zbioru $\mathrm{U} = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{n}\}$ określa się mianem obiektów;

A - niepusty, skończony zbiór atrybutów;

$V = \bigcup_{a \in A} V_a$ - Zbiór wszystkich możliwych wartości atrybutów, gdzie V_a jest ziedziną atrybutu $a \in A$;

f - funkcja informacyjna, gdzie $\bigwedge_{\mathrm{a}\in\mathrm{A},\mathrm{x}\in\mathrm{U}} f_x(a) \in \mathrm{V_a}$, $\mathrm{f}\colon \mathrm{U} \times \mathrm{A} \rightarrow \mathrm{V}$.

W pracy przedstawiono teorię zbiorów przybliżonych w odniesieniu do systemów decyzyjnych (ang. decision system), które są szczególnym przypadkiem systemu informacyjnego [Rut05,MroPlo99]. System decyzyjny, nazywany także tablicą decyzyjną (ang. decision table), stanowi uporządkowana piątkę:

$$SD = (U, C, D, V, f) \tag{3.2}$$

gdzie:

C – niepusty, skończony zbiór atrybutów warunkowych, $\mathrm{C} \in \mathrm{A}$, $\mathrm{C} \neq \emptyset$;

D – niepusty, skończony zbiór atrybutów decyzyjnych, $\mathrm{D} \in \mathrm{A}$, $\mathrm{C} \cap \mathrm{D} = \emptyset$, $\mathrm{D} \neq \emptyset$, $\mathrm{C} \cup \mathrm{D} = \mathrm{A}$;

f – funkcja decyzyjna.

Każdy obiekt sytemu decyzyjnego można rozpatrywać jako regułę decyzyjną l. Jeżeli dla każdej pary reguł $l_a \neq l_b$ z równości wszystkich atrybutów warunkowych C wynika równość wszystkich atrybutów decyzyjnych D to reguły określa się mianem deterministycznych:

$$\bigwedge_{\mathrm{l_a},\mathrm{l_b};\, \mathrm{l_a}\neq\mathrm{l_b}} \left(\bigwedge_{c\in C} f_{l_a}(c) = f_{l_b}(c) \rightarrow \bigwedge_{d\in D} f_{l_a}(d) = f_{l_b}(d) \right). \tag{3.3}$$

Jeżeli warunek równości atrybutów nie jest spełniony, to reguły są nie deterministyczne. Wtedy:

$$\bigvee_{\mathrm{l_a},\mathrm{l_b};\, \mathrm{l_a}\neq\mathrm{l_b}} \left(\bigwedge_{\mathrm{c}\in\mathrm{C}} \mathrm{f_{l_a}(c)} = \mathrm{f_{l_b}(c)} \rightarrow \bigvee_{\mathrm{d}\in\mathrm{D}} \mathrm{f_{l_a}(d)} \neq \mathrm{f_{l_b}(d)} \right). \tag{3.4}$$

3.2.2. Zbiory elementarne i aproksymacja zbiorów

Obiekty $x \in U$ mogą być między sobą porównywane poprzez porównywanie wartości atrybutów. Gdy dla zbioru atrybutów $P \subseteq A$ obiekty będą miały jednakowe wartości wszystkich atrybutów to będą nierozróżnialne. Zależność ta nosi nazwę relacji $\tilde{P}$-nierozróżnialności (ang. indiscernibility relation). Jest to relacja określona na przestrzeni $U \times U$, o następującej postaci [Rut05]:

$$x_1 \tilde{P} x_2 \Leftrightarrow \bigwedge_{p\in P} f_{x_1}(p) = f_{x_2}(p), \tag{3.5}$$

gdzie $x_1, x_2 \in U$, $P \subset A$.

Zbiór wszystkich obiektów $x \in U$ będących w relacji $\tilde{P}$ nazywa się klasą abstrakcji relacji $\tilde{P}$-nierozróżnialności lub zbiorem $\tilde{P}$–elementarnym (ang. elementary set). Dla każdego $x \in U$ istnieje dokładnie jeden taki zbiór (ozn. $[x_i]_{\tilde{P}}$), gdzie:

$$[x_i]_{\tilde{P}} = \{x \in U: x_i \tilde{P} x\}. \tag{3.6}$$

Klasy abstrakcji relacji $\tilde{P}$ w przestrzeni U są zbiorami rozłącznymi, a ich rodzinę oznaczamy przez P^* lub U/P.

Niech $X \subset U$ będzie pewnym zbiorem w przestrzeni U. Jeżeli zbiór X jest skończoną sumą zbiorów $\tilde{P}$–elementarnych to nazywa się go zbiorem $\tilde{P}$–dokładnym, w przeciwnym razie zbiór X nazywa się zbiorem $\tilde{P}$–przybliżonym**.** Dla zbioru X określa się parę precyzyjnych zbiorów nazywanych dolną i górną aproksymacją.

$\tilde{P}$–dolną aproksymacją zbioru X nazywa się taki zbiór $\underline{\tilde{P}}X$, którego elementy są obiektami klas abstrakcji będących podzbiorami zbioru X:

$$\underline{\tilde{P}}X = \{x \in U: [x]_{\tilde{P}} \subseteq X\}. \tag{3.7}$$

$\tilde{P}$–górną aproksymacją zbioru X nazywa się taki zbiór $\overline{\tilde{P}}X$, którego elementy są obiektami klas abstrakcji posiadających część wspólną ze zbiorem X:

$$\overline{\tilde{P}}X = \{x \in U: [x]_{\tilde{P}} \cap X \neq \emptyset\}. \tag{3.8}$$

Na podstawie górnej i dolnej aproksymacji można wyznaczyć dodatkowe charakterystyki zbioru X, jak: obszar pozytywny, obszar negatywny oraz obszar brzegowy.

$\tilde{P}$–pozytywny obszar zbioru X odpowiada dolnej aproksymacji zbioru:

$$Pos_{\tilde{P}}(X) = \underline{\tilde{P}}X\ . \tag{3.9}$$

$\tilde{P}$–negatywny obszar zbioru X jest zbiorem obiektów należących do tych klas abstrakcji, o których możemy powiedzieć, że nie są podzbiorami zbioru X:

$$Neg_{\tilde{P}}(X) = \mathrm{U} \backslash \overline{\tilde{P}}X\ . \tag{3.10}$$

$\tilde{P}$–brzegowy obszar zbioru X jest różnicą zbiorów górnej i dolnej aproksymacji:

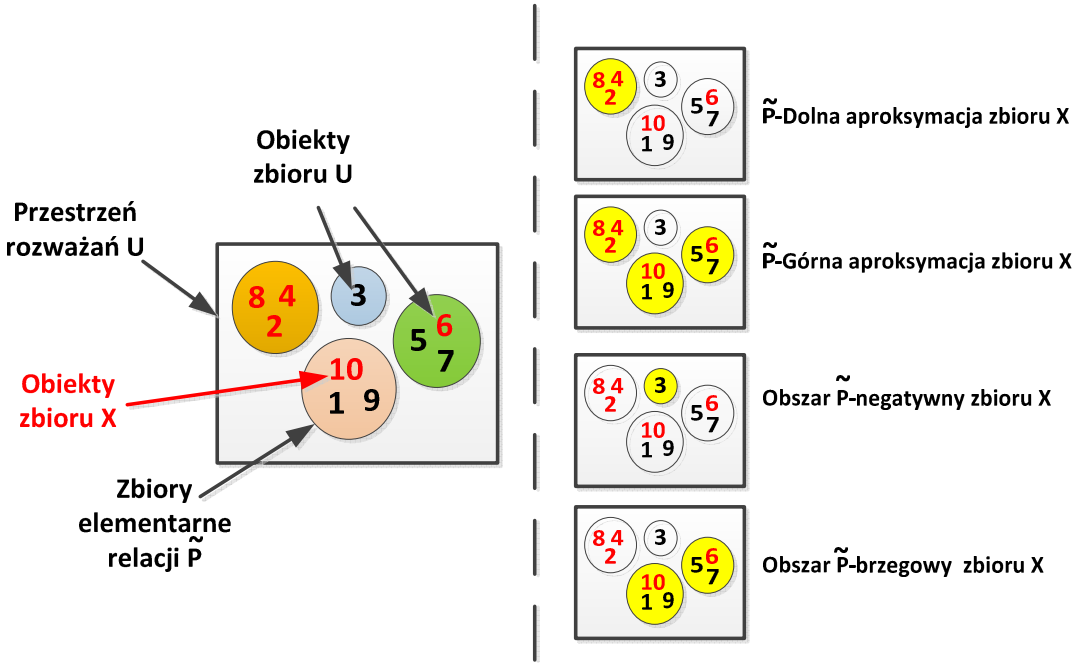
$$Bn_{\tilde{P}}(X) = \overline{\tilde{P}}X \backslash \underline{\tilde{P}}X\ . \tag{3.11}$$

Aproksymację zbioru X w przestrzeni U można przedstawić także w postaci liczbowej. Wartość wyrażoną wzorem:

$$\mu_{\tilde{P}}(X) = \frac{\underline{\tilde{P}}X}{\overline{\tilde{P}}X} \tag{3.12}$$

nazywa się $\tilde{P}$–dokładnością aproksymacji.

Graficzną interpretację pojęć teorii zbiorów przybliżonych zaprezentowano na rys. 3.1. Przedstawiono przestrzeń rozważań U, której elementami są liczby całkowite z przedziału $< 1{,}10 >$. Na przestrzeni U wyznaczono cztery klasy abstrakcji relacji $\tilde{P}$–nierozróżnialności (zbiory elementarne). Klasy abstrakcji oznaczono okręgami. Na przestrzeni U zdefiniowano zbiór X, będący zbiorem liczb parzystych $X = \{2{,}4{,}6{,}8{,}10\}$.



Rys. 3.1. Aproksymacja zbioru X

$\tilde{P}$–dolną aproksymacją zbioru X są te zbiory elementarne, które w całości pokrywają się z obiektami zbioru X. Ponieważ zbiór X jest zbiorem liczb parzystych, to dolną aproksymację stanowi tylko zbiór elementarny zawierający cyfry 2,4,8.

Zbiór $\overline{\tilde{P}}X$, będący $\tilde{P}$–górną aproksymacją zbioru X zawiera te klasy abstrakcji relacji $\tilde{P}$, które posiadają ze zbiorem X przynajmniej jeden element wspólny. W prezentowanym przykładzie będą to zbiory elementarne zawierające: cyfry 2,4,8, cyfry 5,6,7 oraz liczby 1,9,10. Zbiór elementarny zawierający cyfrę 3 stanowi obszar negatywny zbioru X w relacji $\tilde{P}$.

3.2.3. Aproksymacja rodziny zbiorów systemu decyzyjnego

Wartości atrybutów decyzyjnych dzielą przestrzeń U na rodzinę rozłącznych zbiorów $D^* = X = \{X_1, X_2, \dots, X_n\}$, będących klasami abstrakcji relacji $\tilde{D}$-nierozróżnialności. Ponieważ obiekty będące w relacji $\tilde{D}$-nierozróżnialności, przyjmują takie same wartości decyzji, to obiekty te są nierozróżnialne nawet, gdy różnią się wartościami atrybutów warunkowych zbioru C.

Problem aproksymacji przedstawiony w rozdziale 3.2.2, można uogólnić dla rodziny zbiorów. Niech $P \subseteq C$ będzie podzbiorem atrybutów warunkowych.

$\tilde{P}$–dolną aproksymacją rodziny zbiorów X nazywa się zbiór $\underline{\tilde{P}}X$, gdzie:

$$\underline{\tilde{P}}X = \{\underline{\tilde{P}}X_1, \underline{\tilde{P}}X_2, \dots, \underline{\tilde{P}}X_n\}\,. \tag{3.13}$$

$\tilde{P}$–górną aproksymacją rodziny zbiorów X nazywa się zbiór $\overline{\tilde{P}}X$, gdzie:

$$\overline{\tilde{P}}X = \{\overline{\tilde{P}}X_1, \overline{\tilde{P}}X_2, \dots, \overline{\tilde{P}}X_n\}\,. \tag{3.14}$$

$\tilde{P}$–pozytywnym obszarem rodziny zbiorów X nazywa się zbiór $Pos_{\tilde{P}}(X)$, opisany zależnością:

$$Pos_{\tilde{P}}(X) = \bigcup_{X_i \in X} Pos_{\tilde{P}}(X_i)\,. \tag{3.15}$$

$\tilde{P}$–negatywnym obszarem rodziny zbiorów X nazywa się zbiór $Neg_{\tilde{P}}(X)$, definiowany jako:

$$Neg_{\tilde{P}}(X) = U \setminus \bigcup_{X_i \in X} \overline{\tilde{P}} X_i \,. \tag{3.16}$$

$\tilde{P}$–brzegowym obszarem rodziny zbiorów X nazywa się zbiór $Bn_{\tilde{P}}(X)$, określony zależnością:

$$Bn_{\tilde{P}}(X) = \bigcup_{X_i \in X} Bn_{\tilde{P}}(X_i) \,. \tag{3.17}$$

Do liczbowej charakterystyki aproksymacji rodziny zbiorów definiuje się współczynnik jakości oraz współczynnik dokładności aproksymacji.

$\tilde{P}$–jakość aproksymacji rodziny zbiorów X wyznacza się z zależności:

$$\gamma_{\tilde{P}}(X) = \frac{\overline{\overline{Pos_{\tilde{P}}(X)}}}{\overline{\overline{U}}} \,. \tag{3.18}$$

$\tilde{P}$–dokładność aproksymacji rodziny zbiorów X wyraża się wzorem:

$$\beta_{\tilde{P}}(X) = \frac{\overline{\overline{Pos_{\tilde{P}}(X)}}}{\sum_{X_i \in X} \overline{\overline{\tilde{P} X_i}}} \,. \tag{3.19}$$

Powyższe współczynniki spełniają warunek:

$$0 \le \beta_{\tilde{P}}(X) \le \gamma_{\tilde{P}}(X) \le 1 \,. \tag{3.20}$$

3.2.4. Poprawność budowy systemu decyzyjnego

Przedstawione powyżej pojęcia aproksymacji rodziny zbiorów można wykorzystać do weryfikacji poprawności budowy systemu decyzyjnego. W tym celu wprowadza się pojęcie stopnia zależności. Stopień zależności zbioru atrybutów decyzyjnych D od zbioru atrybutów warunkowych C określony jest wyrażeniem:

$$k = \gamma_{\tilde{C}}(D^*) \,. \tag{3.21}$$

Jeżeli poszczególne obiekty przestrzeni U można jednoznacznie przypisać do odpowiednich klas decyzyjnych, to zbiór atrybutów decyzyjnych zależy od zbioru atrybutów warunkowych w stopniu równym 1. Zatem:

$$\gamma_{\tilde{P}}(D^*) = 1 \,. \tag{3.22}$$

Gdy $\gamma_{\tilde{C}}(D^*) = 1$, to tablica decyzyjna jest dobrze określona, a wszystkie jej reguły są deterministyczne. Gdy $\gamma_{\tilde{C}}(D^*) < 1$, to w tablicy decyzyjnej występują reguły niedeterministyczne. Wtedy tablica decyzyjna jest źle określona. Tablicę taką można poprawić poprzez usunięcie reguł niedeterministycznych lub poprzez rozszerzenie zbioru atrybutów warunkowych C.

Istotnym parametrem dla oceny poprawości budowy systemu decyzyjnego jest także znormalizowany współczynnik istotności podzbioru atrybutów warunkowych $C' \subset C$, określony wyrażeniem:

$$\sigma_{(C,D)}(\mathrm{C}') = \frac{\gamma_{\tilde{C}}(D^*) - \gamma_{\tilde{C''}}(D^*)}{\gamma_{\tilde{C}}(D^*)} \tag{3.23}$$

gdzie $C'' = C \setminus C'$.

Jeżeli wartość współczynnika σ dla podzbioru atrybutów $C' \subset C$ będzie równa 0 to podzbiór C' można usunąć bez wpływu na aproksymację rodziny zbiorów D^*.

Jeżeli tablica decyzyjna jest dobrze określona dla zbioru atrybutów warunkowych C to mogą istnieć zbiory atrybutów $P \subset C$, wystarczające do jednoznacznego określenia

odpowiednich klas decyzyjnych. Zbiory takie określa się mianem reduktów względnych. Tablica decyzyjna może mieć więcej niż jeden redut względny lub może nie mieć ich wogóle.

Reduktem względnym zbioru atrybutów C ze względu na zbiór atrybutów D (tzw. D–reduktem) nazywa się każdy D–niezależny zbiór $P \subset C$, dla którego spełniony jest warunek [Rut05,MroPlo99]:

$$\tilde{P} = \tilde{C} . \tag{3.24}$$

Zbiór atrybutów $P \subseteq C$ jest niezależny ze względu na zbiór atrybutów D (D–niezależny), jeśli dla każdego $P_1 \subset P$ zachodzi zależność:

$$Pos_{\tilde{P}}(D^*) \neq Pos_{\tilde{P}_1}(D^*) . \tag{3.25}$$

Pojęcia reduktu względnego nie należy mylić z pojęciem reduktu. Reduktem zbioru atrybutów $P \subseteq A$ nazywa się każdy niezależny zbiór $P_1 \subset P$, dla którego:

$$\tilde{P}_1 = \tilde{P} . \tag{3.26}$$

Przy czym, zbiór atrybutów $P \subseteq A$ nazywa się niezależnym w danym systemie informacyjnym, jeżeli dla każdego $P_1 \subset P$ zachodzi warunek:

$$\tilde{P}_1 \neq \tilde{P} . \tag{3.27}$$

Dowolny podzbiór zbioru atrybutów warunkowych $C' \subset C$ nazywa się przybliżonym D–reduktem zbioru C. Dla każdego zbioru można wyznaczyć błąd przybliżenia jako zależność:

$$\mathcal{E}_{(C,D)}(\mathrm{C}') = \frac{\gamma_{\tilde{C}}(D^*) - \gamma_{\widetilde{C'}}(D^*)}{\gamma_{\tilde{C}}(D^*)}. \tag{3.28}$$

Atrybut $c \in C$ jest nieusuwalny ze zbioru atrybutów warunkowych C ze względu na D (D–nieusuwalny), jeżeli [MroPlo99]:

$$Pos_{\tilde{C}}(D^*) \neq Pos_{\widetilde{C\setminus\{c\}}}(D^*), \tag{3.29}$$

lub co równoważne

$$\gamma_{\widetilde{C\setminus\{c\}}}(D^*) < \gamma_{\tilde{C}}(D^*) . \tag{3.30}$$

Zbiór wszystkich nieusuwalnych atrybutów nosi nazwę D–rdzenia i określony jest wyrażeniem [MroPlo99]:

$$CORE_D(C) = \{c \in C: Pos_{\tilde{C}}(D^*) \neq Pos_{\widetilde{C\setminus\{c\}}}(D^*) \tag{3.31}$$

Dla tablicy decyzyjnej dobrze określonej D–rdzeń atrybutów warunkowych można wykorzystać do wyznaczenia D–reduktów.

3.2.5. Macierz, tablica oraz funkcja rozróżnialności dla systemu decyzyjnego

Uzupełnieniem opisu relacji nierozróżnialności jest macierz rozróżnialności (ang. discernibility matrix). Macierz rozróżnialności systemu decyzyjnego zawiera informacje o atrybutach, które rozróżniają każdą parę obiektów należących do różnych klas decyzyjnych. Macierz rozróżnialności można zdefiniować w następujący sposób:

$$M_D(SD) = \left(m_{ij}\right)_{nxn}, \tag{3.32}$$

gdzie $m_{ij} = \left\{c \in C: f_{x_i}(c) \neq f_{x_j}(c) \wedge f_{x_i}(d) \neq f_{x_j}(d)\right\}$, $i,j = 1,2,..,n$ oraz $n = \overline{\overline{U}}$.

Macierz rozróżnialności posiada następujące właściwości:

- Każdy element macierzy jest zbiorem;
- Elementy macierzy leżące na głównej przekątnej są zbiorami pustymi: $m_{ij} = \varnothing$ dla $i = j$;
- Macierz jest symetryczna względem głównej przekątnej: $m_{ij} = m_{ji}$;
- Rozmiar macierzy odpowiada liczbie obiektów tablicy decyzyjnej: $n = \overline{\overline{U}}$.

Ponieważ zawartości komórek macierzy rozróżnialności nie są typami prostymi to macierz charakteryzuje się dużą złożonością pamięciową. Odpowiednikiem macierzy rozróżnialności jest tablica rozróżnialności (ang. discernibility table) określona zależnością:

$$T_D(SD) = \left(t_{(ij),k}\right)_{(i,j)xm} \tag{3.33}$$

gdzie:

$$\bigwedge_{c\in C} t_{(ij,k)} = \begin{cases} 0;\ f_{x_i}(c) = f_{x_j}(c) \vee f_{x_i}(d) = f_{x_j}(d) \\ 1;\ f_{x_i}(c) \neq f_{x_j}(c) \wedge f_{x_i}(d) \neq f_{x_j}(d) \end{cases}, i,j = 1,2,..,n\ ,\ i > j,\ k = 1,2,..m\ . \tag{3.34}$$

Poszczególne elementy tablicy T_D przyjmują wartości 0 lub 1. Pozwala to na korzystniejszą implementację programową, gdyż poszczególne wiersze tabeli mogą być reprezentowane w postaci bitowej.

Wiedzę zawartą w macierzy lub w tablicy rozróżnialności można przedstawić za pomocą funkcji rozróżnialności (ang. discernibility function). Jest to funkcja boolowska, która każdemu atrybutowi ze zbioru atrybutów warunkowych przypisuje zmienną boolowską c^*. Jeżeli zbiór atrybutów macierzy lub tablicy rozróżnialności jest pusty to funkcji przypisuje się stałą Boolowską 1:

$$f_D(SD) = \cap \left\{ \cup \left(t_{(ij),k} : 1 \leq j \leq i \leq n\ \wedge\ 1 \leq k \leq m\ \wedge t_{(ij),k} \neq \varnothing \right) \right\} \tag{3.35}$$

lub

$$f_D(SD) = \cap \left\{ \cup \left(m_{i,j} : 1 \leq j \leq i \leq n\ \wedge m_{i,j} \neq \varnothing \right) \right\} \tag{3.36}$$

Funkcję rozróżnialności można uprościć m.in. stosując prawo pochłaniania. Funkcję rozróżnialności stosuje się przy rozwiązywaniu wielu problemów teorii zbiorów przybliżonych. Jednym z głównych zastosowań jest zadanie wyznaczania wszystkich reduktów systemu informacyjnego .

3.3. Przybornik Rough Sets Analysis Toolbox

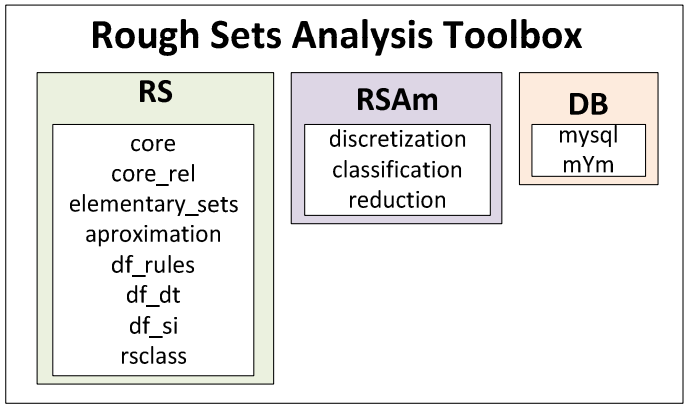
Głównym zadaniem realizowanym przez Rough Sets Analysis Toolbox (RSA Toolbox) jest zastosowanie teorii zbiorów przybliżonych do redukcji przestrzeni atrybutów i klasyfikacji danych. Strukturę narzędzia RSAToolbox przedstawiono na rysunku 3.2.

Moduł RS zawiera programy do przeprowadzania analiz przy użyciu teorii zbiorów przybliżonych. Umożliwiają one m.in. poszukiwanie zbiorów elementarnych, określanie aproksymacji zbiorów i rodziny zbiorów, wyznaczanie rdzenia atrybutów i reduktów, a także generowanie macierzy, tablicy i funkcji rozróżnialności. W module RS zawarto także programy dla opracowanego przez autorkę klasyfikatora wykorzystującego połączenie reguł decyzyjnych i metody k-najbliższych sąsiadów. Każdy z programów jest zapisany w postaci funkcji, co umożliwia niezależne uruchamianie w innych programach środowiska MATLAB. Wybrane programy przygotowano w dwóch wersjach:

jednostanowiskowej i rozproszonej. Wersje jednostanowiskowe programów omówiono w rozdziale 3.4 natomiast wersje rozproszone w rozdziale 3.7.

Moduł RSAm jest zbiorem programów służących do realizacji procesu uczenia nadzorowanego. Umożliwia poszukiwanie optymalnego zbioru atrybutów oraz optymalnego klasyfikatora. Podstawowymi programami redukcji atrybutów oraz klasyfikacji są implementacje teorii zbiorów przybliżonych zawarte w module RS. Jednakże, w module RSAm, uwzględniono również możliwość wykorzystania funkcji klasyfikacyjnych dostępnych w narzędziu Statistic Toolbox, jak: Naiwny klasyfikator Bayesa (NaiveBayes), Analiza dyskryminacyjna (classify), Drzewa decyzyjne (classregree). Zaimplementowano także algorytmy redukcji wymiaru (np.: selekcja metodą korelacji, analiza głównych składowych) oraz dyskretyzacji atrybutów metodami EWD [Ci00,StaSzy06,199], CAIM [KuCi04,Szy09b], CACC [TsLe08,Szy09b]. W implementacji algorytmów CAIM i CACC wykorzystano funkcje udostępnione w centrum wymiany plików MATLAB [wCAIM09,wCACC09]. Wprowadzono zmianę w kodzie programu poprzez zamianę instrukcji warunkowych i pętli sterujących na operacje arytmetyczne wykonywane bezpośrednio na pełnych wektorach lub macierzach (rozdział 3.5.1).

Moduł DB zawiera zbiór funkcji umożliwiających komunikację środowiska MATLAB z bazą danych MySql. Moduł DB wykorzystywany jest w programach rozproszonych, co pozwala na przyspieszenie obliczeń dla analiz wielowymiarowych.



Rys. 3.2. Struktura narzędzia RSA Toolbox

3.4. Moduł RS

3.4.1. Zbiory elementarne

Do wyznaczania zbiorów elementarnych opracowano funkcję elementary_sets. Argumentami wejściowymi funkcji są: zbiór danych oraz zbiór atrybutów definiujący relację $\tilde{P}$-nierozróżnialności. Wynikiem działania programu są zmienne zawierające zbiory elementarne, ich liczbę oraz wektor określający przynależność każdego z obiektów do klasy abstrakcji. Schemat działania programu przedstawiono na listingu 3.1.

W pierwszym kroku wyznaczono możliwe klasy abstrakcji relacji $\tilde{P}$-nierozróżnialności, a wynik zapisano w zmiennej ElementarySets. W drugim kroku określono przynależność obiektów zbioru DataSet do utworzonych zbiorów elementarnych. Wykorzystano w tym celu funkcję ismember, która zwraca dwa wektory:

- IsEsMemeber – wektor binarny o rozmiarze odpowiadającym liczbie obiektów macierzy DataSet. Wartość 1 określa wystąpienie wiersza macierzy DataSet w macierzy ElementarySets;
- ObjectsLocation - wektor, który dla każdego wiersza macierzy DataSet przechowuje indeks równoważnego wiersza w macierzy ElementarySets.

Listing 3.1. Realizacja funkcji wyznaczającej zbiory elementarne (elementary_sets.m)

```
1   Dane wejściowe:
2     DataSet[xn,xm] – Zbiór  uczący;  xn – liczba obiektów, xm – liczba atrybutów warunkowych;
3     AttrList – Zbiór cech określających relację P̃. Nazwy cech odpowiadają numerom kolumn macierzy X.
4
5   Dane wyjściowe:
6     ElementarySets – lista zbiorów elementranych
7     NumOfElementarySets – liczba utworzonych zbiorów elementarnych;
8     ObjectsLocation[xn,1] – wektor określajacy przynależność obiektu do zbioru elementarnego
9
10  Krok 1.
11        ElementarySets =unique(DataSet[ (:,AttrList),'rows');
12  Krok 2.
13        [IsEsMemeber,ObjectsLocation] = ismember(DataSet[ (:,AttrList), ElementarySets, 'rows');
14  Krok 3.
15        NumOfElementarySets=size(ElementarySets,1);
```

Ponieważ macierz ElementarySets utworzono na podstawie macierzy DataSet, to wektor IsEsMemeber zawiera tylko wartości 1. Jeden obiekt zbioru DataSet może należeć tylko do jednego zbioru elementarnego. Obiekty posiadające ten sam indeks wiersza będą należały do tej samej klasy abstrakcji.

W ostatnim kroku (linia 14) wyznaczono liczbę utworzonych zbiorów elementarnych. Maksymalna liczba tworzonych zbiorów elementarnych odpowiada liczbie wierszy macierzy DataSet. Sytuacja taka będzie miała miejsce, gdy wszystkie wiersze macierzy DataSet będą rozróżnialne ze względu na zadany zbiór cech (AttrList).

3.4.2. Aproksymacja zbioru, rodziny zbiorów

Do rozwiązywania grupy zadań związanych z pojęciami aproksymacji opracowano funkcję aproximation. Argumentami wejściowymi funkcji są: zbiór danych (DataSet), zbiór atrybutów decyzyjnych (AtrybutyDecyzyjne), zbiór atrybutów warunkowych (AtrybutyWarunkowe). Elementy wektorów AtrybutyDecyzyjne i AtrybutyWarunkowe zawierają indeksy kolumn macierzy DataSet. Atrybuty decyzyjne wyznaczają rodzinę rozłącznych zbiorów $D^* = X = \{X_1, X_2, \dots, X_n\}$. Atrybuty warunkowe definiują relację $\tilde{P}$–nierozróżnialności.

Dla każdego ze zbiorów: $X_1, X_2, \dots, X_n$ program wyznacza:
- $\tilde{P}$–dolną i $\tilde{P}$–górną aproksymację ;
- obszary: $\tilde{P}$–pozytywny, $\tilde{P}$–brzegowy, $\tilde{P}$–negatywny;
- współczynnik dokładności aproksymacji $\mu_{\tilde{P}}(X_i)$.

Dla rodziny zbiorów $D^* = X$ program wyznacza:
- obszary: $\tilde{P}$–pozytywny, $\tilde{P}$–brzegowy, $\tilde{P}$–negatywny;
- współczynnik $\tilde{P}$ –jakości aproksymacji $\gamma_{\tilde{P}}(X)$;
- współczynnik $\tilde{P}$–dokładności aproksymacji $\beta_{\tilde{P}}(X)$.

Schemat realizacji programu aproximation przedstawiono na listingu 3.2. Wyznaczone charakterystyki przechowywane są w tablicy strukturalnej o nazwie Aproksymacja. Głównymi polami tablicy są:

- ZE – przechowuje wyniki aproksymacji dla zbiorów elementarnych;
- RZ - przechowuje wyniki aproksymacji rodziny zbiorów;
- ObjectsLocationForAproxAttr – zawiera informację o przynależności obiektów do klas decyzyjnych.

Listing 3.2. Realizacja programu aproksymacji (aproximation.m)

```
Dane wejściowe:
   DataSet [xn,xm] – Zbiór  uczący;  xn – liczba obiektów, xm – liczba atrybutów warunkowych;
   AtrybutyDecyzyjne - Zawiera numery kolumn odpowiadających atrybutom decyzyjnym zbioru  D̃
   AtrybutyWarunkowe - Zawiera numery kolumn odpowiadających atrybutom zbioru P̃

Dane wyjściowe:
   Aproksymacja – struktura zawierająca wyniki obliczeń

Krok 1.
        [ElementarySets, ObjectsLocationForDecAttrBin,NumOfElementarySets]= …
            elementary_sets(DataSet, AtrybutyDecyzyjne);
Krok 2.
        ObjectsLocationForDecAttrBin=classtobin(ObjectsLocationForDecAttr, NumOfElementarySets);
Krok 3.
        [ElementarySets, ObjectsLocationForAproxAttrBin,NumOfElementarySets]=…
            elementary_sets(DataSet, AtrybutyWarunkowe);
Krok 4.
        ObjectsLocationForAproxAttrBin=classtobin(ObjectsLocationForAproxAttrBin, NumOfElementarySets);
Krok 5.
        Dla każdego i-tego zbioru elementarnego relacji D-nierozróżnialności
Krok 5.1.
        IndeksyObiektowZbioruDecyzyjnego=find(ObjectsLocationForDecAttrBin(:,i)==1)';
        ZE(i).IndeksyObiektowZbioruDecyzyjnego=IndeksyObiektowZbioruDecyzyjnego;
Krok 5.2.
        temp=repmat(ObjectsLocationForDecAttrBin(:,i),1,NumOfElementarySetsForAproxAttr);
        temp=ObjectsLocationForAproxAttrBin & temp;
Krok 5.3.
        LiczbaWspolnychObiektow=sum(temp);
        LiczbaObiektowWZbiorzeElementarnym=sum(ObjectsLocationForAproxAttrBin);
Krok 5.4.
    ZE(i).IndeksyZbiorowElementarnychDolnejAproksymacji =…
    ~(sign(LiczbaObiektowWZbiorzeElementarnym-LiczbaWspolnychObiektow));
    ZE(i).IndeksyObiektowDolnejAproksymacji=…
    find((sum((ObjectsLocationForAproxAttrBin(:,IndeksyZbiorowElementarnychDolnejAproksymacji)),2))==1);
Krok 5.5
    ZE(i).IndeksyZbiorowElementarnychGornejAproksymacji=logical(sign(LiczbaWspolnychObiektow));
    ZE(i).IndeksyObiektowGornejAproksymacji=…
    find((sum((ObjectsLocationForAproxAttrBin(:,IndeksyZbiorowElementarnychGornejAproksymacji)),2))==1)
Krok 5.6
    ZE(i).PosAproxAttr =IndeksyObiektowDolnejAproksymacji;
Krok 5.7
    ZE(i).BnAproxAttr =setdiff(IndeksyObiektowGornejAproksymacji, IndeksyObiektowDolnejAproksymacji) ;
Krok 5.8
        ZE(i).NegAproxAttr =setdiff(IndeksyObiektow,IndeksyObiektowGornejAproksymacji) ;
Krok 5.9
        ZE(i).uAproxAttr=…
            (size(IndeksyObiektowDolnejAproksymacji,2))/(size(IndeksyObiektowGornejAproksymacji,2));
Krok 5.10
        IndObDolnejAproksymacjiRodzinyZbiorow=…
```

```
50                union(IndeksyObiektowDolnejAproksymacji,IndObDolnejAproksymacjiRodzinyZbiorow);
51            IndObGornejAproksymacjiRodzinyZbiorow=…
52                union(IndeksyObiektowGornejAproksymacji,IndObGornejAproksymacjiRodzinyZbiorow);
53            LiczbaObiektowGornAprokRodzZb=…
54                LiczbaObiektowGornAprokRodzZb+(size(ZE(i).IndeksyObiektowGornejAproksymacji,2));
55    Krok 6.
56            RZ.PosAproxAttr =IndObDolnejAproksymacjiRodzinyZbiorow;
57            RZ.BnAproxAttr =…
58                setdiff(IndObGornejAproksymacjiRodzinyZbiorow, IndObDolnejAproksymacjiRodzinyZbiorow);
              RZ.NegAproxAttr=setdiff(IndeksyObiektow,IndObGornejAproksymacjiRodzinyZbiorow) ;
59            RZ.gammaAproxAttr =(size(RZ.PosAproxAttr,2))/NumOfObjects;
60            RZ.betaAproxAttr =(size(RZ.PosAproxAttr,2))/LiczbaObiektowGornAprokRodzZb;
61    Krok 7.
62            Aproksymacja. RZ= RZ;
63            Aproksymacja. ZE = ZE;
64            Aproksymacja. ObjectsLocationForAproxAttrBin= ObjectsLocationForAproxAttrBin;
65
```

W pierwszym kroku wyznaczono zbiory elementarne dla relacji $\tilde{D}$–nierozróżnialności. Funkcja elementary_sets, omówiona w rozdziale 3.4.1, zwraca wektor ObjectsLocationForDecAttr zawierający informację o przynależności obiektów do klas. Wektor przekształcany jest na postać macierzy binarnej (linia 13). Operację konwersji realizuje funkcja classtobin, omówiona w rozdziale 3.4.13. Każda z kolumn macierzy ObjectsLocationForDecAttrBin odpowiada jednej klasie. Wartość 1 w komórce macierzy oznacza, że obiekt wskazany przez numer wiersza, jest przypisany do klasy wskazanej przez numer kolumny. W krokach 3 i 4 wyznaczono klasy abstrakcji dla zbioru atrybutów warunkowych tworzących relację $\tilde{P}$-nierozróżnialności.

W kroku 5 zrealizowano aproksymację każdego zbioru z rodziny zbiorów $X = \{X_1, X_2, \dots, X_n\}$. Założono, że atrybuty decyzyjne tworzą mniejszą rodzinę zbiorów elementarnych niż atrybuty warunkowe. Założenie to wykorzystano przy dobieraniu zmiennej sterującej pętli for wskazując na liczbę zbiorów dla atrybutu decyzyjnego. Umożliwia to przeprowadzenie większej liczby operacji w sposób wektorowy.

W celu sprawdzenia ile obiektów ze zbioru aproksymującego zawiera się w danej klasie decyzyjnej przeprowadzono dwie operacje. Pierwsza polega na skopiowaniu wektora klasy decyzyjnej w liczbie odpowiadającej zbiorom elementarnych realacji $\tilde{P}$–nierozróżnialności (Krok 5.2). Następnie w celu znalezienia wspólnych obiektów otrzymanej macierzy temp oraz macierzy ObjectsLocationForAproxAttrBin przeprowadzono logiczną operację AND. Jeżeli liczba obiektów w danej kolumnie odpowiada liczbie obiektów zbioru elementarnego realacji $\tilde{P}$, to zbiór elementarny jest częścią dolnej aproksymacji. Porównananie zrealizowano jako różnicę macierzy LiczbaObiektowWZbiorzeElementarnym oraz LiczbaWspolnychObiektow. W wyniku otrzymano wartość 0 dla tych zbiorów elementarnych, gdzie liczba obiektów macierzy temp i zbioru elementarnego jest taka sama (Krok 5.3). Funkcję sign zastosowano do konwersji macierzy do postaci zero-jedynkowej (Krok 5.4). Negacja macierzy umożliwia jednoznaczną identyfikację zbiorów elementarnych. Komórki macierzy posiadające wartość 1 odpowiadają zbiorom elementarnym tworzącym dolną aproksymację zbioru X_i. W linii 33 wyznaczono indeksy obiektów należące do dolnej aproksymacji. Ponieważ zbiory elementarne są rozłączne to w wyniku sumowania wartości w wierszach otrzymuje się jeden wektor. Wartości komórek wektora równe 1 wskazują na indeksy obiektów dolnej aproksymacji.

Do wyznaczenia górnej aproksymacji zbioru X_i wykorzystano wektor LiczbaWspolnychObiektow. Jeżeli liczba znalezionych obiektów części wspólnej zbiorów jest większa lub równa wartości 1 to zbiór ten jest częścią górnej aproksymacji (Krok 5.5).

W kroku 5.6, na mocy wzoru 3.9, określono obszar $\tilde{P}$–pozytywny. W kroku 5.7, na mocy wzoru 3.11, wyznaczono obszar $\tilde{P}$–brzegowy. Natomiast w kroku 5.8, na mocy wzoru 3.10, wyznaczono obszar $\tilde{P}$–negatywny.

Zgodnie ze wzorem 3.12 obliczono współczynnik dokładności aproksymacji (Krok 5.9). Współczynnik odpowiada ilorazowi liczby elementów dolnego przybliżenia i liczby elementów górnego przybliżenia.

W kroku 6 wyznaczono zbiory cech oraz współczynniki aproksymacji dla rodziny zbiorów D^*. Obszar $\tilde{P}$–pozytywny rodziny zbiorów odpowiada obiektom należącym do górnej aproksymacji rodziny zbiorów (linia 56, wzór 3.15). Obszar $\tilde{P}$–brzegowy wyznaczono jako różnicę zbiorów będących elementami górnej i dolnej aproksymacji rodziny zbiorów (linia 57, wzór 3.17). Obszar $\tilde{P}$–negatywny odpowiada obiektom, które nie należą do górnej aproksymacji rodziny zbiorów (linia 69, wzór 3.16).

Współczynnik $\tilde{P}$–jakości aproksymacji rodziny zbiorów wyznaczono na podstawie zależności 3.18 (linia 60), a współczynnik $\tilde{P}$–dokładności na podstawie wzoru 3.19 (linia 61).

3.4.3. Rdzeń

Wyznaczanie rdzenia polega na sprawdzeniu, jak na aproksymację zbioru wpływa usunięcie każdego atrybutu z relacji $\tilde{P}$-nierozróżnialości. Do okreslenia rdzenia atrybutów opracowano funkcję core. Argumentami wejściowymi funkcji są: zbiór danych oraz zbiór atrybutów relacji $\tilde{P}$–nierozróżnialności. Schemat realizacji programu przedstawiono na listingu 3.3.

Listing 3.3. Realizacja programu wyznaczania rdzenia atrybutów (core.m)

```
1   Dane wejściowe:
2       DataSet – Zbiór  uczący;  xn – liczba obiektów, xm – liczba atrybutów warunkowych;
3       AtrybutyWarunkowe – Zawiera numery kolumn odpowiadające atrybutom warunkowym zbioru P̃
4
5   Dane wyjściowe:
6       rdzen – struktura zawierająca wyniki aproksymacji;
7
8   Krok 1.
9       [ElementarySets,ObjectsLocation, NumOfElementarySets]= elementary_sets(DataSet, Atrybuty);
10      ObjectsLocationForDecAttrBin=classtobin(ObjectsLocation,NumOfElementarySets);
11      N1=NumOfElementarySets;
12      O1=ObjectsLocationForDecAttrBin;
13  Krok 2.
14      Dla każdego i-tego atrybutu
15  Krok 2.1
16      AtrybutyMod=setdiff(Atrybuty, Atrybuty (1,i));
17  Krok 2.2.
18      [ElementarySets,ObjectsLocation,NumOfElementarySets]=elementary_sets(DataSet, AtrybutyMod);
19      ObjectsLocationForDecAttrBin=classtobin(ObjectsLocation,NumOfElementarySets);
20  Krok 2.3
21      if NumOfElementarySets~=N1
22          AtrNieusuwalne(1,i)=1;
23      end
24  Krok 2.4
25      if NumOfElementarySets==N1
26          RoznicaZbiorow=setdiff(O1(:,1: NumOfElementarySets)',
27          ObjectsLocationForDecAttrBin(:,1: NumOfElementarySets)','rows');
28      end
29  Krok 2.5
30      if RoznicaZbiorow
```

```
31                      AtrNieusuwalne(1,i)=1;
32              end
33      Krok 3.
34              rdzen=find(AtrNieusuwalne==1);
```

3.4.4. Rdzeń względny

Rdzeń względny jest zbiorem wszystkich atrybutów nieusuwalnych względem innego zbioru atrybutów. W analizie systemów decyzyjnych zależność ta sprowadza się do wyznaczenia D-rdzenia, czyli zbioru wszystkich D-nieusuwalnych atrybutów warunkowych (wzór 3.31). Schemat realizacji programu core_rel, opracowanego do wyznaczania rdzenia względnego przedstawiono na listingu 3.4.

Listing 3.4. Realizacja programu wyznaczania rdzenia względnego (core_rel.m)

```
1       Dane wejściowe:
2               DataSet – Zbiór  uczący;  xn – liczba obiektów, xm – liczba atrybutów warunkowych;
3               AtrybutyDecyzyjne – Zawiera numery kolumn odpowiadające atrybutom decyzyjnym zbioru D̃
4               AtrybutyWarunkowe – Zawiera numery kolumn odpowiadające atrybutom warunkowym zbioru P̃
5
6       Dane wyjściowe:
7               rdzen_wzgledny – wektor zawierający indeksy atrybutów nieusuwalnych;
8
9       Krok 1.
10              wynik_all= aproximation(DataSet,AtrybutyDecyzyjne,AtrybutyWarunkowe)
11      Krok 2.
12              Dla każdego i-tego atrybutu
13      Krok 2.1
14              Atrybuty=setdiff(AtrybutyWarunkowe, AtrybutyWarunkowe(1,i));
15      Krok 2.2.
16               wynik_temp= aproximation(DataSet,AtrybutyDecyzyjne,Atrybuty)
17      Krok 2.3.
18               if wynik_temp.RZ.gammaAproxAttr<wynik_all.RZ.gammaAproxAttr
19                       AtrNieusuwalne(1,i)=1;
20               end
21      Krok 3.
22               rdzen_wzgledny= find(AtrNieusuwalne==1);
```

Do określenia rdzenia względnego atrybutów wykorzystano funkcję aproximation (rozdział 3.4.2). W kroku pierwszym wyznaczono jakość aproksymacji pełnego zbioru atrybutów warunkowych względem atrybutu decyzyjnego. Współczynnik jakości wykorzystano w kroku drugim do oceny każdego atrybutu ze zbioru $\tilde{P}$. Przy ocenie uwzględniono współczynnik jakości aproksymacji rodziny zbiorów dla relacji $\widetilde{P\{p_i\}}$. Współczynnik wyznaczono w kroku 2.2. Porównanie współczynników zrealizowano w kroku 2.3. i-ty atrybut będzie nieusuwalny ze zbioru, gdy jakość aproksymacji rodziny zbiorów relacji $\widetilde{P\{p_i\}}$ będzie mniejsza od jakości aproksymacji pełnego zbioru atrybutów (Krok 2.3).

3.4.5. Tablica rozróżnialności systemu informacyjnego

Tablica rozróżnialności zawiera informację o atrybutach rozróżniających każde dwa obiekty w zbiorze danych (wzór 3.34). Rozmiar tablicy rozróżnialności zależy od liczby obiektów i liczby atrybutów. Liczba obiektów wpływa na pamięciową złożoność obliczeniową. Wiersze tablicy rozróżnialności odpowiadają porównaniu każdej pary obiektów systemu informacyjnego, a ich liczba wynosi $\frac{n^2-n}{2}$. Reprezentacja tablicy rozróżnialności dla wielowymiarowych zbiorów danych wymaga dużej pojemności pamięci operacyjnej. W praktycznym zastosowaniu wymagającym obliczenia tablicy rozróżnialności (np. wyznaczenie reduktów) wystarczy wyznaczenie aktualnie potrzebnych fragmentów tablicy. Uwzględniono to w opracowywanym programie dt_si służącym do obliczenia tablicy rozróżnialności systemu informacyjnego. Schemat programu zamieszczono na listingu 3.5.

Listing 3.5. Realizacja programu wyznaczania tablicy rozróżnialności (dt_si.m)

```
Dane wejściowe:
        X – Zbiór  uczący;  xn – liczba obiektów, xm – liczba atrybutów warunkowych;
        od – numer atrybutu od którego mają być wyznaczane fragmenty tablicy rozróżnialności
        od – numer atrybutu do którego mają być wyznaczane fragmenty tablicy rozróżnialności

Dane wyjściowe:
        X_DT – fragment tablicy rozróżnialności dla zadanej grupy obiektów sąsiadujących;

Krok 1.
    Dla pierwszego obiektu i=od wykonaj:
Krok 1.1
        X0=X(1:i,:);
        X1=repmat(X(i,:),n-i,1);
        X2=[X0;X1];
Krok 1.2
        X_DT1=X2-X;
Krok 1.3
        X_DT=logical(sign(abs(X_DT1)));

Krok 2.
    Dla pozostałych obiektów wykonaj:
Krok 2.1
        X0=X(1:i,:);
        X1=repmat(X(i,:),n-i,1);
        X2=[X0;X1];
Krok 2.2
        X_DT1=X2-X;
Krok 2.3
        X_DT1=logical(sign(abs(X_DT1)));
Krok 2.4
        X_DT=cat(3,X_DT,X_DT1);
```

Funkcja dt_si wymaga trzech argumentów wejściowych: macierzy X odpowiadającej systemowi informacyjnemu oraz dwóch zmiennych (od,do) określających zakres obiektów, dla których należy wyznaczyć tablicę rozróżnialności. Program prezentowany na listingu 3.5 podzielono na dwie części. Pierwsza dotyczy wyznaczenia fragmentu tablicy rozróżnialności dla pierwszego obiektu (Krok 1). Druga część programu to wyznaczanie fragmentów tablicy dla pozostałych obiektów. Tablica rozróżnialności ma strukturę

trójwymiarową. Każdy indeks trzeciego wymiaru odpowiada porównaniu i-tego obiektu z pozostałymi. W przypadku, gdy funkcja dt_si wykorzystywana jest dla zbioru danych o dużej liczbie obiektów, to zaleca się wywołanie tej funkcji dla każdego obiektu osobno.

Porównanie wartości cech i-tego obiektu z pozostałymi zrealizowano jako różnicę dwóch macierzy X oraz X2. Macierz X2 jest złączeniem dwóch macierzy:

- X0 - kopia macierzy X dla wierszy o indeksach mniejszych i równych od i-tego;
- X1 - macierz w której wszystkie wiersze są jednakowe i odpowiadają wektorowi wartości atrybutów analizowanego obiektu.

W wyniku różnicy macierzy X2 i X utworzono macierz, której elementy o wartościach zero odpowiadają jednakowej wartości atrybutu dla porównywanych wierszy. Pozostałe elementy macierzy zawierają wartości różne od zera i wskazują na atrybuty rozróżniające obiekty. W celu uproszczenia zapisu, elementom rozróżniającym przypisano wartość 1 (Krok 1.3). Zastosowano w tym celu funkcję sign.

W podobny sposób przeprowadzono porównania pozostałych obiektów (Krok 2.1 – Krok 2.3). Wyznaczone fragmenty tablicy rozróżnialności zapisywane są w trzecim wymiarze (Krok 2.4).

W funkcji dt_si2 wprowadzono modyfikację - tablica rozróżnialności wyznaczna jest dla porównania jednego obiektu. Schemat realizacji programu dt_si2 przedstawiono na listingu 3.6. Argumentem wejściowym funkcji dt_si2 jest fragment systemu informacyjnego w którym pierwszy wiersz jest wierszem porównywanym. W przedstawionym programie nie występuje macierz trójwymiarowa. Liczba wierszy fragmentu tablicy rozróżnialności odpowiada liczbie porównań obiektów.

Listing 3.6. Realizacja programu wyznaczania tablicy rozróżnialności (dt_si2.m)

```
1     Dane wejściowe:
2           X[xn,xm] – Zbiór  danych;
3
4     Dane wyjściowe:
5           X_DT – fragment tablicy rozróżnialności odpowiadający określonemu obiektowi;
6
7     Krok 1.
8           X1=repmat(X(1,:),n-1,1);
9     Krok 2.
10          X_DT=X1-X(2:n,:);
11    Krok 3.
12          X_DT=logical(sign(abs(X_DT)));
```

3.4.6. Tablica rozróżnialności systemu decyzyjnego

Tablica rozróżnialności systemu decyzyjnego nie zawiera informacji o obiektach pochodzących z tej samej klasy. W celu wyznaczenia fragmentów tablicy rozróżnialności rozbudowano program dt_si2. Schemat programu dt_sd realizującego wyznaczanie tablicy rozróżnialności systemu decyzyjnego przedstawiono na listingu 3.7.

W przedstawionym listingu fragmenty tablicy rozróżnialności systemu decyzyjnego wyznaczono w sposób przedstawiony dla systemu informacyjnego (Krok 1-Krok 3). Tablica wyznaczana jest na podstawie porównania pierwszego wiersza z pozostałymi, bez względu na przynależność do klasy. Następnie w kroku 4 usuwane są te wiersze tablicy, które odpowiadają porównaniu wartości atrybutów obiektów należących do tej samej klasy.

Funkcję dt_sd wykorzystano do wyznaczania reduktów względnych (rozdział 3.4.9). Z tego powodu uproszczono sposób zapisu tablicy rozróżnialności systemu decyzyjnego:

- usunięto wiersze, które wskazują na to, że porównywane obiekty miały te same wartości dla wszystkich atrybutów (Krok 5);
- usunięto powtórzenia wierszy (Krok 6);
- usunięto wiersze będące nadzbiorami innych wierszy (Krok 7); zastosowano w tym celu funkcję pochłanianie omówioną w rozdziale 3.4.12;

Listing 3.7. Realizacja programu wyznaczania tablicy rozróżnialności systemu decyzyjnego (dt_sd.m)

```
1   Dane wejściowe:
2           X[xn,xm] – Zbiór  danych dla i-tego obiektu;
3           MacierzKlas [xn,1] – Wektor klas;
4
5   Dane wyjściowe:
6           X_DT – fragment tablicy rozróżnialności odpowiadający określonemu obiektowi;
7
8   Krok 1.
9           X1=repmat(X(1,:),n-1,1);
10  Krok 2.
11          X_DT=X1-X(2:n,:);
12  Krok 3.
13          X_DT=logical(sign(abs(X_DT)));
14  Krok4.
15          temp_MacierzKlas=MacierzKlas(2:n);
16          X_DT(temp_MacierzKlas(:)==MacierzKlas(1),:)=[];
17  Krok 5.
18          X_DT( sum(~X_DT,2)==m,:)=[];
19  Krok 6.
20          X_DT=unique(X_DT,'rows');
21  Krok 7.
22          X_DT=pochlanianie(X_DT);
```

3.4.7. Tablica rozróżnialności w zadaniu klasyfikacji

Tablicę rozróżnialności wykorzystano przy wyznaczaniu reguł decyzyjnych klasyfikatora, opartego na teorii zbiorów przybliżonych (rozdział 3.4.10). Reguły decyzyjne klasyfikatora wyznaczono na podstawie analiz fragmentów tablicy rozróżnialności. Każdy fragment tablicy rozróżnialności jest wynikiem porównania jednego obiektu ze wszystkimi obiektami systemu decyzyjnego. Schemat programu dt_rules realizującego wyznaczanie omówionej tablicy rozróżnialności przedstawiono na listingu 3.8.

Listing 3.8. Realizacja programu wyznaczania tablicy rozróżnialności systemu decyzyjnego (dt_rules.m)

```
1   Dane wejściowe:
2           X[xn,xm] – Zbiór  danych dla i-tego obiektu
3           MacierzKlas [xn,1] – Wektor klas
4           ktory – indeks wiersza porównywanego
5
6   Dane wyjściowe:
7           X_DT – fragment tablicy rozróżnialności odpowiadający określonemu obiektowi;
8
9   Krok 1.
10           X1=repmat(X(ktory,:),n-1,1);
11  Krok 2.
12          X_DT=X1-X(2:n,:);
```

```
13    Krok 3.
14          X_DT=logical(sign(abs(X_DT)));
15    Krok4.
16          temp_MacierzKlas=MacierzKlas(2:n);
17          X_DT(temp_MacierzKlas(:)==MacierzKlas(ktory),:)=[];
18    Krok 5.
19          X_DT( sum(~X_DT,2)==m,:)=[];
20    Krok 6.
21          X_DT=unique(X_DT,'rows');
22    Krok 7.
23          X_DT=pochlanianie(X_DT);
```

Struktura listingu 3.8 jest analogiczna do programu wyznaczania tablicy rozróżnialności systemu decyzyjnego. W odróżnieniu od programu 3.7, argumentami wejściowymi funkcji dt_rules są zmienna X oraz MacierzKlas dla pełnego systemu decyzyjnego. Zmienna o nazwie ktory wskazuje na numer wiersza dla którego realizowane jest porównywanie.

3.4.8. Funkcja rozróżnialności systemu informacyjnego

Funkcję rozróżnialności wykorzystano do wyznaczenia wszystkich reduktów systemu informacyjnego. Ponieważ funkcja rozróżnialności jest funkcją boolowską, problem wyznaczania wszystkich reduktów zrealizowano w oparciu o metody wnioskowania boolowskiego. Z teorii funkcji boolowskich wynika, że funkcja rozróżnialności jest monotoniczna [Baz98]. Każdą funkcję monotoniczną można przedstawić w postaci alternatywy wszystkich implikantów pierwszych tej funkcji. Implikanty funkcji rozróżnialności zbudowane są z koniunkcji zmiennych boolowskich, odpowiadających atrybutom systemu informacyjnego. Oznacza to, że każdemu reduktowi systemu informacyjnego odpowiada dokładnie jeden taki implikant. Implikanty pierwsze otrzymano w wyniku minimalizacji funkcji rozróżnialności.

Do wyznaczenia funkcji rozróżnialności systemu informacyjnego opracowano program df_si. Schemat programu przedstawiono na listingu 3.9. W pierwszym kroku, dla danego systemu informacyjnego opisanego zmienną X, wyznaczono zbiory elementarne. Funkcja rozróżnialności wyznaczana jest w etapach odpowiadających budowaniu fragmentów tablicy rozróżnialności (Krok 4.1). Ostateczna postać funkcji rozróżnialności jest iloczynem logicznym funkcji uzyskanych w kolejnych iteracjach (Krok 4.2).

Listing 3.9. Realizacja programu wyznaczania funkcji rozróżnialności systemu informacyjnego (df_si.m)

```
1     Dane wejściowe:
2           X[xn,xm] – macierz reprezentująca system informacyjny
3
4     Dane wyjściowe:
5           redukty – zminimalizowana postać funkcji rozróżnialności - redukty;
6
7     Krok 1.
8           [ElementarySets,ObjectsLocation,NumOfElementarySets]= elementary_sets(X,1:size(X,2));
9     Krok 2.
10          [n,m]=size(ElementarySets);
11    Krok 3.
12          X_DF=true(2^m,1);
13    Krok 4.
14          Dla każdego i-tego zbioru elementarnego (bez ostatniego zbioru), wykonaj:
```

```
15    Krok 4.1.
16          tempX_DF=df_si_step(ElementarySets(i:n,:));
17    Krok 4.2.
18          X_DF=X_DF & tempX_DF;
19    Krok 5.
20          redukty=df_min(X_DF);
```

Fragmenty funkcji rozróżnialności są wyznaczane w programie df_si_step. Schemat programu przedstawiono na listingu 3.10.

Listing 3.10. Realizacja programu wyznaczania fragmentów funkcji rozróżnialności SI (df_si_step.m)

```
1     Dane wejściowe:
2           X[xn,xm] – dla której wyznaczana jest tablica rozróżnialności dotycząca i-tego zbioru elementarnego
3
4     Dane wyjściowe:
5           tempX_DF – zminimalizowana postać funkcji rozróżnialności - redukty;
6
7     Krok 1.
8           tempX_DT=dt_si2(X);
9     Krok 2.
10          if (size(tempX_DT,1)>0)
11              tempX_DF=df_step(tempX_DT);
12          else
13              tempX_DF=true(2^m,1);
14          end
15
```

W kroku pierwszym programu df_si_step wyznaczono fragment tablicy rozróżnialności. Tablica rozróżnialności jest reprezentacją funkcji rozróżnialności w koniunkcyjnej postaci normalnej (ang. conjuctive normal form, $f_{D,CNF}(TD)$). Do zamiany funkcji $f_{D,CNF}(TD)$ na alternatywną postać normalną (ang. disjunctive normal form, $f_{D,DNF}(TD)$) opracowano funkcję df_step, wywoływaną w linii 11. Schemat realizacji programu df_step przedstawiono na listingu 3.11.

Listing 3.11. Realizacja programu konwersji funkcji rozróżnialności do alternatywnej postaci normalnej (df_step.m)

```
1     Dane wejściowe:
2           tempX_DT [xn,xm] – wektor binarny
3
4     Dane wyjściowe:
5           tempX_DF – i-ta funkcja rozróżnialności
6
7     Krok 1.
8           X_TT=tt(xm);
9            [n1,m1]=size(X_TT);
10    Krok 2.
11          tempX_sum=sum(tempX_DT,2);
12    Krok 3.
13          tempX_nDT=~tempX_DT;
14    Krok 4.
15          tempX_DF=true(n1,1);
16    Krok 5.
17          Dla każdego i-tego wiersza tablicy rozróżnialności wykonaj
18    Krok 5.1
19          TEMP=repmat(tempX_nDT(i,:),n1,1);
20    Krok 5.2
```

```
21          itempX_DF=sum(~(or(TEMP,X_TT)),2);
22    Krok 5.3
23          itempX_DF=~(itempX_DF>=tempX_sum(i));
24    Krok 5.4
25          tempX_DF=tempX_DF & itempX_DF;
```

Do konwersji zastosowano tablicę prawdy X_TT. Tablica prawdy generowana jest przy użyciu funkcji tt opisanej w rozdziale 3.4.11. Tablica prawdy zawiera wszystkie możliwe kombinacje atrybutów.

Wyznaczenie funkcji w postaci $f_{D,DNF}(SD)$ sprowadza się do porównania każdego wiersza tablicy rozróżnialności z każdym wierszem tablicy prawdy. Przy porównywaniu wykorzystano następujące własności funkcji rozróżnialności:

1. Jeżeli obiekty nie różnią się między sobą, to funkcji przypisuje się wartość 1;
2. Jeżeli obiekty różnią się, to funkcji przypisuje się wartość 0;
3. Jeżeli atrybut rozróżnia obiekty, to przypisuje się mu wartość 0;
4. Jeżeli dwa obiekty są rozróżniane przez zbiór atrybutów $P_1 \subset P_2$, to oznacza, że będą rozróżniane także przez zbiór atrybutów P_2.

W kroku drugim programu 3.11, dla każdego wiersza tablicy rozróżnialności, wyznaczana jest liczba atrybutów rozróżniających.

Do porównania wierszy tablicy rozróżnialności z wierszami tablicy prawdy wymagana jest negacja tablicy rozróżnialności (Krok 3). Wektor wyjścia tablicy prawdy jest inicjalizowany wartościami 1 (Krok 4). Porównanie przeprowadza się dla każdego wiersza tablicy rozróżnialności (Krok 5). Zastosowano tutaj funkcję logiczną or. Jeżeli i-ty wiersz tablicy rozróżnialności zawiera się w wierszu tablicy prawdy, to liczba atrybutów, które przyjmują wartość 0 jest równa lub większa od liczby atrybutów rozróżniających i-tego wiersza. Aktualna postać funkcji wyjścia jest aktualizowana z zastosowaniem funkcji logicznej and (Krok 5.4).

Po wyznaczeniu wszystkich fragmentów funkcji rozróżnialności należy przeprowadzić minimalizację (Listing 3.9, Krok 5). W tym celu opracowano program df_min przedstawiony na listingu 3.12.

Listing 3.12. Realizacja programu wyznaczania minimalizacji funkcji rozróżnialności (df_min.m)

```
1     Dane wejściowe:
2           X_DF[n,1] – wektor binarny
3     Dane wyjściowe:
4           minX_DF – zminimalizowana postać funkcji rozróżnialności;
5
6     Krok 1.
7           trueDF=(find(X_DF)-1)';
8     Krok 2.
9           trueDF =de2bi(trueDF);
10    Krok 3.
11          minX_DF=pochlanianie(trueDF);
```

W realizacji programu df_min do minimalizacji funkcji rozróżnialności wykorzystano prawo absorpcji. Argumentem wejściowym jest postać funkcji wyznaczona w programie df_si. Na jej podstawie odczytywana jest alternatywna macierzowa postać normalna funkcji rozróżnialności (Krok 2). Wiersze macierzy odpowiadają składnikom funkcji, przy czym wartość 1 w komórce wskazuje na wystąpienie danego atrybutu w składniku. W ostatnim kroku realizowana jest operacja pochłaniania. Szczegóły

programu pochłanianie zamieszczono w rozdziale 3.4.12. Argumentem wyjściowym programu jest macierz binarna, będąca reprezentacją funkcji rozróżnialności w alternatywnej postaci normalnej.

3.4.9. Funkcja rozróżnialności systemu decyzyjnego

Funkcję rozróżnialności systemu decyzyjnego wykorzystano do wyznaczania reduktów względnych systemu decyzyjnego. Opracowany w tym celu program df_sd przedstawiono na listingu 3.13. Program posiada dwa argumenty wejściowe. Pierwszy odpowiada części warunkowej systemu decyzyjnego, a drugi części decyzyjnej.

Listing 3.13. Realizacja programu wyznaczania funkcji rozróżnialności systemu decyzyjnego (df_sd.m)

```
1   Dane wejściowe:
2       X[xn,xm] – macierz reprezentująca system informacyjny
3       KX – macierz klas
4
5   Dane wyjściowe:
6       redukty – zminimalizowana postać funkcji rozróżnialności - redukty;
7
8   Krok 1.
9       X_DF=true(2^xm,1);
10  Krok 2.
11      Dla każdego i-tego zbioru (bez ostatniego zbioru), wykonaj:
12  Krok 2.1.
13      tempX_DF=df_si_step(X(i:xn,:));
14  Krok 2.2.
15      X_DF=X_DF & tempX_DF;
16  Krok 3.
17      redukty=df_min(X_DF);
```

Podobnie jak dla systemu informacyjnego funkcja rozróżnialności systemu decyzyjnego wyznaczana jest krokowo. Zastosowano tutaj program df_sd_step (listing 4.14). Tablica rozróżnialności wyznaczana jest względem tych obiektów, które należą do różnych klas decyzyjnych. Do wyznaczenia tablicy rozróżnialności opracowano funkcję dt_sd omówioną w rozdziale 3.4.6.

Listing 3.14. Realizacja programu wyznaczania fragmentów funkcji rozróżnialności SD (df_sd_step.m)

```
1   Dane wejściowe:
2       X[xn,xm] – dla której wyznaczana jest tablica rozróżnialności dotycząca i-tego zbioru elementarnego
3       KX – macierz klas
4
5   Dane wyjściowe:
6       tempX_DF – zminimalizowana postać funkcji rozróżnialności - redukty;
7
8   Krok 1.
9       tempX_DT=dt_sd(X,KX);
10  Krok 2.
11      if (size(tempX_DT,1)>0)
12          tempX_DF=df_step(tempX_DT);
13      else
14          tempX_DF=true(2^xm,1);
15      end
```

3.4.10. Funkcja rozróżnialności w zadaniu klasyfikacji

Funkcję rozróżnialności można wykorzystać do wyznaczenia reguł decyzyjnych (listing 3.15). W takim przypadku funkcja rozróżnialności wyznaczana jest oddzielnie dla każdego obiektu. Z tego powodu obiekt, w każdej iteracji, musi być porównywany ze wszystkimi pozostałymi (listing 3.16).

Listing 3.15. Realizacja programu wyznaczania funkcji rozróżnialności dla reguł decyzyjnych (df_rules.m)

```
Dane wejściowe:
        X[xn,xm] – macierz reprezentująca system informacyjny
        KX– macierz klas

Dane wyjściowe:
        reguly – zminimalizowana postać funkcji rozróżnialności - redukty;

Krok 1.
        Dla każdego i-tego zbioru wykonaj:
Krok 1.1
        reguly_temp=df_rules_step(X, KX,i);
Krok 1.2.
        reguly=[reguly ; reguly_temp];
Krok 2.
        Usuń powtarzające się reguły
```

Listing 3.16. Realizacja programu wyznaczania fragmentów funkcji rozróżnialności reguł decyzyjnych (df_rules_step.m)

```
Dane wejściowe:
        X[xn,xm] – dla której wyznaczana jest tablica rozróżnialności dotycząca i-tego zbioru elementarnego
        KX– macierz klas
        ktory

Dane wyjściowe:
        reguly – zminimalizowana postać funkcji rozróżnialności - redukty;

Krok 1.
        tempX_DT=dt_rules(X, KX, ktory)
Krok 2.
        tempX_DF= df_step(tempX_DT);
Krok 3.
        X_DF_MIN=df_min(tempX_DF);
Krok 4.
        X_DF_MIN(X_DF_MIN==0)=NaN;
Krok 5.
        wiersz_mac=repmat(X(ktory,:),size(X_DF_MIN,1),1);
         wiersz_klas=repmat(KX(ktory),size(X_DF_MIN,1),1);
         reguly=[wiersz_mac.*X_DF_MIN wiersz_klas];
```

3.4.11. Tablica prawdy

Do wyznaczenia tablicy prawdy opracowano program tt. Schemat programu przedstawiono na listingu 3.17. Argumentem wejściowym funkcji jest liczba atrybutów. Do wyznaczenia tablicy prawdy wykorzystano funkcję de2bi, która konwertuje liczbę

w postaci dziesiętnej na liczbę w postaci binarnej zapisaną jako wektor. Najbardziej znaczący bit liczby binarnej znajduje się z prawej strony.

Listing 3.17. Realizacja programu wyznaczania tablicy prawdy (tt.m)

```
1   Dane wejściowe:
2           xm – liczba atrybutów;
3
4   Dane wyjściowe:
5           X_TT – macierz tablicy prawdy;
6
7   X_TT=logical(de2bi(0:1:( (2^ xm) -1),'right-msb'));
8
```

Przedstawiony algorytm pozwala na wygenerowanie tablicy prawdy w bardzo krótkim czasie. Przy 20 atrybutach wygenerowanie macierzy zajmuje około 2,6s.

Funkcja charakteryzuje się jednak wysoką pamięciową złożonością obliczeniową. Liczba wierszy tabeli prawdy rośnie wykładniczo wraz ze wzrostem liczby atrybutów. Tablica prawdy, przy dwudziestu atrybutach, posiada ponad milion wierszy. Liczba atrybutów jakie można analizować przy zastosowaniu modułu RS zależna jest od następujących elementów:
- parametry techniczne komputera obliczeniowego;
- przydział zasobów pamięci dla środowiska MATLAB;
- maksymalny rozmiar zmiennej środowiska MATLAB.

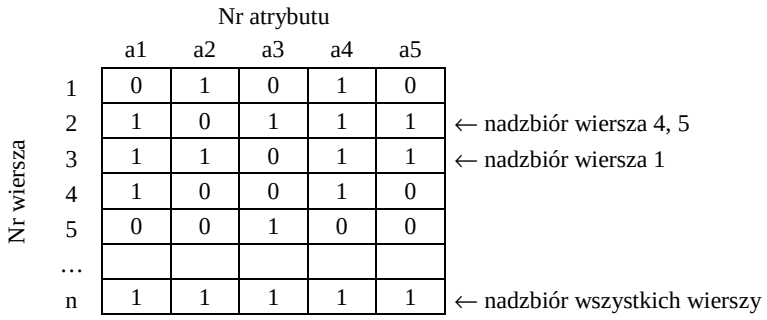
3.4.12. Prawo absorpcji

Prawo absorpcji wykorzystano do uproszczenia funkcji rozróżnialności. Schemat algorytmu realizującego to zadanie przedstawiono na listingu 3.18. Argumentem wejściowym funkcji jest macierz binarna. Realizacja funkcji pochłaniania polega na usunięciu wierszy, będących nadzbiorami innych wierszy (rys. 3.3).

Listing 3.18. Realizacja programu prawa pochłaniania (pochlanianie.m)

```
1   Dane wejściowe:
2           macierz_do_redukcji [xd,yd] – Macierz wejściowa ;
3
4   Dane wyjściowe:
5           macierz_zredukowana – macierz wyjściowa;
6
7   Krok 1.
8           Dopóki  macierz_do_redukcji zawiera wiersze
9   Krok 1.1.
10          [min_w,min_idx]=min(sum(macierz_do_redukcji,2));
11  Krok 1.2.
12          wiersz_min=macierz_do_redukcji(min_idx,:);
13  Krok 1.3.
14          macierz_zredukowana=[macierz_zredukowana; wiersz_min];
15  Krok 1.4.
16          macierz_min=repmat(wiersz_min, size(macierz_do_redukcji,1),1);
17  Krok 1.5.
18          temp_mac=and(macierz_min,macierz_do_redukcji);
19  Krok 1.6.
20          macierz_do_redukcji(sum(temp_mac,2)==min_w,:)=[];
```

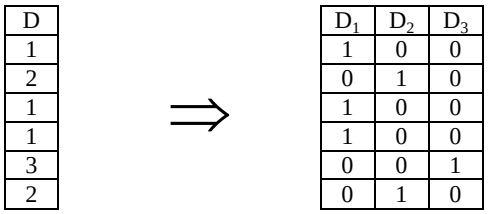
W algorytmie przedstawionym na listingu 3.18, wiersze macierzy wejściowej są sortowane w taki sposób, aby pierwszy wiersz zawierał informację o najmniejszej liczbie atrybutów (Krok 1.1). Wiersz o najmniejszej liczbie atrybutów jest zapamiętywany jako wiersz zmiennej macierz_zredukowana (Krok 1.3). Następnie tworzona jest nowa macierz - macierz_min, która zawiera kopię wiersza minimalnego, w liczbie odpowiadającej aktualnemu rozmiarowi zmiennej macierz_do_redukcji. Poszukiwanie nadzbiorów wierszy realizowane jest przy użyciu iloczynu logicznego (Krok 1.5). Jeżeli atrybut wiersza minimalnego występuje w nadzbiorze, to wynikiem operacji logicznej będzie wartość 1. Gdy liczba komórek o wartości 1, odpowiada liczbie atrybutów zmiennej wiersz_min to wiersz macierzy temp_mac jest nadzbiorem. Wiersze, dla których ten warunek jest spełniony, mogą zostać usunięte z macierzy macierz_do_redukcji (Krok 1.6). Poszukiwanie nadzbiorów jest realizowane do momentu, aż macierz_do_redukcji nie będzie posiadała żadnych wierszy.



Rys. 3.3. Przykłady nadzbiorów wierszy w tablicy rozróżnialności

3.4.13. Konwersja klas (Classtobin)

Opracowana funkcja classtobin służy do konwersji klas zapisanych w postaci wektora na macierz o wartościach 0 i 1. Interpretację graficzną realizacji funkcji classtobin przedstawiono na rysunku 3.4. Liczba kolumn macierzy odpowiada liczbie możliwych klas. Wartość „1” w i-tym wierszu i j-tej kolumnie macierzy wskazuje na przypisanie i-tego obiektu do j-tej klasy.



Rys. 3.4. Przykład zastosowania funkcji classtobin

Schemat algorytmu realizującego funkcję przedstawiono na listingu 3.19. Wejściowy wektor klas porównywany jest z każdym wektorem klasy decyzyjnej. Operacja ta realizowana jest jako różnica dwóch macierzy (Krok 1):

- temp - zawiera kopię wektora klas; Liczba kolumn odpowiada liczbie klas decyzyjnych;
- ObjectsLocationBin - zawiera klasy decyzyjne; Wartości w kolumnach odpowiadają numerowi klasy decyzyjnej.

W wyniku różnicy macierzy (Krok 2), zmienna ObjectsLocationBin, przyjmuje:

- wartości równe 0, dla obiektów należących do danej klasy decyzyjnej;

- wartości różne od zera, gdy przynależność obiektu do klasy jest błędna.

W celu konwersji macierzy na macierz zero-jedynkową, w której wartość 1 odpowiada przynależności obiektu do klasy, zastosowano negację funkcji sign (Krok 3).

Listing 3.19. Realizacja programu konwersji klas (classtobin.m)

```
1     Dane wejściowe:
2           ObjectsLocation;
3           NumOfGroups – Liczba klas decyzyjnych
4
5     Dane wyjściowe:
6           ObjectsLocationBin – macierz klas w postaci zero-jedynkowej;
7
8     Krok 1.
9           temp=repmat(ObjectsLocation(1:xn,:),1,NumOfGroups);
11          ObjectsLocationBin =repmat(=[1:1:NumOfGroups],size(temp,1),1);
12    Krok2.
13          ObjectsLocationBin = ObjectsLocationBin-temp;
14    Krok3.
15          ObjectsLocationBin =~sign(abs(ObjectsLocationBin));
```

3.5. Moduł RSAm

Moduł RSAm umożliwia przeprowadzenie procesu uczenia nadzorowanego. Proces uczenia zrealizowano w sposób blokowy umożliwiający niezależne uruchamianie następujących zadań: redukcja zbioru atrybutów, dyskretyzacja wartości atrybutów, klasyfikacja obiektów. Zautomatyzowanie procesu obliczeń pozwala na poszukiwanie optymalnych zbiorów atrybutów oraz na dobór odpowieniego klasyfikatora (rys. 4.7).

Funkcją startową modułu RSAm jest program rsamrun.m (Listing A.1). Do uruchomienia programu rsamrun.m wymagane jest zdefiniowanie następujących zmiennych:

- option_f_type – określa wybór metod redukcji cech.
- option_discretization – określa metodę dyskretyzacji dla zadania redukcji zbioru atrybutów oraz zadania klasyfikacji. Typy dyskretyzacji oraz stosowane oznaczenia zamieszczono w rozdziale 3.5.1.
- option_classification_types – określa wybór metod klasyfikacji. Możliwe jest uruchomienie jedej funkcji rsamrun dla kilku metod klasyfikacji. Konfiguracja taka ma znaczenie w przypadku zadań, w których czas procesu dyskretyzacji lub redukcji jest złożony obliczeniowo.
- DataSetsNamesList – określa listę nazw zbioru uczącego i zbiorów walidacyjnych.
- DataSetNo – określa numer zestawu zbioru testowego, odpowiadający zbiorowi testowemu DataSetsNamesList, dla którego będą przeprowadzane obliczenia. Numer zbioru jest widoczny w strukturach wyjściowych dyskretyzacji, redukcji i klasyfikacji. Przyjęto oznaczenie DSx, gdzie x oznacza wartość zmiennej DataSetNo.
- AttrList – określa listę atrybutów warunkowych. Lista atrybutów ma postać binarną, gdzie wartość 1 wskazuje na wybranie danego atrybutu do analiz, natomiast 0 wskazuje na jego pominięcie. Rozmiar listy atrybutów powinien odpowiadać liczbie atrybutów w zbiorze uczącym, przekazywanym przez zmienną DBSource. Lista atrybutów nie może uwzględniać atrybutu decyzyjnego.
- AttrSetNo - określa numer zbioru atrybutów, odpowiadający zmiennej AttrList, dla którego przeprowadzane są obliczenia. Numer widoczny jest w strukturach wyjściowych

dyskretyzacji, redukcji i klasyfikacji. Przyjęto oznaczenie ASx, gdzie x oznacza wartość zmiennej AttrSetNo.

- ClassAttr - określa numer atrybutu odpowiadający klasie decyzyjnej. Wartość powinna odpowiadać numerowi atrybutu w zbiorze uczącym, przekazywanym przez zmienną DBSource.
- DBTyp – określa typ źródła danych wykorzystywanego w obliczeniach. Zmienna DBTyp przyjmuje jedną z dwóch wartości:
    - 0 – zbiory danych, których nazwy zdefiniowano w zmiennej DataSetsNamesList, przekazywane są do funkcji rsamrun z przestrzeni roboczej środowiska MATLAB.
    - 1 – zbiory danych, których nazwy zdefiniowano w zmiennej DataSetsNamesList, przekazywane są do funkcji rsamrun poprzez bazę danych.
- DBSource – określa źródło danych wykorzystywane w obliczeniach. W przypadku, gdy zmienna DBTyp przyjmuje wartość 0, to zmienna DBSource ma postać struktury. Nazwy pól w strukturze DBSource odpowiadają nazwom zbiorów w zmiennej DataSetsNamesList i przechowują macierze danych źródłowych. W przypadku, gdy zmienna DBTyp przyjmuje wartość 1, to zmienna DBSource przechowuje informację o nazwie bazy danych.

Wyniki zwracane przez funkcję rsamrun są tablicami strukturalnymi. Każda z tablic odpowiada jednemu z trzech elementów programu: dyskretyzacji, klasyfikacji, redukcji. Pierwsza tablica strukturalna to rsam_discretization. Zawiera informację o granicach dyskretyzacji wykorzystanych w analizie. Druga tablica strukturalna - rsam_reduction, zawiera zredukowane zbiory atrybutów. Najbardziej rozbudowaną postać ma trzecia tablica strukturalna rsam_classification, która zawiera wyniki klasyfikacji. Szczegółowe informacje na temat budowy wymienionych zmiennych zamieszczono w rozdziale 3.5.1, 3.5.2. oraz 3.5.3.

Analiza wyników zwracanych przez funkcję rsamrun umożliwia modyfikację parametrów optymalizacji i ponowne uruchomienie programu rsamrun. W tym celu przygotowano przykładowe kody źródłowe zawierające zbiór instrukcji pozwalający na wielokrotne uruchamianie programu rsamrun:

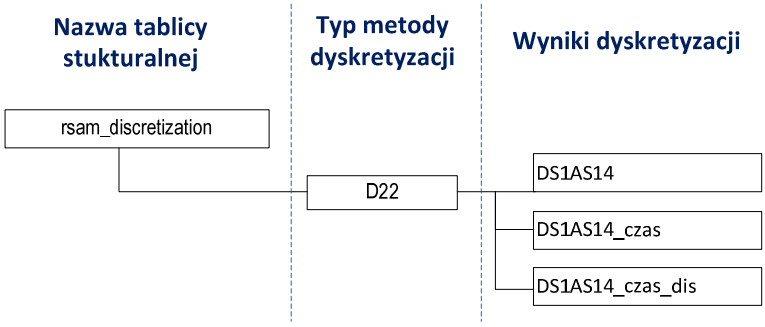
- rsaminit.m – zawiera deklarację zmiennych potrzebnych do uruchomienia programu rsamprerun (listing A.2);
- rsamprerun – realizuje jednostanowiskowe, iteracyjne uruchamianie programu rsamrun dla parametrów zdefiniowanych w pliku rsaminit (listing A.3);
- dist_rsamprerun – realizuje rozproszone uruchamianie programu rsamrun dla parametrów zdefiniowanych w pliku rsaminit (listing A.4).

3.5.1. Dyskretyzacja wartości atrybutów

W module RSAm zaimplementowano następujące metody dyskretyzacji:
- dyskretyzacja równej szerokości (EWD),
- dyskretyzacja CAIM,
- dyskretyzacja CACC.

Algorytmy dyskretyzacji zaimplementowano w postaci funkcji. Pozwala to na ich wykorzystanie, niezależnie od uruchomienia programu rsamrun.

W przypadku dyskretyzacji z zastosowaniem modułu RSAm, należy zdefiniować zmienną option_discretization. Do oznaczenia wybranej metody dyskretyzacji zastosowano notację przedstawioną w tabeli 3.1. Funkcja rsamrun zwraca tablicę strukturalną o nazwie rsam_discretization, przechowującą granice dyskretyzacji oraz czas realizacji obliczeń. Strukturę tablicy przedstawiono na rys. 3.5.



Rys. 3.5. Zagnieżdżona tablica strukturalna dla zadania dyskretyzacji

Granice dyskretyzacji przechowywane są w zmiennej DSxASy, która odpowiada zbiorowi testowemu nr x, analizowanego względem zbioru atrybutów nr y. Czas realizacji dyskretyzacji zapisywany jest w zmiennej DSxASy_czas. W przypadku gdy zastosowano dyskretyzację rozproszoną, to w zmiennej DSxASy_czas_dis zamieszczane są czasy obliczeń każdego z węzłów.

Granice dyskretyzacji wyznaczane są na podstawie próby uczącej. Przy walidacji modelu, wartości dyskretne zbioru walidacyjnego są wyznaczane na podstawie wcześniej obliczonych granic.

Tabela 3.1. Oznaczenia metod dyskretyzacji

Oznaczenie stosowane w definiowaniu zmiennych wejściowych	Oznaczenie stosowane w nazewnictwie zmiennych wyjściowych	Opis
0	0	Brak dyskretyzacji
1	D1W5	Dyskretyzacja EWD o 5 przedziałach
1	D1W10	Dyskretyzacja EWD o 10 przedziałach
1	D1W20	Dyskretyzacja EWD o 20 przedziałach
1	D1W30	Dyskretyzacja EWD o 30 przedziałach
21	D21	CAIM - wersja jednostanowiskowa
22	D22	CAIM - wersja rozproszona
31	D31	CACC - wersja jednostanowiskowa
32	D32	CACC - wersja rozproszona

Dyskretyzacja EWD

Do realizacji dyskretyzacji równej szerokości opracowano program Dyskretyzacja_EWD. Argumentami wejściowymi programu są: macierz o wartościach rzeczywistych oraz liczba przedziałów dyskretyzacji. Liczba przedziałów dyskretyzacji jest jednym z parametrów optymalizacji przy poszukiwaniu najlepszego klasyfikatora. Schemat działania programu zamieszczono na listingu 3.20.

Dyskretyzacja EWD jest przykładem dyskretyzacji globalnej. Granice atrybutów są wyznaczane niezależnie od siebie. Dla każdego atrybutu poszukiwana jest wartość minimalna oraz wartość maksymalna. W kolejnym kroku wyznaczana jest szerokość przedziału dyskretyzacji (Krok 2.2) oraz pierwsza granica (Krok 2.4). Pozostałe granice wyznaczane są w sposób iteracyjny (Krok 2.5).

Listing 3.20. Realizacja programu dyskretyzacji EWD (Dyskretyzacja_EWD.m)

```
Dane wejściowe:
        MacierzCiagla,IloscPrzedzialow

Dane wyjściowe:
        MacierzDyskretna,GraniceDyskretyzacji

Krok 1.
        MacierzDyskretna=zeros(n,F);
        GraniceDyskretyzacji=zeros(IloscPrzedzialow-1,F);
Krok 2.
        Dla każdego p atrybutu wykonaj:
Krok 2.1.
        v_min=min(MacierzCiagla(:,p));
        v_max=max(MacierzCiagla(:,p));
Krok 2.2
        v_dl=(v_max-v_min)/IloscPrzedzialow;
Krok 2.3
        v_gr=v_min+v_dl;
Krok 2.3
        Dla każdego i-tego przedziału wykonaj
Krok 2.3.1
        GraniceDyskretyzacji(i,p)=v_gr;
Krok 2.3.2
        v_gr=v_gr+v_dl;
Krok 3.
        MacierzDyskretna(:,p) = dyskretyzacja_podstawianie(MacierzCiagla(:,p),GraniceDyskretyzacji(:,p) );

```

Dyskretyzacja CAIM

Do realizacji dyskretyzacji metodą CAIM zaimplementowano program Discretization_CAIM. Schemat algorytmu dyskretyzacji CAIM przedstawiono w pracy [KuCi04]. Dyskretyzacja CAIM, zaimplementowana w module RSAm, została opracowana na podstawie źródła [wCAIM09]. W programie wprowadzono modyfikację polegającą na zamianie pętli sterujących i instrukcji warunkowych na obliczenia przeprowadzane bezpośrednio na macierzach. Wprowadzone zmiany wynikają z wysokiej złożoności obliczeniowej wyznaczania granic przedziałów. Ponieważ możliwe granice generowane są jako wartości pośrednie pomiędzy poszczególnymi sąsiednimi wartościami rzeczywistymi, to im większa jest liczba unikatowych wartości w dziedzinie atrybutu, tym więcej jest granic inicjalizacyjnych. Problem ten jest szczególnie widoczny przy macierzach o dużej liczbie wierszy.

Algorytm budowania macierzy kwantyzacji za pomocą pętli sterujących przedstawiono na listingu 3.21. Wartość C wskazuje na liczbę klas. Klasy zapisane są jako ostatnie kolumny w macierzy OriginalData. Każda kolumna odpowiada jednej klasie. Jeżeli obiekt należy do danej klasy decyzyjnej, to element macierzy przyjmuje wartość 1. W przeciwnym razie elementem macierzy jest wartość 0. Taką postać kolumn można uzyskać przy użyciu funkcji classtobin (rozdział 3.4.13).

Listing 3.21. Kod źródłowy programu wyznaczania macierzy kwantyzacji z zastosowaniem pętli sterujących (DiscretWithInterval.m)

```
1   Dane wejściowe
2           OriginalData  - zbiór danych
3           DiscretInterval – granice przedziałów dyskretyzacji
4           Column - dyskretyzowany atrybut
5           C - liczba klas
6
7   Dane wyjściowe
8           DiscretData – zdyskretyzowany wektor danych
9           QuantaMatrix – macierz kwantyzacji
10
11  M = size( OriginalData,1 );
12  k = length( DiscretInterval );
13  F = size( OriginalData,2 ) - C;
14  DiscretData = zeros( M,1 );
15  %Discrete the continuous data upon OriginalData
16  for p = 1:M
17     for t = 1:k
18        if OriginalData( p,Column ) <= DiscretInterval( t )
19           DiscretData( p ) = t-1;
20           break;
21        elseif OriginalData( p,Column ) > DiscretInterval( k )
22           DiscretData( p ) = k;
23        end
24     end
25  end
26
27  CState = C;
28  FState = length( DiscretInterval ) + 1;
29  QuantaMatrix = zeros( CState,FState );
30  for p = 1:M
31     for q = 1:C
32        if OriginalData( p,F+q ) == 1
33           Row = q;
34           Column = DiscretData( p )+1;
35           QuantaMatrix( Row,Column ) = QuantaMatrix( Row,Column ) + 1;
36        end
37     end
38  end
39
```

Wersję zmodyfikowaną, wykorzystującą operacje macierzowe, zamieszczono na listingu 3.22. Wprowadzona modyfikacja polega na wykorzystaniu funkcji histc do zliczania obiektów występujących w każdym przedziale dyskretyzacji (linia 17). Funkcję wykorzystano także do wyliczania wierszy macierzy kwantyzacji (linia 21). Do wyliczania pojedynczego wiersza macierzy kwantyzacji, odpowiadającego jednej klasie decyzyjnej, zastosowano pętlę for.

Listing 3.22. Realizacja programu wyznaczania macierzy kwantyzacji z zastosowaniem operacji macierzowych (DiscretWithInterval.m)

```
1   Dane wejściowe
2           OriginalData  - zbiór danych
3           DiscretInterval – granice przedziałów dyskretyzacji
4           Column - dyskretyzowany atrybut
5           C - liczba klas
```

```
6
7       Dane wyjściowe
8               DiscretData – zdyskretyzowany wektor danych
9               QuantaMatrix – macierz kwantyzacji
11
12      M = size( OriginalData,1 );
13      k = length(DiscretInterval);
14      F = size( OriginalData,2 ) - C;
15      DiscretData = zeros( M,1 );
16
17      granice=[-Inf DiscretInterval Inf];
18      [quanta_all,DiscretData]=histc(OriginalData( :,Feature ) ,granice);
19      FState = length( DiscretInterval ) + 1;
20      QuantaMatrix1 = zeros( FState+1, C);
21      for q = 1:S
22      [QuantaMatrix1(:,q),discdatatest1]=histc(OriginalData(OriginalData( :,F+q ) == 1, Column) ,granice);
23      end
24      QuantaMatrix1(k+2,:)=[];
25      QuantaMatrix=QuantaMatrix1';
```

Zmodyfikowano także programy do wyznaczania współczynnika *caim*. Algorytm oryginalny przedstawiono na listingu 3.23, natomiast jego modyfikację na listingu 3.24.

Listing 3.23. Realizacja programu wyznaczania współczynnika CAIM z zastosowaniem pętli sterujących (.m)

```
1       Dane wejściowe
2               OriginalData  - zbiór danych
3               DiscretInterval – granice przedziałów dyskretyzacji
4               Column - dyskretyzowany atrybut
5               C - liczba klas
6
7       Dane wyjściowe
8
9       k = length(DiscretInterval);
11      [ DiscretData,QuantaMatrix ] = DiscretWithInterval(OriginalData,C,Feature,DiscretInterval );
12      SumQuantaMatrix = sum( QuantaMatrix,1 );
13      CAIMValue = 0 ;
14
15      for p = 1:k
16        if max( QuantaMatrix(:,p) ) > 0
17          CAIMValue = CAIMValue + ( max( QuantaMatrix(:,p) ) )^2/SumQuantaMatrix(p) ;
18        end
19      end
20       CAIMValue = CAIMValue/k ;
```

Listing 3.24. Realizacja programu wyznaczania współczynnika CAIM z zastosowaniem operacji macierzowych (.m)

```
1       Dane wejściowe
2               OriginalData  - zbiór danych
3               DiscretInterval – granice przedziałów dyskretyzacji
4               Feature - dyskretyzowany atrybut
5               S- liczba klas
6
7       Dane wyjściowe
8
9       k = length( DiscretInterval )+1;
```

```
10    [ DiscretData,QuantaMatrix ] = DiscretWithInterval( OriginalData,S,Feature,DiscretInterval );
11    SumQuantaMatrix = sum( QuantaMatrix,1 );
12    MaxQuantaMatrix=max(QuantaMatrix);
13    CAIMi=MaxQuantaMatrix./SumQuantaMatrix;
14    CAIMi=CAIMi.*MaxQuantaMatrix;
15    CAIMValue=nansum(CAIMi)/k;
```

Do wyznaczenia współczynnika *caim* wykorzystano macierz kwantyzacji. Wartość współczynnika *caim* obliczono zgodnie z zależnością 2.19. Wartość q_{+k} odpowiada sumie obiektów w każdym przedziale dyskretyzacji (linia 3). Wartość max_k odpowiada zmiennej MaxQuantaMatrix - maksymalnej liczbie obiektów w każdym przedziale dyskretyzacji (linia 13). Ostateczną postać sumy, występującą w liczniku zależności 2.19, zrealizowano jako sumę wszystkich elementów wektora CAIMi (linia 15).

Dyskretyzacja CACC

Dyskretyzację metodą CACC zrealizowano w programie Discretization_CACC. Schemat algorytmu dyskretyzacji CACC przedstawiono w pracy [TsLe08]. Dyskretyzacja CACC, zaimplementowana w module RSAm, została opracowana na podstawie źródła [wCACC09]. Analogicznie jak w dyskretyzacji metodą CAIM, w programie wprowadzono modyfikację polegającą na zamianie pętli sterujących i instrukcji warunkowych na obliczenia wektorowe. Zmianę wprowadzono przy budowaniu macierzy kwantyzacji oraz przy wyznaczaniu współczynnika *cacc*. Algorytm budowania macierzy kwantyzacji jest taki sam jak zaprezentowany w metodzie CAIM. Oryginalną wersję programu do obliczania współczynnika *cacc*, wykorzystującą pętlę sterujące przedstawiono na listingu 3.25. Wersję zmodyfikowaną, wykorzystującą operacje macierzowe, zamieszczono na listingu 3.26.

Listing 3.25. Realizacja programu wyznaczania współczynnika CACC z zastosowaniem pętli sterujacych (.m)

```
1     Dane wejściowe
2             OriginalData  - zbiór danych
3             DiscretInterval – granice przedziałów dyskretyzacji
4             Column - dyskretyzowany atrybut
5             C - liczba klas
6
7     Dane wyjściowe
8             CACCValue
9
10    M = size( OriginalData,1 );
11    k = length( DiscretInterval );
12    [ DiscretData,QuantaMatrix ] = DiscretWithInterval( OriginalData,C,Feature,DiscretInterval );
13    RowQuantaMatrix = sum( QuantaMatrix,2 );
14    ColumnQuantaMatrix = sum( QuantaMatrix,1 );
15    CACCValue = 0 ;
16
17    for p = 1:C
18       for q = 1:k
19          if RowQuantaMatrix( p ) > 0 && ColumnQuantaMatrix( q ) > 0
20             CACCValue = CACCValue + ( QuantaMatrix( p,q ) )^2/( RowQuantaMatrix( p )*ColumnQuantaMatrix( q )) ;
21          end
22       end
23    end
24    CACCValue = M*( CACCValue-1 )/log2(k+1) ;
25
```

Listing 3.26. Realizacja programu wyznaczania współczynnika CACC z zastosowaniem pętli sterujacych (.m)

```
1    Dane wejściowe
2            OriginalData  - zbiór danych
3            DiscretInterval – granice przedziałów dyskretyzacji
4            Feature - dyskretyzowany atrybut
5            C - liczba klas
6
7    Dane wyjściowe
8            CACCValue
9
11   M = size( OriginalData,1 );
12   k = length( DiscretInterval )+1;
13   [ DiscretData,QuantaMatrix ] = DiscretWithInterval( OriginalData,C,Feature,DiscretInterval );
14   RowQuantaMatrix = sum( QuantaMatrix,2 );
15   ColumnQuantaMatrix = sum( QuantaMatrix,1 );
16   CACCValue = 0 ;
17   y2=0;
18   for p = 1:C
19   y1=((QuantaMatrix( p,:).^2)./ColumnQuantaMatrix)/RowQuantaMatrix(p);
20   y2=y2+y1;
21   end
22   y2=sum(y2);
23   y2 = M*( y2-1 )/log(k) ;
24   CACCValue=sqrt(y2/(y2+M));
```

Współczynnik *cacc* wyznaczano na podstawie zależności 2.20 z uwzględnieniem 2.21. Do wyznaczania współczynnika *cacc* wykorzystano macierz kwantyzacji. Schemat wyznaczania macierzy jest analogiczny jak dla zadania wyznaczania współczynnika *caim* (listing 3.21, 3.22). Do wyliczenia współczynnika *cacc* wykorzystano zarówno liczbę obiektów w każdym przedziale dyskretyzacji q_{i+} (ColumnQuantaMatrix), jak i liczbę obiektów w każdej klasie q_{+k} (RowQuantaMatrix). Ostateczną postać współczynnika *cacc* zapisanego w zmiennej CACCValue, wyznaczono na podstawie zależności 2.20.

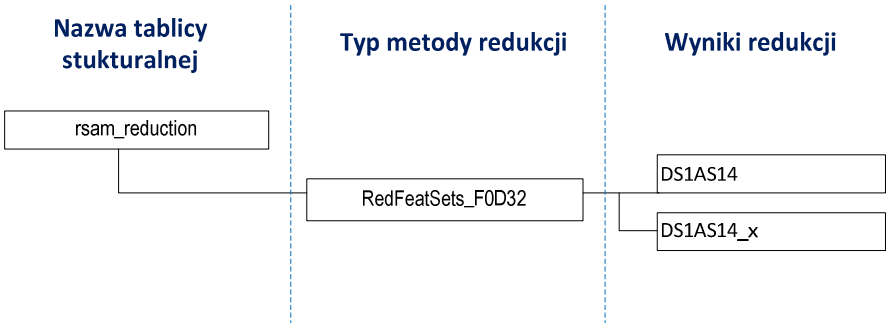
3.5.2. Redukcja przestrzeni atrybutów

W module RSAm zaimplementowano metody redukcji przestrzeni atrybutów przedstawione w rozdziale 2.4. Oznaczenia metod redukcji, wymagane przy definicji zmiennej option_f_types, zamieszczono w tabeli 3.2.

Tabela 3.2. Oznaczenia metod redukcji

Nr metody	Oznaczenie	Opis
0	F0	Brak redukcji
1	F1	Współczynnik korelacji - corrAA
21	F22	Zbiory przybliżone – wersja jednostanowiskowa
22	F22	Zbiory przybliżone – wersja rozproszona
6	F6	Selekcja sekwencyjna
3	F3	Analiza głównych składowych
8	F8	Współczynnik korelacji - corrAC

Zmienne będące wynikiem zadania redukcji przestrzeni atrybutów są zapisywane w tablicy strukturalnej rsam_reduction, której postać przedstawiono na rysunku 3.6.



Rys. 3.6. Zagnieżdżona tablica strukturalna dla zadania redukcji

Podzbiory atrybutów uzyskane w wyniku redukcji pełnego zbioru atrybutów zapisano w tablicy DSxASy. Tablica ma postać binarną, w której wiersze odpowiadają utworzonym podzbiorom, a kolumny atrybutom zbioru ASy. Wartość 1 będąca elementem macierzy DSxASy oznacza wystąpienie danego atrybutu w zbiorze. Ostatni wiersz macierzy DSxASy zawiera tylko wartości 1, co odpowiada pełnemu zbiorowi atrybutów. Umożliwia to porównywanie efektywności klasyfikacji przeprowadzonej na pełnym zbiorze atrybutów z wynikami uzyskanymi dla każdego z utworzonych podzbiorów. Opis podzbiorów atrybutów przechowywany jest w zmiennej DSxASy_x.

Redukcja metodą zbiorów przybliżonych

W redukcji przestrzeni cech metodą zbiorów przybliżonych zastosowano metodę wyznaczania reduktów względnych omówioną w rozdziale 3.4.8. Dla zadania redukcji jednostanowiskowej wykorzystano funkcję df_sd (listing 3.13), natomiast dla redukcji rozproszonej funkcję dist_df_sd (listing A.4).

Redukcja metodą analizy korelacyjnej

W zadaniu redukcji uwzględniającego analizę korelacyjną opracowano algorytmy corr-AA oraz corr-AC, przytoczone w rozdziale 2.42. Tablica strukturalna RedFeatSets_F1Dx, będąca wynikiem procesu redukcji, zawiera dodatkowo pole DSxASy_cm przechowujące macierz korelacji dla redukowanego zbioru atrybutów. Zmienna DSxASy_x zawiera wartości progowe współczynników korelacji, dla których uzyskano poszczególne redukty. Odpowiadają one wierszom zmiennej DSxASy.

Analiza głównych składowych

Analiza głównych składowych (PCA) jest przykładem zadania transformacji cech. W redukcji metodą PCA wykorzystano funkcję princomp. Przed zastosowaniem metody PCA należy przeprowadzić standaryzację danych. Aby można było przeprowadzić klasyfikację na obiektach nowego układu współrzędnych, należy przekształcić odpowiednio przekształcić dane walidacyjne. W tym celu, w trakcie procesu uczenia, zapamiętywane są wartości potrzebne do standaryzacji, jak: średnia oraz odchylenie. Do wyznaczenia współrzędnych czynnikowych przypadków wykorzystano zmienną coefs.

3.5.3. Klasyfikacja

Zaimplementowane w module RSAm metody klasyfikacji omówiono w rozdziale 2.5. Oznaczenia wykorzystywanych metod zamieszczono w tabeli 3.3.

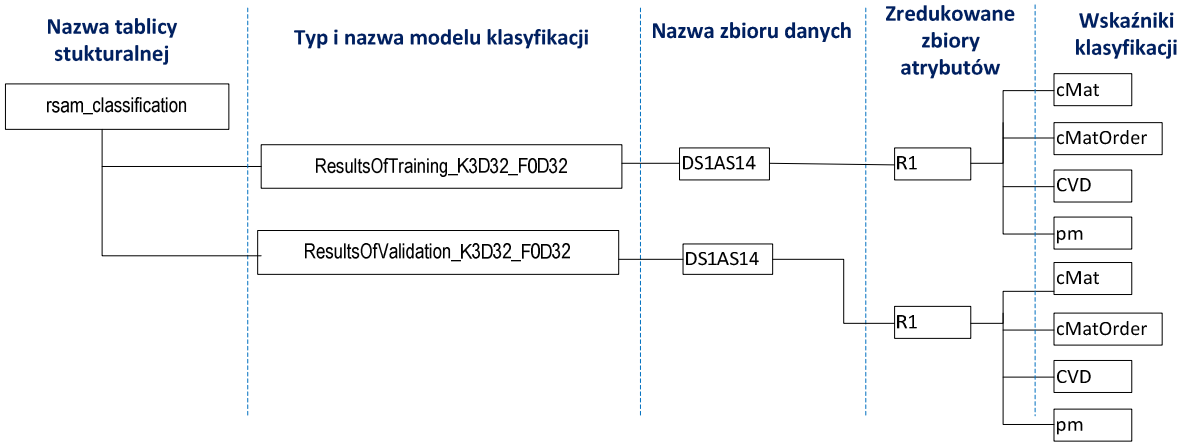
Tabela 3.3. Oznaczenia metod klasyfikacji

Nr metody	Oznaczenie	Opis
1	$K1$	Naiwny klasyfikator Bayesa
21	$K21$	Liniowa analiza dyskryminacyjna
23	$K23$	Kwadratowa analiza dyskryminacyjna
3	$K3$	Drzewa decyzyjne
411	$K411$	Zbiory przybliżone – wersja jednostanowiskowa
421	$K421$	Zbiory przybliżone – wersja rozproszona

W procesie klasyfikacji wyodrębniono dwa etapy: budowanie modelu na podstawie próby uczącej oraz weryfikacja modelu na próbie walidacyjnej. Jeżeli klasyfikację poprzedzono procesem redukcji cech, to klasyfikacja jest przeprowadzana dla każdego z utworzonych zestawów cech.

Wynikiem uczenia nadzorowanego są dwie tablice strukturalne (rys. 3.7):

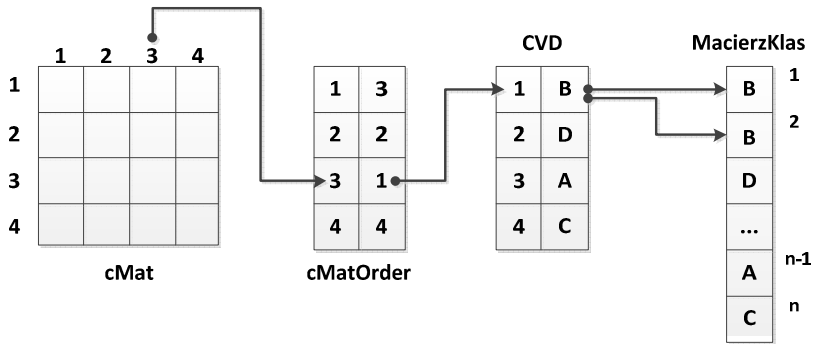
- ResultOfTraining_KxD2y_FzDr – zawiera wyniki weryfikacji klasyfikatora na próbie uczącej;
- ResultsOfValidation_KxD2y_FzDr – zawiera wyniki weryfikacji klasyfikatora na próbie walidacyjnej.



Rys. 3.7. Zagnieżdżona tablica strukturalna dla zadania klasyfikacji

Wynikiem zadania klasyfikacji są miary jakości klasyfikatora przedstawione w rozdziale 2.6. Wyznaczone współczynniki są zapisywane w polu o nazwie pm. Wartości współczynników wyznaczono na podstawie macierzy pomyłek, zapisanej w polu cMat. Informacje o etykietach wierszy i kolumn macierzy cMat, zapisane są w polu cMatOrder.

Wartości atrybutu decyzyjnego mogą być różnego typu (liczbowe, tekstowe). W celu ujednolicenia zapisu klasy aktualne wartości są przekształcane na wartości liczbowe. Słownik konwersji oznaczeń zapisano w zmiennej CVD (rys. 3.8).



Rys. 3.8. Schemat odczytu etykiet klas dla macierzy pomyłek

W nazewnictwie tablic strukturalnych wykorzystano notację przedstawioną w tabelach 3.1, 3.2, 3.3. Na przykład, skrót K21D22 oznacza klasyfikację metodą liniowej analizy dyskryminacyjnej, do której wykorzystano dane dyskretyzowane metodą CAIM. Inny skrót - F6D1W20, oznacza, że na analizowanym zbiorze przeprowadzono selekcję cech metodą sekwencyjną, opartą na danych zdyskretyzowanych metodą równej szerokości o 20 przedziałach. W module RSAm założono możliwość stosowania różnych metod dyskretyzacji dla zadań redukcji i zadań klasyfikacji. Dlatego zapisywane modele składają się z dwóch członów o konstrukcji KxDy_FwDz, gdzie x,y,w,z odpowiadają numerowi operacji zgodnie z tabeleami 3.1, 3.2, 3.3. W przypadku redukcji atrybutów metodą transformacji cech, klasyfikację należy przeprowadzić na nowych zmiennych. W tym celu wykorzystano oznaczenie KxFwDz_FwDz. Na przykład, zastosowanie analizy głównych składowych bez dyskretyzacji danych, a następnie przeprowadzenie na tym zbiorze klasyfikacji metodą liniowej analizy dyskryminacyjnej, opisane zostanie w sposób następujący: K21F3D0_F3D0.

Klasyfikator RS

Opracowany algorytm klasyfikacji polega na określeniu miary podobieństwa rozpoznawanego obiektu do reguł decyzyjnych zbudowanych na podstawie próby uczącej. Każda reguła traktowana jest jak obiekt w przestrzeni xm wymiarowej, gdzie xm odpowiada liczbie atrybutów warunkowych zbioru uczącego. Dla każdego obiektu zbioru poddanego klasyfikacji wyznacza się odległość od obiektu reprezentującego regułę. Jednymi z możliwych miar, mających zastosowanie dla cech ilościowych, są miary odległościowe np. Euklidesowa, miejska (Manhattan), które są szczególnym przypadkiem metryki Minkowskiego opisanej zależnością [Grabc03,MicSte81]:

$$d(\underline{a}_j, \underline{a}_k) = \left[\sum_{i=1}^{p}(v_i^j - v_i^k)^l\right]^{1/l}, \tag{3.37}$$

gdzie:

$d(\underline{a}_j, \underline{a}_k)$ - odległość pomiędzy wektorami atrybutów odpowiadających j-temu i k-temu obiektowi,

l - liczba zależna od miary odległości (np.: Euklidesowa p=2, miejska p=1),

p - wymiar przestrzeni atrybutów (liczba atrybutów opisujących obiekt).

Do klasyfikacji nowych obiektów zastosowano reguły decyzyjne uzyskane w analizie systemu decyzyjnego. Nowy obiekt przypisywany jest do jednej z klas decyzyjnych w oparciu o poniżej przedstawione zasady:

1. Wartości atrybutów nowego obiektu odpowiadają dokładnie jednej regule deterministycznej. W tym przypadku predykcja jest jednoznaczna, nie wymaga definiowania dodatkowych zależności.

2. Wartości atrybutów nowego obiektu odpowiadają atrybutom dokładnie jednej reguły niedeterministycznej. Sytuacja taka nie jest jednoznaczna, gdyż występuje możliwość przypisania analizowanego obiektu do różnych klas. Ponieważ algorytm maksymalizuje czułość to obiekt przypisywany jest do klasy o większym znaczeniu.
3. Wartości atrybutów nowego obiektu pasują do więcej niż jednej reguły decyzyjnej. W przypadku, gdy reguły wskazują na tą samą klasę, to liczba reguł nie ma znaczenia. Gdy reguły wskazują na różne klasy, to problem można sprowadzić do sumarycznego wyznaczenia liczby reguł w ramach każdej z klas, a następnie przypisaniu nowego obiektu do klasy charakteryzującej się większym prawdopodobieństwem wystąpienia.
4. Wartości atrybutów nowego obiektu nie pasują do żadnej reguły decyzyjnej. W celu określenia klasy decyzyjnej nalezy odnaleźć reguły znajdujące się „najbliżej" wartości atrybutów nowo klasyfikowanego obiektu. „Bliskość" wyznaczona jako miara odległości, pozwala na znalezienie klas leżących w sąsiedztwie. Prawdopodobieństwo trafienia właściwej klasy jest w takim przypadku większe niż losowe przypisanie obiektu do jednej z decyzji.

Uwzględniając powyższe zasady opracowano algorytm klasyfikacji wykorzystujący ideę klasyfikacji minimalno-odległościowej. Na podstawie najmniejszej odległości wybiera się możliwą klasę decyzyjną stosując metodologię klasyfikacji tymczasowych. Koncepcja klasyfikacji tymczasowych pozwala na dokładniejszą analizę zachowania klasyfikatora. Metodologię klasyfikacji tymczasowej dla problemu klasyfikacji binarnej przedstawiono w tabeli 3.4.

Tabela 3.4. Tymczasowa macierz pomyłek

<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="8">Klasa obiektu wg klasyfikatora</td></tr>
<tr><td>Klasa1</td><td>Klasa2</td><td>Klasa00</td><td>Klasa01</td><td>Klasa02</td><td>Klasa10</td><td>Klasa11</td><td>Klasa12</td></tr>
<tr><td rowspan="2">Rzeczywista klasa obiektu</td><td>Klasa1</td><td>TP_1</td><td>Err_1</td><td>Err_1</td><td>TP_1</td><td>Err_1</td><td>Err_1</td><td>Err_1</td><td>Err_1</td></tr>
<tr><td>Klasa2</td><td>Err_2</td><td>TP_2</td><td>TP_2</td><td>Err_2</td><td>TP_2</td><td>TP_2</td><td>TP_2</td><td>TP_2</td></tr>
<tr><td colspan="2"></td><td colspan="2">Reguły wskazują na jedną klasę</td><td colspan="6">Reguły wskazują na różne klasy</td></tr>
<tr><td colspan="2"></td><td colspan="5">Odległość = 0</td><td colspan="3">Odległość =min ∧ Odległość ≠ 0</td></tr>
</table>

W tymczasowej macierzy pomyłek wiersze odpowiadają rzeczywistym klasom obiektów, a kolumny klasom wyznaczonym przez klasyfikator. Każda komórka macierzy zawiera liczbę obiektów jaka w wyniku zadania klasyfikacji została przypisana do danej klasy w odniesieniu do rzeczywistej klasy obiektu. Każdy z klasyfikowanych obiektów zostaje przypisany do jednej z ośmiu klas tymczasowych wykorzystując poniższe reguły:

- Klasa1 – klasyfikowany obiekt pasuje do jednej lub wielu reguł decyzyjnych, ale wszystkie reguły wskazują na klasę Klasa1,
- Klasa2 – klasyfikowany obiekt pasuje do jednej lub wielu reguł decyzyjnych, ale wszystkie reguły wskazują na klasę Klasa2,
- Klasa0x – klasyfikowany obiekt pasuje do wielu reguł decyzyjnych, reguły wskazują na różne klasy, przy czym:
  - *Klasa00* – liczba reguł dla każdej z możliwych klas jest taka sama,
  - *Klasa01* – liczba reguł jest większa dla klasy Klasa1,
  - *Klasa02* – liczba reguł jest większa dla klasy Klasa2,

- Klasa1x – klasyfikowany obiekt nie pasuje do żadnej z reguł decyzyjnych, do dalszej analizy wybrano te reguły do których odległość jest najmniejsza, przy czym:
  - *Klasa10* – liczba reguł dla każdej z możliwych klas jest taka sama,
  - *Klasa11* – liczba reguł jest większa dla klasy Klasa1,
  - *Klasa12* – liczba reguł jest większa dla klasy Klasa2.

W kolejnym kroku tymczasową macierz pomyłek przekształca się do macierzy binarnej. Problem ostatecznej klasyfikacji sprowadza się do interpretacji klas, które nie zostały jednoznacznie określone (Klasa0x, Klasa1x). W tym celu należy wskazać, którą z rzeczywistych klas binarnych należy traktować jako klasę Pozytywną, a którą Negatywną (tab. 3.5). Klasa Pozytywna rozumiana jest jako klasa wyróżniona, która charakteryzuje się szczególnym znaczeniem w analizowanym zjawisku, np. wystąpienie choroby. W narzędziu RSA zaproponowano następujący schemat przekształceń maksymalizujący czułość klasyfikacji (liczebność klasy Pozytywnej):

- O obiektach zakwalifikowanych do Klasa00 nie można jednoznacznie powiedzieć, do której klasy należą dlatego obiekty przypisano klasy pozytywnej (P).
- O obiektach zakwalifikowanych do Klasa01 nie można jednoznacznie powiedzieć, że należą do tej klasy, jednak na tą klasę wskazuje większa liczba reguł. Dlatego obiekty przypisano do klasy negatywnej (N).
- O obiektach zakwalifikowanych do Klasa02 nie można jednoznacznie powiedzieć, że należą do tej klasy, jednak na tą klasę wskazuje większa liczba reguł. Dlatego obiekty przypisano do klasy pozytywnej (P).
- O obiektach zaklasyfikowanych do Klasa1x nie można jednoznacznie powiedzieć do której klasy należą. W celu zwiększenia czułości klasyfikatora obiekty przypisano do klasy pozytywnej (P).

Tabela 3.5. Binarna macierz pomyłek

		Klasa obiektu wg klasyfikatora	
		N (Klasa1)	P (Klasa2)
Rzeczywista klasa obiektu	N (Klasa1)	TN	FP
	P (Klasa2)	FN	TP
		Klasa01	Klasa00
		Klasa11	Klasa02
			Klasa10
			Klasa12

Do budowania reguł decyzyjnych wykorzystano metodę przedstawioną w rozdziale 3.4.10, wykorzystującą funkcję rozróżnialności. W module RSAm, budowę klasyfikatora realizuje funkcja df_rules. Do klasyfikacji nowych obiektów, na podstawie utworzonych reguł decyzyjnych, przygotowano funkcję rsclass. W zależności od zadeklarowanego typu obliczeń klasyfikacja może być realizowana w wersji jednostanowiskowej lub rozproszonej.

W wyniku uczenia nadzorowanego metodą RS, tablica strukturalna ResultOfTraining_KxD2y_FzDr, zawiera dodatkowe pola:

- cMatRS – tymczasowa macierz pomyłek;
- DSxASy_reguly – zbiór reguł klasyfikatora;
- DSxASy_czas_reguly – czas generowania reguł klasyfikatora w wersji jednostanowiskowej lub rozproszonej;

- DSxASy_czas_klas – czas klasyfikacji nowych przypadków w wersji jednostanowiskowej lub rozproszonej
- DSxASy_pred – rzeczywisty wynik klasyfikacji, na podstawie którego budowana jest tymczasowa macierz pomyłek.

Inne klasyfikatory

Klasyfikator NaiveBayes zaprojektowano do zadań uczenia nadzorowanego, w których atrybuty w zbiorach danych są od siebie niezależne. Przeprowadzone analizy pokazują, że miary jakości klasyfikacji osiągają wysokie wartości także, gdy warunek niezależności nie jest spełniony. Metoda klasyfikacji Naiwnego Bayesa do budowy modelu klasyfikatora wykorzystuje funkcję NaiveBayes.fit, natomiast do walidacji funkcję predict. Na podstawie zbioru uczącego, klasyfikator przeprowadza estymację parametrów rozkładu. W zadaniu klasyfikacji nowych obiektów, wyznaczane jest prawdopodobieństwo przynależności obiektu do każdej klas. Obiekt zostaje przypisany do tej klasy, która charakteryzuje się najwyższym prawdopodobieństwem.

Klasyfikacja oparta na analizie dyskryminacyjnej realizowana jest przy użyciu funkcji classify. Pozwala to na przeprowadzanie klasyfikacji z wykorzystaniem różnych funkcji dyskryminacyjnych między inynmi:

- liniowa - do każdej klasy dopasowuje wielowymiarową gęstość normalną, z sumaryczną estymatą kowariancji.
- kwadratowa - dopasowuje wielowymiarową gęstość normalną z kowariancją estymowaną dla każdej klasy.
- mahalanobisa - wykorzystuje miarę odległości Mahalanobisa z kowariancją estymowaną dla każdej klasy.

Klasyfikacja z zastosowaniem drzew decyzyjnych wykorzystuje do budowania modelu funkcję classregtree, natomiast do walidacji treeval.

Szczegółowe informacje na temat zastosowanych metod klasyfikacji są dostępne w dokumentacji przybornika Statistica.

3.6. Moduł DB

Rozproszona realizacja algorytmów wymaga przesyłania dużej liczby zbiorów danych. W celu przyspieszenia realizacji obliczeń dla danych wielowymiarowych, opracowano moduł DB. Wykorzystano w tym celu bazę danych MySql.

W module DB zaimplementowano biblioteki umożliwiające komunikację środowiska MATLAB z bazą danych. Wykorzystano w tym celu interfejs mYm v1.36. Jest to zbiór bibliotek stanowiący nakładkę na pierwszą wersję interfejsu 'MySQL and Matlab' autorstwa R. Almgren [wMysql,wMym]. Interfejs 'MySQL and Matlab' umożliwia przeprowadzanie na bazie danych operacji DDL (ang. Data Definition Language) oraz DML (ang. Data Manipulation Language). Interfejs mYm umożliwia dodatkowo zapisywanie macierzy środowiska MATLAB w postaci obiektów BLOB w bazie danych. Zapisywane macierze są kompresowane, co zwiększa szybkość zapisu i odczytu. Dodatkową zaletą zapisywania macierzy w postaci obiektów BLOB jest fakt, iż wartości atrybutów zachowują dokładność zapisu z przestrzeni roboczej środowiska MATLAB.

3.7. Model obliczeń rozproszonych

Przedstawiony w pracy problem redukcji przestrzeni atrybutów przy użyciu metody RS jest złożony obliczeniowo. Złożoność zależy zarówno od liczby obiektów w zbiorze jak i od liczby cech. Liczba obiektów wpływa na rozmiar tablicy rozróżnialności. Od liczby cech zależy wielkość tablicy prawdy opisującej wszystkie możliwe kombinacje wystąpień atrybutów. Liczba wierszy tablicy prawdy rośnie wykładniczo wraz ze wzrostem zbioru atrybutów.

Przy tworzeniu tablicy rozróżnialności systemu informacyjnego porównywane są wszystkie obiekty. Złożoność obliczeniowa wyznaczenia macierzy rozróżnialności jest rzędu $O(n^2)$, gdzie n jest liczbą obserwacji. Na szybkość wyznaczenia wartości funkcji w tablicy prawdy ma wpływ jej rozmiar, który rośnie wykładniczo wraz ze wzrostem liczby analizowanych atrybutów. Wyznaczenie wartości funkcji rozróżnialności w zaproponowanym algorytmie, polega na porównaniu każdego wiersza tablicy rozróżnialności z każdym wierszem tablicy prawdy, co daje złożoność obliczeniową rzędu $O(2^p n^2)$, gdzie p jest liczbą cech.

Wysoka złożoność obliczeniowa związana jest także z klasyfikacją obiektów metodą RS. Czas obliczeń zależy od liczby obiektów w klasyfikowanym zbiorze oraz od liczby reguł klasyfikatora. Złożoność obliczeniowa wynika z wyznaczania odległości klasyfikowanego obiektu od każdej z reguł decyzyjnych.

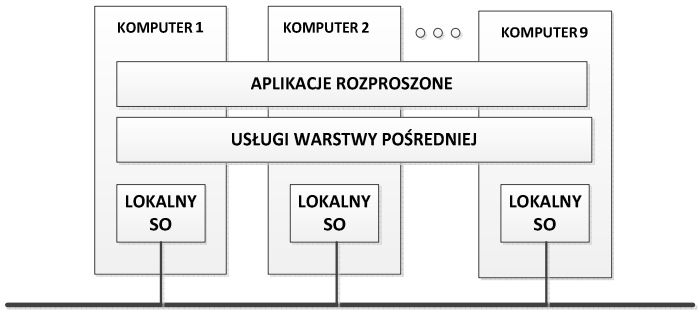
Do rozwiązania problemu złożoności obliczeniowej zastosowano system rozproszony [TanSte06,Gro03,KarNie01,CouDol99]. Zastosowany system jest homogeniczny. Składa się z dziewięciu niezależnych komputerów. Każdy z komputerów posiada jednakowe parametry techniczne:

– Procesor: Intel Core 2 Quad CPU Q6600, 2.4GHz,

– Pamięć 3,5 GB RAM,

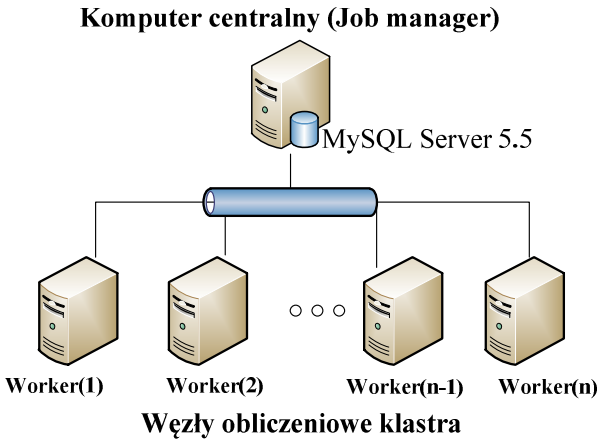
– System Operacyjny: Windows XP 32bit.

System rozproszony zrealizowano poprzez warstwę oprogramowania, umieszczoną logicznie pomiędzy warstwa aplikacji, a warstwą systemu operacyjnego. Schemat przedstawiono na rys. 3.9.



Rys. 3.9. System rozproszony jako warstwa pośrednia oprogramowania (ang. middleware) [Gro03]

Do realizacji systemu rozproszonego wykorzystano komponenty środowiska obliczeniowego MATLAB Parallel Computing Toolbox oraz Distributed Computing Server. Poszczególne komputery połączono stosując topologię drzewa (rys. 3.10). Jeden z komputerów, stanowiący element centralny klastra (ang. job manager), jest odpowiedzialny za nadzorowanie i zarządzanie zadaniami. Pozostałe komputery stanowią węzły obliczeniowe klastra (ang. workers). Węzły komunikują się wyłącznie z elementem centralnym.



Rys. 3.10. Klaster komputerów w postaci topologii drzewa wykorzystany do realizacji obliczeń równoległych

Zadania (ang. task) realizowane przez węzły obliczeniowe są generowane i grupowane na komputerze centralnym, tworząc tzw. obiekty pracy (ang. jobs). Zadania realizowane przez poszczególne węzły klastra mogą się od siebie różnić. Każdy z węzłów po wykonaniu zadania odsyła wynik do menadżera, a ten z kolei przydziela następne wolne zadanie. Stacja robocza wykonuje tylko jedno zadanie w jednej chwili.

Pierwotny sposób wykorzystania środowiska rozproszonego w narzędziu RSA przedstawiono w pracach [SzySta08, Szy07b]. Przeprowadzone eksperymenty obliczeniowe pokazały, iż część czasu w obliczeniach równoległych środowiska MATLAB zajmuje przesyłanie macierzy do węzłów klastra. Dlatego wprowadzono modyfikację w stosunku do wcześniej publikowanych prac [SzySta08, Szy07b]. Element centralny klastra wyposażono w bazę danych, w której przechowywane są macierze wykorzystane przy budowie klasyfikatorów. Zastosowano w tym celu bazę danych MySQL (rozdział 3.6). Zastosowanie bazy danych w narzędziu RSA uprościło system komunikatów pomiędzy komputerem centralnym a węzłami roboczymi. Komunikaty składają się jedynie z informacji sterujących, jak na przykład numer obiektu dla którego mają być przeprowadzone obliczenia.

W narzędziu RSA zaimplementowano dwa poziomy zrównoleglenia obliczeń. Pierwszy poziom obliczeń to rozproszone uruchamianie modułu RSAm. Zaletą wprowadzonego rozwiązania jest zarządzanie analizami z poziomu komputera centralnego. Umożliwia to automatyczne gromadzenie wyników obliczeń w jednej przestrzeni roboczej MATLABA, bez konieczności ręcznego uruchamiania programu na stacjach roboczych.

Drugi poziom rozproszenia dotyczy obliczeń szczegółowych, jak redukcja przestrzeni cech metodą RS, czy równoległa klasyfikacja metodą RS. W zadaniu selekcji cech metodą RS wprowadzono zrównoleglenie na poziomie wyznaczania tablicy rozróżnialności. Na każdym z węzłów obliczeniowych wyznacza się odpowiednie fragmenty tablicy rozróżnialności, a następnie dla każdego z fragmentów tablicy rozróżnialności wyznacza się funkcję rozróżnialności. Po wykonaniu obliczeń, wyniki przekazywane są do komputera centralnego. Poszczególne zadania obliczeniowe realizowane są niezależnie od siebie, co znacząco wpływa na możliwość równoległej implementacji.

Kody źródłowe programów realizujących algorytmy rozproszone zamieszczono w załączniku A. Kody przedstawiono w postaci uproszczonej, aby zwiększyć czytelność algorytmów. Rozproszona realizacja obliczeń wymaga zdefiniowania dwóch plików. Pierwszy, o charakterze sterującym, zawiera zadania dla managera zadań. Przykładem może być program `dist_df_sd` (listing A.5) realizujący rozproszne wyznaczanie funkcji rozróżnialności. W pliku sterującym wyodrębnio pięć części:

- zdefiniowanie managera zadań,

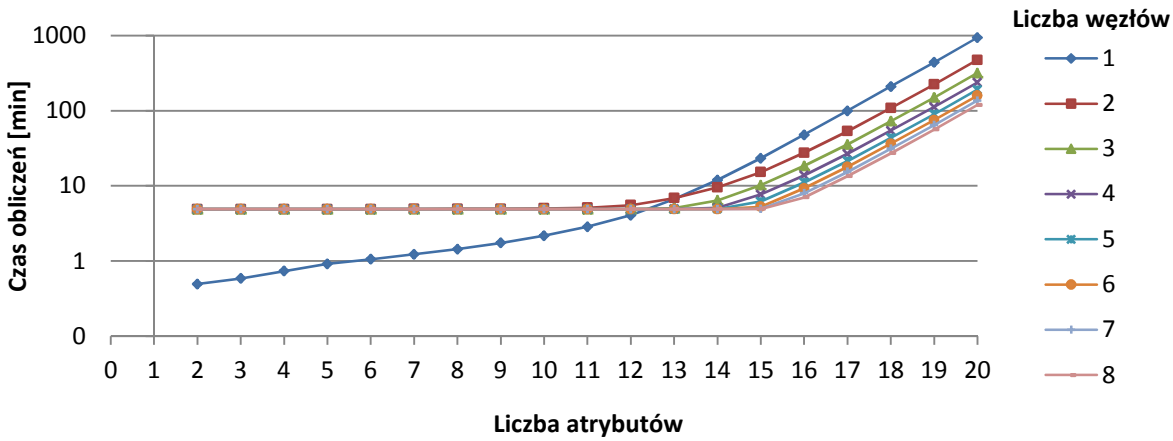
- inicjalizacja obiektu pracy,
- wysłanie obiektu pracy do managera zadań,
- odbieranie wyników obliczeń,
- usunięcie obiektu pracy.

Drugi plik, zdefiniowany do obliczeń rozproszonych, uruchamiany jest bezpośrednio na węzłach klastra. Zawiera listę operacji wymaganych do realizacji zadania. Instrukcje realizowane na węzłach odwołują się bezpośrednio do funkcji zdefiniowanych w narzędziu RSAToolbox. Przykładem jest program dist_df_sd_step (listing A.6), który składa się z dwóch części:

- pobranie macierzy z bazy danych,
- realizacja obliczeń.

3.7.1. Wyznaczanie reduktów względnych

Rozproszone wyznaczanie reduktów względnych zrealizowano na drugim poziomie zrównoleglenia. Analizę wyznaczania reduktów względnych przeprowadzono dla zbioru uczącego zawierającego ok. 18 tysięcy obiektów. Analizę przeprowadzono dla różnej liczby atrybutów decyzyjnych. Ze względu na złożoność obliczeniową, analizy przeprowadzono na maksymalnej liczbie 20 cech. Zależność czasu obliczeń od liczby cech przedstawiono na rys. 3.13. Na osi odciętych zaznaczono liczbę atrybutów warunkowych dla których przeprowadzano obliczenia, natomiast na osi rzędnych czas obliczeń w skali logarytmicznej. Każda z zamieszczonych charakterystyk odpowiada innej liczbie węzłów klastra, dla której przeprowadzono obliczenia.



Rys. 3.11. Czas wyznaczania reduktów względnych w zależności od liczby atrybutów

Przedstawione na rysunku 3.11 charakterystyki pokazują, że zastosowanie obliczeń rozproszonych w porównaniu z obliczeniami jednostanowiskowymi (1 węzeł) przynosi widoczne korzyści dopiero przy 14 atrybutach. Im większa liczba atrybutów tym zysk czasowy z wprowadzenia obliczeń równoległych jest większy. Począwszy od piętnastu atrybutów, czas realizacji wzrasta prawie dwukrotnie przy każdym kolejnym atrybucie, niezależnie od liczby węzłów. Zadanie poszukiwania reduktów dla zbioru 20 atrybutów zajmuje około 940 minut (1 węzeł). Stosując dwa węzły klastra czas można skrócić już o prawie o połowę. Zastosowanie 8 węzłów pozwala na wyznaczenie reduktów w niecałe 120 minut (około 8-krotonie krócej).

Czasy obliczeń dla liczby węzłów większej od 2, pokrywają się w zakresie liczby atrybutów od 1 do 11. Wynika to z faktu, iż czas komunikacji komputera centralnego

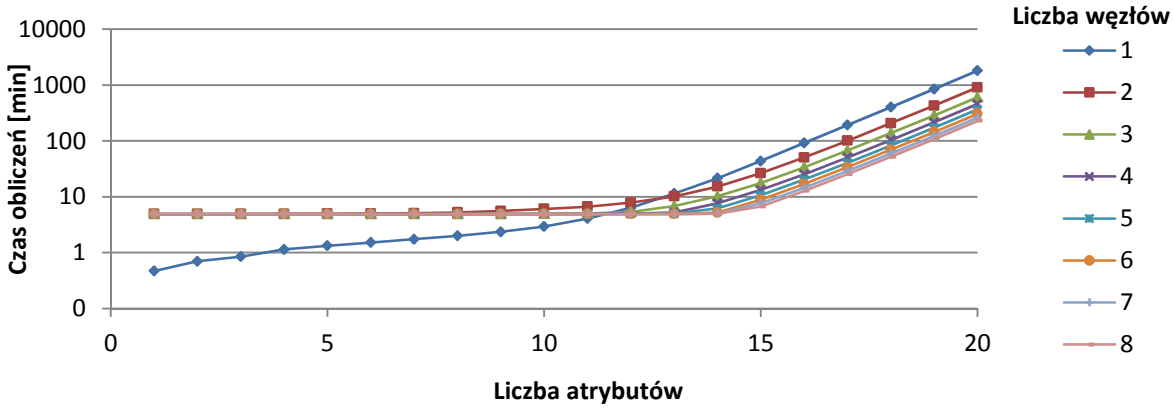
z węzłami przewyższa czas obliczeń wykonanych na poszczególnych. Czas komunikacji zależy od czasu przesyłania zadania i odbierania wyników oraz od czasu potrzebnego na pobranie danych z bazy.

3.7.2. Klasyfikacja metodą RS

Na zadanie klasyfikacji składają się dwa etapy obliczeń. Pierwszy związany jest z budowaniem modelu klasyfikatora. Drugi etap to klasyfikacja nowych przypadków na podstawie utworzonych reguł decyzyjnych. Każdy z etapów obliczeń może być realizowany w sposób równoległy. W pierwszym przypadku, na złożoność obliczeniową wpływa liczba atrybutów oraz liczba obiektów w danych uczących. W drugim etapie czas obliczeń zależy od liczby obiektów w danych uczących oraz od liczby reguł decyzyjnych.

Na rysunku 3.12 przedstawiono charakterystykę czasu obliczeń przy wyznaczaniu reguł decyzyjnych. Kształt charakterystyk jest analogiczny do zależności czasowych przedstawionych na rys. 3.11. Należy jednak zwrócić uwagę na czasy obliczeń. Wyznaczenie reduktów dla zbioru dziesięciu atrybutów, w wersji jednostanowiskowej, zajmuje ok. 3min, natomiast przy wyznaczaniu reguł decyzyjnych jedynie 2 min. Przy wyznaczaniu reduktów dla 14 atrybutów obliczenia zajmują 11 minut, natomiast przy budowaniu reguł decyzyjnych potrzeba już 21 minut.

W zadaniu poszukiwania reguł decyzyjnych pierwszy zysk czasu obliczeń rozproszonych, w porównaniu z jednostanowiskowymi, osiąga się przy 12 atrybutach. Wraz ze wzrostem liczby atrybutów zysk jest coraz większy.



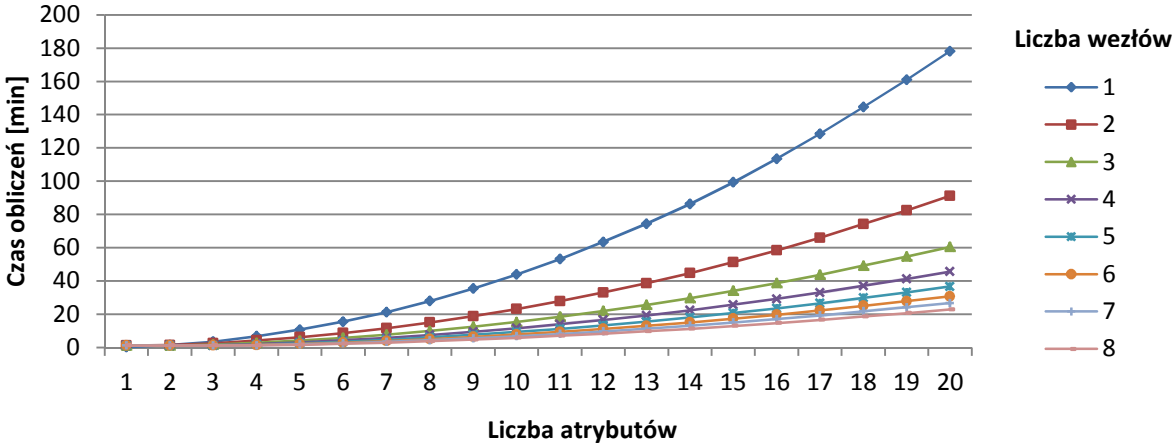
Rys. 3.12. Czas wyznaczania reguł decyzyjnych w zależności od liczby atrybutów

Należy zauważyć, że przedstawione wyniki są charakterystyczne dla analizowanego rozkładu wartości i klas. Przy budowaniu tablicy rozróżnialności, w systemach decyzyjnych porównuje się wartości atrybutów obiektów należących do różnych klas. Wynika z tego, że rozmiar tablicy rozróżnialności wykorzystywanej przy wyznaczaniu funkcji rozróżnialności zależy od gęstości klas. W analizowanym problemie tablica rozróżnialności wyznaczona dla i-tego obiektu, należącego do klasy komórek nowotworowych, będzie większa od tablicy rozróżnialności wyznaczanej dla j-tego obiektu należącego do klasy komórek zdrowych.

Ponieważ w zbiorze 97% komórek jest zdrowych, to w 97% analiz maksymalny rozmiar tablicy rozróżnialności będzie odpowiadał 3% wierszy zbioru uczącego. Dla próby uczącej liczącej 18000 przypadków, w której stosunek klas jest 3:97, będzie 17460 analiz przeprowadzonych dla tablicy rozróżnialności o maksymalnej liczbie wierszy wynoszącej 540 i 540 analiz przeprowadzonych dla tablicy rozróżnialności o maksymalnie 17460

wierszach. Zależność tą należy uwzględnić przy projektowaniu rozproszonego systemu decyzyjnego.

W zaprojektowanym klasyfikatorze z zastosowaniem zbiorów przybliżonych, czas zadania klasyfikacji zależy od liczby przypadków walidacyjnych oraz od liczby reguł decyzyjnych klasyfikatora. Na liczbę reguł decyzyjnych wpływa liczba atrybutów opisujących obiekty zbioru uczącego. Czasy obliczeń przy zastosowaniu różnej liczby węzłów klastra przedstawiono na rys. 3.13. Czasy obliczeń wyznaczono na zbiorze walidacyjnym składającym się z 4600 obiektów.

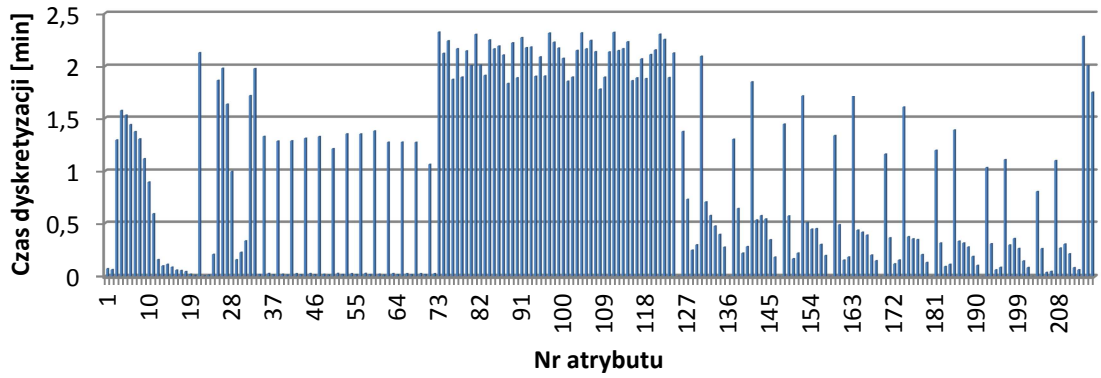


Rys. 3.13. Czas klasyfikacji metodą RS w zależności od liczby atrybutów

Obserwując powyższą charakterystykę widać, że zysk czasu obliczeń jest zauważalny już przy 3 atrybutach. Pierwszy znaczący zysk otrzymano przy 6 atrybutach. Czas realizacji obliczeń jednostanowiskowych wynosi w tym przypadku ok. 15 minut. Przy zastosowaniu 2 węzłów klastra czas ten maleje już o połowę. Przy zastosowaniu przynajmniej 6 węzłów czas skraca się od 3 do 2 min. Niewielkie różnice czasowe pojawiające się przy zbiorach od 6 do 8 atrybutów, wskazują, że przy klasyfikacji dużych zbiorów danych, w systemach rozproszonych o dużej liczbie węzłów, celowym jest tworzenie podklastrów. Zbiór walidacyjny można wtedy podzielić w stosunku odpowiadającym liczbie podklastrów i przeprowadzać obliczenia równolegle.

3.7.3. Dyskretyzacja atrybutów

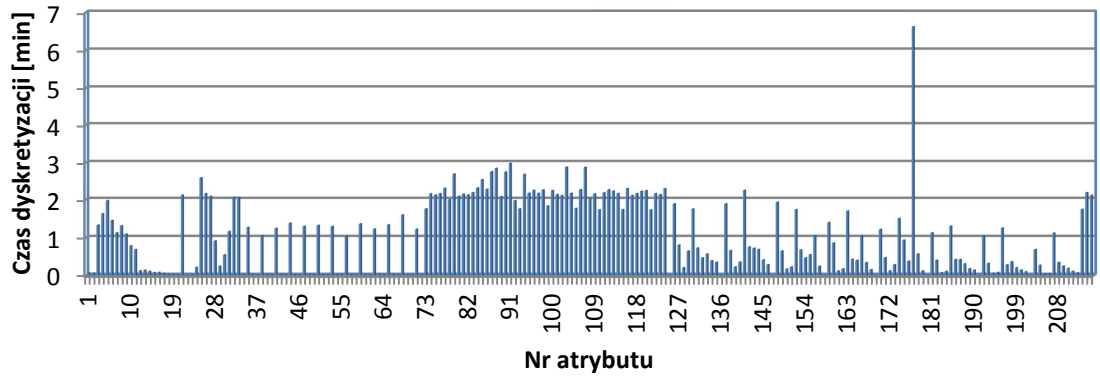
W algorytmach dyskretyzacji metodą CACC oraz CAIM obliczenia są przeprowadzane osobno dla każdego z analizowanych atrybutów. Wyznaczanie współczynnika kryterium dla każdej z możliwych granic przedziałów, przy dużej zmienności atrybutu może się okazać bardzo czasochłonne. Na rys 3.14. zamieszczono czasy poszukiwania granic przedziałów dyskretyzacji metodą CAIM. Charakterystykę przedstawiono w postaci słupków kolumnowych, gdzie każdy słupek odpowiada jednemu atrybutowi. Przedstawiony wykres dotyczy zbioru komórek raka pęcherza moczowego.



Rys. 3.14. Histogram czasów dyskretyzacji pojedynczych atrybutów metodą CAIM dla zbioru 215 atrybutów

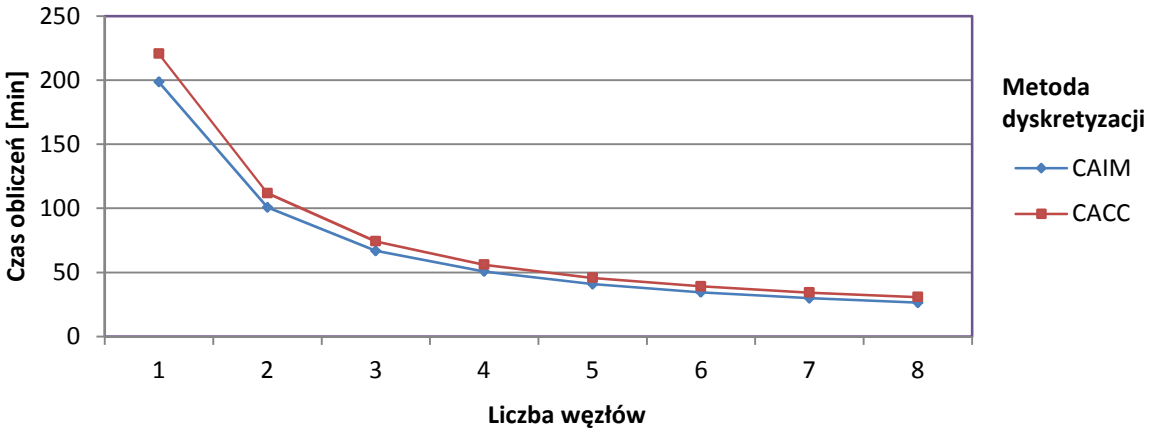
Jak pokazano na rys 3.14, czas obliczeń jest bardzo zróżnicowany dla poszczególnych atrybutów. Maksymalny czas obliczeń, jaki zaobserwowano, wynosi 2.3 minuty. Ponieważ atrybuty dyskretyzowane są kolejno, to w przypadku realizacji jednostanowiskowej dyskretyzacja okazuje się bardzo czasochłonna. Czas potrzebny na przeprowadzenie dyskretyzacji 215 atrybutów wynosi 199 minut.

Również dyskretyzacja metodą CACC charakteryzuje się wysoką złożonością czasową (rys. 3.15). Dla większej liczby atrybutów, czas ten przewyższa czas dyskretyzacji metodą CAIM. Najdłuższy czas wynosi 6.5 minuty. Podobnie jak w przypadku metody CAIM, zaobserwowano zróżnicowanie w czasach obliczeń.



Rys. 3.15. Histogram czasów dyskretyzacji pojedynczych atrybutów metodą CACC dla zbioru 215 atrybutów

Zaobserwowane zależności skłaniają do zastosowania rozproszonego modelu obliczeń. Najbardziej odpowiednie w tym przypadku jest dyskretyzowanie każdego z atrybutów na osobnym węźle klastra. Wyniki przeprowadzonych badań przedstawiono na rys. 3.16.



Rys. 3.16. Czas dyskretyzacji 212 atrybutów metodą CAIM, CACC

Na przedstawionym wykresie widać, że zastosowanie obliczeń rozproszonych pozwala na znaczące przyspieszenie czasu dyskretyzacji. Zastosowanie dwóch węzłów skraca ten czas już o połowę. Największe zyski otrzymano dla dyskretyzacji zrealizowanej na 8 węzłach, która w przypadku metody CAIM wynosiła 26 minut, natomiast w przypadku metody CACC - 30 minut.

3.8. Podsumowanie

W rozdziale przedstawiono podstawowe problemy z obszaru teorii zbiorów przybliżonych opracowane na podstawie bardzo szerokiej literatury [WaMa99,Dom04,Paw83,Rut05,MroPlo99,Baz98,PawSlo07,PaSk07]. Jak pisze prof. A.Skowron [IszNowIn10] od pierwszej publikacji prof. Z. Pawlaka nt. zbiorów przybliżonych do chwili obecnej opublikowano jedynie w języku angielskim ponad 4000 publikacji.

Przedstawiono opracowany przybornik RSAToolbox dla środowiska obliczeniowego MATLAB. Narzędzie składa się z trzech modułów:

- RS – zbiór programów opracowanych dla zagadnienia zbiorów przybliżonych, umożlwiający m.in. wyznaczanie zbiorów elementarnych (elementary_sets.m), określanie aproksymacji zbiorów i rodziny zbiorów (aproximation.m), wyznaczanie rdzenia atrybutów (core.m, core_rel.m), czy generowanie funkcji rozróżnialności (df_sd.m). Programy zrealizowano wykorzystując operacje wektorowe i macierzowe, charakterystyczne dla środowiska MATLAB;
- RSAm – zbiór programów ułatwiajacych realizację procesu uczenia nadzorowanego w sposób zautomatyzowany, co umożliwia poszukiwanie optymalnego zbioru atrybutów oraz optymalnego klasyfikatora. W procesie uczenia można wykorzystać implementacje teorii zbiorów przybliżonych zawarte w module RS, a także funkcje klasyfikacyjne narzędzia Statistic Toolbox (Naiwny klasyfikator Bayesa, Analiza dyskryminacyjna, rzewa decyzyjne);
- DB – zbiór interfejsów programowych służących do połączenia środowiska MATLAB z bazą danych. Moduł wykorzystano w obliczeniach równoległych do skrócenia czasów transmisji danych pomiędzy węzłami klastra.

Ze względu na wysoką złożoność obliczeniową, tak czasową jak i pamięciową, niezbędne jest stosowanie obliczeń równoległych. Zastosowano w tym celu komponenty środowiska MATLAB: Parallel Computing Toolbox oraz Distributed Computing Server.

Wykorzystanie operacji wektorowych i macierzowych w środowisku MATLAB pozwala na znaczne przyspieszenie operacji związanych z analizą danych wielowymiarowych.

Przy wyznaczaniu reduktów względnych korzyści z zastosowania obliczeń równoległych są widoczne już przy 13 atrybutach. Przy małej liczbie atrybutów w procesie obliczeniowym istotną rolę odgrywa czas niezbędny do transmisji danych między węzłami klastra. Przy większej liczbie atrybutów (większej lub równej 15) czas obliczeń jest eksponencjalnie zależny od liczby atrybutów zgodnie z zależnością:

$$\tau[min] \approx \frac{24 \cdot 2^{(p-15)}}{w^{0,87}}, \tag{3.38}$$

gdzie:

p – liczba atrybutów,

w – liczba węzłów.

Wzrost liczby węzłów klastra powoduje niemal odwrotnie proporcjonalny spadek czasu obliczeń (wykładnik równy 0,87).

Podczas klasyfikacji na podstawie reguł decyzyjnych korzyści z wprowadzenia obliczeń równoległych są widoczne już przy 6 atrybutach. Czas obliczeń jest potęgowo zależny od liczby atrybutów (z wykładnikiem równym 1.5), zgodnie z zależnością:

$$\tau[min] \approx \frac{16}{w}[1 + 0.18(p-6)^{1.5}], \tag{3.39}$$

gdzie:

p – liczba atrybutów,

w – liczba węzłów.

Zależność 3.39 jest słuszna dla liczby atrybutów większej lub równej 6. Wzrost liczby węzłów klastra powoduje odwrotnie proporcjonalny spadek czasu obliczeń.

Zdaniem autorki opracowane narzędzie w znaczącym stopniu może ułatwić zastosowanie teorii zbiorów przybliżonych do rozwiązywania zadań o dużej liczbie cech i bardzo dużej liczbie przypadków. Wykorzystanie modułu RSAm, pozwala na zautomatyzowanie obliczeń rozproszonych. Problem ten jest istotny szczególnie w analizie danych wielowymiarowych, dla których pojedyncze uruchomienie programu jest procesem charakteryzującym się wysoką złożonością czasową

ROZDZIAŁ 4

Diagnostyka medyczna nowotworów

4.1. Diagnostyka medyczna

Systemy diagnostyczne stosowane w medycynie należą do grupy systemów doradczych. Mogą mieć charakter systemów konsultacyjnych lub mogą być ważnym elementem diagnostyki laboratoryjnej. W pierwszym przypadku, funkcjonując w oparciu o odpowiednio przygotowaną bazę wiedzy medycznej, mają na celu głównie wspomaganie przy rozpoznaniu choroby na podstawie objawów. W drugim, ze względu na specyficzność pracy laboratorium wynikającą z małej trwałości materiałów podlegających badaniu, a także ważności wyników badań, stanowią bodziec do rozwoju i udoskonalenia systemów diagnostyki cechujących się wysoką niezawodnością [Rus02,Jan02,Kul02,SzyWol02, Cie02,Tad09,DulPie00,Hal99].

Postęp nauki w dziedzinach takich jak automatyka i robotyka oraz informatyka stwarza ogromne możliwości dalszego rozwoju medycznych systemów diagnostyki, procesów pomiaru sygnałów, analizy danych, procesów obrazowania diagnostycznego, czy klasycznych badań analitycznych [Rus02,Jan02,Kul02,SzyWol02,RudGra07]. Nieustannie rozwijanym kierunkiem jest wykorzystanie sztucznej inteligencji oraz komputerowych systemów baz danych. Od kilkunastu lat rośnie zainteresowanie metodami związanymi z maszynowym uczeniem się na podstawie przykładów (ang. machine learning) i odkrywaniem wiedzy na podstawie danych (ang. knowledge discovery). Umożliwiają one poszukiwanie zależności charakterystycznych dla analizowanych danych oraz ich czytelną interpretację. Zagadnienia diagnostyki medycznej mogą być rozpatrywane jako systemy ekspertowe [Bub90,Joz98,Rus02,Jan02,Kul02], sieci neuronowe [RutSta02,Cie02, TadPal07,Rut00,Szc00], czy logika rozmyta [RutSta02, Str02,Cie02, KowObu09,Szc00]. Na uwagę zasługuje także metoda zbiorów przybliżonych, której pierwsze zastosowania dotyczyły problematyki medycznej [KomPol99,PawSlo02].

Ważną rolę w diagnostyce odgrywają techniki obrazowania medycznego oraz związane z nimi przetwarzanie, analiza i automatyczne wspomaganie rozpoznawania obrazów. Przykładami obrazów są: zdjęcia rentgenowskie [TadOgi09a,Rum03], MRI (obrazowanie rezonansem magnetycznym, ang. Magnetic Resonance Imaging) [Wlo09,HadFig09], USG (obrazy ultrasonograficzne, ang. ultrasonography) [Zaz09, Prz03], tomogramy (tomografia komputerowa) [CzoZaz09, TadOgi09b,NowKac03], itp. Przetwarzanie obrazu ma na celu zwiększenie jakości obrazu (np. zwiększanie kontrastu, zaznaczanie krawędzi) oraz wydzielenie tych obiektów, które są interesujące z medycznego punktu widzenia. Etap analizy prowadzi do zdefiniowania właściwości (cech) ułatwiających skuteczne podejmowanie decyzji. Często są to informacje, których nie można bezpośrednio dostrzec [Tad97,GrzHip02,MarKor00,JawKan09,DziMar07, HreKor07,TomPuc07].

Jednym z działów medycyny, w którym coraz częściej wykorzystuje się osiągnięcia współczesnych technologii jest patologia. Przedmiotem badań tej specjalności jest analiza

i rozpoznawanie zmian chorobowych zachodzących w organizmie człowieka na podstawie substratu morfologicznego. Zastosowanie kamer CCD pozwala na przenoszenie obrazów mikroskopowych do pamięci komputera, gdzie następnie są poddawane dalszej obróbce [SzyWol02,StrzZie02,Guz05,PleLoe01].

Diagnostyka mikroskopowa różni się od innych obszarów diagnostyki obrazowej (rentgenodiagnostyka, tomografia komputerowa itp.). W odróżnianiu od innych badań, celem diagnostyki mikroskopowej nie jest opis lokalizacji anatomicznej schorzenia, lecz określenie charakteru obserwowanych zmian. Zasady rozpoznawania obrazów mikroskopowych opierają się o kryteria dotyczące struktury całego obrazu lub zespołów cech morfologicznych wyróżnionych obiektów. W innych metodach diagnostyki obrazowej poszukuje się wyróżniających, pojedynczych ognisk lub obszarów, które są niezgodne z normami morfologicznymi danego narządu (pod względem kształtu, wielkości, gęstości optycznej struktury). Dlatego w obrazach mikroskopowych nie można zastosować metody rozpoznawania polegającej na porównywaniu i substrakcji obrazu wzorcowego i obrazu diagnozowanego [PleLoe01,ZieStr03,Nie03,Zi03].

4.2. Nowotwór pęcherza moczowego

Rak pęcherza moczowego jest jednym z coraz częściej wykrywanych nowotworów [BorSie04]. Występuje on przeważnie u osób w wieku starszym (60 -70 lat) i stanowi czwarty co do częstości występowania nowotwór złośliwy u mężczyzn, a ósmy u kobiet. Guzy pęcherza można podzielić na kilka grup. Najczęstszym typem raka jest rak z nabłonka przejściowego (ang. transitional cell carcinoma) stanowiący 90% wszystkich przypadków. Kolejne istotne grupy to rak płaskonabłonkowy i rak gruczołowy [BorSie04].

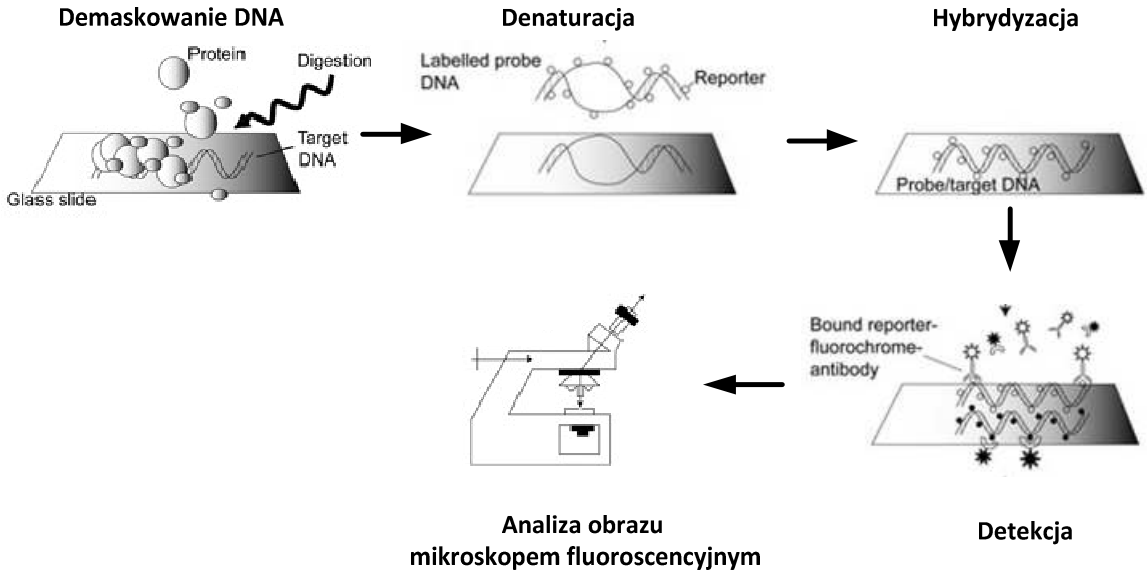
Aktualnie są stosowane różne metody badań umożliwiające rozpoznanie choroby:

- posiew moczu (laboratoryjne testy na obecność bakterii),
- cytologia moczu (mikroskopowe badania komórek wyeliminowanych z pęcherza),
- cytometria przepływowa (pomiar charakterystycznych fizycznych lub chemicznych cech komórek),
- cystoskopia (badanie pęcherza moczowego przy użyciu wziernika),
- biopsja (pobranie fragmentów tkanki do analizy komórek rakowych i identyfikacji typu nowotworu),
- urografia dożylna (wstrzykiwanie do krwiobiegu kontrastowego barwnika i przeprowadzenie zdjęcia rentgenowskiego) [BorCon03, wAbot10].

Interesującym kierunkiem badań we wczesnej diagnostyce nowotworów jest zastosowanie biomarkerów. Biomarkery są to substancje produkowane przez nowotwory lub wytwarzane przez organizm w reakcji na obecność nowotworów w organizmie. Wykrywanie komórek rakowych z wykorzystaniem biomarkerów możliwe jest poprzez badanie krwi, moczu czy tkanki. Analizy takie mogą być powiązane z innymi badaniami jak cystoskopia, gdy monitorowanie pewnych części układu moczowego jest utrudnione bądź niemożliwe [BorCon03].

Jedną z metod wykorzystania markerów jest fluorescencyjna hybrydyzacja in situ (FISH) [OliFre05,HubKul01]. Do zastosowania metody FISH wymagane jest rozdzielenie podwójnej helisy (denaturacja DNA, rys. 4.1) poprzez wysuszenie preparatu na szkiełku mikroskopowym. W metodzie FISH markerem jest sekwencja DNA, którą uwidacznia się przez hybrydyzację z sondą fluorescencyjną. Zastosowanie znaczników o różnych kolorach emisji, umożliwiło hybrydyzację wielu sond z jednym chromosomem. Znaczniki fluorescencyjne o różnych kolorach (różne długości fal) włącza się do nukleotydów lub

bezpośrednio do cząsteczki DNA. Do wykrywania wykorzystywane są mikroskopy fluorescencyjne [Bro01,ZajWis03,OliFre05].

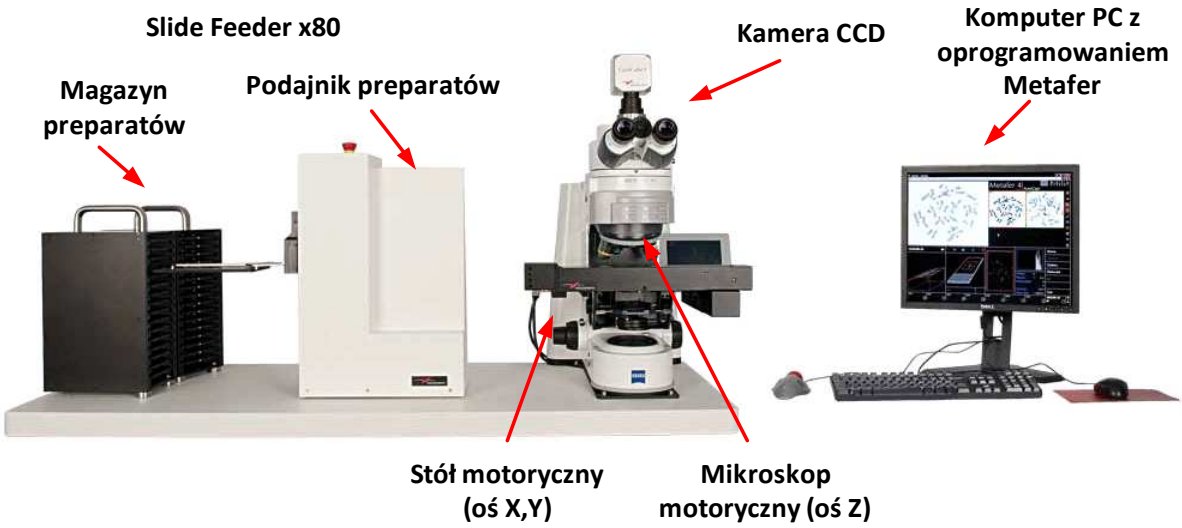


Rys. 4.1. Schemat FISH [opracowano na podstawie OliFre05]

W preparatach mikroskopowych nie są obserwowalne naturalne obrazy komórek. W wyniku różnych procedur i czynności związanych z przygotowaniem preparatu (np. utrwalenie, odwodnienie, zjawiska fizykochemiczne i reakcje podczas barwienia) komórki, ulegają zmianom i deformacjom. Pojedyncze komórki są prawie przezroczyste i muszą być odpowiednio uwidocznione, aby były dostrzegalne pod mikroskopem. Stosuje się w tym celu różne sposoby barwienia, obejmujące cały zakres światła widzialnego. Każdy z nich w odmienny sposób uwidacznia poszczególne składniki komórek co wpływa na ich formę morfologiczną. W przypadku metody FISH zastosowano barwnik DAPI (dichlorowodorek 4,6-diamino-2-fenyloindolu) [DanRon05,DanRon07,ZieStr03].

4.2.1. Obrazowanie przy użyciu systemu skaningowego

W diagnostyce mikroskopowej wykorzystuje się tzw. systemy skaningowe. Przykładem systemu skaningowego jest „Metafer" firmy „MetaSystems" (rys. 4.2) [PleLoe01,Guz05,HubLor98,HubKul01,GuzSzy06]. Głównymi elementami systemu „Metafer" są: kamera CCD zamontowana na motorycznym mikroskopie oraz automatyczny, 8-mio pozycyjny stół skaningowy. W rozbudowanej wersji dostępny jest także automatyczny magazyn SlideFeeder, który obsługuje analizę od 80 do 880 preparatów. Składa się on z centralnego podajnika, który pobiera preparaty z elementów magazynujących i przekazuje je na stół motoryczny.



Rys. 4.2. Komponenty systemu "Metafer" [5] [Guz05]

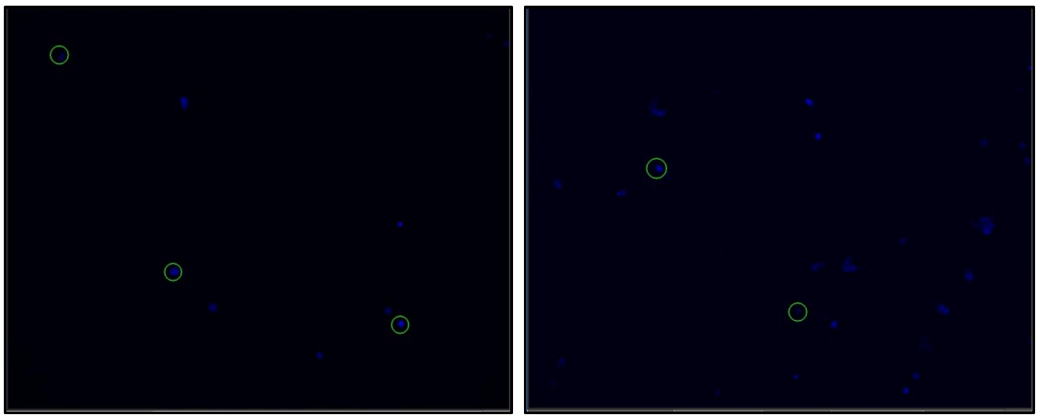
System „Metafer" rozpoczyna analizę znajdującego się pod mikroskopem preparatu od jego podziału na pola skanowania. Wielkość pola wyrażona jest w mikrometrach i jest stosunkiem rozdzielczości kamery CCD do powiększenia użytego w mikroskopie. Ostrość preparatu dobierana jest automatycznie. Pobieranie obrazu odbywa się na podstawie wcześniej zdefiniowanych parametrów skanowania (powiększenie, liczba detektorów FPA (ang. Focal Plane Array), liczba kanałów kolorów i in.). Obraz jest pobierany osobno dla każdego ze zdefiniowanych kanałów kolorów (rys. 4.3).

Rys. 4.3. Pobieranie obrazu za pomocą mikroskopu fluorescencyjnego[5]

Pobrany za pomocą kamery obraz pola preparatu przesyłany jest do komputera. Metodyka wykonywania rozmazów cytologicznych powoduje, że komórki często tworzą zlepy, grupy, skupiska, wzajemnie nakładając się na siebie [ZieStr03]. Największe z nich mają tendencję do rozmieszczania się na obwodzie preparatu. Przy analizie powinno się brać pod uwagę wszystkie pola, gdyż losowy wybór pól może prowadzić do zafałszowań statystycznych dotyczących wskaźników ilościowych opisujących badaną populację komórkową.

[5] http://www.metasystems-international.com

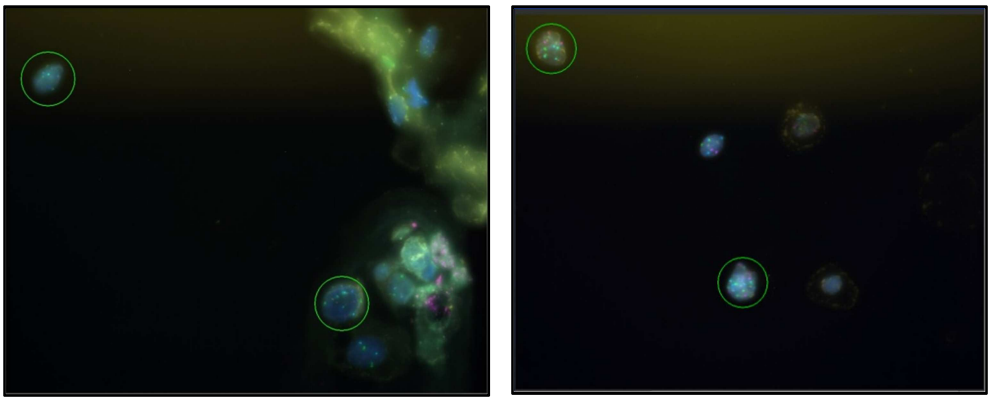
Przykłady obrazów pojedynczych pól preparatów zamieszczono na rysunkach 4.4 i 4.5. Na rysunkach tych okręgami zaznaczono komórki nowotworowe.



Rys. 4.4 Obrazy pojedynczego pola preparatu widoczne w kanale DAPI przy 10-krotnym powiększeniu.

Rysunek 4.4 przedstawia obrazy dwóch pól preparatu widoczne tylko w kanale DAPI, przy skanowaniu z 10-krotnym powiększeniem. Kanał ten jest widziany jako niebieski. Analiza kanału DAPI służy do wyszukiwania potencjalnych komórek nowotworowych na podstawie kształtu, obszaru, intensywności i innych cech.

Wydzielenie komórek nowotworowych w preparacie badanym w kanale DAPI stanowi wstępną diagnostykę nowotworu pęcherza moczowego. Obiekty wskazane na pierwszym etapie są następnie badane przy pomocy metody FISH.

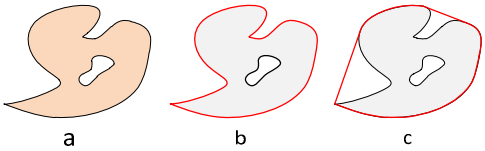


Rys. 4.5 Obrazy pojedynczego pola preparatu widoczne we wszystkich kanałach kolorów przy 40-krotnym powiększeniu

Na rys 4.5. zamieszczono obrazy dwóch zeskanowanych pól widoczne w różnych kanałach kolorów. Naświetlanie kanałami kolorów umożliwia zliczanie sygnałów, czyli tzw. spotów. W zależności od liczby spotów określa się czy komórka jest nowotworowa.

Liczba komórek w analizowanych preparatach może być bardzo duża. Przy skanowaniu każdego pola preparatu różnymi kolorami, badanie trwałoby kilka, a nawet kilkanaście godzin. Dlatego ważnym etapem jest skanowanie wstępne w kanale DAPI, które pozwala na wybranie pól preparatu w których mogą znajdować się komórki nowotworowe.

Do automatycznej oceny typu komórki wykorzystuje się parametry morfometryczne. Każdy parametr morfometryczny stanowi liczbowy opis obiektu morfologicznego (rys. 4.6). Może on dotyczyć takich właściwości jak liczebność obiektów, ich wielkość, kształt, właściwości optyczne, tekstura czy topologia. Wybór parametrów zależy od rodzaju badanego materiału genetycznego.



Rys. 4.6. Przykładowe parametry geometryczne obiektów:
a) pole powierzchni b) obwód c) obwód wypukły [StrzZie02]

Podstawowymi parametrami morfometrycznymi w diagnostyce mikroskopowej komórek pęcherza moczowego, określonymi przez ekspertów, są: całkowita/względna powierzchnia komórki (ang. absolute/relative cell area), całkowita/względna/ /radialna/centralna intensywność komórki (ang. absolute/ relative/radial/center cell intensity), obwód (ang. Circumference), współczynnik kształtu (ang. aspect ratio), nieregularność (ang. irregularity), kolistość (ang. roundness), głębia wklęsłości (ang. concavity depth), promień obrysu (ang. contour radius), ziarnistość (ang. granularity), centralny/radialny moment intensywności (ang. center/radial distance moment), oraz liczba obiektów na określonym poziomie intensywności (ang. the number of objects at a given intensity). Zestaw parametrów powiększony jest o współczynniki, wartości średnie i odchylenia parametrów podstawowych.

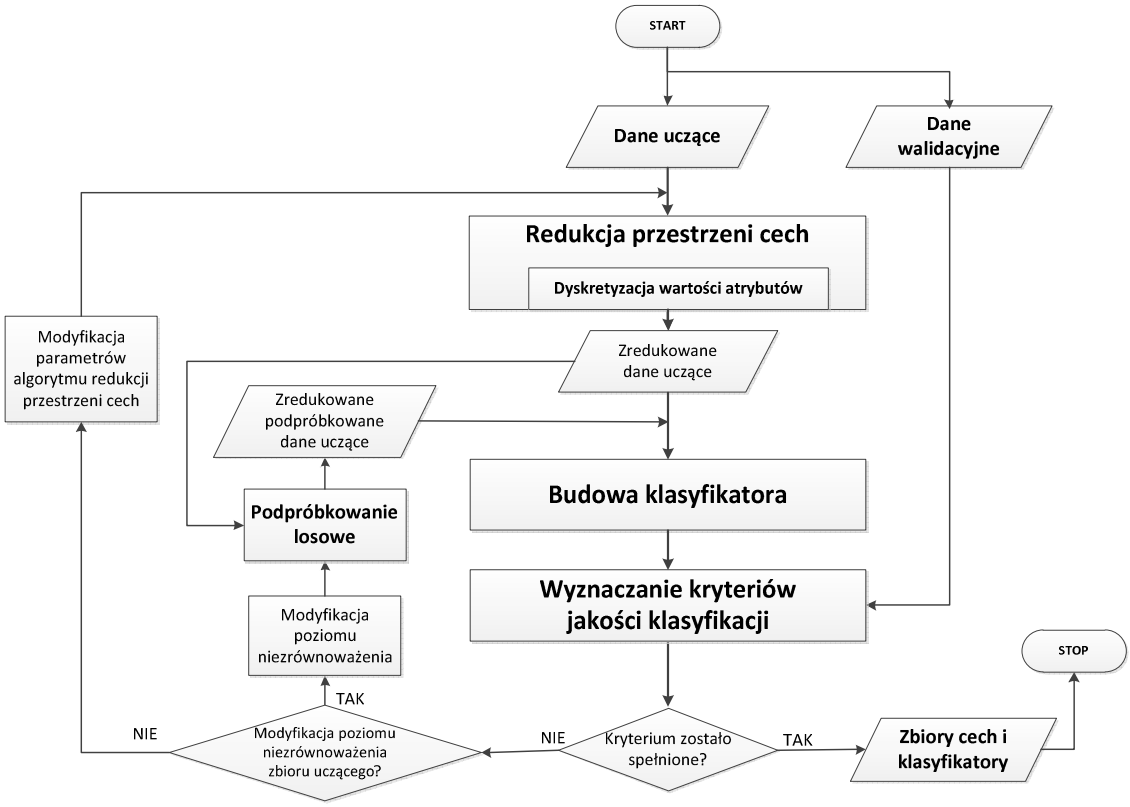
Ostatecznie uzyskano zbiór danych opisany przez 212 parametrów, nazywanych dalej cechami. Dziedziną każdej cechy jest zbiór liczb zmiennoprzecinkowych, z dokładnością zapisu do 10-ciu cyfr znaczących. Każdy z obiektów występujący w zbiorze przypisano do jednej z dwóch klas: komórki zdrowe (ozn. „0”) lub komórki nowotworowe (ozn. „1”). Ocena przynależności obiektów do odpowiedniej klasy została przeprowadzona przez eksperta, który w badanym zbiorze wskazał tylko 640 komórek nowotworowych. Jest to istotna uwaga dla analizowanego zbioru, gdyż poszukiwana klasa (komórki nowotworowe) stanowi niecałe 3% wszystkich dostępnych danych.

Analiza komórek w polu preparatu, pod kątem wyznaczenia wartości każdej z cech, jest kosztowna czasowo. Im mniejsza liczba cech, będących kryterium oceny komórki, tym proces skanowania wstępnego będzie szybszy. Dlatego poszukuje się metod pozwalających na wybór takich cech lub zbiorów cech, które pozwalają na trafną ocenę typu komórek. W załączniku B zestawiono cechy komórek wyznaczone w systemie „Metafer”.

4.3. Charakterystyka prowadzonych analiz

Ze względu na to, że liczba możliwych kombinacji wszystkich cech wynosi 2^{212}-1, czyli ok. 10^{64}, poddanie analizie wszystkich zbiorów jest niemożliwe. Z tego powodu zaproponowano iteracyjny proces określenia zbioru cech opisujących komórki (rys. 4.7).

W pierwszej kolejności iteracyjnego procesu ustala się zbiór danych uczących i walidacyjnych. W analizie komórek nowotworu pęcherza moczowego utworzony zbiór danych uczących składa się z 18370 obiektów (ozn. dane uczące), natomiast zbiór danych walidacyjnych zawiera 4592 obiekty (ozn. dane walidacyjne).



Rys. 4.7. Analiza zbiorów cech i dobór klasyfikatora

W kolejnym kroku rozpoczyna się redukcja przestrzeni cech. Poszukiwanie optymalnego zbioru cech przeprowadzono następującymi metodami:

- Selekcja metodą korelacji atrybut – atrybut (ozn. corr-AA),
- Selekcja metodą korelacji atrybut – klasa (ozn. corr-AC),
- Redukcja metodą analizy głównych składowych (ozn. PCA),
- Selekcja metodą zbiorów przybliżonych (ozn. RS).

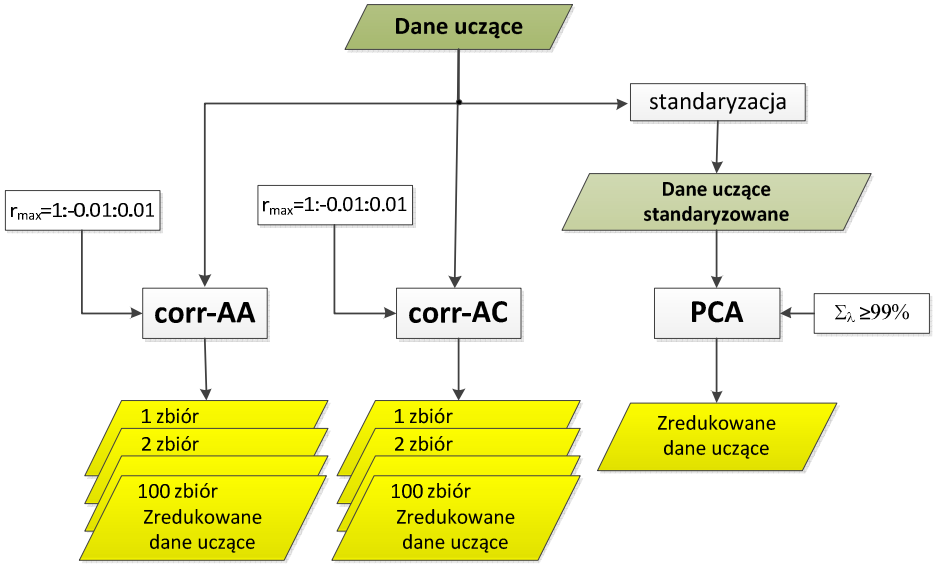
Selekcję metodą korelacji corr-AA, corr-AC oraz redukcję metodą PCA przeprowadzono na pełnym zbiorze cech (rys. 4.8). W przypadku metod corr-AA, corr-AC poszczególne zredukowane zbiory uczące, uzyskano zmieniając wartość graniczną współczynnika korelacji w zakresie <1,0.01> z krokiem -0.01. Uzyskano w ten sposób 100 zbiorów cech, które charakteryzują się liczebnością cech od 212 do 2 (rys. 4.10). W metodzie corr-AA selekcja jest przeprowadzana na podstawie wzajemnej korelacji poszczególnych cech. W metodzie selekcji corr-AC uwzględnia się dodatkowo korelację z atrybutem decyzyjnym. Zbiór danych wielowymiarowych jest w pierwszej kolejności porządkowany malejąco względem współczynnika korelacji między poszczególnymi cechami a atrybutem decyzyjnym. Następnie dla określonej wartości progowej współczynnika korelacji zostają usunięte cechy, których korelacja z poprzednimi cechami jest większa od wartości granicznej.

W przypadku redukcji metodą PCA uzyskano jeden zbiór cech. Rozmiar zbioru zależy od zadanego poziomu zmienności wyjaśnionej przez składowe główne. Redukcja zbioru cech metodą PCA, ze względu na jej właściwości, została poprzedzona procesem standaryzacji.

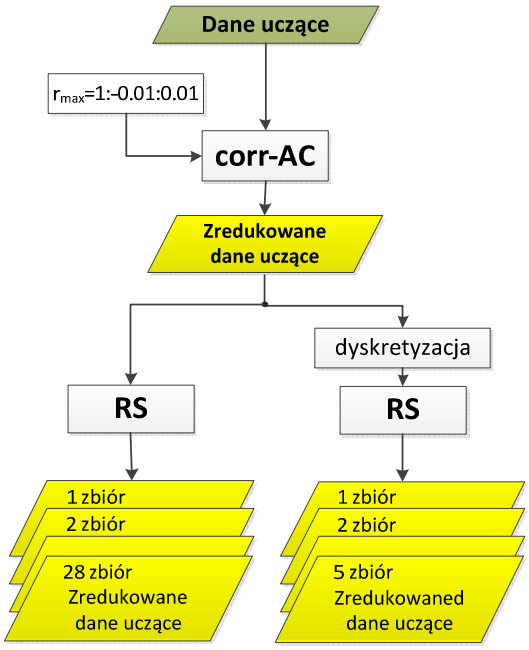
Algorytm redukcji przestrzeni cech RS (rys. 4.9) charakteryzują się wysoką złożonością obliczeniową. Z tego względu analiza danych uczących komórek nowotworu

pęcherza moczowego jest możliwa jedynie dla zbioru o liczbie atrybutów mniejszej od 20 (graniczna wartość współczynnika korelacji wynosi 0.45). Z tego powodu, dla algorytmu RS, zaproponowano metodę 2-etapową. Na pierwszym etapie redukcja zbioru cech jest przeprowadzana na podstawie wzajemnej korelacji poszczególnych cech oraz korelacji z atrybutem decyzyjnym (metoda corr-AC). Następnie dla wybranego zbioru przeprowadza się redukcję metodą RS. W przypadku metody RS etap redukcji może być poprzedzony dyskretyzacją wartości cech. W analizowanym problemie zastosowano następujące metody dyskretyzacji:

- dyskretyzacja równej szerokości o liczbie przedziałów: 5, 10, 20, 30, 40,50
- dyskretyzacja CAIM,
- dyskretyzacja CACC



Rys. 4.8. Redukcja przestrzeni cech metodami corr-AA, corr-AC, PCA.



Rys. 4.9. Redukcja przestrzeni cech metodą RS.

W kolejnym kroku procesu iteracyjnego (rys. 4.7), na podstawie danych uczących o zredukowanej liczbie cech, przeprowadza się proces nadzorowanego uczenia klasyfikatora (budowa klasyfikatora) dla różnych metod klasyfikacji. W przeprowadzonej analizie zastosowano pięć algorytmów klasyfikacji:

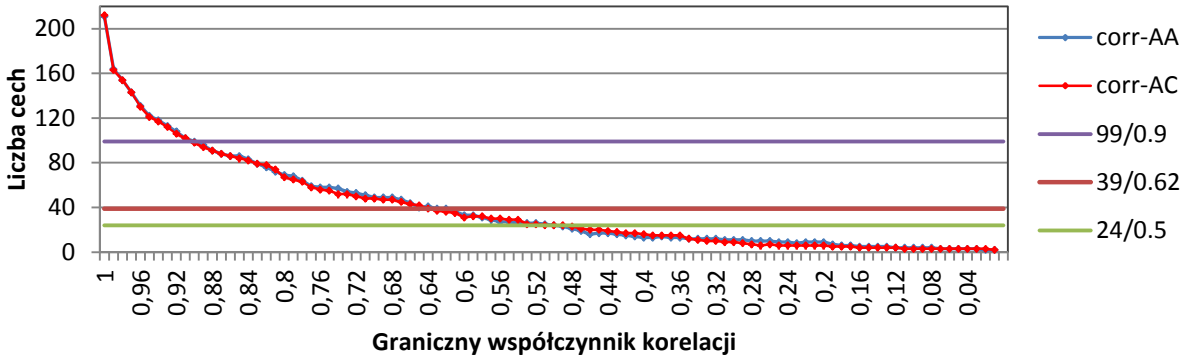
- Zbiory przybliżone (RS),
- Naiwny Klasyfikator Bayesa (NB),
- Liniowa analiza dyskryminacyjna (LDA),
- Kwadratowa analiza dyskryminacyjna (QDA),
- Drzewa decyzyjne (DT).

Następnie na podstawie zbioru walidacyjnego dokonuje się klasyfikacji komórek i wyznacza kryteria jakości klasyfikacji. Jeżeli wyznaczona jakość klasyfikacji nie spełnia oczekiwań, następuje modyfikacja zbioru cech lub zmiana poziomu niezrównoważenia i cały proces powtarza się.

4.4. Redukcja zbioru cech metodą corr-AA, corr-AC, PCA

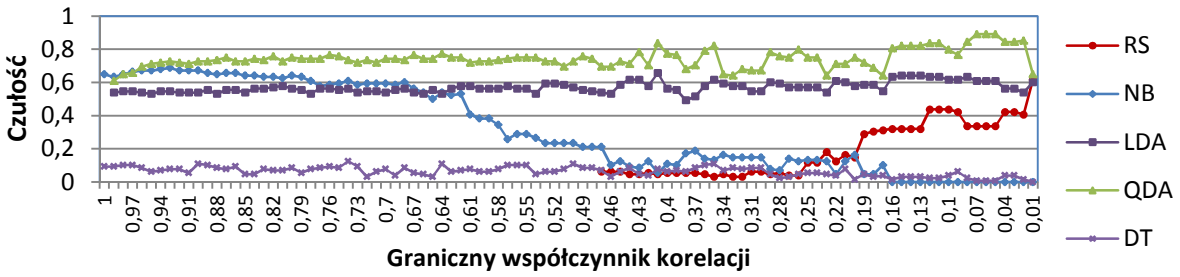
4.4.1. Redukcja zbioru cech metodą korelacji corr-AA

Redukcję zbioru cech metodą korelacji - corr-AA przeprowadzono na zbiorze 212 cech. Wartość graniczną współczynnika korelacji, względem którego eliminowano poszczególne cechy, zmieniano co 0.01 w zakresie wartości bezwzględnych od 1 do 0.01. Po każdym kroku selekcji przeprowadzano klasyfikację. Jakość klasyfikacji oceniano poszukując maksimum funkcji $FMaxSen$ zdefiniowanej równaniem 2.50. W celu wyznaczenia ekstremum globalnego obliczenia wykonano dla pełnego zakresu wartości granicznej współczynnika korelacji. Na rys. 4.10 przedstawiono liczebność zbiorów cech względem granicznego współczynnika korelacji.

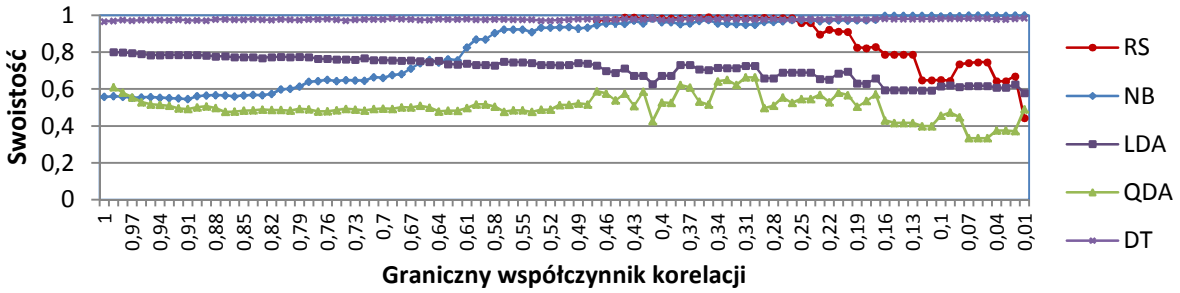


Rys. 4.10. Zależność liczby cech od współczynnika korelacji w metodzie selekcji corr-AA i corr-AC

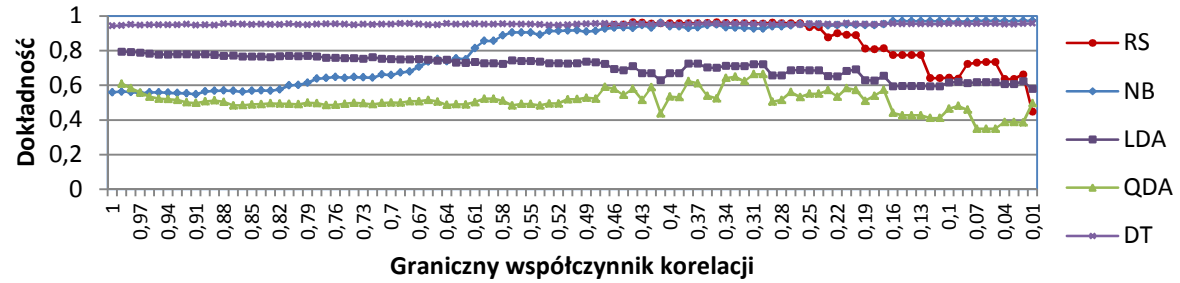
Oceniając każdy zbiór cech uzyskano przestrzeń możliwych wyników klasyfikacji. Wyniki klasyfikacji przedstawiono na rysunkach 4.11-4.14.



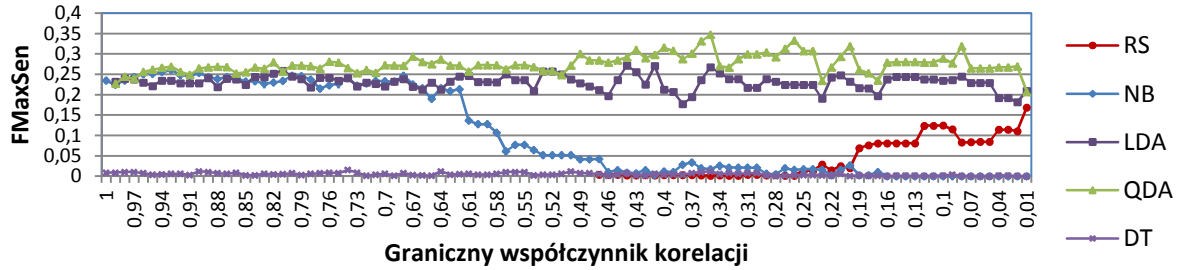
Rys. 4.11. Charakterystyka czułości klasyfikatorów dla redukcji cech metodą corr-AA



Rys. 4.12. Charakterystyka swoistości klasyfikatorów dla redukcji cech metodą corr-AA



Rys. 4.13. Charakterystyka dokładności klasyfikatorów dla redukcji cech metodą corr-AA



Rys. 4.14. Charakterystyka funkcji FMaxSen dla redukcji cech metodą corr-AA

Analizując wykres czułości klasyfikacji widać, że dla pełnego zbioru cech (graniczny współczynnik korelacji $r = 1$), najlepsze właściwości predykcyjne otrzymano dla klasyfikatora NB oraz QDA. Czułość tych klasyfikatorów wynosi ok. 60%. Niewiele gorszą czułość, wynoszącą 53%, odnotowano dla metody LDA. Klasyfikator LDA różni się jednak wartością współczynnika swoistości. Wynosi ona 80%, podczas gdy dla klasyfikatorów NB i QDA – ok.60%. Pomimo występującej różnicy wszystkie trzy

klasyfikatory posiadają zbliżoną wartość funkcji celu. Wartość ta waha się w granicach 0.22-0.23, przy czym wartość większą uzyskano dla metody klasyfikacji LDA.

Na charakterystykach 4.11-4.14 zaobserwowano, iż klasyfikacja metodami LDA i QDA osiąga wysokie wartości wskaźników niezależnie od liczebności zbioru cech. Z kolei jakość klasyfikacji metodą NB jest maksymalna tylko przy selekcji cech powyżej współczynnika korelacji 0.62. Taka wartość współczynnika odpowiada zbiorowi cech zawierającemu 39 elementów. Wraz ze zmniejszaniem się liczby cech, czułość klasyfikatora maleje. Spadek czułości związany jest ze wzrostem swoistości. Wzrost współczynnika swoistości oznacza, że klasyfikator wykrywa coraz więcej komórek zdrowych i równocześnie popełnia większy błąd w predykcji komórek nowotworowych.

Weryfikację wyznaczonych zbiorów cech przeprowadzono także algorytmami opartymi na regułach: DT i RS. Ze względu na ograniczenia związane ze złożonością pamięciową, zastosowanie metody RS jest możliwe dopiero przy granicznej wartości współczynnika korelacji r=0.47. Zbiór cech uzyskany dla takiej wartości progowej zawiera 19 elementów, a czułość wynosi 6%. Czułość dla klasyfikacji metodą RS zwiększa się wraz ze zmniejszaniem liczby cech. Zależność tą widać także na charakterystyce funkcji $FMaxSen$. Na takie zachowanie klasyfikatora RS ma wpływ liczba reguł decyzyjnych modelu klasyfikującego. Im większa jest liczba cech, tym większa jest liczba możliwych reguł. Ponieważ liczba obiektów z klasy komórek zdrowych jest zdecydowanie większa, to większe jest prawdopodobieństwo wytypowania klasy komórek zdrowych w przestrzeni reguł decyzyjnych.

Klasyfikator DT umożliwia klasyfikację także na pełnym zbiorze cech. Niestety jakość przeprowadzonej klasyfikacji jest bardzo niska. Dla zadanego zbioru cech uzyskano 9% czułość, przy 97% swoistości. Mała wartość współczynnika czułości jest również obserwowalna na wykresie funkcji celu, gdzie wartość funkcji zbliżona jest do zera. Dla klasyfikatora DT nie obserwuje się żadnej poprawy jakości klasyfikacji przy zmniejszaniu zbioru cech. Amplituda zmian współczynnika czułości, w zależności od liczby cech, nie przekracza 12%.

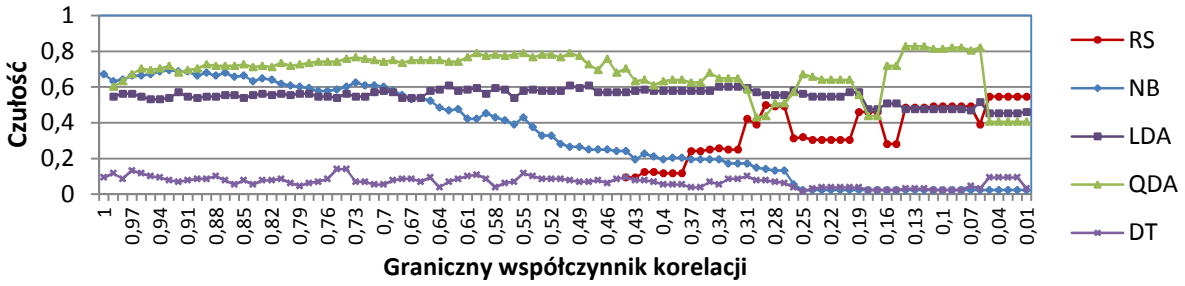
Na charakterystyce współczynnika dokładności (rys. 4.13) obserwuje się liczbę poprawnych klasyfikacji w stosunku do całego zbioru danych. Dokładność nie odzwierciedla rozkładu liczby obiektów w poszczególnych klasach. O problemie tym sygnalizowano w rozdziale 2.7.2 (tab. 2.4). Porównując charakterystyki swoistości (rys. 4.12) i dokładności (rys. 4.13) można zauważyć, że są one do siebie bardzo zbliżone. Wynika to z proporcji liczby obiektów należących do różnych klas. Komórki zdrowe stanowią prawie 97% przypadków. Dlatego, gdy żadna komórka nowotworowa nie zostanie wykryta, dokładność klasyfikacji będzie wynosiła prawie 97%.

Przeprowadzone analizy pokazują, że zmniejszenie liczby cech nie zawsze musi prowadzić do pogorszenia możliwości klasyfikacji zbioru. Zależność tą widać na wykresie funkcji $FMaxSen$ (rys. 4.14), gdzie klasyfikator QDA osiąga maksimum globalne przy współczynniku korelacji – 0.36. Klasyfikator ten charakteryzuje się 82% czułością i 51% swoistością. Osiągnięta wartość funkcji $FMaxSen$, wynosząca 0.35, stanowi maksimum globalne dla redukcji cech metodą corr-AA.

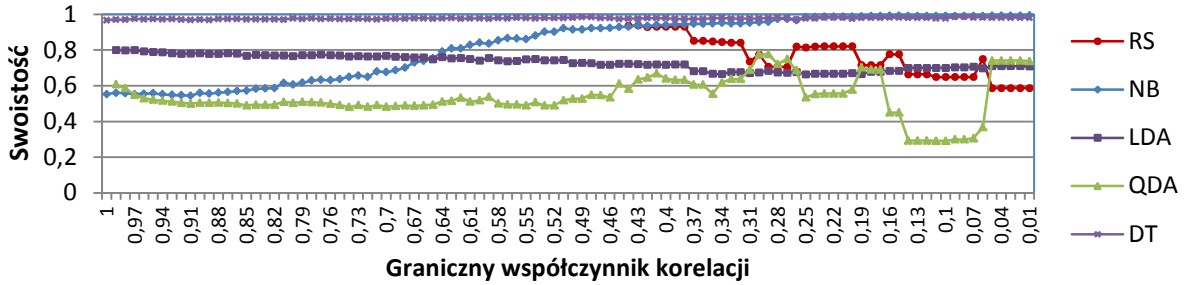
4.4.2. Redukcja zboru cech metodą korelacji corr-AC

W metodzie selekcji corr-AC wartość progową współczynnika korelacji, względem którego eliminowano poszczególne cechy, zmieniano co 0.01 w zakresie wartości bezwzględnych od 1 do 0.01. Otrzymano w ten sposób różne liczebności zbiorów cech. Charakterystykę zależności liczby cech od wartości progowej współczynnika korelacji zamieszczono na rys. 4.10. Charakterystyka pokrywa się w dużym stopniu

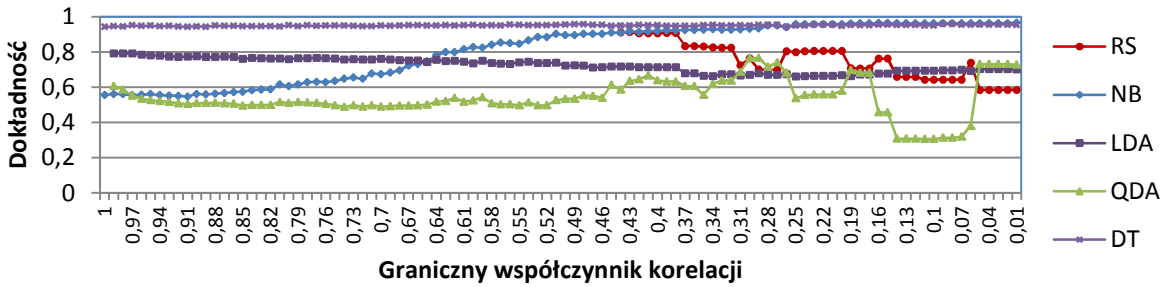
z charakterystyką wykreśloną dla metody corr-AA. Utworzone zbiory cech poddano ocenie poprzez klasyfikację. Wyniki miar jakości klasyfikacji zamieszczono na rysunkach 4.15-4.18.



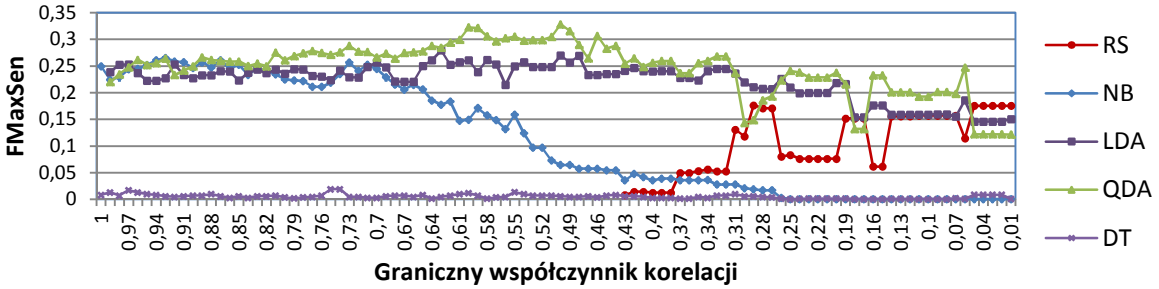
Rys. 4.15. Charakterystyka czułości klasyfikatorów dla redukcji cech metodą corr-AC



Rys. 4.16. Charakterystyka swoistości klasyfikatorów dla redukcji cech metodą corr-AC



Rys. 4.17. Charakterystyka dokładności klasyfikatorów dla redukcji cech metodą corr-AC



Rys. 4.18. Charakterystyka funkcji FMaxSen dla redukcji cech metodą corr-AC

Najlepsze wskaźniki jakości otrzymano dla metod QDA oraz LDA. Maksymalne wartości funkcji $FMaxSen$ uzyskano dla klasyfikatora QDA dla wartości progowej współczynnika korelacji 0.5 oraz 0.6, co odpowiada zbiorom cech o liczebności

24 oraz 31. Wartość funkcji celu dla klasyfikatora QDA wynosiła 0.328 przy czułości ok. 80% i swoistości 52% .

Maksymalna czułość dla klasyfikatora LDA nie przekraczała 60%. Najwyższą uzyskana wartość funkcji celu wyniosła 0.284 dla wartości progowej współczynnika korelacji 0.63.

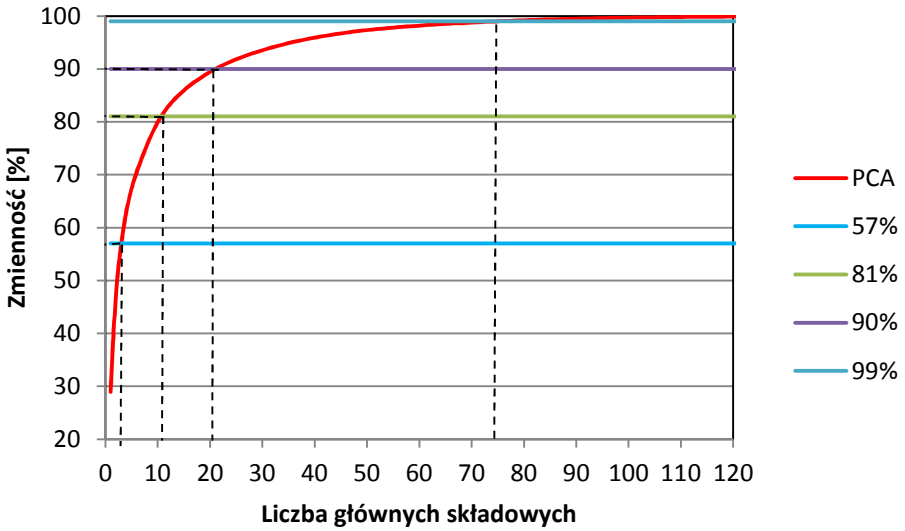
Dla zbiorów cech o dużej liczebności elementów również wysokimi wartościami miar jakości klasyfikacji charakteryzuje się klasyfikacja metodą NB. Wykres cechuje się podobnym trendem jak klasyfikator NB dla selekcji corr-AA. Wraz ze zmniejszaniem się liczby cech maleje czułość klasyfikacji.

Metoda corr-AC okazała się skuteczniejsza dla klasyfikatora RS. Czułość klasyfikacji rośnie wraz ze zmniejszaniem się liczby atrybutów w zbiorze. Czułość jest ok. 12% wyższa w stosunku do metody corr-AA i wynosi 54%. Wysoką wartość funkcji $FMaxSen$, wynoszącą 0.17 osiągnięto dla zbioru 3 cech. Wartość funkcji $FMaxSen$ była w tym przypadku wyższa od klasyfikatorów LDA i QDA .

Najgorszymi właściwościami klasyfikującymi charakteryzuje się klasyfikator DT. Maksymalna czułość klasyfikacji wynosiła ok. 15%. Odpowiada to znikomej wartości funkcji $FMaxSen$, wynoszącej 0.019.

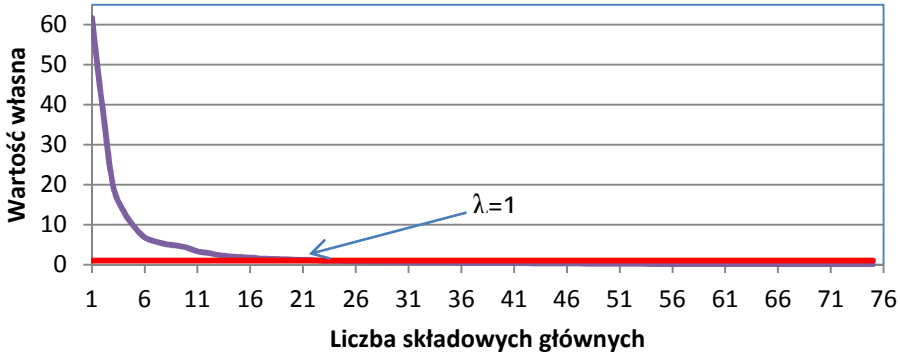
4.4.3. Redukcja zbioru cech metodą analizy głównych składowych (PCA)

Zastosowanie metody PCA prowadzi do transformacji przestrzeni cech, w wyniku której uzyskuje się nowy układ współrzędnych. Osie nowego układu odpowiadają nowym cechom - składowym głównym. Każda z nowoutworzonych cech wyjaśnia pewną część zmienności zbioru oryginalnego. Największą część zmienności reprezentuje pierwsza składowa. Zależność tę przedstawiono na rysunku 4.19. Krzywa PCA, odpowiada skumulowanej zmienności dla zbioru składowych głównych. 57% zmienności uzyskuje się dla trzech pierwszych składowych. Oznacza to, że jeżeli taki poziom zmienności byłby satysfakcjonujący, to pierwotny zbiór cech można zredukować do 3 elementów. Do zachowania zmienności o wartości 81% wystarczy analiza jedenastu pierwszych składowych głównych, a przy zmienności 90% - zaledwie dwudziestu dwóch składowych. Zachowanie zmienności oryginalnego zbioru na poziomie 99% wymaga rozpatrzenia 75 pierwszych głównych składowych.



Rys. 4.19. Skumulowany wykres zmienności wyjaśnianej przez składowe główne

Liczbę składowych głównych można także określić korzystając z kryterium wartości własnej (rys. 4.20) przedstawionego w rozdziale 2.4.1. Według tego kryterium, każda z wybranych składowych musi posiadać wartość własną przynajmniej równą 1. W analizowanym zbiorze powyższy warunek spełniają dwadzieścia dwie składowe główne.



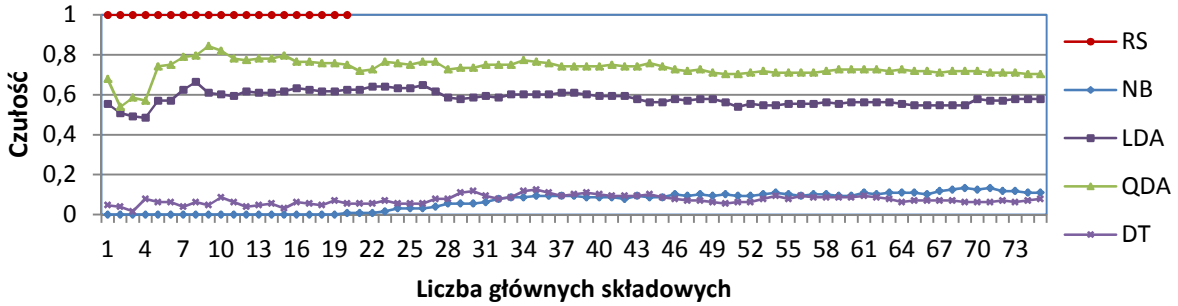
Rys. 4.20. Charakterystyka kryterium wartości własnej

W badanym zbiorze danych występuje bardzo mała liczba komórek nowotworowych. Z tego powodu do dalszych analiz wykorzystano pierwsze kryterium wyboru liczby składowych głównych. Aby zapewnić zmienność danych na poziomie 99%, klasyfikację przeprowadzono analizując 75 możliwych zbiorów. Liczba elementów w zbiorach odpowiada liczbie kolejnych składowych głównych.

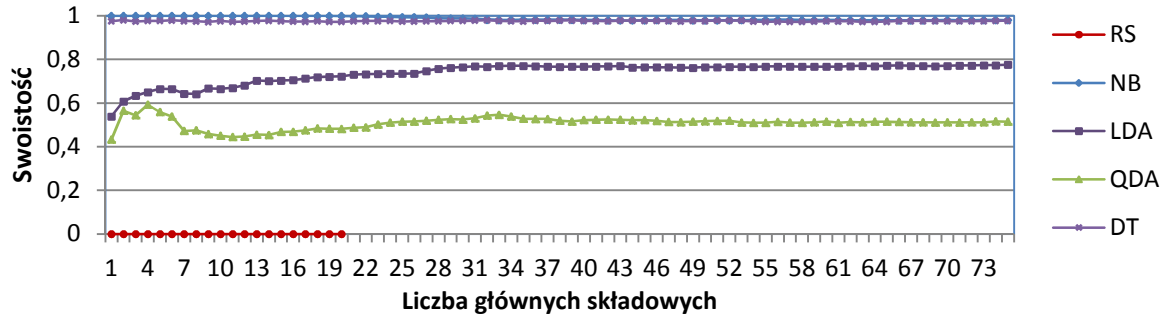
Charakterystyki otrzymanych wyników klasyfikacji przedstawiono na wykresach 4.21-4.24. Najlepsze wskaźniki jakości uzyskano dla klasyfikacji metodami LDA i QDA. Na rys. 4.21 przedstawiającym charakterystykę czułości widać wahania wskaźnika dla zbiorów cech zawierających od 1 do 5 składowych głównych. W tym zakresie parametrów, spadek współczynnika czułości powoduje wzrost współczynnika swoistości. Przy dodawaniu kolejnych składowych obserwuje się lokalny wzrost współczynnika czułości. W obszarze tym uzyskano maksimum globalne funkcji celu dla redukcji cech metodą PCA. Zbiór cech zawiera dziewięć pierwszych składowych głównych. Odpowiada to uzyskaniu 78% zmienności zbioru. Maksymalną wartość funkcji celu, wynoszącą 0.32, osiągnięto przy zastosowaniu klasyfikacji metodą QDA.

Transformacja układu współrzędnych nie przyniosła żadnych korzyści dla klasyfikatora RS. Chociaż klasyfikator ten miał czułość na poziomie 100%, to jego swoistość wyniosła 0. Ze względu na złożoność obliczeniową analizę metodą RS przeprowadzono dla do 20-stu składowych głównych.

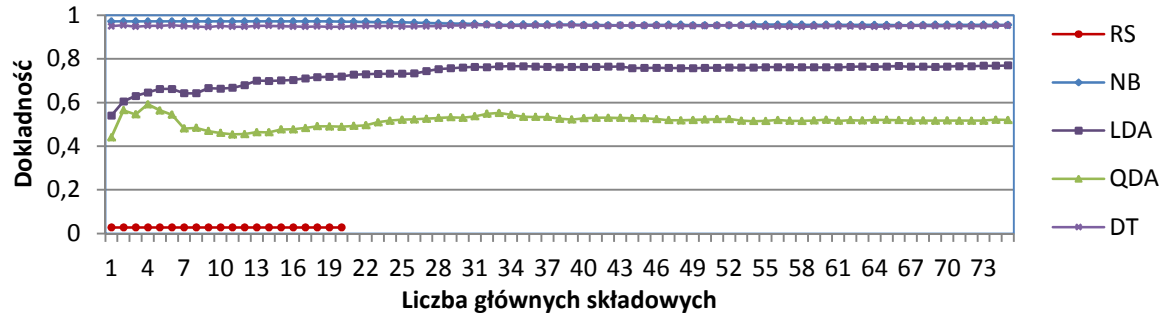
Zastosowanie metody NB i DT nie poprawiło wyników jakości klasyfikacji. Pomimo, iż wraz ze wzrostem liczby składowych głównych obserwuje się wzrost czułości klasyfikatorów, to swoistość uzyskana dla każdej z metod była zbliżona do 97%. Wartość funkcji celu wyznaczona dla klasyfikacji NB i DT była zbliżona do wartości 0.01.



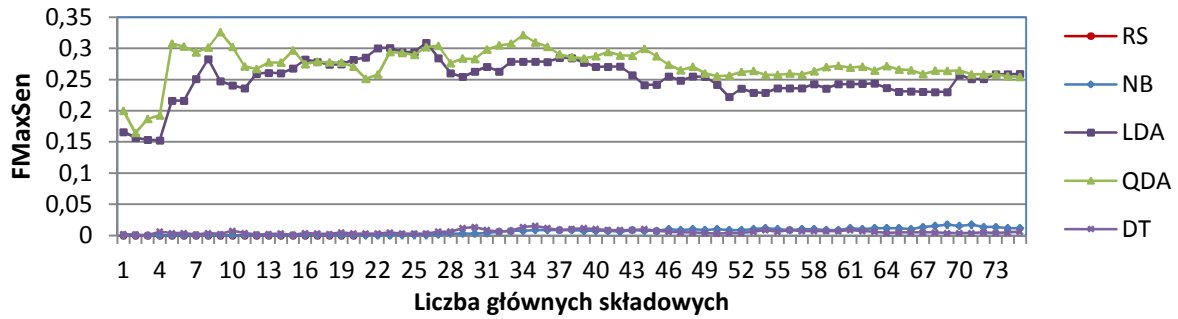
Rys. 4.21. Charakterystyka czułości klasyfikatorów dla redukcji cech metodą PCA



Rys. 4.22. Charakterystyka swoistości klasyfikatorów dla redukcji cech metodą PCA



Rys. 4.23. Charakterystyka dokładności klasyfikatorów dla redukcji cech metodą PCA



Rys. 4.24. Charakterystyka funkcji FMaxSen klasyfikatorów dla redukcji cech metodą PCA

4.5. Redukcja zbioru cech metodą RS

Analizując wyniki redukcji zbioru cech metodami corr-AA, corr-AC widać, że metoda klasyfikacji RS jest skuteczna jedynie dla małej liczby cech. Z rys. 4.14 i 4.18 wynika, że dla granicznej wartości współczynnika korelacji poniżej 0.4 (co odpowiada liczbie cech równej 16), funkcja $FMaxSen$ przyjmuje maksymalne wartości ok. 0.15. Ponadto, dla małej liczby cech, także algorytmy QDA i LDA zachowują swoje właściwości klasyfikacyjne. Dla tych algorytmów zaobserwowano spadek FMaxSen z ok. 0.33 dla liczby 24 cech do wartości ok. 0.25 dla 3 cech.

Z tego powodu postanowiono poddać analizie dalszą redukcję zbioru cech z zastosowaniem algorytmu RS. Selekcja cech metodą RS pozwala na znalezienie wszystkich podzbiorów cech, które dzielą przestrzeń obiektów na zbiory elementarne w taki sam sposób, jak redukowany zbiór cech. Ze względu na wysoką złożoność obliczeniową metody RS analizę parametrów morfometrycznych komórek zrealizowano na zbiorze o zredukowanej liczbie cech będącym wynikiem selekcji corr-AC.

Wykaz cech wybranych do analiz zamieszczono w tabeli 4.1. Cechy uporządkowano w kierunku malejącym względem korelacji z atrybutem decyzyjnym. Cecha oznaczona numerem 1 jest najbardziej skorelowana, a oznaczona numerem 16 jest najmniej skorelowana. Kolumna nr 2 w tabeli 4.1. odpowiada numerowi cechy w inicjalnym zbiorze danych (Załącznik B).

Tabela 4.1. Zbiór cech wybrany do redukcji metodą RS

Nr cechy w zestawieniu	**Nr cechy w oryginalnym zbiorze**	**Nazwa cechy**
1	40	Maximum Intensity, Relative Spot Area 1/X (X=320)
2	210	Standard Deviation of Object Intensity x=90, y=300
3	24	Minimum Radius of Contour in 1/10 µm
4	104	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z, X=0.3, Y=-1, Z=-1
5	193	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels) x=80, y=300
6	80	Center Distance Moment, Distance Exponent X, Intensity Exponent Y, X=2, Y=1.5
7	188	Standard Deviation of Object Intensity
8	33	Total Relative Concavity Area (0..1)
9	27	Aspect Ratio of Cell (Short Axis / Long Axis, 0..1)
10	13	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels) x=90, y=300, z=2
11	149	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels), X=40, Y=300
12	128	Mean of Relative Object Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels) X=20, Y=300 Z=0
13	90	Radial Distance Moment, Distance Exponent X, Intensity Exponent Y (X=1, y=2)
14	173	Standard Deviation of Relative Object Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels) x=60, y=300
15	214	Standard Deviation of Distance to other Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels) x=90, y=300
16	192	Standard Deviation of Distance to other Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels) x=70, y=300

4.5.1. Wybór zbioru cech metodą zbiorów przybliżonych (RS)

Selekcja cech metodą zbiorów przybliżonych pozwala na znalezienie różnych grup cech, które opisują system decyzyjny w stopniu porównywalnym do pełnego zbioru cech. Opis algorytmu poszukiwania reduktów względnych realizującego procedurę selekcji przedstawiono w rozdziale 3.4.9. W wyniku zastosowania algorytmu utworzono 28 zbiorów cech. Wykaz utworzonych zbiorów wraz z numerami cech wchodzących w skład zbioru zamieszczono w tabeli 4.2.

Tabela 4.2. Zredukowane zbiory cech wyznaczone metodą poszukiwania reduktów względnych

Nr zbioru	Lista cech
1	9
2	6
3	8 13
4	1 13
5	4 8
6	1 4
7	1 12
8	12 13
9	4 12
10	13 14
11	4 14
12	7 13
13	4 7
14	13 16

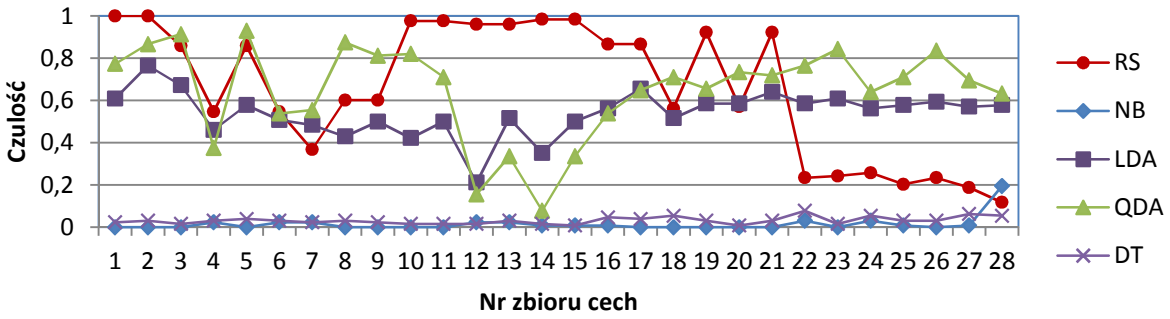
Nr zbioru	Lista cech
15	4 16
16	3 11 13
17	3 4 11
18	5 10 13
19	3 5 13
20	4 5 10
21	3 4 5
22	1 3 8 10 11
23	3 8 10 12 14
24	1 3 10 11 14
25	3 10 11 12 14
26	3 5 8 10 12
27	3 5 10 11 12

Do oceny każdego z wyznaczonych zbiorów zastosowano różne metody klasyfikacji. Wyniki zamieszczono na rysunkach 4.25 – 4.28. Zbiory posortowano w kolejności zwiększającej się liczby cech. Zbiór oznaczony numerem 28 odpowiada pełnemu zbiorowi cech, zawierającemu 16 elementów.

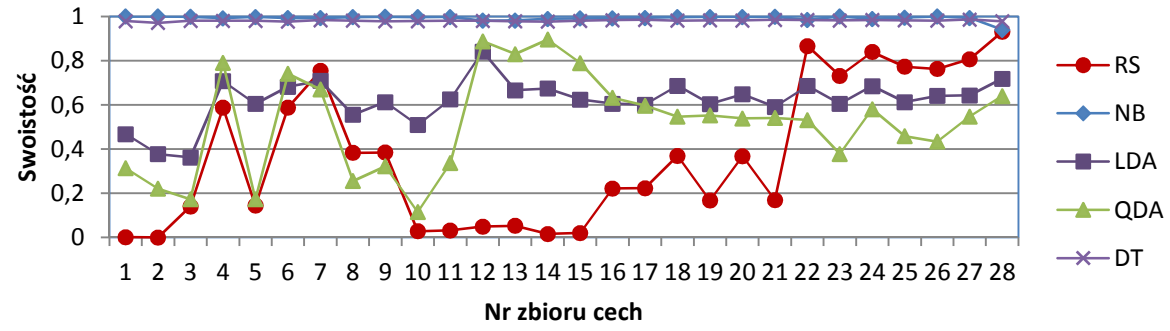
Najlepszymi właściwościami charakteryzują się zbiory zawierające 5 cech. Wartości funkcji celu, uzyskane metodami klasyfikacji LDA i QDA, były w tym zakresie największe. Maksymalną wartość funkcji $FMaxSen$ uzyskano dla zbioru numer 22 przy klasyfikacji metodą QDA. Wartość ta wynosi 0.31 i odpowiada czułości wynoszącej 76% oraz swoistości 53%.

Najgorszymi właściwościami, niezależnie od klasyfikatora, charakteryzował się zbiór cech o numerze 12. Zdolność klasyfikacji na tym zbiorze osiągała wysoką swoistość przy czułości wynoszącej niecałe 20%.

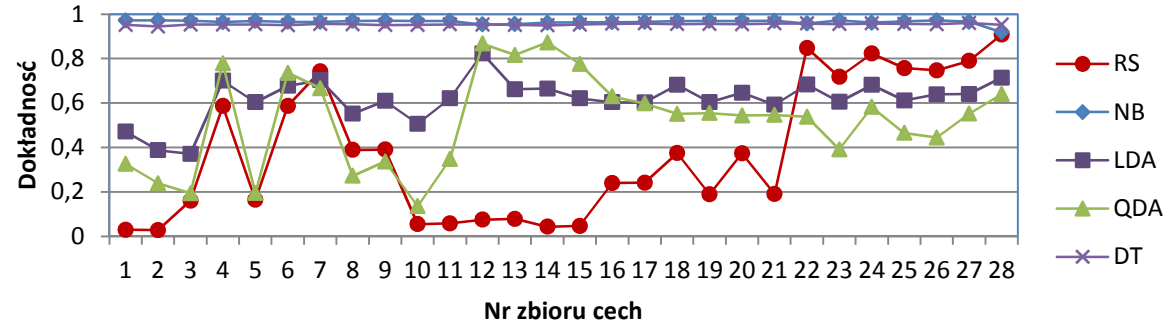
Dla klasyfikatora RS najlepsze wskaźniki otrzymano dla zbiorów trzech cech. Największą wartość funkcji celu, wynoszącą ok. 0.17, osiągnięto dla zbiorów cech o numerach 16 i 17.



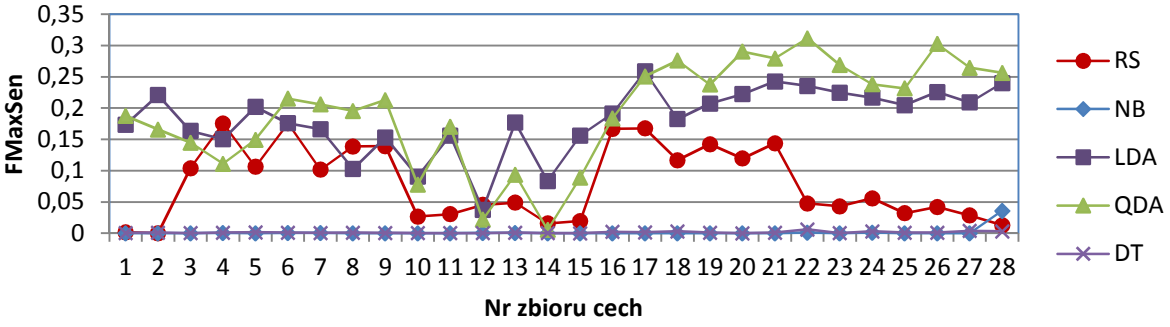
Rys. 4.25. Charakterystyka czułości klasyfikatorów dla redukcji cech metodą RS



Rys. 4.26. Charakterystyka swoistości klasyfikatorów dla redukcji cech metodą RS



Rys. 4.27. Charakterystyka dokładności klasyfikatorów dla redukcji cech metodą RS



Rys. 4.28. Charakterystyka funkcji celu klasyfikatorów dla redukcji cech metodą RS

4.5.2. Dyskretyzacja wartości cech

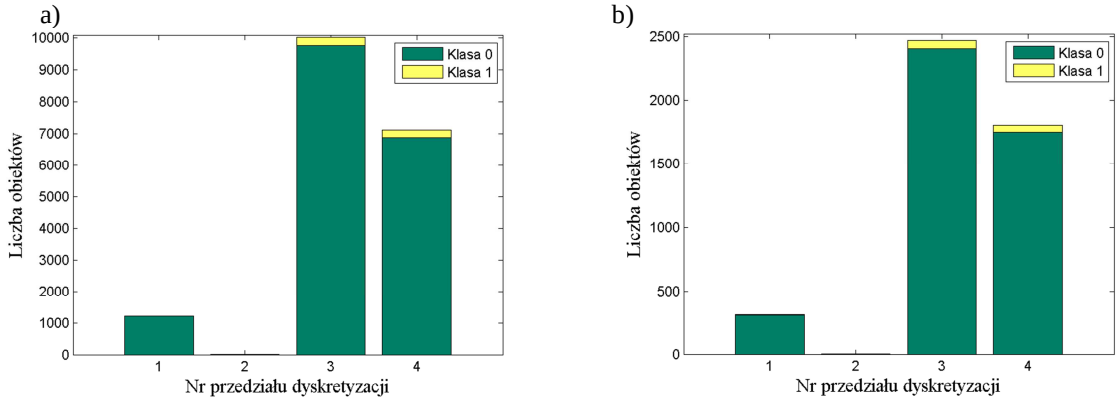
Zbiory przybliżone umożliwiają wyznaczenie takich cech lub zbiorów cech w systemach decyzyjnych, dla których system będzie dobrze określony. Gdy cechy posiadają wartości typu rzeczywistego, to w zbiorze tym mogą wystąpić pojedyncze cechy pozwalające na rozróżnienie klas. Nie oznacza to jednak, że klasyfikator zbudowany na podstawie takiej cechy będzie funkcjonował prawidłowo także dla danych walidacyjnych. Aby zapobiec sytuacji, w której redukty składają się z pojedynczych cech, wprowadzono dyskretyzację. Dyskretyzacja ogranicza liczbę możliwych wartości cechy, a tym samym wymusza poszukiwanie zestawów cech.

Analizowano wpływ dyskretyzacji na możliwości selekcji atrybutów metodą zbiorów przybliżonych. W tym celu wykorzystano trzy metody dyskretyzacji:

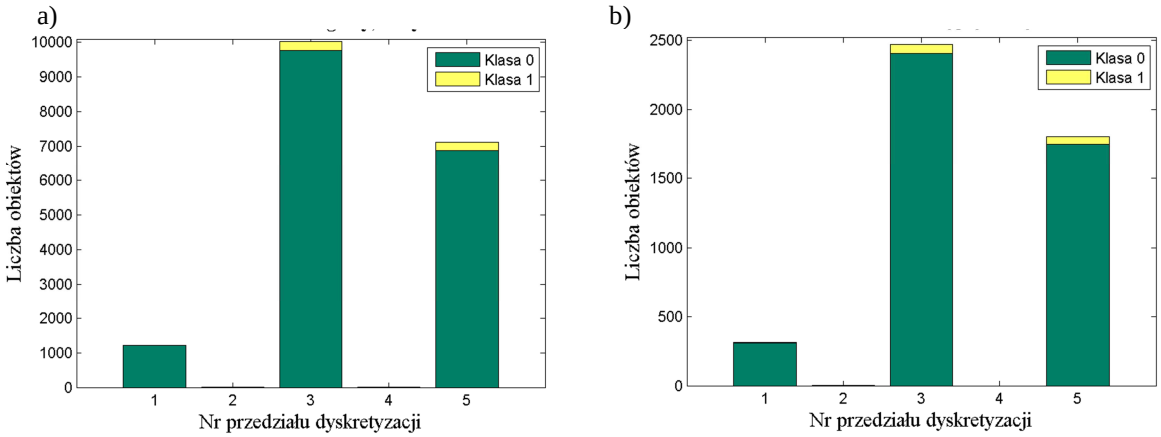
- dyskretyzacja równej szerokości (EWD) o liczbie przedziałów: 5, 10, 20, 30, 40, 50;
- dyskretyzacja CAIM;
- dyskretyzacja CACC.

Granice dyskretyzacji każdej z cech wyznaczono na podstawie zbioru danych uczących. Zastosowanie osobnych granic dyskretyzacji, dla próby uczącej i walidacyjnej, może doprowadzić do sytuacji, w której obiekty, posiadające takie same wartości zarówno dla zbioru uczącego jak i walidacyjnego, zostaną przypisane do różnych przedziałów dyskretyzacji. W przypadku klasyfikatorów działających na podstawie reguł decyzyjnych, błędny wybór przedziału może wpłynąć na wybór błędnej reguły decyzyjnej, co ostatecznie może doprowadzić do błędnej klasyfikacji.

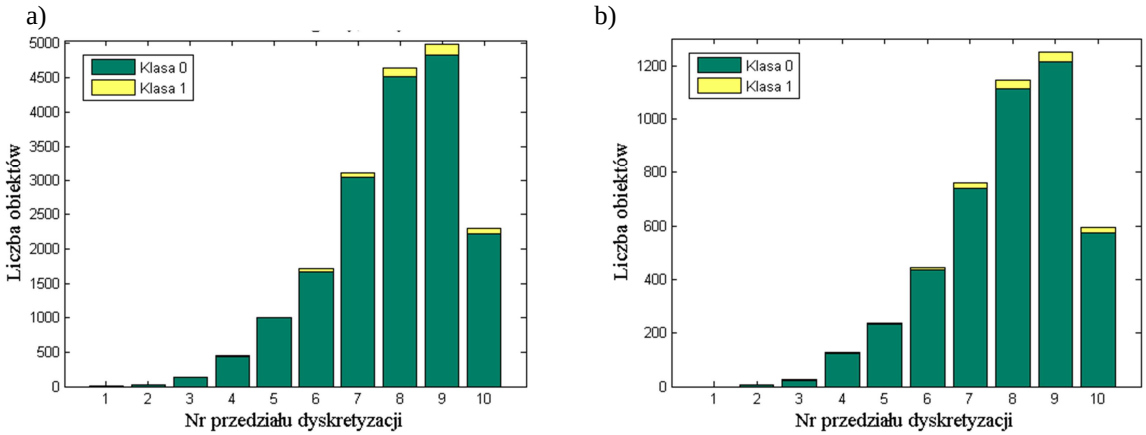
Na rysunkach 4.29-4.32 zamieszczono wyniki dyskretyzacji CAIM, CACC, EWD dla współczynnika kształtu komórki (cecha nr 27). Na zamieszczonych histogramach widać, że rozkłady wartości w próbie uczącej i walidacyjnej są podobne. Komórki określone przez eksperta jako komórki zdrowe oznaczono kolorem ciemnym (klasa 0), natomiast komórki określone jako nowotworowe oznaczono kolorem jasnym (klasa 1). W przypadku metody EWD, przy wyznaczaniu przedziałów dyskretyzacji, nie brano pod uwagę przynależności obiektu do klasy. Z kolei w metodach CAIM i CACC uwzględniono typ klasy zdefiniowanej w zbiorze uczącym. W metodzie EWD komórki nowotworowe znajdują się w każdym z przedziałów dyskretyzacji, przy czym zdecydowana większość znajduje się w przedziałach o największych liczebnościach obiektów. W metodach CAIM, CACC rozkład klas jest nierównomierny. Utworzone przedziały dyskretyzacji nie wyodrębniły w sposób zdecydowany komórek nowotworowych.



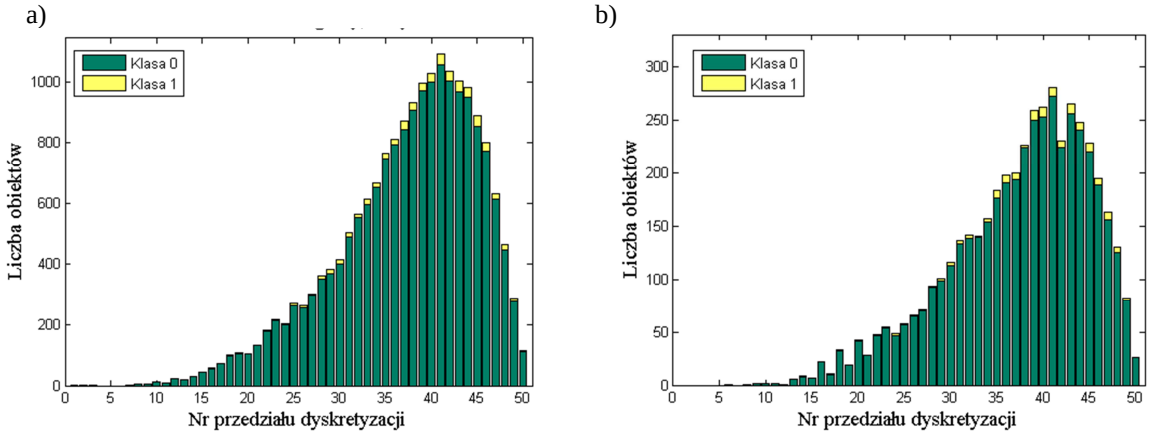
Rys. 4.29. Histogram dyskretyzacji metodą CAIM na przykładzie cechy nr 27 - Współczynnik kształtu komórki: a) zbiór uczący b) zbiór walidacyjny



Rys. 4.30. Histogram dyskretyzacji metodą CACC na przykładzie cechy nr 27 - Współczynnik kształtu komórki: a) zbiór uczący b) zbiór walidacyjny



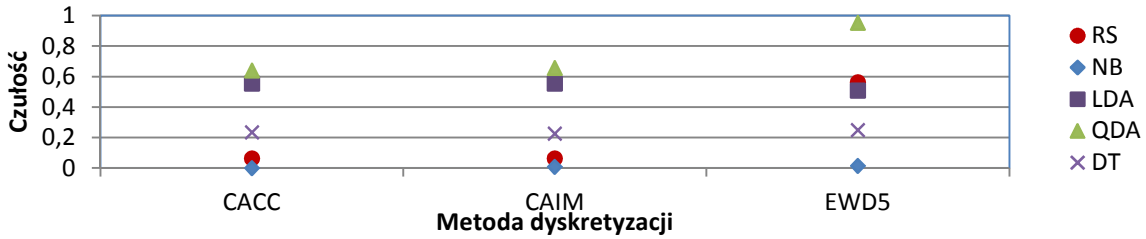
Rys. 4.31. Histogram dyskretyzacji metodą EWD o 10 przedziałach na przykładzie cechy nr 27 - Współczynnik kształtu komórki : a) zbiór uczący b) zbiór walidacyjny



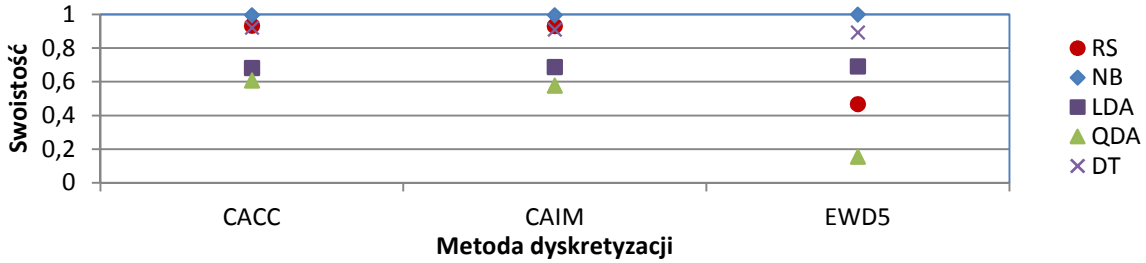
Rys. 4.32. Histogram dyskretyzacji metodą EWD o 50 przedziałach na przykładzie cechy nr 27 - Współczynnik kształtu komórki: a) zbiór uczący b) zbiór walidacyjny

4.5.3. Wybór zbioru cech metodą RS dla dyskretyzacji EWD5, CAIM, CACC

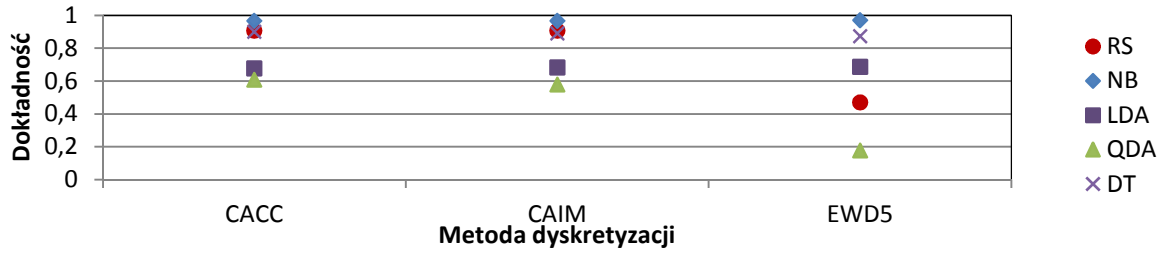
Zastosowanie dyskretyzacji EWD5, CAIM oraz CACC prowadzi do utworzenia cech charakteryzujących się małą liczbą możliwych wartości. Zastosowanie selekcji cech metodą zbiorów przybliżonych na wartościach dyskretnych pokazuje, że dla danego systemu decyzyjnego nie jest możliwa redukcja zbioru cech. Wyniki klasyfikacji przedstawione na rys. 4.33-4.36 dotyczą klasyfikacji na podstawie zbioru 16 cech.



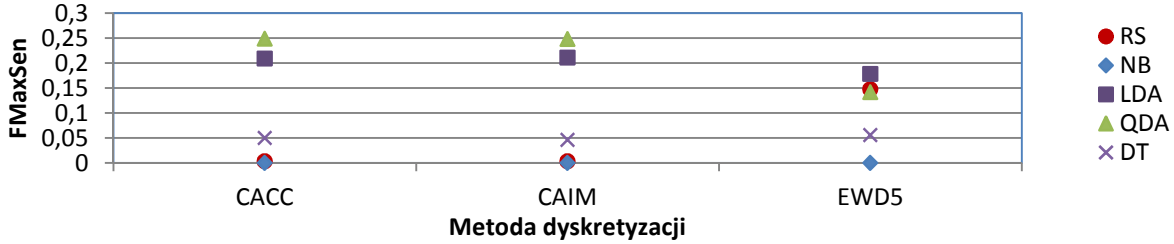
Rys. 4.33. Czułość klasyfikatorów dla cech dyskretyzowanych metodami: EWD5, CACC, CAIM



Rys. 4.34. Swoistość klasyfikatorów dla cech dyskretyzowanych metodami: EWD5, CACC, CAIM



Rys. 4.35. Dokładność klasyfikatorów dla cech dyskretyzowanych metodami: EWD5, CACC, CAIM



Rys. 4.36. Wartość funkcji FMaxSen klasyfikatorów dla cech dyskretyzowanych metodami: EWD5, CACC, CAIM

Najwyższą wartość funkcji celu, wynoszącą 0.248 uzyskano dla danych dyskretyzowanych metodą CACC i CAIM przy klasyfikacji LDA i QDA. Uzyskana wartość jest zbliżona do funkcji celu przy klasyfikacji danych pierwotnych. Czułość

uzyskana dla dyskretyzacji CACC wyniosła 64% przy swoistości 60%. Niewiele większą czułością – 66%, charakteryzuje się klasyfikacja danych dyskretyzowanych metodą CAIM. Uzyskana swoistość tego modelu była niższa (58%), jednak nie wpłynęło to na obniżenie wartości funkcji celu.

Wyniki klasyfikacji danych dyskretyzowanych metodą równej szerokości (EWD5) zależą od typu zastosowanej metody. Lepsze wartości funkcji celu uzyskano dla metody LDA. Uzyskana wartość 0.17, charakteryzowała się czułością wynoszącą 50% i swoistością - 69%.

Wysoką wartość funkcji celu otrzymano dla klasyfikatora RS. Przy zastosowaniu metody EWD5 na zbiorze 16-stu cech, uzyskano wartość funkcji $FMaxSen$ wynoszącą 0.14. Klasyfikator charakteryzował się czułością wynoszącą 56% przy swoistości 46%.

Z punktu widzenia klasyfikatora RS najlepsze wyniki (porównywalne z danymi ciągłymi dla trzech cech – rys .4.28) uzyskuje się dla metody dyskretyzacji EWD5 przy 16 cechach ($FMaxSen \approx 0.17$).

Metody dyskretyzacji CACC i CAIM pozwalają uzyskać wysoką skuteczność klasyfikacji z zastosowaniem klasyfikatorów QDA i LDA, natomiast dla klasyfikatora RS dają bardzo niską skuteczność klasyfikacji ($FMaxSen \approx 0$).

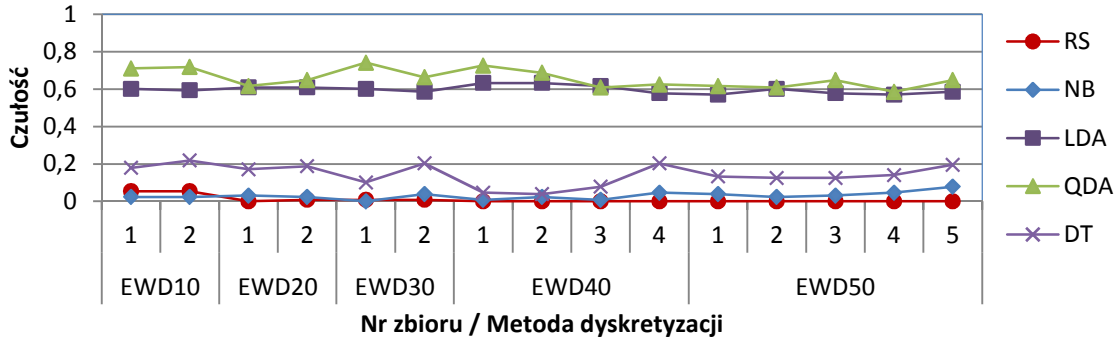
4.5.4. Wybór zbioru cech metodą RS dla dyskretyzacji EWD10-EWD50

Zastosowanie dyskretyzacji równej szerokości o liczbie przedziałów 10, 20, 30 pozwoliło na znalezienie pojedynczych reduktów. Zastosowanie dyskretyzacji równej szerokości o 40 i 50 przedziałach pozwoliło na wyznaczenie większej liczby reduktów. Dla metody EWD40 uzyskano trzy podzbiory cech, natomiast w metodzie EWD50 - cztery podzbiory. Cechy będące elementami reduktów przedstawiono w tabeli 4.3. Znalezione redukty charakteryzują się różnymi liczebnościami. Im większa jest gęstość dyskretyzacji, tym liczba elementów w redukcie jest mniejsza.

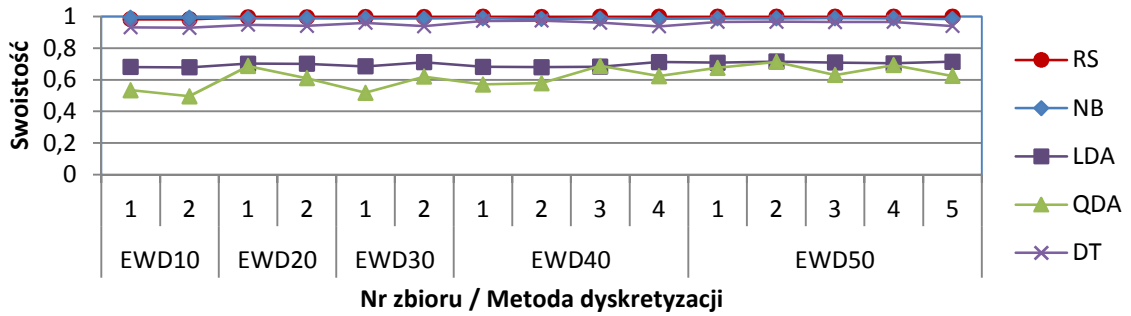
Tabela 4.3. Zbiory cech utworzone metodą RS dla dyskretyzacji EWD10, EWD20 i EWD30

Metoda dyskretyzacji	Nr zbioru	Lista cech
EWD10	1	1 2 3 4 5 6 7 8 9 10 11 12 13 14 16
	2	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
EWD20	1	1 2 3 4 5 6 7 8 9 10 11 12 13
	2	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
EWD30	1	1 3 4 5 6 8 9 10 11 12 13 14
	2	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
EWD40	1	1 3 4 5 8 9 10 11 12 13
	2	1 3 4 5 9 10 11 12 13 14
	3	1 3 4 5 7 9 10 11 12 13
	4	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
EWD50	1	1 2 3 4 6 9 10 12 13
	2	1 2 3 4 9 10 11 12 13
	3	1 2 3 4 9 10 12 13 14
	4	1 2 3 4 5 9 10 12 13
	5	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

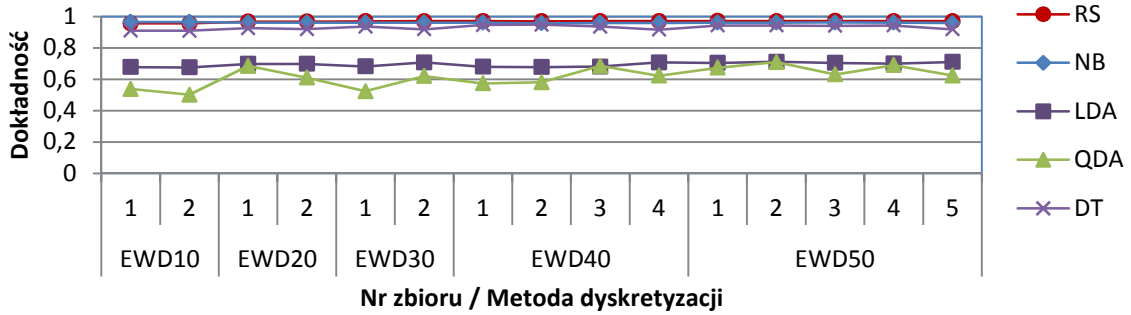
Na rys. 4.37-4.40 zamieszczono charakterystyki miar jakości klasyfikacji dla utworzonych zbiorów cech. Metody dyskretyzacji oraz odpowiadające im numery zbiorów cech opisano na osi odciętych.



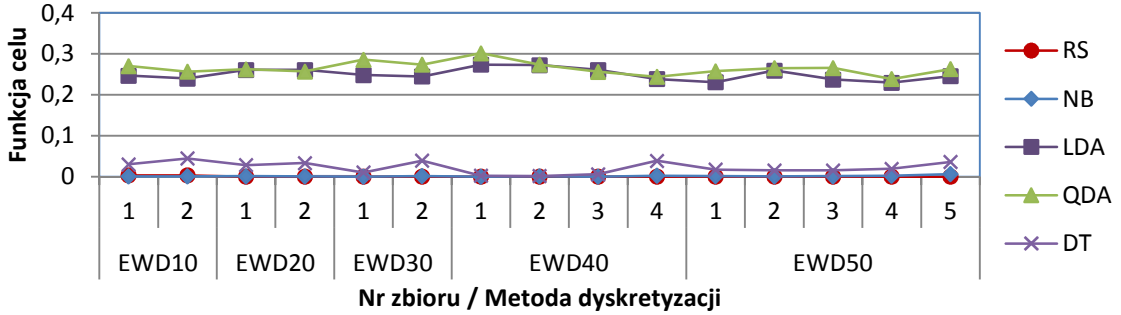
Rys. 4.37. Charakterystyka czułości klasyfikatorów dla cech dyskretyzowanych metodami: EWD10-EWD50



Rys. 4.38. Charakterystyka swoistości klasyfikatorów dla cech dyskretyzowanych metodami: EWD10-EWD50



Rys. 4.39. Charakterystyka dokładności klasyfikatorów dla cech dyskretyzowanych metodami: EWD10-EWD50



Rys. 4.40. Charakterystyka funkcji FMaxSen klasyfikatorów dla cech dyskretyzowanych metodami: EWD10-EWD50

Wyznaczone zbiory cech zostały najlepiej ocenione klasyfikatorami QDA i LDA. Wyniki są do siebie zbliżone, niezależnie od stosowanej metody dyskretyzacji. Maksymalną wartość funkcji celu uzyskano przy klasyfikatorze QDA dla reduktu nr 1 wyznaczonego przy dyskretyzacji EWD40. Wyniosła ona 0.3, co odpowiada 72% czułości i 57% swoistości. Uzyskana wartość przewyższyła wskaźnik otrzymany dla zbioru 16 cech o wartościach ciągłych, który wyniósł 0.25.

Przy zastosowaniu klasyfikacji metodą DT uzyskano niewielkie wartości współczynnika czułości. Maksymalną czułość, wynoszącą 20%, otrzymano dla pełnego zbioru cech. Otrzymana wartość jest niezależna od zastosowanej metody dyskretyzacji. Czułości tej odpowiada wartość funkcji celu wynosząca 0.03.

Najgorsze miary jakości uzyskano dla klasyfikacji NB i RS. Przeprowadzona klasyfikacja charakteryzowała się niską czułością, wynoszącą prawie 0. Takiej wartości czułości odpowiada wysoki wskaźnik swoistości. Ostatecznie wyznaczona wartość funkcji celu była zbliżona do zera.

4.6. Analiza wpływu próbkowania losowego na efektywność klasyfikacji

Zbiór komórek nowotworu pęcherza charakteryzuje się niezrównoważonym rozkładem przypadków względem klas. Zachowanie takie może wpływać niekorzystnie na algorytmy klasyfikacji. W celu zbadania wpływu rozkładu danych na wykorzystane w pracy algorytmy klasyfikacji zastosowano metodę podpróbkowania losowego. Polega ona na utworzeniu nowego zbioru danych, który zawiera wszystkie przypadki z komórek nowotworowych oraz losowo wybrane przypadki ze zbioru komórek zdrowych.

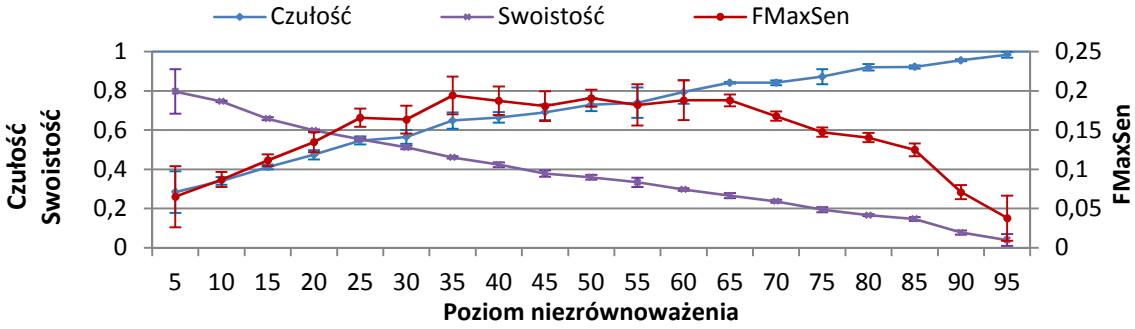
Do analizy wybrano zredukowany zbiór cech przedstawiony w tab. 4.1. W zbiorze tym klasyfikatory RS, NB, DT charakteryzowały się bardzo niskimi wartościami funkcji $FMaxSen$. Dla klasyfikatora DT uzyskano najmniejszą wartość, wynoszącą $FMaxSen$=0.002. Dla klasyfikatora RS wartość wyniosła $FMaxSen$=0.01, natomiast dla klasyfikatora NB uzyskano $FMaxSen$=0.03.

Analizę przeprowadzono dla różnych liczebności przypadków klasy komórek nowotworowych w zbiorze. Zawartość procentową komórek nowotworowych w zredukowanych danych uczących zmieniano w zakresie 5%-95%. Ponieważ liczba komórek nowotworowych nie ulega przy tym zmianie, to wraz ze wzrostem procentowej zawartości komórek nowotworowych maleje liczebność zbioru uczącego. Zbiór walidacyjny pozostaje niezmieniony.

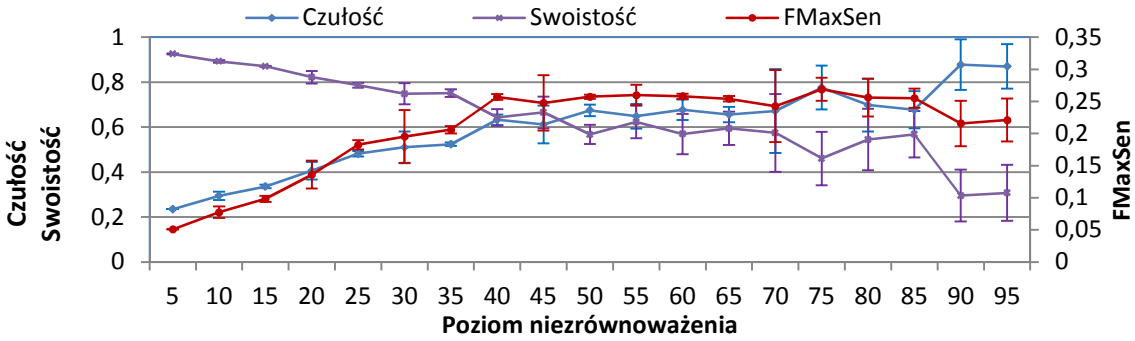
Na rysunkach 4.41-4.45 przedstawiono miary jakości klasyfikatorów (swoistość, czułość) w funkcji poziomu niezrównoważenia. Zamieszczono także charakterystykę funkcji celu $FMaxSen$, będącej kryterium wyboru najlepszego klasyfikatora komórek nowotworowych. Dla powtórzonych kilkukrotnie analiz wyznaczono wartości średnie i odchylenia standardowe.

Rozpatrując charakterystyki czułości i swoistości można wyróżnić dwie grupy klasyfikatorów. Pierwszą grupę stanowią klasyfikatory RS (rys. 4.41), NB (rys. 4.42) oraz DT (rys. 4.45). Są one silnie zależne od poziomu niezrównoważenia zbioru uczącego. Wraz ze wzrostem zawartości komórek nowotworowych w zbiorze rośnie czułość klasyfikacji. Wzrostowi czułości towarzyszy spadek swoistości.

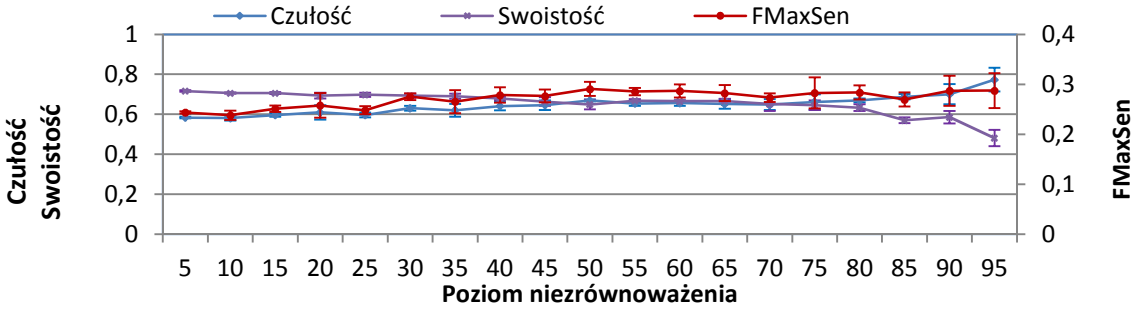
Drugą grupę tworzą klasyfikatory LDA (rys. 4.43) oraz QDA (rys. 4.44). Charakterystyka czułości i swoistości ulega tutaj niewielkim odchyleniom przy zmianie poziomu niezrównoważenia. Modyfikacja rozkładu klas wpływa w niewielkim stopniu na zmianę jakości klasyfikacji.



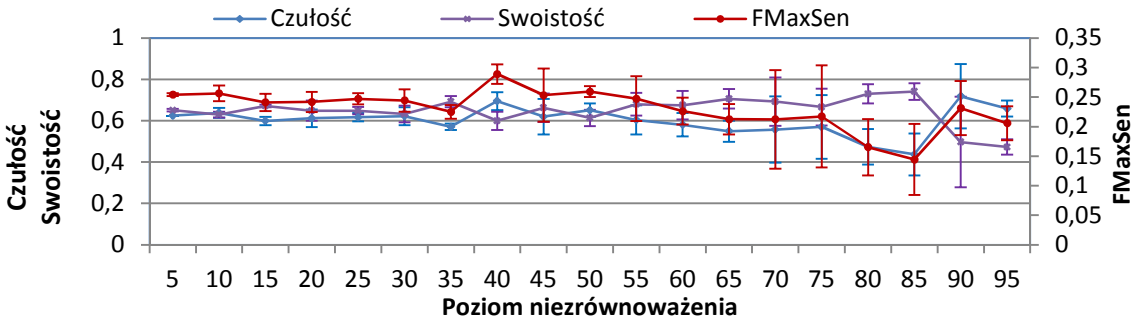
Rys. 4.41. Charakterystyka zmian miar klasyfikatora RS dla różnych poziomów niezrównoważenia przy zbiorze szesnastu cech.



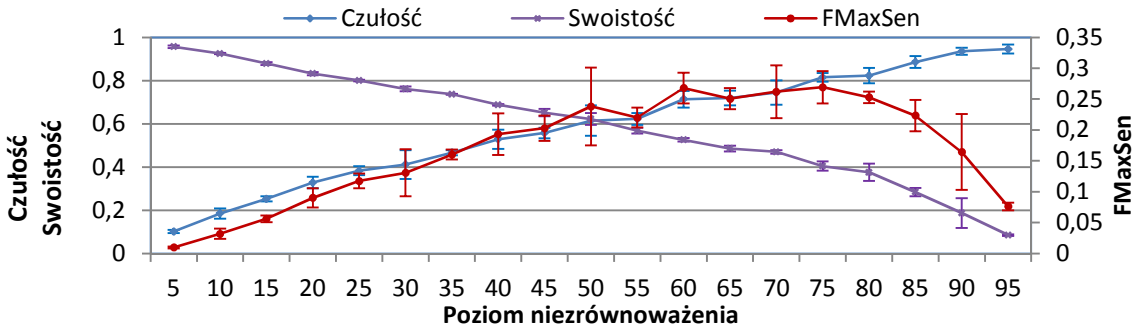
Rys. 4.42. Charakterystyka zmian miar klasyfikatora NB dla różnych poziomów niezrównoważenia przy zbiorze szesnastu cech.



Rys. 4.43. Charakterystyka zmian miar klasyfikatora LDA dla różnych poziomów niezrównoważenia przy zbiorze szesnastu cech.



Rys. 4.44. Charakterystyka zmian miar klasyfikatora QDA dla różnych poziomów niezrównoważenia przy zbiorze szesnastu cech.

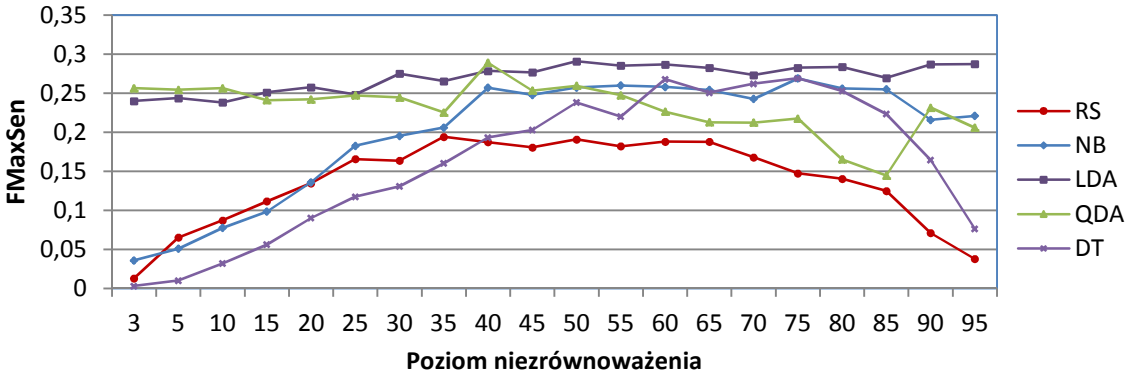


Rys. 4.45. Charakterystyka zmian miar klasyfikatora DT dla różnych poziomów niezrównoważenia przy zbiorze szesnastu cech.

Na rys. 4.46 zamieszczono charakterystykę zmian funkcji $FMaxSen$ dla różnych klasyfikatorów na różnym poziomie niezrównoważenia. Zmiana poziomu niezrównoważenia w różnym stopniu wpływa na poprawę efektywności klasyfikacji. Najlepszy wskaźnik, wynoszący 0.29 uzyskano dla klasyfikatora LDA przy poziomie niezrównoważenia zbioru uczącego 50%. Wskaźnik ten jest o 0.04 wyższy od miary $FMaxSen$ uzyskanej dla zbioru początkowego. Bardzo wysoki przyrost wartości funkcji celu osiągnięto dla charakterystyk RS, NB, DT. Wymienione klasyfikatory są silnie zależne od rozkładu danych, a przy klasyfikacji zbioru inicjalnego wyznaczone wartości funkcji $FMaxSen$ były zbliżone do zera.

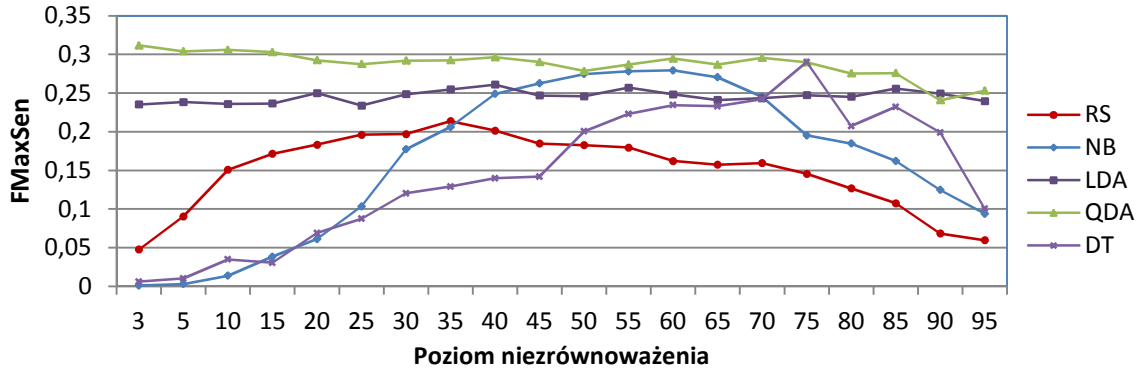
W przypadku modelu RS wartość funkcji celu stopniowo rośnie. Przy poziomie niezrównoważenia ok. 35 % osiągnięto maksimum funkcji $FMaxSen$=0.2. Przyrost czułości zaobserwowany w tym zakresie wynosi około 0.17, osiągając wartość 0.63. Wzrost czułości związany jest ze spadkiem swoistości do wartości 0.49.

Znaczną poprawę wartości funkcji celu zaobserwowano także dla klasyfikatorów DT, NB. Początkowe wartości funkcji celu zbliżone były do zera. Zmiana poziomu niezrównoważenia do 50% pozwala na osiągnięcie funkcji celu o wartości 0.25. Odpowiada to czułości i swoistości wynoszących około 0.6.

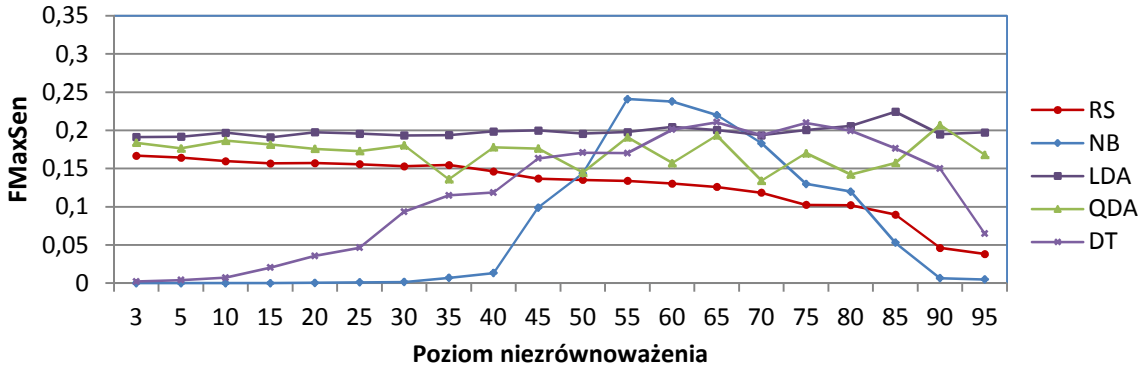


Rys. 4.46. Zmiany funkcji FMaxSen dla różnych poziomów niezrównoważenia przy zbiorze szesnastu cech.

Na rysunku 4.63-3.64 zamieszczono przyrost bezwzględny funkcji $FMaxSen$ dla zbioru pięciu i trzech cech (zbiory nr 22 i nr 6 z tab. 4.2). Zamieszczone charakterystyki pokazują, że zmniejszenie rozmiaru przestrzeni cech opisujących obiekty, przy wzroście stopnia niezrównoważenia zbioru uczącego, wpływa niekorzystnie na jakość klasyfikacji. Szczególnie jest to zauważalne dla klasyfikatora RS, przy którym odnotowano stopniowe pogarszanie się wartości funkcji celu dla zbioru trzech cech.



Rys. 4.47. Zmiany funkcji FMaxSen dla różnych poziomów niezrównoważenia przy zbiorze pięciu cech (na przykładzie reduktu nr 22 dla selekcji RS).



Rys. 4.48. Zmiany funkcji FMaxSen dla różnych poziomów niezrównoważenia przy zbiorze trzech cech (na przykładzie reduktu nr 16 dla selekcji RS).

4.7. Podsumowanie

Przedstawiony w pracy problem diagnostyki nowotworu pęcherza moczowego dotyczy ilościowej analizy preparatów cytologicznych w systemie skaningowym. Wykrywanie komórek nowotworowych przeprowadza się w dwóch etapach:

- wstępne wydzielenie w preparacie komórek, które mogą być komórkami nowotworowymi, z zastosowaniem barwnika DAPI i analizy ilościowych cech komórek;
- precyzyjne wskazanie komórek nowotworowych z zastosowaniem metody FISH, dla komórek wytypowanych na etapie wstępnym.

W pracy skupiono się na etapie wstępnej klasyfikacji komórek nowotworowych na podstawie parametrów morfometrycznych. Cechy ilościowe, dla których przeprowadzono wstępną diagnostykę, można podzielić na:

- pierwotne - parametry morfometryczne uzyskane na podstawie analizy obrazów mikroskopowych, np. powierzchnia, obwód, nieregularność, kolistość;
- wtórne - wyznaczone na podstawie cech pierwotnych, np. stosunek minimalnego promienia obrysu komórki do maksymalnego promienia obrysu komórki.

Zbiór danych uzyskany z systemu skaningowego zawiera 212 cech. Komórki uznane przez eksperta jako „nowotworowe” stanowią zaledwie 3% całego zbioru. Stawia to bardzo wysokie wymagania dla algorytmów uczenia nadzorowanego. Klasyfikacja

komórek może być wykonana na podstawie wszystkich wyznaczonych cech obiektów. Jednak, dla większości klasyfikatorów, jest to niecelowe ze względu na:

- silne skorelowanie analizowanych cech,
- wysoką złożoność obliczeniową niektórych algorytmów klasyfikacji (np. metoda RS, która charakteryzuje się wykładniczą złożonością obliczeniową).

Z tego powodu ważnym zadaniem jest określenie zbioru cech dla poszczególnych algorytmów klasyfikacji, a także dobór klasyfikatora. Dla wstępnej klasyfikacji komórek istotne jest, aby klasyfikator przede wszystkim maksymalizował czułość, przy jednocześnie wysokiej swoistości. Zapewnia to wysoką rozróżnialność komórek nowotworowych przy możliwe największej rozróżnialności komórek zdrowych. W celu optymalnego określenia zbioru cech przyjęto funkcję celu $FMaxSen$ opisaną zależnością 2.50.

Redukcję liczby cech przeprowadzono w dwóch etapach. W pierwszym analizowano pełen zestaw 212 cech. Zastosowano dwie metody selekcji (corr-AA, corr-AC) oraz jedną metodę ekstrakcji (PCA). Na drugim etapie redukcji przestrzeni cech, wykorzystano metodę opartą na teorii zbiorów przybliżonych (RS). Ze względu na złożoność obliczeniową metod RS oraz niską skuteczność klasyfikatora RS dla dużej liczy cech (rys. 4.18), do analizy wybrano zbiór składający się z 16 cech, uzyskany metodą selekcji corr-AC.

Metoda selekcji corr-AA eliminuje cechy skorelowane z pozostałymi cechami. Metoda corr-AC dodatkowo uwzględnia korelację cech z klasą. Eliminowane cechy są silnie skorelowane z pozostałymi cechami oraz najmniej skorelowane z atrybutem decyzyjnym. Zależność liczby cech od współczynnika korelacji w algorytmach corr-AA i corr-AC jest porównywalna (rys. 4.10). Obie metody cechują się prostą implementacją i wysoką szybkością działania. W powiązaniu z algorytmami klasyfikacji QDA i LDA dają zadowalające wyniki klasyfikacji komórek nowotworowych pęcherza moczowego dla badanego zbioru danych.

Metoda PCA polega na wyznaczeniu nowego układu współrzędnych na podstawie kombinacji liniowej wszystkich cech. Liczba składowych głównych odpowiada liczbie cech w zbiorze pierwotnym. Redukcję zbioru cech przeprowadza się poprzez odrzucenie tych składowych, które powiązane są z najmniejszymi wartościami własnymi. Stosuje się w tym celu różne kryteria. Do najczęściej stosowanych zalicza się kryterium wartości własnej (rys. 4.19) oraz kryterium zmienności (rys. 4.20). Powyższe kryteria wykorzystano do analizy parametrów morfometrycznych.

Utworzone zbiory cech poddano ocenie poprzez zastosowanie różnych metod klasyfikacji. Pierwszym klasyfikatorem wybranym do analiz jest klasyfikator NB będący przykładem klasyfikatora probabilistycznego, wykorzystującego statystyczną wiedzę o zbiorze uczącym. Ze względu na założenie niezależności statystycznej zmiennych, klasyfikator NB jest szczególnie odpowiedni przy dużej liczbie wymiarów przestrzeni zmiennych wejściowych.

Zbiory oceniano także korzystając z metod analizy dyskryminacyjnej (LDA, QDA). Ich celem jest znalezienie takich funkcji w przestrzeni cech, które możliwie najlepiej rozróżniają klasy występujące w zbiorach danych.

Do oceny zbiorów cech zastosowano również klasyfikację przy użyciu drzew decyzyjnych. Reguły klasyfikacyjne odpowiadające gałęziom drzewa umożliwiają czytelną interpretację wyników analiz.

Ostatnim klasyfikatorem, który wykorzystano do analizy parametrów morfometrycznych jest opracowany i zaimplementowany przez autorkę klasyfikator RS. Klasyfikator RS opiera się na tworzeniu zbioru reguł decyzyjnych. Reguły nie są w żaden sposób usystematyzowane, jak to ma miejsce w przypadku analizy drzew decyzyjnych.

Każdą regułę należy interpretować jako obiekt w przestrzeni, której rozmiar określa liczba cech. Klasyfikację nowych przypadków przeprowadza się porównując położenie obiektów w stosunku do obiektów odpowiadających regułom decyzyjnym. Do wyznaczenia odległości zastosowano miarę Euklidesową.

Dyskretyzacja wartości cech w analizowanym zbiorze metodami CACC i CAIM doprowadza do utworzenia czterech i pięciu przedziałów i może być stosowana jedynie dla klasyfikatorów QDA i LDA. Zastosowanie tych metod dyskretyzacji dla klasyfikatora RS daje bardzo negatywne wyniki. Zastosowanie dyskretyzacji wartości cech metodą EWD dla klasyfikatora RS może dać pozytywne efekty (np. rys. 4.36, dyskretyzacja EWD5, 16 cech). Dyskretyzacja wartości cech daje także pozytywne efekty związane z obniżeniem czasu obliczeń.

Klasyfikatory RS, NB, DT są silnie zależne od rozkładu obiektów względem klas. Przy klasyfikacji zbioru 16 cech wyznaczone wartości funkcji $FMaxSen$ były zbliżone do zera. Poprawa wskaźników możliwa jest dzięki modyfikacji poziomu niezrównoważenia. Dla zbioru niezrównoważonego na poziomie 35%, klasyfikator RS osiąga wartość 0.2. Natomiast dla klasyfikatora NB oraz DT osiągnięto wartość funkcji ok. 0.25.

Zestawienie najważniejszych wyników analiz parametrów morfometrycznych dla wstępnej diagnostyki nowotworu pęcherza moczowego zamieszczono w tabeli 4.4.

Tabela 4.4. Porównanie najlepszych wyników analizy parametrów morfometrycznych

LP	**Metoda selekcji**	**Metoda klasyfikacji**	**Liczba cech**	**Nr zbioru/wsp.kor**	**Czułość**	**Swoistość**	**FMaxSen**
1	corr-AA	QDA	12	0,35	**0,820**	0,515	**0,347**
2	corr-AA	QDA	10	0,26	0,796	0,524	0,333
3	corr-AA	LDA	17	0,44	0,617	0,712	0,271
4	corr-AA	LDA	14	0,41	0,656	0,628	0,270
5	corr-AA	LDA	12	0,35	0,617	0,703	0,268
6	corr-AA	NB	35	0,68	0,601	0,682	0,246
7	corr-AA	NB	113	0,93	0,687	0,551	0,260
8	corr-AA	RS	2	0.01	0,617	0,443	0,168
9	corr-AC	QDA	24	0,50	0,789	0,527	**0,328**
10	corr-AC	LDA	23	0,48	0,609	0,726	0,269
11	corr-AC	NB	52	0,73	0,625	0,656	0,256
12	corr-AC	RS	3	0,05-0,02	0,546	0,587	0,175
13	corr-AC	RS	6	0,27	0,492	0,705	0,170
14	PCA	QDA	9	9	**0,843**	0,458	**0,326**
15	PCA	QDA	34	34	0,773	0,537	0,321
16	PCA	LDA	26	26	0,648	0,735	0,309
17	PCA	LDA	8	8	0,664	0,641	0,283
18	RS	QDA	5	22	0,765	0,531	**0,311**
19	RS	QDA	5	26	**0,835**	0,433	0,302
20	RS	LDA	3	17	0,656	0,600	0,258
21	RS	RS	2	4	0,546	0,587	0,175
22	RS + CACC/CAIM	QDA	16	1	0,640	0,607	0,249
23	RS+EWD30	QDA	12	1	0,742	0,517	0,285
24	RS+EWD40	QDA	10	1	0,726	0,570	**0,300**

Najkorzystniejszą metodą oceny zredukowanych zbiorów cech, dla zdefiniowanej funkcji celu $FMaxSen$ (wzór 2.50), jest metoda QDA. Dla klasyfikacji metodą QDA uzyskano wysokie wartości funkcji $FMaxSen$, niezależnie od zastosowanej metody redukcji liczby cech. Wartość maksymalną uzyskano dla zbioru 12 cech wyznaczonego metodą corr-AA. Klasyfikator charakteryzuje się czułością wynoszącą 82% oraz swoistością 51,5%. Uzyskana wartość funkcji $FMaxSen$ jest wyższa od wartości otrzymanej dla inicjalnego zbioru cech, dla którego wartość funkcji jest zbliżona do 0.25. Odpowiada to około 67% czułości i 55% swoistości.

Wysokie wartości funkcji $FMaxSen$ otrzymano także dla metody PCA. Najlepszymi właściwościami klasyfikacyjnymi charakteryzują się metody QDA i LDA. Najlepsze wyniki uzyskano dla klasyfikacji QDA przy zbiorach zawierających przynajmniej 5 składowych głównych. Początkowo obserwuje się wzrost współczynnika czułości, który dla zbioru 9 atrybutów osiąga maksimum wynoszące 84.3%. Zbiór charakteryzuje się swoistością wynoszącą 45%, przy wartości funkcji $FMaxSen$ równej 0.326. Wraz ze wzrostem liczby składowych czułość zaczyna powoli spadać, osiągając minimum wynoszące 70%, odpowiadające funkcji celu $FMaxSen$=0.25.

W wyniku zastosowania selekcji cech metodą zbiorów przybliżonych otrzymano zbiory cech będące dokładnymi aproksymacjami zbioru wstępnego. Oznacza to, że system decyzyjny zdefiniowany przy użyciu wyznaczonych atrybutów jest dobrze określony. Przeprowadzone analizy pokazują, że pomimo tego jakość klasyfikacji jest zmienna i może okazać się nieskuteczna dla niektórych zbiorów cech. Zależność tą przedstawiono na rys. 4.28. Wśród utworzonych zbiorów wyznaczono jednak takie, które charakteryzują się wysoką wartością funkcji $FMaxSen$. Najwyższe wartości otrzymano przy klasyfikacji metodą QDA. Uzyskana wartość funkcji $FMaxSen$ wynosi 0.311, dla czułości 76.5% oraz swoistości 53.1%. Należy tutaj zwrócić szczególną uwagę na fakt, że zbiór składa się tylko z 5 cech.

Zastosowanie metody klasyfikacji NB do analizy parametrów morfometrycznych komórek daje zadowalające wyniki dla znacznej liczby cech. Przy 113 cechach uzyskano czułość wynoszącą 83.7% i swoistość 55%. Wskaźniki odpowiadają funkcji $FMaxSen$ wynoszącej 0.26. W przypadku selekcji cech metodą corr-AA wyznaczona funkcja celu zaczyna znacząco spadać przy liczbie cech wynoszącej 39. Dla metody corr-AC wartość funkcji celu maleje już przy 48 cechach. Z tego powodu metoda ta nie jest skuteczna na drugim etapie selekcji cech.

Przy klasyfikacji metodą RS najlepsze wyniki otrzymano dla zbiorów charakteryzujących się małą liczbą cech. Maksymalna wartość funkcji celu, jaką uzyskano nie przekracza wartości 0.175. Współczynnik ten nie jest tak wysoki jak przy klasyfikacjach metodami QDA czy LDA. Należy jednak zauważyć, że opracowany przez autorkę algorytm generowania reguł dla analizy parametrów morfometrycznych komórek osiąga lepsze miary jakości niż klasyfikator regułowy oparty na drzewach decyzyjnych.

Najgorszymi właściwościami klasyfikacyjnymi dla analizowanego zbioru danych charakteryzuje się algorytm drzew decyzyjnych. Cechuje go czułość nieprzekraczająca 15%, co przy wysokiej swoistości, prowadzi do uzyskania wartości funkcji $FMaxSen$ zbliżonej do 0.

Podsumowując, najkorzystniejsze wyniki selekcji cech oraz klasyfikacji, dla wstępnej diagnostyki pęcherza moczowego uzyskano w następujących przypadkach:

LP	Metoda selekcji	Metoda klasyfikacji	Liczba cech	Czułość	Swoistość	FMaxSen
1	corr-AA	QDA	12	0,820	0,515	0,347
2	corr-AC	QDA	24	0,789	0,527	0,328
3	PCA	QDA	9	0,843	0,458	0,326
4	RS	QDA	5	0,765	0,531	0,311

Przedstawione wyniki pokazują, że analiza parametrów morfometrycznych komórek uzyskanych za pomocą systemu skaningowego daje możliwość wstępnej diagnostyki nowotworu pęcherza moczowego.

ROZDZIAŁ 5

Podsumowanie

5.1. Najważniejsze rezultaty

W pracy doktorskiej podjęto zadanie opracowania metody wstępnej klasyfikacji komórek na podstawie parametrów morfometrycznych uzyskanych z systemu skaningowego dla wczesnego wykrywania nowotworu pęcherza moczowego.

Zbiór danych uzyskany od firmy MetaSystems zawiera blisko 23000 obiektów opisanych przez 212 parametrów morfometrycznych i charakteryzuje się wysokim poziomem niezrównoważenia (ok. 3% wszystkich komórek stanowią komórki nowotworowe).

Szczególną uwagę zwrócono na opracowanie metody klasyfikacji komórek, pozwalającej na maksymalizację czułości, przy jednocześnie możliwe wysokiej swoistości. Zapewnia to wysoką wykrywalność komórek nowotworowych przy jednoczesnej wysokiej rozróżnialności komórek zdrowych.

Przy wstępnej klasyfikacji komórek na podstawie parametrów morfometrycznych najistotniejsze jest wykrycie możliwe wszystkich komórek nowotworowych, przy ograniczonym błędzie klasyfikacji polegającym na zakwalifikowaniu komórki zdrowej jako komórki nowotworowej. Z tego względu, jako funkcję celu podczas poszukiwania optymalnego algorytmu klasyfikacji, przyjęto wyrażenie opisane zależnością:

$$FMaxSen = sen^2 * spe\,. \tag{5.1}$$

Na podstawie otrzymanych wyników można stwierdzić, że:

- najkorzystniejsze wyniki wstępnej klasyfikacji komórek nowotworowych pęcherza moczowego uzyskuje się z zastosowaniem algorytmów QDA i LDA. Ze względu na niską złożoność obliczeniową tych algorytmów można je stosować do klasyfikacji na postawie dużej (nawet większej od 200) liczby cech;
- metoda klasyfikacji nadzorowanej z zastosowaniem zbiorów przybliżonych i metody k-najbliższych sąsiadów z maksymalizacją czułości klasy mniejszościowej jest skutecznym algorytmem klasyfikacji komórek nowotworowych pęcherza moczowego, jednak ze względu na wysoką złożoność obliczeniową, może być stosowany przy bardzo ograniczonej liczbie cech (mniej niż 16);
- algorytm NB jest mniej skutecznym algorytmem klasyfikacji i daje pozytywne wyniki jedynie przy znacznej liczbie cech;
- algorytm DT jest nieskutecznym algorytmem klasyfikacji komórek nowotworowych pęcherza moczowego.

Ze względu na znaczne niezrównoważenie zbioru komórek, podczas poszukiwania algorytmu klasyfikacji celowe jest zastosowanie wstępnego przetwarzania zbioru danych z zastosowaniem techniki podpróbkowania losowego. Zmiana poziomu niezrównoważenia

w różnym stopniu wpływała na poprawę efektywności klasyfikacji. Najlepszy wynik uzyskano dla klasyfikatora LDA przy poziomie niezrównoważenia 50%. Wartość funkcji $FMaxSen$ jest o 0.05 wyższa od miary $FMaxSen$ uzyskanej dla zbioru inicjalnego. Bardzo wysoki przyrost wartości funkcji osiągnięto też dla klasyfikatorów RS, NB, DT. Klasyfikatory te są silnie zależne od rozkładu danych, a przy klasyfikacji zbioru inicjalnego wyznaczone wartości funkcji $FMaxSen$ były zbliżone do zera.

Spośród 212 cech, którymi są opisane poszczególne komórki, znaczna ich część jest ze sobą ściśle skorelowana. Oznacza to, że celowa jest redukcja liczebności zbioru cech. Redukcję przeprowadzono w dwóch etapach: początkowo zastosownao algorytmy corr-AA, corr-AC, PCA, a następnie algorytm RS. Na podstawie uzyskanych wyników można stwierdzić, że połączenie algorytmów corr-AC do wstępnej selekcji zbioru cech z algorytmem RS do ostatecznego wyboru cech daje istotne rezultaty. W wyniku redukcji zbioru cech wstępną klasyfikację komórek nowotworowych można przeprowadzić na podstawie kilku (3-5) cech. Do podstawowych cech wyróżniających komórki nowotworowe należy zaliczyć: całkowitą/względną powierzchnię komórki, obwód, współczynnik kształtu.

Analizowane dane charakteryzują się ciągłością dziedziny. Ma to negatywny wpływ na czas obliczeń z zastosowaniem teorii zbiorów przybliżonych. Dlatego w pracy zbadano także wpływ dyskretyzacji na zastosowanie algorytmów wykorzystujących teorię zbiorów przybliżonych. Wykorzystano dwie grupy metod dyskretyzacji. Pierwsza, której przykładem jest dyskretyzacja równej szerokości, w procesie tworzenia granic nie uwzględnia przynależności obiektów do klas. Druga grupa metod (algorytmy CAIM oraz CACC) wyznacza wartości granic w taki sposób, aby w każdym z przedziałów dyskretyzacji maksymalizować liczbę obiektów należących do jednej klasy.

Zastosowanie dyskretyzacji przy selekcji cech lub klasyfikacji metodą RS skraca czas obliczeń. Dyskretyzacja wpływa także na liczbę zredukowanych zbiorów cech, im większa gęstość dyskretyzacji tym większa jest liczba reduktów systemu decyzyjnego (tab. 4.3.). Dodatkowo, wraz ze wzrostem gęstości dyskretyzacji, maleje liczba cech wchodzących w skład reduktu.

Obliczenia przeprowadzono z wykorzystaniem autorskiego narzędzia Rough Sets Analysis Toolbox stanowiącego zbiór programów dla środowiska MATLAB. Przy projektowaniu przybornika główny nacisk położono na algorytmy i programy dla teorii zbiorów przybliżonych. Opracowano programy dla obliczeń w wersji jednostanowiskowej oraz rozproszonej. Do realizacji systemu rozproszonego wykorzystano komponenty środowiska obliczeniowego MATLAB Parallel Computing Toolbox oraz Distributed Computing Server. Model rozproszony rozbudowano o bazę danych MySql. Skrócono w ten sposób czas przesyłania danych do węzłów klastra. Przeprowadzone obliczenia pokazują, że wprowadzenie zrównoleglenia znacząco przyspiesza czas obliczeń przy większej liczbie cech. Pozwala to także na przeprowadzanie analiz na zbiorach charakteryzujących się dużą liczebnością obiektów.

Zdaniem autorki opracowanie algorytmów i programów wykorzystujących operacje wektorowe i macierzowe do wspomagania diagnostyki z zastosowaniem zbiorów przybliżonych oraz zastosowanie obliczeń równoległych pozwala na znaczne skrócenie czasu obliceń.

Zastosowanie teorii zbiorów przybliżonych do selekcji cech dla analizowanego zbioru komórek daje bardzo interesujące rezultaty. Uzyskane zbiory charakteryzują się małą liczebnością zbioru cech i jednocześnie wysoką zdolnością klasyfikacji.

Zastosowanie zbiorów przybliżonych w zadaniu klasyfikacji komórek nowotworowych daje umiarkowane efekty. Znacznie korzystniejsze pod tym względem są algorytmy dyskryminacyjne QDA i LDA.

5.2. Kierunki dalszych badań

W trakcie prowadzonych analiz stwierdzono kilka problemów, których rozwiązanie wykracza poza zakres pracy.

Zastosowanie obliczeń rozproszonych znacząco przyspiesza realizację problemów rozpoznawania wzorców z zastosowaniem zbiorów przybliżonych. Równoległa realizacja jest celowa przy dużej liczbie cech lub przy dużej liczbie obiektów. Na czas realizacji wpływają także czasy komunikacji pomiędzy węzłami. Z tego powodu interesującym kierunkiem dalszych badań jest poszukiwanie innych architektur równoległych do rozwiązywania złożonych zadań klasyfikacji.

Przeprowadzone badania pokazują, że wykorzystanie tablicy prawdy do poszukiwania wszystkich reduktów jest skuteczne. Zastosowanie operacji macierzowych upraszcza stosowanie takiego rozwiązania. Niestety przy generowaniu tablicy prawdy występują znaczące ograniczenia pamięciowe. Problem ten stanowi możliwy kierunek badań, który można rozwiązać poprzez obliczenia na rozproszonej tablicy prawdy. Wymagany jest do tego klaster obliczeniowy, charakteryzujący się krótkimi czasami komunikacji.

Bibliografia

[AmSa03] Amitava R., Sankar K.: *Fuzzy discretization of feature space for a rough set classifier*. Pattern Recognition Letters. Vol. 24, 2003, pp.895–902.

[Arm78] Armitage P.: *Metody statystyczne w badaniach medycznych*. Państwowy Zakład Wydawnictw Lekarskich. Warszawa, 1978.

[Bat08] Bator M.: *Automatyczna detekcja zmian nowotworowych w obrazach mammograficznych z wykorzystaniem dopasowania wzorców i wybranych narzędzi sztucznej inteligencji*. Instytut Podstawowych Problemów Techniki PAN. 2008. Praca doktorska pod kierunkiem prof. dr hab.inż. Mariusz Jacek Nieniewski.

[Baz98] Bazan J.G.: Metody wnioskowań aproksymacyjnych dla syntezy algorytmów decyzyjnych. Politechnika Warszawska. Wydział Matematyki, Informatyki i Mechaniki. Rozprawa doktorska pod kierunkiem prof. dr hab. Andrzeja Skowrona. Warszawa, 1998r.

[BazSzc00] Bazan J., Szczuka M.: RSES and RSESlib - *A Collection of Tools for Rough Set Computations*. Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing (RSCTC 2000), Banff, Canada, October 16-19, 2000, 74-81, in. Technical report CS-2000-07, September 2000, University of Regina (2000).

[Be04] Beynon M.: *Stability of continuous value discretization: an application within rough set theory*. International Journal of Approximate Reasoning. Vol. 35, 2004, pp. 29–53.

[BorCon03] Borkowska E, Constantinou M, Binka-Kowalska A., Kałużewski B.: *Diagnostyka raka pęcherza moczowego przy użyciu metody MSSCP (eksony 5-8 genu P53) i testu UroVysion*. I Konferencja użytkowników DNA Pointer System, Warszawa, 2003.

[BorSie04] Borowka A., Siedlecki P.: *Nowotwory układu moczowo-płciowego* . Praca zbiorowa. Opracowanie przygotowane przez zespół ekspertów Polskiego Towarzystwa Urologicznego i Polskiego Towarzystwa Onkologii Klinicznej. 2004.

[Bro01] Brown T.A.: *Genomy*, Wydawnictwo Naukowe PWN. Warszawa, 2001.

[BrzSwi08] Brzostowski K., Świątek J.: *Adaptacyjny algorytm wyboru scenariusza oparty na wiedzy eksperta*. Sterowanie i automatyzacja: Aktualne problemy i ich rozwiązania. Red. Malinowski K., Rutkowski L. Akademicka Oficyna Wydawnicza Exit. Warszawa, 2008. s.275-284.

[Bub90] Bubnicki Z.: *Wstęp do systemów ekspertowych*. Państwowe Wydawnictwo Naukowe. Warszawa, 1990.

[Bub01] Bubnicki Z.: *Application of learning algorithms and uncertain variables in knowledge-based pattern recognition*. Artificial Life and Robotics, 2001, Volume 5, Number 2, pp. 67-71.

[Byr02] Byrski M.: *Data Mining w bazie Oracle 9i*. VIII Konferencja Użytkowników i Deweloperów Oracle - PLOUG. Październik 2002. s. 112-122.

[Cat66] Cattell R. B.: The scree test for the number of factors. Multivariate Behavioral Research, Vol.1,1966, pp.245-276.

[Cha10] - Chawla N.: Data Mining for Imbalanced Datasets: An Overview. Data Mining and Knowledge Discovery Handbook. Maimon O. Rokach L., 2010, Part 6, 875-886.

[ChWo95] Ching J.Y., Wong A.K.C., Chan K.C.C.: *Class-dependent discretization for inductive learning from continuous and mixed-mode data*. IEEE Transactions on Pattern Analysis and Machine Inteligence, V.17/7, 1995, pp. 641-651.

[Cho02] Cholewa W.: *Systemy doradcze w diagnostyce technicznej*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str. 543-580.

[ChoKos02] Cholewa W., Kościelny J.: *Wprowadzenie*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W.; Wydawnictwa Naukowo-Techniczne; Warszawa 2002; str. 3-27.

[Ci00] Cichosz P.: *Systemy uczące się*, Wydawnictwo Naukowo-Techniczne, Warszawa 2000.

[Cie02] Cierniak R.: *Inteligentne systemy obliczeniowe wspomagające zadania diagnostyczne w medycynie*. Systemy komputerowe i teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 343-360.

[CiPe98] Cios K.J., Pedrycz W., Swiniarski R.: *Data mining methods for knowledge discovery*. Kluwer Academic Publishers, 1998.

[CiPe07] Cios K.J., Pedrycz W., Swiniarski R.: *Data Mining: A Knowledge Discover Approach.* Springer, 2007.

[Ciu05] Ciupke K.: *A comperative study on methods of reduction and selection of infromation in technical diagnostics*. Mechanical Systems and Signal Processing. Vol. 19, I. 5, 2005, pp 919-938.

[CiuUrb07] Ciupke K., Urbanek G.: *Methods of features selection in inverse model identification.* Fault Diagnosis and Fault Tolerant Control. Edit. Korbicz J., Patan K., Kowal M. Academic Publishing House Exit, Warszawa 2007. s.145-152.

[CouDol99] Coulouris G., Dollimore J., Kindberg T.: *Systemy rozproszone : podstawy i projektowanie*. tł. Zdzisław Płoski. - Warszawa : Wydaw-a Naukowo-Techniczne, 1999.

[CzoZaz09] Czopek K., Zazulak M.: *Wirtualne rozwarstwienie człowieka, czyli co potrafi tomografia komputerowa*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 389-394

[DanRon05] Daniely M., Rona R., Kaplan T., Olsfanger S., Elboim L., Zilberstiena Y., Freibergera A.,Kidronc D. Lew S., Leibovitch I.: *Combined analysis of morphology and fluorescence in situ hybridization significantly increases accuracy of bladder cancer detection in voided urine samples*. Urology. Vol. 66, I.6, 2005, pp. 1354-1359.

[DanRon07] Daniely M., Rona R., Kaplan T., Olsfanger S., Elboim L., Freiberger A., Lew S., Leibovitch I.: *Combined morphologic and fluorescence in situ hybridization analysis of voided urine samples for the detection and follow-up of bladder cancer in patients with benign urine cytology*. Cancer Cytopathology. 2007, Vol. 111, I.6, pp. 517–524.

[DasLiu97] Dash M., Liu H.: *Feature Selection for Classification*. Intelligent Data Analysis. 1997. Vol. 1. pp. 131-156.

[Dom04] Dominik A., *Analiza danych z zastosowaniem teorii zbiorów przybliżonych*. Praca Magisterska, Opiekun pracy: dr inż.. Roman Podraza, Politechnika Warszawka, Wydział Elektroniki i Technik Informacyjnych, Instytut Informatyki. Rok akademicki 2003/2004.

[Dud07] Duda J.T.: *Pozyskiwanie wzorców diagnostycznych w komputerowych analizach sprawności urządzeń*. Diagnostyka Procesów i Systemów. Pod red. Korbicz J., Patan K., Kowal M. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2007, s.1-16.

[DoKo95] Dougherty J., Kohavi R., Sahami M.: *Supervised and Unsupervised Discretization of Continuous Features*. Machine Learnining: Proceedings of the Twelfth International Conference, 1995, Morgan Kaufmann Publishers, pp. 194-202.

[DraSwi08] Drapała J., Świątek J.: *Dynamiczne sieci neuronowe jako globalnie optymalny model systemu złożonego - zbieżność algorytmu uczenia*. Sterowanie i automatyzacja: Aktualne problemy i ich rozwiązania. Red. Malinowski K., Rutkowski L. Akademicka Oficyna Wydawnicza Exit. Warszawa, 2008. s.155-165.

[DulPie00] Dulewicz A., Piętka B.D., Jaszczak P.: *Komputerowa metoda wspomagania diagnostyki nowotworów pęcherza moczowego*. Obrazowanie biomedyczne. Biocybernetyka i inżynieria biomedyczna 2000. Tom 8. red. Chmielewski L., Kulikowski J., Nowakowski A. s. 883-920.

[DziMar07] Dziekan Ł., Marciniak A., Obuchowicz A.: *Segmentation of colour cytological images using type-2 fuzzy sets*. Fault Diagnosis and Fault Tolerant Control. Edit. Korbicz J., Patan K., Kowal M. Academic Publishing House Exit, Warszawa 2007. s.263-270.

[FaPr97] Fawcett T., Provost F.: *Adaptive Fraud Detection. Journal Data Mining and Knowledge Discovery*. Kluwer Academic Publishers HinghamVol. 1/3, 1997, pp. 291-316.

[Faw06] Fawcett T.: *An introduction to ROC analysis*. Pattern Recognition Letters. V. 27,2006,pp. 861–874.

[FeGa11] - Fernández A., García S., Herrera F.: Addressing the Classification with Imbalanced Data: Open Problems and New Challenges on Class Distribution. Hybrid Artificial Intelligent Systems. Lecture Notes in Computer Science, 2011, Vol.6678/2011, pp.1-10.

[GaSa10] García V., Sánchez J.S., Mollineda R.A.: Exploring the Performance of Resampling Strategies for the Class Imbalance Problem. Trends in Applied Intelligent Systems Lecture Notes in Computer Science, 2010, Volume 6096/2010, pp.541-549.

[Gat98] Gatnar E.: *Symboliczne metody klasyfikacji danych*. Wydawnictwo Naukowe PWN, Warszawa, 1998.

[GloPat07] Głowacki G., Patan K., Korbicz J.: *Nonlinear principal component analysis in fault diagnosis*. Fault Diagnosis and Fault Tolerant Control. Edit. Korbicz J., Patan K., Kowal M. Academic Publishing House Exit, Warszawa 2007. s.211-218.

[Grabo03] Grabowski S.: *Konstrukcja klasyfikatorów minimalnoodległościowych o strukturze sieciowej*. Politechnika Łódzka, Wydział Elektroniki i Elektrotechniki. Rozprawa doktorska pod kierunkiem prof. dr hab. inż. Dominika Sankowskiego, Łódź, 2003.

[Grabc03] Grąbczewski K.: *Zastosowanie kryterium separowalności do generowania reguł klasyfikacji na podstawie baz danych*. Instytut Badań Systemowych, Polska Akademia Nauk, Rozprawa doktorska pod kierunkiem prof. dra hab. Włodzisława Ducha, Warszawa, 2003.

[Gro03] Grochowski L.: *Rozproszone systemy informatyczne*. Warszawa, Elipsa - Dom Wydaw., 2003.

[GrzHip02] Grzymała-Busse J., Hippe Z., Bajcar S.: *Ocena ryzyka zagrożenia czerniakiem skóry na podstwie wybranych cech obrazów dermatoskopowych*. Systemy komputerowe i teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 213-224.

[GuEl03] Guyon I., Elisseeff A.: *Empirical InferenceAn Introduction to Variable and Feature Selection*. Journal of Machine Learning Research. Vol. 3, 2003, pp. 1157-1182.

[Guz05] Guz T.: *Poprawa efektywności klasyfikatora „Box Classifier" w systemie „Metafer"*. XIII Konferencja „Sieci i Systemy Informatyczne", Łódź, 2005.

[GuzSzy06] Guz T., Szydłowska E.: *Genetic Algorithm as a Method of Feature Selection for the Purpose of Cancer Cell Classification in the Automatic Scanning System "Metafer"*. Materiały konferencyjne. ICYR 2006, 18-20.09.2006, Zielona Góra.

[HadFig09] Haduch J., Figiel H.: Pasowicz M.: *Obrazowanie magnetyczno-rezonansowe*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 419-440.

[Ha99] Hall M.: *Correlation-based Feature Selection for Machine Learning*. Department of Computer Science. The Universuty of Waikato. Doctoral Thesis. Supervisor: Lloyd Smith. Hamilton, NewZealand. April 1999.

[Hal99] Hallinan J.: Detection of Malignancy AssociatedChanges in Cervical Cells Using Statistical and Evolutionary Computation Techniques. The University of Queensland. Centre for Sensor Signal and nformation Processing. 1999.

[HaMa05] Hand D., Mannila H., Smyth P.: *Eksploracja danych*. Wydawnictwa Naukowo-Techniczne WNT, 2005.

[HreKor07] Hrebień M., Korbicz J.: *Segmentation of cytological images by combined Hough, evolutionary and watershed algorithms*. Fault Diagnosis and Fault Tolerant Control. Edit. Korbicz J., Patan K., Kowal M. Academic Publishing House Exit, Warszawa 2007. s.271-278.

[HuMo98] Huan L., Motoda H.: *Feature Transformation And Subset Selection*. Intelligent Systems and their Applications, 1998, Vol. 13, Issue 2, pp. 26 – 28.

[HuaSet97] Huan Liu, Setiono. R.: *Feature selection via discretization*. IEEE Transactions on Knowledge and Data Engineering. 1997.Vol.9, I.4. pp.642 - 645.

[HubLor98] Huber R., Lörch Th., Kulka U.,Braselmann H.,Bauchinger M.: *Technical report: automated classification of first and second cycle metaphases*. Mutation Research/Genetic Toxicology and Environmental Mutagenesis. Elsevier Science. Vol: 419, No 1-3 , 1998, pp. 27-32

[HubKul01] Huber R., Kulka U., Lörch Th., Braselmann H., Engert D., Figel M., Bauchinger M.: *Technical report: application of the Metafer2 fluorescence scanning system for the analysis of radiation-induced chromosome aberrations measured by FISH-chromosome painting*. Mutation Research/Genetic Toxicology and Environmental Mutagenesis. Subscribed Journal Mutation Research/Genetic Toxicology and Environmental Mutagenesis. Vol. 492, No.1-2, 2001. pp. 51-57.

[IszNowIn10] Iszkowski W., Nowak J., Skowron A., Stroiński M., Tadeusiwicz R., Węglarz J., Wiatr K.: Wczoraj, dziś i jutro polskiej informatyki. Red. R. Tadeusiewicz. Wydaw. Polskie Towarzystwo Informatyczne, 2010.

[Ja93] Jajuga K.: *Statystyczna analiza wielowymiarowa*. Wydawnictwo PWN, Warszawa, 1993.

[Jap00] Japkowicz N.: *Learning from Imbalanced Data sets: A Comparison of Various Strategies*. In Proceedings of the AAAI'2000 Workshop on Learning from Imbalanced Data Sets, Austin, TX.

[JapMy95] Japkowicz N., Myers C., Gluck M.: *A Novelty Detection Approach to Classification*. In Proceedings of the Fourteenth Joint Conference on Artificial Intelligence,1995, pp.518-523.

[JeWro02] Jeżewski J.,Wróbel J., Horoba K.: *Monitorowanie zagrożeń płodu wspomagane komputerem*. Systemy komputerowe i teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 97-119.

[Jan02] Janecki J.: *System komputerowego wspomagania diagnostyki laboratoryjnej*. Systemy komputerowe i teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 68-81.

[JawKan09] Jaworek J., Kańtoch E.: *Wstępny przegląd problematyki komputerowego przetwarzania i analizy obrazów w systemach diagnostyki medycznej*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 357-368.

[Joz98] Józefczyk J.: *Rozpoznawanie i zastosowania biomedyczne. Problemy automatyki i informatyki*. Zakład Narodowy im. Osolińskich - Wydawnictwo Polskiej Akademii Nauk. Warszawa, 1998.

[Kai60] Kaiser H.: *The application of electronic computers to factor analysi*s. Educational and Psychological Measurement, Vol.20, 1960, pp.141-151.

[KarNie01] Karbowski A., Niewiadomskia-Szynkiewicz E.: *Obliczenia Równoległe i Rozproszone*. ofic. Wyd. Pol. Warszawskiej, Warszawa, 2001.

[KoJo97] Kohavi R., John G.: *Wrappers for Feature Subset Selection*. Artificial Intelligence Vol. 97, 1997, pp.273-324.

[KomPol99] Komorowski J., Polkowski L., Skowron A.: *Rough Sets: A Tutorial*. Lecture Notes of the 11th European Summer School in Logic, Language and Information (ESSLLI). UTRECHT UNIVERSITY, 1999.

[KorCwi05] Kornacki J., Ćwik J.: *Statystyczne systemy uczące się*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2005.

[Kos01] - Kościelny J.M.: *Diagnostyka zautomatyzowanych procesów przemysłowych*. Akademicka Oficyna Wydawnicza Exit, Warszawa, 2001.

[Kos02a] Kościelny J.: *Modele w diagnostyce procesów*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str. 29-56.

[Kos02b] Kościelny J.: *Metodologia diagnostyki procesów*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str. 57-114.

[Kos02c] Kościelny J.: *Zastosowanie logiki rozmytej w diagnostyce procesów przemysłowych*. Materiały konferencyjne. Red. Bubnicki Z., Korbicz J., XIV Krajowa Konferencja Automatyki, Zielona Góra, 24-27 czerwca 2002. s.589-594.

[KosSyf02] Kościelny J.M., Syfert M.: *Zastosowanie logiki rozmytej w diagnostyce*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str. 383-426.

[KowObu09] Kowal M., Obuchowicz A.: *Cytological image segmentation using fuzyy clustering*. Diagnosis of Processes and Systems. Edit. Zdzisław Kowalczuk. Pomeranian Science and Technology Publishers PWNT, Gdańsk 2009. pp. 283-290.

[KowBia02] Kowalczuk Z., Białaszewski T.: *Algorytmy genetyczne w wielokryterialnej optymalizacji obserwatorów detekcyjnych*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str. 465-512.

[Krz90] Krzyśko M.: *Analiza dyskryminacyjna*. Wydawnictwa Naukowo-Techniczne. Warszawa, 1990.

[KrzWol08] Krzyśko M., Wołyński W., Górecki T., Skorzybut M.: *Systemy uczące się. Rozpoznawanie wzorców, analiza skupień i redukcja wymiarowości*. Wydawnictwa Naukowo-Techniczne. Warszawa, 2008.

[KuHo98] Kubat M., Holte R.C., Matwin S.: *Machine Learning for the Detection of Oil Spills in Satellite Radar Images*. Journal Machine Learning - Special issue on applications of machine learning and the knowledge discovery process. Vol.30/2-3, 1998, pp.195-215.

[KulŁuk09] Kulczycki P., Łukasik S.: *Redukcja wymiaru i liczności próby dla potrzeb syntezy statystycznego układu wykrywania uszkodzeń*. Systemy wykrywające, analizujące i tolerujące usterki. Red. Zdzisław Kowalczuk. Pomorskie Wydawnictwo Naukowo-Techniczne. Gdańsk, 2009. s.139-146.

[Kul00] Kulikowski J.: *Rozpoznawanie obrazów*. Obrazowanie biomedyczne. Biocybernetyka i inżynieria biomedyczna 2000. Tom 8. red. Chmielewski L., Kulikowski J., Nowakowski A. s. 193-238.

[Kul02] Kulikowski J.L.: *Uzyskiwanie wiedzy z baz danych medycznych. Systemy komputerowe i teleinformatyczne w służbie zdrowia*. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 132-164.

[KurCio03] Kurgan L., Cios K.: *Fast Class-Attribute Interdependence Maximization (CAIM) Discretization Algorithm*. Proceedings of the 2003 International Conference on Machine Learning and Applications (ICMLA'03), Los Angeles, CA, U.S.A., CSREA Press, pp. 30-36.

[KuCi04] Kurgan L., Cios K.: *CAIM Discretization Algorithm*, IEEE Transactions on Knowledge and Data Engineering, V.16/2,2004, pp. 145-153.

[Kwi07] Kwiatkowski W.: *Metody automatycznego rozpoznawania wzorców*. Wydawnictwo Bel Studio Sp. Z o.o., Warszawa, 2007.

[Lar06] Larose D.T.: *Odkrywanie wiedzy z danych. Wprowadzenie do eksploracji danych*. Wydawnictwo PWN, Warszawa 2006.

[Lar08] Larose D.T.: *Metody i modele eksploracji danych*. tłum.: Anna Wilbik, Wydawnictwo Naukowe PWN 2008.

[MacSte11] Maciejewski T., Stefanowski J.: *Local Neighbourhood Extension of SMOTE for Mining Imbalanced Data*. In Proc. IEEE Symposium on Computational Intelligence and Data Mining CIDM 2011. Within 2011 IEEE SSCI, Paris, 11-15 April 2011, IEEE Press, 104-111.

[Mal02] Malina W.: *Podstawy automatycznej klasyfikacji obrazów*. Wydawnictwo Politechniki Gdańskiej. Gdańsk 2002.

[MarKor00] Marciniak A., Korbicz J., Kuś J.: *Wstępne przetwarzanie danych*. Biocybernetyka i inzynieria biomedyczna 2000. Sieci neuronowe. Tom 6. Akademicka Oficyna Wydawnicza Exit, Warszawa, 2000. s.29-71.

[MarKor02] Marciniak A., Korbicz J.: *Metody rozpoznawania obrazów w diagnostyce*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str. 513-542.

[MiKwa06] Michalak K., Kwaśnicka H.: *Correlation-based feature selection strategy in classification problems*. Int. J. Appl. Math. Comput. Sci., 2006 , Vol. 16, No. 4, 503–511.

[MicSte81] Michalski R., Stepp R., Diday E.: *A recent advance in data analysis: clustering objects into classes characterized by conjuctive concepts*. Progress in Pattern recognition, 1981, Vol. 1, pp. 33–56.

[MocTom09] Moczulski W., Tomasik P., Wachla D., Szulim R.: *Inteligentny system diagnostyki wspomagania sterowania procesów przemysłowych DIASTER*. Systemy wykrywające, analizujące i tolerujące usterki. Red. Zdzisław Kowalczuk. Pomorskie Wydawnictwo Naukowo-Techniczne. Gdańsk, 2009. s.65-76.

[MroPlo99] Mrózek A., Płonka L.: *Analiza danych metodą zbiorów przybliżonych*. Akademicka Oficyna Wydawnicza PLJ,Warszawa, 1999.

[Nie03] Niemiewski M.: *Rekonstrukcja i segmentacja obrazów w morfologii matematycznej* . Biocybernetyka i inżynieria biomedyczna. T8. Obrazowanie biomedyczne. Red. L. Chmielewski, J. Kulilkowski, A.Nowakowski. Warszawa 2003. Akadem. Oficyna. Wydaw. Exit, s.83-125.

[NowKac03] Nowakowski A., Kaczmarek M., Hryciuk M.: *Tomografia termiczna*. Biocybernetyka i inżynieria biomedyczna. T8. Obrazowanie biomedyczne. Red. L. Chmielewski, J. Kulilkowski, A.Nowakowski. Warszawa 2003. Akadem. Oficyna. Wydaw. Exit, s.615-696.

[Ngu97] Nguyen Hung Son: *Discretizaton of Real Value Attributes: A boolean reasoning approach*. Department Mathematics, Computer Science and Mechanics. Thesis Supervisor: prof. dr hab. A.Skowron.Warsaw University. 1997.

[NgSk95] Nguyen Son H., Skowron A.: *Quantization Of Real Value Attributes - Rough Set and Boolean Reasoning Approach*. 1995. Proc. of the Second Joint Annual Conference on Information Sciences, Wrightsville Beach, North Carolina.

[ObuKor02] Obuchowicz A., Korbicz J.: *Metody ewolucyjne w projektowaniu systemów diagnostycznych*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str. 279-310.

[Ogi04] Ogiela M.R.: *Strukturalne metody rozpoznawania obrazów w kognitywnej analizie zobrazowań medycznych*.Uczelniane Wydawnictwa Naukowo-Dydaktyczne, Kraków 2004.

[OgiTad09] Ogiela M.R., Tadeusiewicz R., *Automatyczna interpretacja obrazów, czyli obrazowanie medyczne wzbogacone sztuczną inteligencją*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 547-580.

[Ohr99] Øhrn A.: *Discernibility and Rough Sets in Medicine: Tools and Applications*. PhD thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway. Advisor: Professor Jan Komorowski. NTNU report 1999:133.

[OliFre05] Oliveira A.M., French C.A.: *Applications of Fluorescence in Situ Hybridization in Cytopathology*. Acta Cytologica. Vol. 49,No.6, 2005 pp.

[Ora03] *Oracle Data Mining Concepts*. 10g Release 1 (10.1); Par No. b10698-01. December 2003

[PatKor02] Patan K., Korbicz J., Mrugalski M.: *Sztuczne sieci neuronowe w układach diagnostyki*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str. 311-352.

[Paw81a] Pawlak Z.: *Classification of objects by means of attributes: an approach to inductive inference*. Prace IPI PAN, ICS PAS REPORTS, No.429, January 1981, Warszawa.

[Paw81b] Pawlak Z.: Rough Sets: basic notions, Prace IPI PAN, 431, Warszawa, Luty 1981, ISSN 0138-0648.

[Paw81c] Pawlak Z.: *Rough relations*. Prace IPI PAN, 435, Warszawa, Luty 1981.

[Paw83] Pawlak Z.: *Systemy informacyjne. Podstawy teoretyczne*. Wydawnictwa Naukowo-Techniczne, Warszawa, 1983.

[PaSk07] Pawlak Z., Skowron A.: *Rudiments of rough sets*. Information Sciences, Vol. 177 (2007), pp. 3–27.

[PawSlo07] Pawlak Z., Słowiński R.: *Zbiory przybliżone we wspomaganiu decyzji.* Techniki informacyjne w badaniach systemowych. Red. Kulczycki P., Heryniewicz O., Kacprzyk J. Wydawnictwa Naukowo-Techniczne. Warszawa 2007. s. 137-158.

[PawSlo02] D271 - Pawlak Z., Słowiński K., Stefanowski J.: *Teoria zbiorów przybliżonych w analizie danych medycznych*. Systemy komputerowe i teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 253-269.

[Pie03] Pieczyński A.: *Reprezentacja wiedzy w diagnostycznym systemie ekspertowym.* Monografia. Lubelskie Towarzystwo Naukowe w Zielonej Górze, 2003.

[PioSta11] Piotrowska E., Stanisławski W.: *Zastosowanie Rough Sets Analysis Toolbox pakietu MATLAB w zadaniach rozpoznawania wzorców.*. XVII Krajowa Konferencja Automatyki, Kielce, 2010, Sterowanie i Automatyzacja: Aktualne problemy i ich rozwiązania, ……., s. …….

[PleLoe01] Plesch A., Loerch T.: *Metafer a Novel Ultra High Throughpt Scanning System for Rare Cell Detection and Automatic Interphase FISH Scoring*. Early Prenatal Diagnosis, Fetal Cells and DNA in the Mother, Present State and Perspectives. 12th Fetal Cell Workshop, Prague, May 2001, pp. 329-339

[PreSlo98] Predki B., Słowiński R., Stefanowski J., Susmaga R., Wilk S.: *ROSE - Software Implementation of the Rough Set Theory*. Rough Sets and Current Trends in Computing. Lecture Notes in Computer Science, 1998, Vol.1424/1998, pp.605-608.

[Prz03] Przytulska M.: *Komputerowa analiza obrazów wentrykulograficznych serca*. Biocybernetyka i inżynieria biomedyczna. T8. Obrazowanie biomedyczne. Red. L. Chmielewski, J. Kulilkowski, A.Nowakowski. Warszawa 2003. Akadem. Oficyna. Wydaw. Exit, s.445-474.

[Ros58] Rosenblatt F.: *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*. Cornell Aeronautical Laboratory, Psychological Review, 1958, v65, No. 6, pp. 386–408.

[Roz79] Rozin B.B.: *Teoria rozpoznawania obrazów w badaniach ekonomicznych*. tł. Grzegorz Napiórkowski, Państwowe Wydawnictwo Naukowe, Warszawa 1979.

[RudGra07] Rudowski R., Grabowski M., Kownacki Ł., Piotrowska-Kownacka D.: *Zastosowania infromatyki w diagnostyce technicznej*. Diagnostyka Procesów i Systemów. Pod red. Korbicz J., Patan K., Kowal M. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2007, s.17-36.

[Rut00] Rutkowska D.: *Inteligentne systemy obliczeniowe i sztuczna inteligencja*. Biocybernetyka i inżynieria biomedyczna 2000. Sieci neuronowe. Tom 6. Akademicka Oficyna Wydawnicza Exit, Warszawa, 2000. s.765-783

[Rut07] Rutkowska D.: *Metody rozmyto-neuronowe w zastoswoaniu do analizy i przetwarzania danych*. Techniki informacyjne w badaniach systemowych. Red. Kulczycki P., Heryniewicz O., Kacprzyk J. Wydawnictwa Naukowo-Techniczne. Warszawa 2007. s. 137-158.

[RutSta02] Rutkowska D., Starczewski A., Nowicki R.: *Sieci neuronowe i logika rozmyta w systemach medycznych*. Systemy komputerowe i teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 287-310.

[Rum03] Rumiński J.: *Rentgenowska tomografia komputerowa*. Biocybernetyka i inżynieria biomedyczna. T8. Obrazowanie biomedyczne. Red. L. Chmielewski, J. Kulilkowski, A.Nowakowski. Warszawa 2003. Akadem. Oficyna. Wydaw. Exit, s.241-305.

[Rus02] Ruszkowski J.: *Systemy z bazą wiedzy ekspertów dla wspomagania diagnostyki różnicowej w praktyce lekarza pierwszego kontaktu*. Systemy komputerowe i teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 49-68.

[Rut05] Rutkowski L.: *Metody i techniki sztucznej inteligencji*. Wydawnictwo Naukowe PWN, Warszawa 2005.

[Sho99] Shomali A.: *Rozpoznawanie mówcy na podstawie długookresowego histogramu amplitud*. Rozprawa doktorska pod kierunkiem prof. zw. dr hab. inż.. Ryszarda Tadeusiewicza. Akademia Górniczo-Hutnicza w Krakowie. Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki. Kraków, 1999.

[SinNgu99] Sinh Hoa, NguyenThi.: *Regularity analysis and its applications in Data Mining*. DoctoralThesis. Thesis supervisor: dr hab. Bohdan S. Chlebus. Warsaw University, Faculty of Mathematics, Warsaw,1999.

[Slo10] Ślot K.: *Rozpoznawanie biometryczne*. Wydawnictwa Komunikacji i Łączności, Warszawa 2010.

[Slo08] Ślot K.: Wybrane zagadnienia biometrii. Wydawnictwa Komunikacji i Łączności, Warszawa, 2008.

[SobMal78] Sobczak W., Malina W.: *Metody selekcji informacji*. Wydawnictwa Naukowo-Techniczne. Warszawa, 1978.

[Sob08] Sobolewski A.: *Selekcja cech w diagnostyce eksploatacyjnej uszkodzeń wirnika indukcyjnego*. Sterowanie i automatyzacja: Aktualne problemy i ich rozwiązania. Red. Malinowski K., Rutkowski L. Akademicka Oficyna Wydawnicza Exit. Warszawa, 2008. s.328-337.

[SomPud02] Somol P., Pudil P.: *Feature selection toolbox*. Pattern Recognition. Vol. 35, Issue 12, December 2002, pp. 2749-2759.

[StaSzy06] Stanisławski W., Szydłowska E.: *Analiza narzędzia Data Mining ORACLE 10g do klasyfikacji komórek nowotworowych w cytometrycznym systemie skaningowym*, XII Konferencja Użytkowników i Deweloperów Oracle - PLOUG, Zakopane, Październik 2006. s. 252-263.

[St07] Stanisz A.: *Przystępny kurs statystyki z zastosowaniem STATISTICA PL na przykładach z medycyny Tom 3. Analizy Wielowymiarowe*. StatSoft Polska Sp. z o.o., Kraków 2007.

[Sta06] Stanisz A.: *Przystępny kurs statystyki z zastosowaniem Statistica PL na przykładach z medycyny. Tom 1. Statystyki podstawowe*. Wydawca: StatSoft Polska, Kraków, 2006.

[StWi05] J.Stefanowski, S.Wilk: Combining Rough Sets and Rule based Classifiers for Handling Imbalanced Data. In: Czaja L. (ed.) Proceedings of Concurrency, Specification and Programming CS&P 2005 Conference, vol. 2, 2005, 497-508.

[Str02] Straszecka E.: *Możliwości wspomagania decyzji w medycynie z zastosowaniem teorii Dempstera-Shafera i zbiorów rozmytych*. Systemy komputerowe i teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 269-287.

[StrzZie02] Strzelecki M., Zieliński K.W.: *Komputerowa analiza obrazu biomedycznego. Wstęp do morfometrii i patologii ilościowej*. Wydawnictwo Naukowe PWN, Warszawa-Łódź 2002.

[Szc00] Szczepaniak P.S.: *Sieci neuronowe i logika rozmyta w medycynie* - przegląd zastosowań. Biocybernetyka i inżynieria biomedyczna 2000. Sieci neuronowe. Tom 6. Akademicka Oficyna Wydawnicza Exit, Warszawa, 2000. s.617-633.

[Szy07a] Szydłowska E.: *Klasyfikacja komórek rakowych z wykorzystaniem technik eksploracji danych*. Zeszyty Naukowe. Elektryka. Politechnika Opolska.2007, Vol. 320, z. 58, s. 159-176.

[Szy07b] Szydłowska E.: *Implementacja równoległa algorytmu selekcji atrybutów z zastosowaniem teorii zbiorów przybliżonych*, IX International PhD Workshop, OWD'2007, 20-23 Październik 2007, s.227-232.

[Szy07c] Szydłowska E.: *Algorytmy selekcji atrybutów w zadaniach eksploracji danych*. XIII Konferencja Użytkowników i Deweloperów Oracle - PLOUG, Kościelisko 2007.s. 261-277.

[Szy08] Szydłowska E.: *Implementation of dimensionality reduction method in analysis of cell morphometric features*. X International PhD Workshop, OWD'2008, 18–21 October 2008. pp.129-132.

[Szy09a] Szydłowska E.: *Discretization of cells morphometric features*. XI International PhD Workshop, OWD'2009, 17-20 October 2009, s.158-159.

[Szy09b] Szydłowska E.: *Data preprocessing using attribute discretization*, III Środowiskowe Warsztaty Doktorantów Politechniki Opolskiej, Głuchołazy, 24-26.06.2009, z.62 Elektryka, Nr 329/2009, s.67-68.

[SzySta08] Szydłowska E., Stanisławski W.: *Teoria zbiorów przybliżonych w zastosowaniu do redukcji cech morfometrycznych*. XVI Krajowa Konferencja Automatyki, Szczyrk, 2008, Sterowanie i Automatyzacja: Aktualne problemy i ich rozwiązania, Wyd. Exit, s. 688-695.

[SzyWol02] - Szymaś J., Stefanowski J., Wolf G., Papierz W., Jarosz B., Nowak M.: *Internetowy system telepatologiczny sprzężony z obrazową bazą danych*. Systemy komputerowe i

teleinformatyczne w służbie zdrowia. Akademicka Oficyna Wydawnicza EXIT. Warszawa, 2002, str. 132-164.

[Tad88] Tadeusiewicz R.: *Sygnał mowy*. Wydaw. Komunikacji i Łączności, Warszawa, 1988.

[Tad09] Tadeusiewicz R.: *Bo to najważniejsze…* Podstawy inżynierii biomedycznej, Tom 1. Pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 9-44.

[Tad91] Tadeusiewicz R., Flasiński M.: *Rozpoznawanie obrazów*. Wydawnictwo Naukowe PWN, Warszawa 1991.

[Tad97] Tadeusiewicz R., Korohoda P.: *Komputerowa analiza i przetwarzanie obrazów*. Wydaw. Fundacji Postępu Telekomunikacji, Kraków, 1997.

[TadOgi02] Tadeusiewicz R., Ogiela M.: *Automatyczne rozumienie obrazów*. XIV Krajowa Konferencja Automatyki, Zielona Góra, 24-27 czerwca 2002. s.51-60.

[TadOgi07] Tadeusiewicz R., Ogiela M.: *New diagnostics perspectives connected with a concept of automatic pattern understanding*. Fault Diagnosis and Fault Tolerant Control. Edit. Korbicz J., Patan K., Kowal M. Academic Publishing House Exit, Warszawa 2007. s.43-49.

[TadOgi09a] Tadeusiewicz R., Ogiela M.: *Klasyczne obrazowanie rentgenowskie. Technika z przeszłością i z przyszłością*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 373-388.

[TadOgi09b] Tadeusiewicz R., Ogiela M.: *Komputer pozwala widzieć więcej i lepiej. Tomografia i jej odmiany*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 395-406.

[TadPal07] Tadeusiewicz R., Paliwoda-Pękosz G., Lula P.: *Metody sztucznej inteligencji i ich zastosowania w ekonomii i zarządzaniu*. Wydawnictwa Uniwersytetu w Krakowie. Kraków, 2007.

[TanSte06] Tanenbaum A.S., Steen M.: *Systemy rozproszone : zasady i paradygmaty*. tłum. Płoski Z., Warszawa,Wydawnictwa Naukowo-Techniczne, 2006.

[Tim07] Timofiejczuk A.: *Zastosowanie algorytmów ewolucyjnych w procesie wnioskowania* diagnostycznego. Diagnostyka Procesów i Systemów. Pod red. Korbicz J., Patan K., Kowal M. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2007, s.101-108.

[TomPuc07] Tomczyk A., Puchała D., Stokfiszewski K., Szczepaniak P.: *System wspomagający diagnostykę obrazową*. Diagnostyka Procesów i Systemów. Pod red. Korbicz J., Patan K., Kowal M. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2007, s.133-140.

[Tra09] Trawińska A.: *Zastosowanie do celów biometrycznych analizy mowy wykonywanej przy użyciu narzędzi inżynierii biomedycznej*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 253-275.

[TsLe08] Tsai C.J., Lee C.I., Yang W.P.: *A discretization algorithm based on Class-Attribute Contingency Coefficient*. Information Sciences, Volume 178, Issue 3, 2008, pp. 714-731.

[VukCur06] Vuk M., Curk T.: *ROC Curve, Lift Chart and Calibration Plot*. Metodoloski zvezki, Vol. 3, No. 1, 2006, pp. 89-108.

[WajWoj09] Wajs W., Wojtowicz H., Wojtowicz J.: *Klasyfikacja odcisków palców przy użyciu algorytmów inteligencji obliczeniowej w diagnostyce medycznej*. Systemy wykrywające, analizujące i tolerujące usterki. Red. Zdzisław Kowalczuk. Pomorskie Wydawnictwo Naukowo-Techniczne. Gdańsk, 2009. s.251-258.

[Wal07] Walaszek-Babiszewska A.: *Construction of Fuzzy Models Using Probability Measures of Fuzzy Events*, in Proc.13th IEEE/IFAC Int. Conf. on Methods and Models in Automation and Robotics, MMAR 2007, Szczecin, Poland, 661-666.

[WalBla08] Walaszek-Babiszewska A., Błaszczyk K., Czabak A.: *Budowa rozmytych modeli procesów stochastycznych przy użyciu reguł asocjacji*. Sterowanie i automatyzacja: Aktualne problemy i ich rozwiązania. Red. Malinowski K., Rutkowski L. Akademicka Oficyna Wydawnicza Exit. Warszawa, 2008. s.x-x

[WaMa99] Walczak B., Massarat D.L.: *Rough sets theory*. Chemometrics and Intelligent Laboratory Systems, Elsevier Science, Vol. 47, No 1, 1999, pp. 1-16.

[WitKor02] Witczak M., Korbicz J.: *Programowanie genetyczne w diagnostyce uszkodzeń i identyfikacji nieliniowych systemów dynamicznych*. Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania. Pod red. Korbicz J., Kościelny J. M., Kowalczuk Z., Cholewa W., WNT, Warszawa 2002, str 427-464.

[Wlo09] Włodarczyk M.: *Rezonans magnetyczny*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 407-418.

[WnuSyf09] Wnuk P., Syfert M., Kościelny J.: *Inteligentny system diagnostyki wspomagania sterowania procesów przemysłowych DIASTER*. Systemy wykrywające, analizujące i tolerujące usterki. Red. Zdzisław Kowalczuk. Pomorskie Wydawnictwo Naukowo-Techniczne. Gdańsk, 2009. s.57-64.

[Wsz09] Wszołek W.: *Analiza dźwięków mowy do celów medycznych*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 229-252.

[XuKrz92] Xu L., Krzyżak A., Suen C.Y.: *Methods of combining multiple classifiers and their applications to handwriting recognition*. IEEE Transactions on Systems, Man and Cybernetics. 1992, ol.22, No.3, pp.418-435.

[ZajWis03] Zając M., Wiśniewska M.: *Zastosowanie fluoroscencyjnej hybrydyzacji in situ (FISH) w identyfikacji zmian materiału genetycznego u osób z niepełnosprawnością intelektualną*. Nowiny Lekarskie 2003, 72, 1, s. 9-13.

[Zaz09] Zazulak M.: *Obrazowanie dźwiękiem, czyli USG*. Podstawy inżynierii biomedycznej, Tom 1, pod red. R. Tadeusiewicza, P. Augustyniaka, Wydawnictwa AGH, Kraków 2009. str. 509-512.

[ZhHu04] Zhang G., Hu L, Jin W.: *Discretization of Continous Attributes in Rough Set Theory and Its Application*, CIS 2004, LNCS 3314, pp.1020-1026.

[Zi03] Zieliński K.: *Parametry morfometryczne wykorzystywane w pomiarach biomedycznych*. Biocybernetyka i inżynieria biomedyczna. T8. Obrazowanie biomedyczne. Red. L. Chmielewski, J. Kulilkowski, A.Nowakowski. Warszawa 2003. Akadem. Oficyna. Wydaw. Exit, s.165-177.

[ZieStr03] Zieliński K., Strzelecki M.: *Wybrane zagadnienia ocen ilościowych i przetwarzania obrazów*. Biocybernetyka i inżynieria biomedyczna. T8. Obrazowanie biomedyczne. Red. L. Chmielewski, J. Kulilkowski, A.Nowakowski. Warszawa 2003. Akadem. Oficyna. Wydaw. Exit, s.4.

Zasoby sieci Internet

[wAbot10] D158 - Strona internetowa Abbott Laboratories Inc
http://www.abbottmolecular.com/UroVysion_5181.aspx (z dnia 2010-01-18)

[wCAIM09] Matlab File Exchange, CAIM Discretization Algorithm by Guangdi Li, 04 Jun 2009 (http://www.mathworks.com/matlabcentral/fileexchange/24344-caim-discretization-algorithm).

[wCACC09] Matlab File Exchange, Discretization algorithms: Class-Attribute Contingency Coefficient by Guangdi Li, 04 Jun 2009
http://www.mathworks.com/matlabcentral/fileexchange/24343-discretization-algorithms-class-attribute-contingency-coefficient).

[wMym] Dokumentacja biblioteki mYm: http://sourceforge.net/projects/mym/

[wMysql] Dokumentacja biblioteki mysql: http://www.mmf.utoronto.ca/resrchres/mysql/

[wStat10a] Strona internetowa. Glosariusz statystyczny,

[wStat10b] Statistica 9. Electronic Manual – StatSoft Polska Sp. z o.o. 2010.

Wykaz symboli i skrótów

A	Zbiór atrybutów (przestrzeń atrybutów)
a_j	j-ty atrybut
C	Zbiór atrybutów warunkowych
c_i	j-ty atrybut warunkowy
D	Zbiór atrybutów decyzyjnych
d_j	j-ty atrybut decyzyjny
P	Podzbiór atrybutów
p	Liczba atrybutów
U	Uniwersum, zbiór obiektów x_i, dziedzina zadania
n	Liczba obiektów uniwersum
t	Liczba obiektów zbioru uczącego
X_i	i-ty podzbiór uniwersum
x_i	i-ty obiekt uniwersum
V_{a_j}	Dziedzina j-tego atrybutu
δ_j	Liczba klas j-tego atrybutu decyzyjnego
I_{a_j}	Zbiór przedziałów dyskretyzacji j-tego atrybutu
$I_{a_j}^k$	k-ty przedział dyskretyzacji j-tego atrybutu
η	Liczba przedziałów dyskretyzacji
v_i^j	Wartość j-tego atrybutu dla i-tego obiektu
M	Macierz rozróżnialności
m_{ij}	Elementy macierze rozróżnialności
T	Tablica rozróżnialności
$t_{ij,k}$	Elementy tablicy rozróżnialności
k	Stopień zależności atrybutów
$\mu_{\tilde{P}}(X)$	$\tilde{P}$-dokładność aproksymacji zbioru
$\beta_{\tilde{P}}(X)$	$\tilde{P}$-dokładność aproksymacji rodziny zbiorów
$\gamma_{\tilde{P}}(X)$	$\tilde{P}$-jakość aproksymacji rodziny zbiorów
f	Funkcja informacyjna
Q	Macierz kwantyzacji (ang. Quanta matrix)
CC	Współczynnik kontyngencji
Err_{AB}	Błąd klasyfikacji (Błędne odrzucenie klasy A poprzez przypisanie do klasy B)
TP_A	Prawidłowe wskazanie klasy A
TN_A	Prawidłowe wskazanie innej klasy niż A
FP_A	Błędne wskazanie klasy A
FN_A	Błędne odrzucenie klasy A
E	Entropia
Z_i	i-ta składowa główna
$\tilde{D}$	Relacja równoważności ze względu na zbiór atrybutów D
S	Odwzorowanie odpowiadające rozpoznawaniu wzorców
$\hat{S}$	Algorytm odwzorowania odpowiadający rozpoznawaniu wzorców

Wykaz rysunków

Wykaz tabel

Wykaz programów źródłowych

Załącznik A. Programy i przykłady

1. Wyznaczanie reduktów z zastosowaniem tablicy prawdy

Niech system informacyjny będzie opisany trzema atrybutami: a_1, a_2, a_3. Niech tabela A.1 będzie tabelą rozróżnialności omawianego systemu informacyjnego.

Tabela A.1. Przykład tablicy rozróżnialności (TD)

a_1	a_2	a_3
1	1	0
0	1	1

Funkcja rozróżnialności w postaci koniunkcyjnej, odpowiadająca tabeli rozróżnialności, ma następującą postać:

$$f_{D,CNF}(TD) = (a_1^* \vee a_2^*) \wedge (a_2^* \vee a_3^*). \quad (A.1)$$

Tablica prawdy, wykorzystana do poszukiwania postaci alternatywnej funkcji rozróżnialności, ma postać przedstawioną w tabeli A.2a. Początkowo funkcja wyjścia tablicy prawdy $f_{0,TT}$ przyjmuje dla wszystkich wierszy wartość 1.

Porównanie pierwszego wiersza tablicy rozróżnialności z tablicą prawdy zamieszczono w tabeli A.2b. Funkcja przyjmuje wartości 0 dla wierszy, które odpowiadają danemu zbiorowi atrybutów rozróżniających. Dla pierwszego wiersza tablicy TD będzie to wiersz 0 i 1 tablicy TT. Podobne porównanie przeprowadzono dla wiersza drugiego tablicy TD (tab. A.2c).

Tabela A.2. Przykład interpretacji szablonów reguł decyzyjnych

a)

	a_1^*	a_2^*	a_3^*	$f_{0,TT}$
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

b)

	a_1^*	a_2^*	a_3^*	$f_{1,TT}$
0	**0**	**0**	0	**0**
1	**0**	**0**	1	**0**
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

c)

	a_1^*	a_2^*	a_3^*	$f_{2,TT}$
0	0	**0**	**0**	**0**
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	**0**	**0**	**0**
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

d)

	f_0	f_1	f_2	f_{TT}
0	1	**0**	**0**	**0**
1	1	**0**	1	**0**
2	1	1	1	1
3	1	1	1	1
4	1	1	**0**	**0**
5	1	1	1	1
6	1	1	1	1
7	1	1	1	1

Ostateczna postać funkcji wyjścia jest iloczynem logicznym funkcji f_{TT} otrzymanych w każdym kroku porównania.

$$f_{TT} = f_{0,TT} \wedge f_{1,TT} \wedge \ldots \wedge f_{n,TT}, \quad (A.2)$$

gdzie n jest liczbą wierszy tablicy rozróżnialności. Dla tabeli A.2d funkcja będzie miała postać:

$$f_{TT} = (a_2^*) \vee (a_2^* \wedge a_3^*) \vee (a_1^* \wedge a_3^*) \vee (a_1^* \wedge a_2^* \wedge a_3^*). \quad (A.3)$$

Stosując prawo pochłaniania dla funkcji f_{TT} otrzymuje się funkcję wyrażoną wzorem.

$$f_{TT} = (a_2^*) \vee (a_1^* \wedge a_3^*). \quad (A.4)$$

2. Interpretacja reguł decyzyjnych

Dla obiektu opisanego tabelą A.3a i szablonem opisanym tabelą A.3b otrzymujemy reguły opisane w tabeli A.3c. Reguły należy odczytywać w następujący sposób: jeżeli atrybut a_1 ma wartość 5 i atrybut a_3 ma wartość 2 to wybierz klasę 2, lub jeżeli atrybut a_2 przyjmuje wartość 4 to wybierz klasę 2.

Tabela A.3. Przykład interpretacji szablonów reguł decyzyjnych

a)

a_1	a_2	a_3	Klasa
5	4	2	2

b)

a_1	a_2	a_3
1	*	1
*	1	*

c)

a_1	a_2	a_3	Klasa
5	*	2	2
*	4	*	2

3. Realizacja programu rsamrun.m

Plik rsamrun.m jest funkcją startową modułu RSAm. Zadaniem funkcji jest uruchomienie programu model_main.m, w którym zaimplementowano proces uczenia nadzorowanego. Program model_main.m może być wywołany bezpośrednio, jednak spowoduje to utworzenie wielu zmiennych tymczasowych w przestrzeni środowiska MATLAB. Zmienne zwracane przez funkcję rsamrun.m zawierają jedynie niezbędne wyniki uczenia nadzorowanego. Ze względu na rozmiary pliku model_main.m, jego kod źródłowy dostępny jest tylko w wersji elektronicznej.

Listing A.1. Przykładowy kod programu rsamrun.m

```
1
2       function [rsam_classification rsam_reduction rsam_discretization]=…
3               rsamrun(option_f_type,option_discretization,option_classification_types,...
4                               DataSetNo,DataSetsNamesList,AttrSetNo,AttrList,ClassAttr,typ_db,nazwa_db)
5
6           model_main
7
8       end
9
```

4. Realizacja programu rsaminit.m

Plik rsaminit.m definuje metody klasyfikacji, redukcji i dyskretyzacji wymagane do wywołania funkcji rsamrun:

- option_f_types - wektor definiujący metody redukcji cech. W programie rsamprerun.m, funkcja rsamrun wywoływana jest w iteracjach dla zmiennej option_f_type, która odpowiada kolejno wartościom wektora option_f_types.
- optionDataSetsNamesList, optionDataSetsNo - macierz optionDataSetsNamesList zawiera listę zestawów nazw zbiorów danych wykorzystanych w modelowaniu. Nazwy zbiorów przechowywane są w polach tekstowych, przy czym, pierwsze pole odpowiada nazwie zbioru uczącego. Kolejne pola tekstowe odpowiadają zbiorom walidacyjnym. W zmiennej optionDataSetsNamesList może być zdefiniowanych wiele zestawów zbiorów danych. Numery zestawów, dla których mają być

przeprowadzone obliczenia, zdefiniowano w wektorze optionDataSetsNo. W programie rsamprerun.m, funkcja rsamrun wywoływana jest w iteracjach dla zmiennej DataSetsNo, która odpowiada kolejnym wartościom wektora optionDataSetsNo. Zmienna DataSetsNo wskazuje na numer wiersza macierzy optionDataSetsNamesList, zawierający nazwy zbiorów, dla których maja być przeprowadzone obliczenia (DataSetsNamesList).

- optionAttrSetsList, optionAttrSetsNo – macierz optionAttrSetsList zawiera listę zbiorów atrybutów wykorzystanych analizach. Każdy wiersz macierzy definiuje inny zbiór cech. Informacja o zbiorach cech, które mają być wykorzystywane w obliczeniach przechowywana jest w wektorze optionAttrSetsNo. W programie rsamprerun.m, funkcja rsamrun wywoływana jest w iteracjach dla zmiennej AttrSetsNo, która odpowiada wartościom wektora optionAttrSetsNo. Zmienna AttrSetsNo wskazuje na numer wiersza macierzy optionAttrSetsList, zawierający zbiór atrybutów, dla którego maja być przeprowadzone obliczenia (AttrList).
- posible_discretization – zawiera listę metod dyskretyzacji, jakie mają być wykorzystane w modelowaniu. W przypadku uruchomienia programu prersamrun.m, zmienna option_discretization, może być kombinacją różnych typów dyskretyzacji dla redukcji i klasyfikacji. Definicje tych metod przechowywane są w macierzach posible_disc_f oraz posible_disc_k. Każda z macierzy może odpowiadać zmiennej posible_discretization. Struktura kolumn wymienionych powyżej trzech macierzy odpowiada strukturze kolumn zmiennej option_discretization. Ostateczna postać macierzy option_discretization wykorzystana w programie prersamrun.m jest kombinacją wystąpień wszystkich wierszy zmiennej posible_disc_f z wierszami zmiennej posible_disc_k w postaci: option_discretization=[posible_disc_f; posible_disc_k].

Istotnym elementem klasyfikacji jest zastosowanie metod walidacji. Walidacja polega na utworzeniu podzbiorów danych uczących i przeprowadzeniu obliczeń dla każdego podzbioru. W tym celu w module RSAm wykorzystano program partycjonowanie. Do budowy zbiorów walidacyjnych zastosowano funkcję cvpartition, będącą elementem narzędzia Statistica Toolbox. W module RSAm zaimplementowano dwie metody walidacji: holdout oraz kfold. Metoda holdout realizuje podział zbioru danych na dwa zbiory: uczący i walidacyjny. Liczba obiektów w zbiorze walidacyjnym stanowi 10% obiektów zbioru wejściowego. Zbiór uczący zapisywany jest w nowej zmiennej o nazwie zcvh01, natomiast zbiór walidacyjny w zmiennej zcvh02. Druga metoda walidacji, kfold, wymaga zdefiniowania liczby podzbiorów. Wynikiem partycjonowania jest utworzenie nowych zmiennych o nazwach zcvhx1, zcvhx2, gdzie x odpowiada numerowi kolejnego zestawu testowego.

Program partycjonowanie przeprowadza podział zbioru zdefiniowanego w zmiennej optionDataSetsNamesList. W wyniku partycjonowania wartość zmiennej optionDataSetsNamesList, jest aktualizowana w taki sposób, aby zawierała nazwy nowoutworzonych zbiorów testowych. Każdorazowe uruchomienie partycjonowania prowadzi do utworzenia zbiorów testowych i walidacyjnych, zawierających inne obiekty. Dlatego, w przypadku poszukiwania modelu optymalnego i wielokrotnego uruchamiania programu rsamprerun, nie zaleca się powtarzania partycjonowania, które zmienia wartości zmiennej optionDataSetsNamesList. W tym celu warto skorzystać z programu rsaminit2.m, który będzie zawierał tylko deklarację wartości zmiennych optionDataSetsNo, optionAttrSetsNo, option_classification_types, option_f_types.

Listing A.2. Przykładowy kod programu rsaminit.m

```
%% ********************************************************************************
% rsaminit.m
% Inicjalizacja zmiennych dla programu rsamprerun.m oraz dist_rsamprerun.m
%% ********************************************************************************
% Metody dyskretyzacji
posible_discretization(1,:)=[0,1,5];
posible_discretization(2,:)=[0,1,10];
posible_discretization(3,:)=[0,1,20];
posible_discretization(4,:)=[0,0,0];
posible_disc_f=posible_discretization;
posible_disc_k=posible_discretization;
% Metody redukcji cech
option_f_types=[0,1,22,3,4];
% Metody klasyfikacji
option_classification_types =[1,23,3];
% Zestawy zbiorów danych
clear optionDataSetsNamesList
optionDataSetsNo=[1,3,5];
optionDataSetsNamesList(:,:,1)=['zcvk01'; 'zcvk02'];
optionDataSetsNamesList(:,:,2)=['zcvk11'; 'zcvk12'];
optionDataSetsNamesList(:,:,3)=['zcvk21'; 'zcvk22'];
optionDataSetsNamesList(:,:,4)=['zcvk31'; 'zcvk32'];
optionDataSetsNamesList(:,:,5)=['zcvk41'; 'zcvk42'];
% Zestawy zbiorów atrybutów
optionAttrSetsNo=1:1:3;
optionAttrSetsList=zeros(3,220);
optionAttrSetsList(1,[3,4,23,24,25,27,28,29,31,32,33,34,35])=1;
optionAttrSetsList(2,3:100)=1;
optionAttrSetsList(3,[3,5,26,38,84,133,188,203,199,206,210])=1;
ClassAttr=2;
% Zrodlo danych
DBTyp=1;
if DBTyp==1
   DBSource='obliczenia';
else
   for i=1:5
      eval(['DBSource.zcvk',num2str(i-1),'1=zcvk',num2str(i-1),'1;']);
      eval(['DBSource.zcvk',num2str(i-1),'2=zcvk',num2str(i-1),'2;']);
   end
end
% Waldacja krzyżowa
optionCrossValidation=2;
if optionCrossValidation~=0
   partycjonowanie
end
% Uruchomienie obliczen
rsamprerun
```

5. Realizacja programu rsamprerun.m

Program rsamprerun.m uruchamiany jest w wersji jednostanowiskowej z możliwością obliczeń rozproszonych dla złożonych zadań (poziom 1 rozproszenia). Przykładowy kod programu zamieszczono poniżej.

Listing A.3. Przykładowy kod programu rsamprerun.m

```
1       j=1;
2       k=1;
3       i=1;
4       for i_n_dis=1:NumOfDis
5             option_discretization=[posible_disc_f(i_n_dis,:);posible_disc_k(i_n_dis,:)]
6             for ftyp=1:size(option_f_types,2)
7               option_f_type=option_f_types(ftyp);
8               for attrset_no=1:size(optionAttrSetsNo,2)
9                  for mod_no=1:size(optionDataSetsNo,2)
10                    DataSetNo = optionDataSetsNo(mod_no);
11                    DataSetsNamesList=optionDataSetsNamesList(:,:,DataSetNo)
12                    AttrSetNo=optionAttrSetsNo(attrset_no);
13                     clear AttrList;
14                     AttrList=optionAttrSetsList(AttrSetNo,:);
15                     [rsam_classification  rsam_reduction rsam_discretization]= ...
16                         rsamrun(option_f_type,option_discretization, option_classification_types,...
17                         DataSetNo,DataSetsNamesList,...
18                         AttrSetNo,AttrList,ClassAttr,...
19                         DBTyp,DBSource);
20                     try
21                       eval(['rsam_classification',num2str(j),'(i,1)=rsam_classification']);
22                       eval(['rsam_reduction',num2str(j),'(i,1)=rsam_reduction']);
23                       eval(['rsam_discretization',num2str(j),'(i,1)=rsam_discretization']);
24                      i=i+1;
25                    catch error_txt
26                       disp(['blad zapisu ... ',error_txt.message]);
27                       komunikat_tresc=['... ... ... Blad zapisu: ',error_txt.message, '. Ustawiono j=',num2str(j+1)];
28                       komunikat
29                       j=j+1;
30                       i=1;
31                       eval(['rsam_classification',num2str(j),'(i,1)=rsam_classification']);
32                       eval(['rsam_reduction',num2str(j),'(i,1)=rsam_reduction']);
33                       eval(['rsam_discretization',num2str(j),'(i,1)=rsam_discretization']);
34                       i=i+1;
35                    end
36                    k=k+1;
37              end
38             end
39          end
40      end
41
```

6. Realizacja programu dist_rsamprerun.m

Program dist_rsamprerun umożliwia uruchomienie programu rsamrun równolegle na węzłach klastra (poziom 1 rozproszenia).

Listing A.4. Przykładowy kod programu dist_rsamprerun.m

```
1       %Adres lub nazwa komputera na którym uruchomiono jobmanagera
2       jobmanager_ip='10.0.5.10';
3       %Nazwa jobmanagera  uruchomionego na adresie jobmanager_ip
4       jobmanager_name='MyJobMgr';
5
6       %Zdefiniowanie maganegra zadan
```

```
7          eval(['jm= findResource(''scheduler'', ''type'',
8      ''jobmanager'',''name'',''',jobmanager_name,''',''LookupURL'','''',jobmanager_ip,''');']);
9
10         j_job = createJob(jm);
11         set(j_job, 'fileDependencies', {'obl_init'});
12
13     %Przygotowanie obiektu pracy
14     k=1;
15     for i_n_dis=1:NumOfDis   % takie dziwne nazwy zmiennych ale lepiej zey sie nie pokrylo z innymi
16
17
18           fig_no=0;
19           option_discretization=[posible_disc_f(i_n_dis,:);
20                                  posible_disc_k(i_n_dis,:)]
21
22           for ftyp=1:size(option_f_types,2)
23
24             option_f_type=option_f_types(ftyp);
25             for attrset_no=1:size(optionAttrSetsNo,2)
26                for mod_no=1:size(optionDataSetsNo,2)
27
28                   for klas_no=1:size(option_classification_types,2)
29                      k
30                      DataSetNo = optionDataSetsNo(mod_no);
31                      DataSetsNamesList=optionDataSetsNamesList(:,:,DataSetNo)
32
33                      AttrSetNo=optionAttrSetsNo(attrset_no)
34
35                      ClassNo = option_classification_types(klas_no)
36
37                       clear AttrList;
38                       AttrList=optionAttrSetsList(AttrSetNo,:);
39
40                       model_parametry(k,:)=[ClassNo option_f_type AttrSetNo DataSetNo posible_disc_f(i_n_dis,2:3)];
41
42                       jtask(k)=createTask(j_job, @obl_init,2, {option_f_type ...
43                          option_discretization ...
44                          ClassNo ...
45                          DataSetNo DataSetsNamesList ...
46                          AttrSetNo AttrList ClassAttr ...
47                          typ_db nazwa_db });
48
49
50                      k=k+1;
51                   end
52                end
53               end
54           end
55
56     end
57              %Wysylanie obiektu pracy
58             submit(j_job);
59           il_zad=k-1;
60
61              %Zbieranie wynikow obiektu pracy
62
63      i=1;
64      j=1;
65      tab_przekroczonych_limitow=[];
66     for k =1:il_zad
67
```

```
68              stan_oczekiwania=waitForState(jtask(k),'finished',600);
69              if stan_oczekiwania==true
70                 try
71                  eval(['model_wyniki',num2str(j),'(i,1)=jtask(k).OutputArguments{1};'])
72                  eval(['model_redukty',num2str(j),'(i,1)=jtask(k).OutputArguments{2};'])
73                  i=i+1;
74                 catch error_txt
75
76                    disp(['blad zapisu ... ',error_txt.message]);
77
78                    j=j+1;
79                    i=1;
80                    eval(['model_wyniki',num2str(j),'(i,1)=jtask(k).OutputArguments{1};'])
81                    eval(['model_redukty',num2str(j),'(i,1)=jtask(k).OutputArguments{2};'])
82                    i=i+1;
83
84                 end
85              else
86                 disp(['blad ... przekroczono czas oczekwiania ']);
87              end
88        end
89
90     %Usuniecie obiektu pracy
91     destroy(j_job);
92
```

7. Realizacja programu dist_df_sd

Listing A.5. Realizacja program rozproszonego wyznaczania funkcji rozróżnialności (dist_df_sd.m)

```
1      function [redukty]= dist_df_sd(X)
2
3         %Adres lub nazwa komputera na którym uruchomiono jobmanagera
4         jobmanager_ip='10.0.5.10';
5         %Nazwa jobmanagera  uruchomionego na adresie jobmanager_ip
6         jobmanager_name='MyJobMgr';
7
8         %Zdefiniowanie maganegra zadan
9         eval(['jm= findResource(''scheduler'', ''type'',
10    ''jobmanager'',''name'','''',jobmanager_name,''','LookupURL','''',jobmanager_ip,''');']);
11
12         %Rozpoczenie realizacji rozproszonej
13
14        [n,m]=size(X);
15        X_DF=true(2^m,1);
16        il_pet=n-1;
17
18        %Zmienne ro optymalizacji czasu wykonania
19        liczebnosc=5000;
20        ile_ob=25;
21        liczba_petli=ceil(il_pet/liczebnosc);
22
23         for p_i=1:liczba_petli
24
25             %Przygotowanie obiektu pracy
26             j_job = createJob(jm);
27             set(j_job, 'fileDependencies', {'dist_df_sd_step.m'});
28
```

```
29                      if p_i==liczba_petli
30                         liczba_ob_od=(p_i-1)*liczebnosc+1;
31                         liczba_ob_do=il_pet;
32                      else
33                         liczba_ob_od=(p_i-1)*liczebnosc+1;
34                         liczba_ob_do=p_i*liczebnosc;
35                      end
36
37                j=1;
38
39                krok=ile_ob;
40                for i=liczba_ob_od:krok:liczba_ob_do
41                      if i+krok>liczba_ob_do
42                         krok=liczba_ob_do-i+1;
43                      end
44                         jtask(j)=createTask(j_job, @dist_df_sd_step,2, {i krok});
45                         j=j+1;
46                end
47
48             %Wysylanie obiektu pracy
49             submit(j_job);
50
51
52              il_zad=j-1;
53
54              %Zbieranie wynikow obiektu pracy
55              for j =1:il_zad
56
57                    j
58                    krok=ile_ob;
59
60                    if j==1
61                       od_w=liczba_ob_od+j-1;
62                       do_w=od_w+krok-1;
63                    else
64
65                       od_w=liczba_ob_od+(j-1)*krok;
66                        if od_w+krok>liczba_ob_do
67                            krok=liczba_ob_do-od_w+1
68                        end
69                       do_w=od_w+krok-1;
70                     end
71
72                    stan_oczekiwania=waitForState(jtask(j),'finished',300);
73
74                    if stan_oczekiwania==true
75                       X_DF=X_DF & jtask(j).OutputArguments{1};
76                    else
77                       disp(['blad obliczen ... przekroczono czas oczekwiania '])
78                    end
79              end
80              %Usuniecie obiektu pracy
81             destroy(j_job);
82          end
83
84          redukty=df_min(X_DF);
85      end
86
87
88
```

Listing A.6 Realizacja programu rozproszonego wyznaczania fragmentów funkcji rozróżnialności (dist_df_sd_step.m)

```
function [X_DF,czas_obliczen]=dist_df_sd_step(i, ile_ob)

    % POBRANIE DANYCH Z BAZY
    addpath(genpath('C:\ep\rsatoolbox'));
    mym( 'open', '10.0.5.10', 'login', 'haslo' );
    mym('use','nazwa_bazy');

    nazwa_tabeli_bazy='macierzklas';
    polecenie=['"SELECT wartosc FROM ',nazwa_tabeli_bazy,' WHERE id_tab ="{Si}"", 1'];
    eval(['tempMacierzKlas=mym(',polecenie,');']);
    MacierzKlas=cell2mat(tempMacierzKlas.wartosc);
    clear tempMacierzKlas

    nazwa_tabeli_bazy='macierzx';
    polecenie=['"SELECT wartosc FROM ',nazwa_tabeli_bazy,' WHERE id_tab ="{Si}"", 1'];
    eval(['tempX=mym(',polecenie,');']);
    X=cell2mat(tempX.wartosc);
    clear tempX

    mym('close');

    % REALIZACJA ZADAŃ

    od_obiektu=i;
    [n,m]=size(X);
    X_DF=true(2^m,1);

    for nri=1:ile_ob

        ktory=od_obiektu+nri-1;

        %Wyznaczanie tablicy rozróżnialności
        tempX_DT=dt_sd(X(ktory:n,:), MacierzKlas(ktory:n));

        % Wyznaczanie funkcji rozróżnialności
        if (size(tempX_DT,1)>0)
            [tempX_DF]= df_step(tempX_DT);
        else
            tempX_DF=true(2^m,1);
        end
        X_DF=X_DF & tempX_DF;
    end

end
```

Załącznik B. Parametry morfometryczne

W załączniku przedstawiono wykaz parametrów morfometrycznych wykorzystanych we wstępnej diagnostyce nowotworu pęcherza moczowego. Załącznik podzielono na trzy części.

W części pierwszej zamieszczono cechy pogrupowane według typu miary. Każdej cesze przypisano numer, który odpowiada indeksowi cechy w systemie Metafer.

W części drugiej przedstawiono opis każdej cechy. Cechy posortowano według przypisanych im numerów.

W części trzeciej zamieszczono wykaz cech użytych w analizach. W przypadku cech sparametryzowanych podano odpowiednie wskaźniki.

1. Grupy cech

Tabela 1. Cechy opisujące rozmiar

Nr cechy w systemie	Nazwa cechy
1	Contour Area
2	Circumference
15	Rel. Area X% Int.
16	Abs. Area X% Int.
86	Mean Contour Radius
87	Min. Contour Radius
88	Max. Contour Radius

Tabela 2. Cechy opisujące kształt

Nr cechy w systemie	Nazwa cechy
107	Aspect Ratio
28	Eccentricity
3	Irregularity
4	Roundness
5	Max. Conc. Depth
6	Max. Conc. Area
7	Tot. Conc. Area

Tabela 3. Cechy opisujące intensywność

Nr cechy w systemie	Nazwa cechy
9	Mean Intensity
10	Intensity S.D.
11	Max. Intens. Rel.
12	Min. Intens. Rel
13	Max. Intens. Abs.
14	Min. Intens. Abs.

Tabela 4. Cechy opisujące rozkład

Nr cechy w systemie	Nazwa cechy
53	Center Dist. Moment
54	Radial Dist. Moment
55	Rel. Center Intens.
56	Rel. Radial Intens.
57	Rel. Granularity
59	Pixel Granularity
60	Granularity SDev

Tabela 4. Cechy opisujące obiekty

Nr cechy w systemie	Nazwa cechy
61	N of Objects
62	Mean Rel. Obj. Area
63	Mean Abs. Obj. Area
64	Sdev Rel. Obj. Area
65	SDev Abs. Obj. Area
66	Mean Obj. Intensity
67	SDev Obj. Intensity
91	Mean Center Dist.
92	SDev Center Dist.
93	Mean Obj. Dist.
94	SDev Obj. Dist.

2. Wykaz cech według numerów przypisanych w systemie Metafer

1 - Contour Area

“Total Area within Contour in µm²” – the number of pixels within the nucleus contour as defined by the object threshold in the counterstain channel is converted to µm² using the known pixel size of the CCD camera and the microscope magnification defined in the parameter set (see above).

2 - Circumference

“Circumference of Contour in µm” – the nucleus contour length calculated in µm.

3 - Irregularity

“Irregularity of Contour (0..1)” – the center of the nucleus and the distance of all contour pixels from the center are calculated. If A is the mean center distance of the 10% contour pixels with the greatest distance from the center, and B is the mean center distance of the 10% contour pixels with the smallest distance from the center, the irregularity is defined by the formula:

$$\sqrt{1-\frac{B}{A}}$$

(in analogy to the calculation of the numerical eccentricity of an ellipse, see below). The feature value is always between 0 and 1 (as A >= B), for a perfect circle the result is 0.

4 - Roundness

"Roundness = 4 Pi Area / Circumference² (0..1)" – this global test value computed from area and circumference also ranges between 0 and 1. The value for a perfect circle is 1.

5 - Max. Conc. Depth

"Maximum Relative Concavity Depth (0..1)" – the system determines the convex hull of the contour and then finds the concavities. The depth of the deepest concavity is divided by the equivalent diameter of the nucleus (which is the diameter of a circle with the same area as the nucleus).

6 - Max. Conc. Area

"Relative Area of Deepest Concavity (0..1)" – the system determines the convex hull of the contour and then finds concavities. The area of the deepest concavity (i.e. the area between the contour and the convex hull) is divided by the total area of the nucleus.

7 - Tot. Conc. Area

"Total Relative Concavity Area (0..1)" – the system determines the convex hull of the contour and then finds concavities. The total area of all concavities (i.e. the total area between the contour and the convex hull) is divided by the total area of the nucleus.

9 - Mean Intensity

"Mean Intensity" – is the total intensity, divided by the number of pixels within the cell image. Pixels with an intensity of 0 are not counted, so you can exclude the background by using the "ApplyMask" operation.

10 - Intensity S.D.

"Standard Deviation of Intensity" – is the standard deviation of the intensity values of all pixels in the cell image, using the same scale as the total and mean intensity. Pixels with an intensity of 0 are not counted, so you can exclude the background by using the "ApplyMask" operation.

11 - Max. Intens. Rel.

"Maximum Intensity, Relative Spot Area 1/X" – for several of the remaining feature parameters a measurement spot is used, and there are two ways to define the size of this spot. The "Relative Spot Area" is calculated by dividing the nucleus area by the parameter value X specified. Use the relative spot area if you expect the structure of interest to expand and shrink with the nucleus as a whole.
The "Max. Intens. Rel." feature is determined by moving the measurement spot over the nucleus, computing the mean intensity within the spot for every position, and detecting the maximum value.

12 - Min. Intens. Rel

Minimum Intensity, Relative Spot Area 1/X" – is the minimum intensity value, measured with relative spot area 1/X.

13 - Max. Intens. Abs.

"Maximum Intensity, Absolute Spot Area X/100 µm²" – the second way to define the measurement spot size is "absolute spot area". In this case the spot area is directly specified in units of 1/100 µm². Use the absolute spot area if you expect the size of structure of interest to stay constant when the size of the nucleus varies.
"Max. Intens. Abs." is the maximum intensity value determined using a measurement spot of this specified absolute area.

14 - Min. Intens. Abs.

"Minimum Intensity, Absolute Spot Area X/100 µm²" – is the minimum intensity value, measured with absolute spot area X/100 µm².

15 - Rel. Area X% Int.

"Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)" – first the intensities of the image are normalized to fill the whole available contrast range. To avoid meaningless measurements for empty images, the gain factor (i.e. the factor by which the contrast is increased) is limited to a maximum of Y% for this step. A second limiting parameter for the upper threshold determination is the number of saturated pixels in the image. If Z is set to zero, this parameter is not used. Then a relative intensity threshold at X% of the full contrast range is set, and the above threshold area is determined. Finally this area is divided by the area of the nucleus to get a relative area.

16 - Abs. Area X% Int.

"Total Absolute Area in µm² at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)" – first the intensities of the image are normalized to fill the whole available contrast range. To avoid meaningless measurements for empty images, the gain factor (i.e. the factor by which the contrast is increased) is limited to a maximum of Y% for this step. A second limiting parameter for the upper threshold determination is the number of saturated pixels in the image. If Z is set to zero, this parameter is not used. Then a relative intensity threshold at X% of the full contrast range is set and the above threshold area is determined and converted to µm².

28 – Eccentricity

"Numerical Eccentricity of Ellipse (0..1)" – the contour of the nucleus is approximated by an ellipse, and the radius along the long (A) and short (B) axis is determined. The numerical eccentricity is defined by the formula:

$$\sqrt{1-\frac{B}{A}}$$

The feature value is always between 0 and 1 (as A>=B), for a perfect circle the result is 0.

53 - Center Dist. Moment

"Center Distance Moment, Distance Exponent X, Intensity Exponent Y" – calculates the normalized sum of intensities of all pixels of a cell multiplied with their distance from the center pixel.

$$\sum \text{Intensity}^{X}\,\text{Distance}^{Y}$$

The exponents define the weighting of intensity and distance, e.g. with X=1 and Y=2 the distance has greater influence.

This feature can be used to distinguish differently stained round cells, e.g. homogeneously stained cells, ringshaped stained cells or cells with higher center intensities.

54 - Radial Dist. Moment

"Radial Distance Moment, Distance Exponent X, Intensity Exponent Y" – calculates the normalized sum of intensities of all pixels of a cell multiplied with their relative distance from the center pixel.

$$\sum \text{Intensity}^X \text{Distance}^Y$$

The exponents define the weighting of intensity and distance, e.g. with X=1 and Y=2 the distance has greater influence.
This feature can be used to differentiate differently stained irregular cells, e.g. tumor cells. So homogeneously stained cells, ringshaped stained cells or cells with higher center intensities can be distinguished.

55 - Rel. Center Intens.

"Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z" – calculates the relative center intensity, specified by the relative radius X and the inner and outer percentile. Possible values for the percentile are 0..100, a value of –1 calculates the mean value.

56 - Rel. Radial Intens.

"Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z" – calculates the relative radial intensity, specified by the relative radius X and the inner and outer percentile. Possible values for the percentile are 0..100, a value of –1 calculates the mean value.

57 - Rel. Granularity

"Granularity, Relative Measurement Distance X" – calculates the maximum difference of gray levels of the center pixel to all pixels in a specified relative distance X.

59 - Pixel Granularity

"Granularity, Measurement Distance X Pixel" – calculates the maximum difference of gray levels of the center pixel to all pixels in a specified distance of X pixels.

60 - Granularity SDev

"Standard Deviation of Last Granularity" – calculates the standard deviation of the last calculated granularity (relative, absolute or pixel).

61 - N of Objects

"Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)" – first the intensities of the image are normalized to fill the whole available contrast range. To avoid meaningless measurements for empty images, the gain factor (i.e. the factor by which the contrast is increased) is limited to a maximum of Y% for this step. Then a relative intensity threshold at X% of the full contrast range is set and the number of objects is determined. This feature can be used to determine the number of objects within a cell.

62 - Mean Rel. Obj. Area

"Mean of Relative Object Area" – calculates the mean relative object area of all objects determined by using the feature "N of Objects".

63 - Mean Abs. Obj. Area

"Mean of Absolute Object Area" – calculates the mean absolute object area of all objects determined by using the feature "N of Objects".

64 - Sdev Rel. Obj. Area

"Standard Deviation of Relative Object Area" – this will calculate the standard deviation of the relative object area of all objects determined by using the feature "N of Objects".

65 - SDev Abs. Obj. Area

"Standard Deviation of Absolute Object Area" – this will calculate the standard deviation of the absolute object area of all objects determined by using the feature "N of Objects".

66 - Mean Obj. Intensity

"Mean of Object Intensity" – this will calculate the mean intensity of all objects determined by using the feature "N of Objects".

67 - SDev Obj. Intensity

"Standard Deviation of Object Intensity" – this will calculate the standard deviation of the intensity of all objects determined by using the feature "N of Objects".

86 - Mean Contour Radius

"Mean Radius of Contour in 1/10 µm" – calculates the mean radius of a cell.

87 - Min. Contour Radius

"Minimum Radius of Contour in 1/10 µm" – measures the minimum radius of a cell.

88 - Max. Contour Radius

"Maximum Radius of Contour in 1/10 µm" – measures the maximum radius of a cell.

91 - Mean Center Dist.

"Mean Distance of Objects to Cell Center" – this will calculate the mean distance of all objects determined by using the feature "N of Objects" to the center of the cell.

92 - SDev Center Dist.
"Standard Deviation of Distance to Cell Center" – this will calculate the mean distance of all objects within the cell image determined by using the feature "N of Objects" to the center of the cell.

93 - Mean Obj. Dist.
"Mean Distance of Objects to other Objects" – calculates the mean distance of all objects within the cell image determined by using the feature "N of Objects" to other objects.

94 - SDev Obj. Dist.

"Standard Deviation of Distance to other Objects" – calculates the standard deviation of the distance of all objects within the cell image determined by using the feature "N of Objects" to other objects within this cell image.

107 - Aspect Ratio

"Aspect Ratio of Cell (Short Axis / Long Axis, 0..1)" – this is the aspect ratio of a cell used for the cell selection. It is calculated by dividing the short axis by the long axis. The feature value is always between 0 and 1, for a perfect circle the result is 1.

113 - Field Number

"Number of Field during Search" – gives the field number in which a cell or object has been detected.

3. Cechy wykorzystane we wstępnej diagnostyce nowotworu pęcherza moczowego

Nr cechy w zbiorze danych	Numer cechy w systemie metafer	Parametry			Nazwa cechy
		Param1	Param2	Param3	
1	1				Total Area within Contour in µm$_c$
2	2				Circumference of Contour in µm
3	15	10.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
4	15	20.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
5	15	30.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
6	15	40.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
7	15	50.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
8	15	60.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
9	15	70.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
10	15	80.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
11	15	90.000	300.000	2.000	Total Relative Area at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
12	16	10.000	300.000	2.000	Total Absolute Area in µm$_c$ at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
13	16	20.000	300.000	2.000	Total Absolute Area in µm$_c$ at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
14	16	30.000	300.000	2.000	Total Absolute Area in µm$_c$ at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
15	16	40.000	300.000	2.000	Total Absolute Area in µm$_c$ at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
16	16	50.000	300.000	2.000	Total Absolute Area in µm$_c$ at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)

Nr cechy w zbiorze danych	Numer cechy w systemie metafer	Parametry			Nazwa cechy
		Param1	Param2	Param3	
17	16	60.000	300.000	2.000	Total Absolute Area in µm² at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
18	16	70.000	300.000	2.000	Total Absolute Area in µm² at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
19	16	80.000	300.000	2.000	Total Absolute Area in µm² at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
20	16	90.000	300.000	2.000	Total Absolute Area in µm² at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
21	86				Mean Radius of Contour in 1/10 µm
22	87				Minimum Radius of Contour in 1/10 µm
23	88				Maximum Radius of Contour in 1/10 µm
24	21	24.000	25.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
25	107				Aspect Ratio of Cell (Short Axis / Long Axis, 0..1)
26	28				Numerical Eccentricity of Ellipse (0..1)
27	3				Irregularity of Contour (0..1)
28	4				Roundness = 4 Pi Area / Circumference² (0..1)
29	5				Maximum Relative Concavity Depth (0..1)
30	6				Relative Area of Deepest Concavity (0..1)
31	7				Total Relative Concavity Area (0..1)
32	9				Mean Intensity
33	10				Standard Deviation of Intensity
34	12	640.000			Minimum Intensity, Relative Spot Area 1/X
35	11	640.000			Maximum Intensity, Relative Spot Area 1/X
36	21	36.000	37.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
37	12	320.000			Minimum Intensity, Relative Spot Area 1/X
38	11	320.000			Maximum Intensity, Relative Spot Area 1/X
39	21	39.000	40.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
40	12	160.000			Minimum Intensity, Relative Spot Area 1/X
41	11	160.000			Maximum Intensity, Relative Spot Area 1/X
42	21	42.000	43.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
43	12	80.000			Minimum Intensity, Relative Spot Area 1/X
44	11	80.000			Maximum Intensity, Relative Spot Area 1/X
45	21	45.000	46.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
46	12	40.000			Minimum Intensity, Relative Spot Area 1/X
47	11	40.000			Maximum Intensity, Relative Spot Area 1/X
48	21	48.000	49.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
49	12	20.000			Minimum Intensity, Relative Spot Area 1/X
50	11	20.000			Maximum Intensity, Relative Spot Area 1/X
51	21	51.000	52.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
52	12	10.000			Minimum Intensity, Relative Spot Area 1/X
53	11	10.000			Maximum Intensity, Relative Spot Area 1/X
54	21	54.000	55.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
55	14	40.000			Minimum Intensity, Absolute Spot Area X/100 µm²
56	13	40.000			Maximum Intensity, Absolute Spot Area X/100 µm²
57	21	57.000	58.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
58	14	80.000			Minimum Intensity, Absolute Spot Area X/100 µm²
59	13	80.000			Maximum Intensity, Absolute Spot Area X/100 µm²
60	21	60.000	61.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
61	14	160.000			Minimum Intensity, Absolute Spot Area X/100 µm²
62	13	160.000			Maximum Intensity, Absolute Spot Area X/100 µm²
63	21	63.000	64.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
64	14	320.000			Minimum Intensity, Absolute Spot Area X/100 µm²
65	13	320.000			Maximum Intensity, Absolute Spot Area X/100 µm²

Nr cechy w zbiorze danych	Numer cechy w systemie metafer	Parametry			Nazwa cechy
		Param1	Param2	Param3	
66	21	66.000	67.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
67	14	640.000			Minimum Intensity, Absolute Spot Area X/100 µm¸
68	13	640.000			Maximum Intensity, Absolute Spot Area X/100 µm¸
69	21	69.000	70.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
70	14	1280.000			Minimum Intensity, Absolute Spot Area X/100 µm¸
71	13	1280.000			Maximum Intensity, Absolute Spot Area X/100 µm¸
72	21	72.000	73.000	1.000	Ratio Value of Feature Variables X and Y (Scale Factor Z = 1.0)
73	53	1.000	1.000		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
74	53	1.500	1.000		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
75	53	2.000	1.000		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
76	53	1.000	1.500		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
77	53	1.500	1.500		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
78	53	2.000	1.500		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
79	53	1.000	2.000		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
80	53	1.500	2.000		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
81	53	2.000	2.000		Center Distance Moment, Distance Exponent X, Intensity Exponent Y
82	54	1.000	1.000		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
83	54	1.500	1.000		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
84	54	2.000	1.000		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
85	54	1.000	1.500		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
86	54	1.500	1.500		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
87	54	2.000	1.500		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
88	54	1.000	2.000		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
89	54	1.500	2.000		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
90	54	2.000	2.000		Radial Distance Moment, Distance Exponent X, Intensity Exponent Y
91	55	0.100	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
92	55	0.200	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
93	55	0.300	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
94	55	0.400	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
95	55	0.500	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
96	55	0.600	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
97	55	0.700	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
98	55	0.800	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
99	55	0.900	-1.000	-1.000	Relative Center Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
100	56	0.100	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
101	56	0.200	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
102	56	0.300	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
103	56	0.400	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
104	56	0.500	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
105	56	0.600	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
106	56	0.700	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
107	56	0.800	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
108	56	0.900	-1.000	-1.000	Relative Radial Intensity, Rel. Radius X, Inner Perc. Y, Outer Perc. Z
109	57	0.320			Granularity, Relative Measurement Distance X
110	60				Standard Deviation of Last Granularity
111	57	0.160			Granularity, Relative Measurement Distance X
112	60				Standard Deviation of Last Granularity
113	57	0.080			Granularity, Relative Measurement Distance X
114	60				Standard Deviation of Last Granularity
115	59	1.000			Granularity, Measurement Distance X Pixel
116	60				Standard Deviation of Last Granularity

Nr cechy w zbiorze danych	Numer cechy w systemie metafer	Parametry			Nazwa cechy
		Param1	Param2	Param3	
117	59	2.000			Granularity, Measurement Distance X Pixel
118	60				Standard Deviation of Last Granularity
119	59	4.000			Granularity, Measurement Distance X Pixel
120	60				Standard Deviation of Last Granularity
121	59	8.000			Granularity, Measurement Distance X Pixel
122	60				Standard Deviation of Last Granularity
123	59	16.000			Granularity, Measurement Distance X Pixel
124	60				Standard Deviation of Last Granularity
125	61	20.000	300.000	0.000	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
126	62				Mean of Relative Object Area
127	64				Standard Deviation of Relative Object Area
128	63				Mean of Absolute Object Area
129	65				Standard Deviation of Absolute Object Area
130	66				Mean of Object Intensity
131	67				Standard Deviation of Object Intensity
132	91				Mean Distance of Objects to Cell Center
133	92				Standard Deviation of Distance to Cell Center
134	93				Mean Distance of Objects to other Objects
135	94				Standard Deviation of Distance to other Objects
136	61	30.000	300.000	0.000	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
137	62				Mean of Relative Object Area
138	64				Standard Deviation of Relative Object Area
139	63				Mean of Absolute Object Area
140	65				Standard Deviation of Absolute Object Area
141	66				Mean of Object Intensity
142	67				Standard Deviation of Object Intensity
143	91				Mean Distance of Objects to Cell Center
144	92				Standard Deviation of Distance to Cell Center
145	93				Mean Distance of Objects to other Objects
146	94				Standard Deviation of Distance to other Objects
147	61	40.000	300.000	0.000	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
148	62				Mean of Relative Object Area
149	64				Standard Deviation of Relative Object Area
150	63				Mean of Absolute Object Area
151	65				Standard Deviation of Absolute Object Area
152	66				Mean of Object Intensity
153	67				Standard Deviation of Object Intensity
154	91				Mean Distance of Objects to Cell Center
155	92				Standard Deviation of Distance to Cell Center
156	93				Mean Distance of Objects to other Objects
157	94				Standard Deviation of Distance to other Objects
158	61	50.000	300.000	0.000	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
159	62				Mean of Relative Object Area
160	64				Standard Deviation of Relative Object Area
161	63				Mean of Absolute Object Area
162	65				Standard Deviation of Absolute Object Area
163	66				Mean of Object Intensity
164	67				Standard Deviation of Object Intensity
165	91				Mean Distance of Objects to Cell Center

Nr cechy w zbiorze danych	Numer cechy w systemie metafer	Parametry			Nazwa cechy
		Param1	Param2	Param3	
166	92				Standard Deviation of Distance to Cell Center
167	93				Mean Distance of Objects to other Objects
168	94				Standard Deviation of Distance to other Objects
169	61	60.000	300.000	0.000	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
170	62				Mean of Relative Object Area
171	64				Standard Deviation of Relative Object Area
172	63				Mean of Absolute Object Area
173	65				Standard Deviation of Absolute Object Area
174	66				Mean of Object Intensity
175	67				Standard Deviation of Object Intensity
176	91				Mean Distance of Objects to Cell Center
177	92				Standard Deviation of Distance to Cell Center
178	93				Mean Distance of Objects to other Objects
179	94				Standard Deviation of Distance to other Objects
180	61	70.000	300.000	0.000	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
181	62				Mean of Relative Object Area
182	64				Standard Deviation of Relative Object Area
183	63				Mean of Absolute Object Area
184	65				Standard Deviation of Absolute Object Area
185	66				Mean of Object Intensity
186	67				Standard Deviation of Object Intensity
187	91				Mean Distance of Objects to Cell Center
188	92				Standard Deviation of Distance to Cell Center
189	93				Mean Distance of Objects to other Objects
190	94				Standard Deviation of Distance to other Objects
191	61	80.000	300.000	0.000	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
192	62				Mean of Relative Object Area
193	64				Standard Deviation of Relative Object Area
194	63				Mean of Absolute Object Area
195	65				Standard Deviation of Absolute Object Area
196	66				Mean of Object Intensity
197	67				Standard Deviation of Object Intensity
198	91				Mean Distance of Objects to Cell Center
199	92				Standard Deviation of Distance to Cell Center
200	93				Mean Distance of Objects to other Objects
201	94				Standard Deviation of Distance to other Objects
202	61	90.000	300.000	0.000	Number of Objects at X% Intensity (Maximum Gain Y%, Upper Thr. with Z Sat. Pixels)
203	62				Mean of Relative Object Area
204	64				Standard Deviation of Relative Object Area
205	63				Mean of Absolute Object Area
206	65				Standard Deviation of Absolute Object Area
207	66				Mean of Object Intensity
208	67				Standard Deviation of Object Intensity
209	91				Mean Distance of Objects to Cell Center
210	92				Standard Deviation of Distance to Cell Center
211	93				Mean Distance of Objects to other Objects
212	94				Standard Deviation of Distance to other Objects