POLSKA AKADEMIA NAUK
INSTYTUT BIOCYBERNETYKI
I INŻYNIERII BIOMEDYCZNEJ

Adam Jóźwik

NIEPARAMETRYCZNE METODY KLASYFIKACJI NADZOROWANEJ

PRACE INSTYTUTU BIOCYBERNETYKI **NR**
WARSZAWA 2013 I INŻYNIERII BIOMEDYCZNEJ ISSN

Składam serdeczne podziękowania i wyrazy wdzięczności:

Panu Profesorowi Wojciechowi Zmysłowskiemu za bardzo wnikliwe przeczytanie pracy, naniesienie swoich uwag i sugestii, dzięki czemu wiele zagadnień zostało przedstawionych w bardziej przejrzystej formie.

Panu Profesorowi Dominikowi Sankowskiemu za życzliwość i przekazanie swoich uwag zwłaszcza dotyczących końcowego rozdziału pracy.

Panu Profesorowi Markowi Darowskiemu za zachętę do napisania tej monografii.

Współautorom moich publikacji, dzięki którym przestawione w monografii metody znalazły zastosowanie

Wszystkim osobom, które okazywały stałe zainteresowanie stanem zaawansowania pracy.

Adam Jóźwik

STRESZCZENIE

Monografia, w przeważającej części swej objętości, zawiera opis metod opracowanych przez jej autora. Przedstawione zostały najpierw ogólne zasady konstruowania klasyfikatorów oraz metody ich oceny. Znaczną część pracy zajmują metody wyznaczania hiperpłaszczyzn rozdzielających pary klas z przeznaczeniem do konstrukcji klasyfikatorów wielo-decyzyjnych o strukturze równoległej, złożonych z klasyfikatorów dwu-decyzyjnych.

Szczegółowo został przedstawiony znany algorytm korekcji błędów, iteracyjny algorytm wyznaczania hiperpłaszczyzny rozdzielającej dwa zbiory oraz rekursywny algorytm badania rozdzielności liniowej dwóch zbiorów opublikowany już wcześniej przez autora niniejszej monografii.

Nowym wynikiem opublikowanym po raz pierwszy w niniejszej pracy jest klasyfikator skonstruowany na podstawie zbioru uczącego, edytowanego dla liniowej rozdzielności pary zbiorów reprezentujących różne klasy. Kolejnym nowym wynikiem jest rozszerzenie rekursywnego algorytmu badania rozdzielności liniowej dwóch zbiorów, które umożliwia weryfikację, czy badane zbiory są rozdzielne liniowo z zadanym „prześwitem”.

Najwięcej uwagi zostało poświęcone metodom minimalno-odległościowym, w tym rozmytej regule *k* najbliższych sąsiadów oraz metodzie konstrukcji klasyfikatorów polegającej na wyznaczaniu obszarów klas. Przedstawiony został algorytm redukcji zbioru odniesienia polegający na wyznaczaniu par obiektów wzajemnie najbliższych oraz nowy algorytm wykorzystujący konwersję zbioru już zredukowanego do nowej sztucznej przestrzeni cech, w celu jego dalszej redukcji. Również nową propozycją jest algorytm kondensacji zbioru odniesienia z zastosowaniem sztucznych cech. Zaproponowana też została, niepublikowana wcześniej, skorygowana reguła *k* najbliższych sąsiadów z przeznaczeniem dla zbiorów uczących zawierających braki wartości niektórych cech.

SPIS TREŚCI

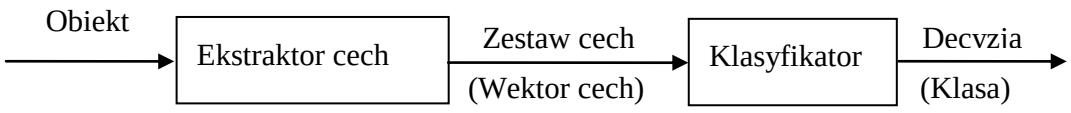
1. WPROWADZENIE

1.1 Informacje ogólne

Klasyfikacja obiektów jest przedmiotem badań dziedziny określanej najczęściej mianem *rozpoznawania obrazów*. Nierzadko mówi się też o rozpoznawaniu wzorców, czy rozpoznawaniu obiektów. Obiekty rozumiane są w bardzo ogólnym sensie. Mogą nimi być różne przedmioty, znaki alfanumeryczne, dźwięki, obrazy optyczne, stany zdrowia, rośliny, zwierzęta, czy osoby. Zakłada się, że obiekty opisane są zestawem cech, zwanym też wektorem cech, które mogą być liczbowe, jakościowe, czy binarne. Jednakże, w niniejszej monografii, jeśli nie zostanie to oddzielnie zaznaczone, zakłada się, że cechy przyjmują wartości liczbowe, rzeczywiste lub całkowite. Wektorom cech odpowiadają punkty w przestrzeni cech, zwykle euklidesowej. W dalszych rozważaniach pojęcia: obiekt opisany zestawem cech $\underline{x}$, wektor cech $\underline{x}$, punkt $\underline{x}$ w przestrzeni cech, będą dla wygody używane zamiennie. O punktach w przestrzeni cech będzie najczęściej mowa w przypadku odwoływania się do interpretacji geometrycznej, w przypadku operacji algebraicznych będzie mowa o wektorach cech.

W niniejszej monografii przedmiotem rozważań będzie wyłącznie klasyfikacja nadzorowana, której reguły decyzyjne są wyprowadzane ze zbioru uczącego, którym jest zbiór obiektów ze znaną przynależnością do klas. Ponadto, w zbiorze tym liczności klas powinny być zgodne ze statystyką ich występowania. Istnieje jeszcze inny rodzaj klasyfikacji, która najczęściej oparta jest na funkcji podobieństwa lub odległości pomiędzy obiektami i wykorzystuje zbiór obiektów, których przynależność nie jest znana. Zadanie klasyfikacji polega na podzieleniu zbioru tych obiektów na pewną liczbę podzbiorów, zadaną z góry bądź nie, w taki sposób, aby obiekty z różnych podzbiorów różniły się możliwie najmocniej, a obiekty wewnątrz tych podzbiorów powinny różnić się od siebie jak najmniej. Jest to klasyfikacja nienadzorowana, która zostanie pominięta w monografii.

W układzie rozpoznającym można wyróżnić dwa podstawowe bloki: blok wydzielania cech, czyli ekstraktor cech i klasyfikator, jak to ilustruje Rys. 1.1.

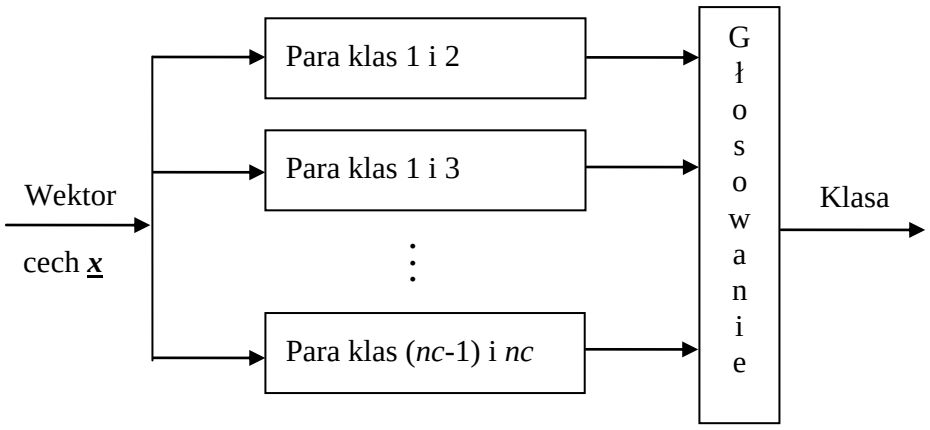


Rys. 1.1. Ogólny schemat układu rozpoznającego

Rozważania przeprowadzone w niniejszej pracy odnosić się będą wyłącznie do klasyfikatora. Klasyfikator realizuje pewną regułę decyzyjną, na podstawie której wskazana zostaje jedna z *nc* rozpatrywanych klas. Reguła decyzyjna może odnosić się do wielu klas lub tylko do dwóch klas. W przypadku, gdy wykorzystana w klasyfikatorze reguła decyzyjna dotyczy tylko dwóch klas lub gdy jest zastosowana reguła wielo-decyzyjna, ale ograniczona do pary klas, to klasyfikator będzie określany jako dwu-decyzyjny.

1.2 Struktury klasyfikatorów

Klasyfikatory dwu-decyzyjne można wykorzystać do konstrukcji klasyfikatorów wielo-decyzyjnych. Jednym z rozwiązań jest klasyfikator równoległy złożony z tylu klasyfikatorów dwu-decyzyjnych ile jest możliwych par klas, a jest ich oczywiście $nc*(nc-1)/2$. Struktura takiego klasyfikatora przedstawiona została na Rys. 1.2.



Rys. 1.2. Klasyfikator równoległy

Każdy ze składowych klasyfikatorów dwu-decyzyjnych oddaje głos na jedną z dwóch klas, dla których został skonstruowany. Natomiast klasyfikator równoległy wskaże na klasę, która uzyska najwięcej głosów. Przypadki niejednoznaczne, określane dalej dla wygody jako remisowe, mogą być rozstrzygane na korzyść klasy najczęściej występującej, jeśli a priori taka informacja jest dostępna, losowo lub do klasy z mniejszym indeksem, jeśli klasy są ponumerowane.

W przypadku większej liczby klas niż dwie klasy, możliwa jest struktura hierarchiczna, np. dwuetapowa. W klasyfikacji dwuetapowej klasy mogą być pogrupowane w makro-klasy. W pierwszym etapie następuje wówczas klasyfikacja do makro-klas, a w drugim etapie rozstrzygnięcie następuje pomiędzy klasami tej makro-klasy, do której w pierwszym etapie obiekt został zaklasyfikowany. W szczególności makro-klasa może składać się tylko z jednej klasy, co na pewno ma miejsce w

przypadku zadania klasyfikacji dotyczącej trzech klas. Wtedy część obiektów rozpoznawana jest już w pierwszym etapie, a część dopiero w drugim.

Etapów klasyfikacji w strukturze hierarchicznej może być więcej niż dwa. Liczba możliwych struktur hierarchicznych szybko rośnie wraz ze wzrostem liczby rozważanych klas. Uciążliwość tego efektu będzie mniejsza, jeśli w pierwszym etapie utworzona zostanie mała liczba makro-klas, ale składających się z większej liczby klas. Kolejny etap może mieć wtedy więcej makro-klas, ale zawierających mniej klas. W końcowym etapie występują już tylko pojedyncze klasy.

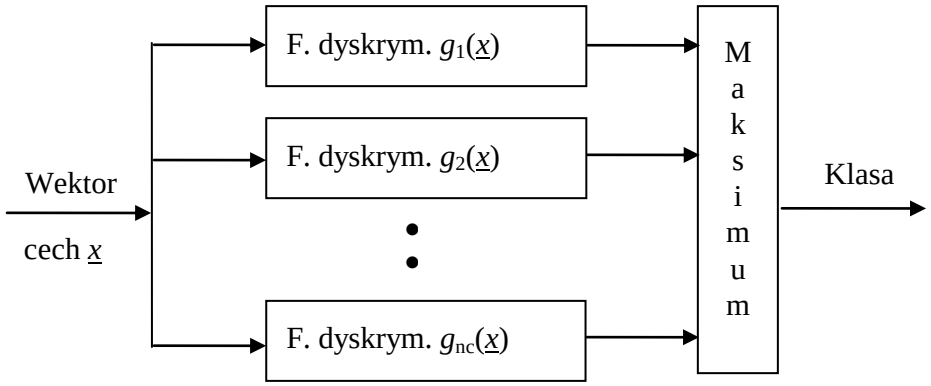
Łatwo zauważyć, że hierarchiczną sekwencję klasyfikacji można uzyskać ze struktury równoległej. Wystarczy tylko zorganizować hierarchicznie głosowanie.

Możliwa jest też struktura wielo-etapowa, w której na każdym etapie rozstrzyga się pomiędzy pojedynczą klasą, a pozostałymi klasami danego etapu. Na przykład, w przypadku zadania dla trzech klas, możemy na pierwszym etapie rozstrzygać, czy obiekt należy do klasy 1 czy też do makro-klasy złożonej z klas 2 i 3. Wskazanie na klasę 1 kończy klasyfikację, zaś wskazanie na makro-klasę uaktywnia drugi etap, który rozstrzyga już tylko pomiędzy klasami 2 i 3. Taki przypadek często występuje w zastosowaniach medycznych, gdy klasa 1 oznacza osoby zdrowe, a klasy 2 i 3 osoby chore. Warto zauważyć, że dla grup osób chorych możemy dysponować innymi zestawami cech niż dla grupy osób zdrowych. W stosunku do klasyfikatorów wykorzystujących jedną regułę wielo-decyzyjną klasyfikatory złożone z wielu dwu-decyzyjnych klasyfikatorów mają tą zaletę, że każdy z klasyfikatorów składowych może działać na innych zestawach cech, co może skutkować wyższą jakością klasyfikacji mierzoną frakcją lub odsetkiem poprawnych decyzji. Również i ten wielo-etapowy schemat podejmowania decyzji może być zrealizowany w strukturze równoległej, pokazanej na Rys. 1.2, jeżeli głosowanie klasyfikatorów składowych zostanie zorganizowane wielo-etapowo. Klasyfikacji wieloetapowej poświęcono wiele miejsca w książce [Kurzyński M., 1997].

1.3 Funkcje dyskryminacyjne

Zasadę działania klasyfikatora wyjaśnia Rys. 1.3., na którym przedstawiona została ogólna budowa klasyfikatora, który zawiera tyle bloków funkcji dyskryminacyjnych ile jest klas.

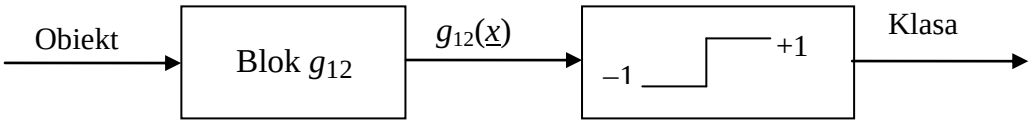
Obiekt, czyli wektor cech $\underline{x}$ zostanie przyporządkowany do tej klasy, dla której funkcja dyskryminacyjna osiąga wartość maksymalną. Ewentualny przypadek remisu, czyli równych wartości funkcji dyskryminacyjnej można rozstrzygać w podobny sposób, jaki miał miejsce w przypadku głosowania klasyfikatorów składowych klasyfikatora równoległego. W najprostszym przypadku funkcje dyskryminacyjne mogą być liniowe.



Rys. 1.3. Ogólna budowa klasyfikatora

Każda z funkcji dyskryminacyjnych $g_j(\underline{x})$, $j=1,2,\ldots,nc$, może być interpretowana jako funkcja podobieństwa obiektu opisanego wektorem cech $\underline{x}$ do obiektów klasy j. W przypadku, gdy funkcje dyskryminacyjne są liniowe, klasyfikator o strukturze jak pokazana na Rys. 1.3, nazywany jest maszyna liniową.

Jeżeli rozważany problem klasyfikacji dotyczy tylko dwóch klas, to nie jest konieczne zastosowanie dwóch bloków funkcji dyskryminacyjnych. Zamiast dwóch funkcji $g_1(\underline{x})$ i $g_2(\underline{x})$ można użyć jednej funkcji $g_{12}(\underline{x})=g_1(\underline{x})-g_2(\underline{x})$. Obiekt zostanie zaklasyfikowany do klasy 1, gdy $g_{12}(\underline{x})\geq 0$ i do klasy 2, gdy $g_{12}(\underline{x})<0$. Struktura klasyfikatora wówczas się upraszcza do postaci jak na Rys. 1.4.



Rys. 1.4. Struktura klasyfikatora dwu-decyzyjnego

Pierwszy blok wylicza wartość funkcji dyskryminacyjnej, a drugi wyznacza jej znak, czyli liczy funkcję $signum(g_{12}(\underline{x}))$ i jeśli przyjmie ona wartość +1, to obiekt zostaje zakwalifikowany do klasy 1, a w przeciwnym przypadku do klasy 2.

Z takich dwu-decyzyjnych klasyfikatorów można zbudować klasyfikator wielo-decyzyjny stosując strukturę z Rys. 1.2. Należy wówczas użyć funkcji $g_{i,j}(\underline{x})$, $i=1,2,\ldots,nc-1$, $j=i+1$, $i+2,\ldots,nc$. W przypadku, gdy $g_{i,j}(\underline{x})\geq 0$ klasyfikator składowy oddaje głos na klasę i, a w przypadku, gdy $g_{i,j}(\underline{x})<0$, to na klasę j. W dalszych rozważaniach w zadaniach, w których liczba klas będzie większa niż dwie klasy, przedmiotem zainteresowania będą głownie klasyfikatory wykorzystujące jedną regułę wielo-decyzyjną lub klasyfikatory o strukturze równoległej, jak na Rys. 1.2.

1.4 Klasyfikator idealny

Podstawowym kryterium oceny jakości klasyfikacji, pomijając klasyfikację rozmytą, która będzie opisana w dalszej części pracy, jest prawdopodobieństwo mylnej decyzji, szacowane frakcją lub odsetkiem mylnych decyzji. Wobec takiego założenia klasyfikator powinien wskazywać na klasę najbardziej prawdopodobną, czyli wskazywać klasę *i*, dla której spełniony jest warunek:

$$p(i/\underline{x})=\max_j p(j/\underline{x}), \tag{1.1}$$

gdzie $p(j/\underline{x})$ jest prawdopodobieństwem wystąpienia klasy j, $j=1,2,\ldots,nc$, pod warunkiem, że obiekt opisany jest wektorem cech $\underline{x}$. Jednakże, rozkłady prawdopodobieństw $p(j/\underline{x})$ dla rozważanych klas j nie są zazwyczaj a priori znane. Mogą być jednak bezpośrednio oszacowane na podstawie zbioru obiektów, których przynależność jest znana. Taki zbiór, jak już wyżej było wspomniane w podrozdziale 1.1, nazywany jest zbiorem uczącym lub zbiorem treningowym i służy on do konstrukcji klasyfikatora, tj. reguły decyzyjnej.

Gdyby gęstości rozkładów prawdopodobieństw $f(\underline{x}/j)$ wektorów cech dla rozważanych klas i prawdopodobieństwa a priori $p(j)$ wystąpienia klas były znane, to prawdopodobieństwa $p(j/\underline{x})$ można by obliczyć ze wzoru [Bayes T., 1973] :

$$p(j/\underline{x})=p(j)*f(\underline{x}/j)/f(\underline{x}). \tag{1.2}$$

Wartość funkcji $f(\underline{x})$ może być obliczona ze znanego wzoru na prawdopodobieństwo całkowite:

$$f(\underline{x})=\sum_{j=1}^{nc} p(j)*f(\underline{x}/j), \tag{1.3}$$

Jednak nie jest to konieczne, gdyż $f(\underline{x})$ w mianowniku wzoru 1.2 można pominąć, ponieważ tylko licznik decyduje, o tym które z prawdopodobieństw $p(j/\underline{x})$ osiągnie wartość maksymalną. Jednak w niektórych zadaniach wyliczenie prawdopodobieństw $p(j/\underline{x})$ może być pożądane, a wtedy trzeba szacować $f(\underline{x})$. Prawdopodobieństwa $p(j)$ oraz gęstości rozkładów prawdopodobieństw $f(\underline{x}/j)$, podobnie jak to miało miejsce w przypadku prawdopodobieństw warunkowych $p(j/\underline{x})$, mogą być oszacowane na podstawie zbioru uczącego. Dla oszacowania prawdopodobieństw $p(j)$, konieczne jest, aby klasy w zbiorze uczącym były reprezentowane w proporcjach zgodnych z częstościami ich występowania.

Tak więc chcąc skonstruować regułę decyzyjną, oferującą możliwie najmniejszy odsetek pomyłek, można na podstawie zbioru uczącego oszacowywać bezpośrednio prawdopodobieństwa $p(j/\underline{x})$ bądź prawdopodobieństwa $p(j)$ i gęstości rozkładów $f(\underline{x}/j)$ i na ich podstawie wyliczać $p(j/\underline{x})$. W przypadku cech, których wartości są wyłącznie całkowitoliczbowe, funkcje gęstości rozkładu prawdopodobieństw $f(\underline{x}/j)$ i $f(\underline{x})$,

występujące we wzorach 1.2 i 1.3, należy zastąpić funkcjami rozkładu odpowiednich prawdopodobieństw $p(\underline{x}/j)$ i $p(\underline{x})$.

Łatwo zauważyć, że klasyfikator działający wg reguły określonej relacją 1.1 ma strukturę zgodną zilustrowaną na Rys. 1.3.

Nie wszystkie z metod, które będą przedstawione w dalszych częściach niniejszej pracy, mieszczą się w opisanych wyżej schematach. Dotąd przyjęte było założenie, że obiekt może należeć tylko do jednej z rozważanych *nc* klas. Jednakże jedna z metod, która będzie zaproponowana, dotyczy klasyfikacji zwanej rozmytą. Polega ona na tym, że obiekt może należeć w różnym stopniu do każdej z klas. Wynikiem klasyfikacji jest wówczas rozmyty wektor przynależności $\underline{v}=[v_1,v_2,...,v_{\mathrm{nc}}]$, gdzie v_j, $j=1,2,\ldots,nc$, oznacza stopień przynależności obiektu do klasy *j*. Błąd *e* pojedynczej klasyfikacji będzie w tym przypadku przyjmował wartości z przedziału [0,0-1,0] i jest obliczany według formuły:

$$e=(\sum_{j=1}^{nc}|\, v_j - w_j \,|)/2, \tag{1.4}$$

gdzie $\underline{v}=[v_1,v_2,...,v_{nc}]$ jest faktycznym wektorem przynależności obiektu do rozważanych klas, a $\underline{w}=[w_1,w_2,...,w_{nc}]$ jest przypisanym wektorem przynależności, wytworzonym przez klasyfikator. W szczególnym przypadku składowe wektora przynależności mogą przyjmować wartości binarne, czyli 0 lub 1. W takim przypadku będziemy mieli do czynienia z klasyfikacja ostrą (nie rozmytą). Klasyfikacja ostra jest więc szczególnym przypadkiem klasyfikacji rozmytej. W dalszych rozważaniach, jeżeli nie będzie zaznaczone jaki typ klasyfikacji jest rozważany, należy przyjąć, że chodzi o klasyfikację ostrą.

1.5 Problem standaryzacji cech

Cechy opisujące rozpoznawane obiekty mogą być wyrażone w różnych jednostkach miary. Rodzaj jednostek nie stwarza żadnych problemów, gdyż i tak w obliczeniach nie biorą one udziału, a tylko liczności tych jednostek reprezentują wartości cech. Jednak, gdy jakaś wielkość będzie wyrażona w małych jednostkach, to będzie reprezentowana w cesze większą liczbą niż gdyby była wyrażona w dużych jednostkach. W obliczeniach znaczenie tych cech, których wartości są wyrażone dużymi liczbami będzie zwykle większe. Z tego powodu zalecana jest standaryzacja cech. Poza tym, zwykle wyniki klasyfikacji z zastosowaniem standaryzacji cech powinny być lepsze. Poniżej rozważone zostaną dwa podstawowe rodzaje standaryzacji, standaryzacja klasyczna i standaryzacja medianowa. Niech zbiór uczący ma postać $X=\{\underline{x}_i\}_{i=1}^{m}$, gdzie $\underline{x}_i=[x_{i,1},x_{i,2,\ldots},x_{i,n}]$ i $x_{i,j}$ jest wartością *j*-tej cechy *i*-tego obiektu. Wówczas standaryzacja klasyczna przebiega wg formuły:

$$x^s_{i,j} = (x_{i,j} - mv_j) / sd_j , \tag{1.5}$$

gdzie mv_j jest wartością średnią j-tej cechy, a sd_j jej odchyleniem standardowym. Obie te wartości obliczane są na podstawie zbioru uczącego. Cechy j dla których sd_j jest równe zeru nie podlegają standaryzacji i powinny być pominięte przy konstruowaniu klasyfikatora.

Inny rodzaj standaryzacji wyraża się podobnym wzorem, z tą różnicą, że średnia wartość cechy zastąpiona zostaje medianą md_j, a odchylenie standardowe medianą mdd_j z wartości absolutnych odchyleń od mediany, czyli:

$$x^s_{i,j} = (x_{i,j} - md_j) / mdd_j . \tag{1.6}$$

Standaryzacja będzie jeszcze przedmiotem rozważań w dalszej części pracy i choć będą używane symbole średniej wartości cech i ich odchyleń standardowych, to otrzymane wnioski będą odnosić się również do standaryzacji medianowej, jeśli nie zostanie to oddzielnie zaznaczone.

2. METODY OCENY JAKOŚCI KLASYFIKACJI

2.1 Szacowanie prawdopodobieństwa mylnej klasyfikacji

Metoda zbioru testującego

Najbardziej naturalną metodą oszacowania prawdopodobieństwa mylnej decyzji klasyfikatora, jest zastosowanie oddzielnego zbioru testującego T, tj. zbioru obiektów, których przynależność jest znana. Wszystkie obiekty ze zbioru T, których jest m_T, są klasyfikowane regułą decyzyjną wyprowadzoną ze zbioru uczącego U. Liczba e_T obiektów mylnie klasyfikowanych podzielona przez liczbę m_T wszystkich obiektów zbioru testującego T, czyli frakcja błędów $er_T=e_T/m_T$ jest oszacowaniem prawdopodobieństwa mylnej klasyfikacji. Rzetelna ocena klasyfikatora wymaga, aby zbiór testujący był wykorzystany dokładnie jeden raz. Zwykle dysponujemy jednym zbiorem obiektów o znanej przynależności do klas i użycie tej metody wymaga podzielenia tego zbioru obiektów na część uczącą U oraz część testującą T. Problemem w tej metodzie jest proporcja pomiędzy liczebnością części uczącej U i testującej T.

Metoda l-krotnej walidacji

Jest to prawdopodobnie najczęściej stosowana metoda oceny prawdopodobieństwa mylnej klasyfikacji. Zbiór U, o liczebności m_U dzielony jest na l możliwie równych części T_l. Każda z tych l części pełni kolejno rolę zbioru testującego, a suma mnogościowa pozostałych części, czyli zbiór $U–T_l$ służy do konstrukcji klasyfikatora. W tej metodzie, wymagającej l-krotnego konstruowania klasyfikatora, każdy z

obiektów zbioru uczącego U jest klasyfikowany jeden raz. Jeżeli e_U z nich jest mylnie klasyfikowanych, to frakcja $er_U=e_U/m_U$ jest oszacowaniem prawdopodobieństwa mylnej decyzji klasyfikatora.

Jak widzimy, metoda ta polega na wielokrotnym zastosowaniu metody zbioru testującego. Im parametr l jest mniejszy, tym mniejsza różnica pomiędzy liczebnościami zbiorów $U–T_l$ a zbiorem U. Można się bowiem spodziewać, że klasyfikator zbudowany na podstawie zbioru U będzie oferował mniejszą frakcję mylnych decyzji niż klasyfikator skonstruowany z użyciem jakiegokolwiek jego podzbioru $U–T_l$. Należy więc oczekiwać, że wyliczona tą metodą frakcja błędów będzie zawyżona, w tym bardziej znaczącym stopniu im liczba l będzie mniejsza. Z drugiej strony duże wartości l wymagają większego nakładu obliczeń, bo należy konstruować więcej klasyfikatorów. Przy wyborze l powinien być wzięty pod uwagę koszt obliczeniowy jednokrotnego konstruowania klasyfikatora. Klasyfikacja nowych obiektów, czyli obiektów spoza zbioru uczącego, będzie przeprowadzana na podstawie klasyfikatora zbudowanego z użyciem całego zbioru U. Najczęściej przyjmowaną wartością jest l=10 [Kohavi R., 1995]. Można też dzielić zbiór danych na 5 mniej więcej równych części i każdą z nich raz użyć w roli zbioru uczącego, a drugi raz jako zbiór testujący.

Metoda minus jednego elementu

Ta metoda jest szczególnym przypadkiem metody l-krotnej walidacji, jeżeli przyjmiemy l=1. Polega ona na klasyfikacji każdego obiektu $\underline{u}_i$ ze zbioru uczącego U, zawierającego m_U obiektów, z użyciem klasyfikatora wyprowadzonego ze zbioru $U–\{\underline{u}_i\}$. Jeżeli przyjmiemy, że liczba pomyłek wynosi e_U, to frakcja $er_U=e_U/m_U$ jest oszacowaniem prawdopodobieństwa mylnej decyzji. Stosowanie tej metody polecane jest szczególnie w przypadkach małych zbiorów uczących, zwłaszcza, gdy konstruowanie wybranego typu klasyfikatora wymaga tylko niewielkiego nakładu obliczeń.

Jej wadą jest konieczność wielokrotnego konstruowania klasyfikatora. Metoda minus jednego elementu stała się popularna począwszy od roku 1965 [Lachenbruch P.A., 1965]. Jednak, autorzy książki [Devijver P.A., Kittler J., 1982] zauważyli, że metoda ta była wcześniej stosowana przez autorów rosyjskich, ale nie udało im się ustalić, kto był pomysłodawcą tej metody.

2.2 Macierze przekłamań

Frakcja błędów jest standardowym kryterium oceny klasyfikatora, ale w niektórych zastosowaniach pożądana jest również informacja o rodzaju pomyłek. Dokładniejszą informację zawierałaby jednowierszowa macierz frakcji błędów $er=\{er_i\}_{i=1}^{nc}$, gdzie er_i

oznacza frakcję pomyłek popełnianych wśród obiektów z klasy *i*. Z tych frakcji, znając liczebności klas m_i wyliczyć można globalną frakcję błędów $er=\sum_{j=1}^{nc} m_j \circ er_i/m$, gdzie m liczbą obiektów klasyfikowanych.

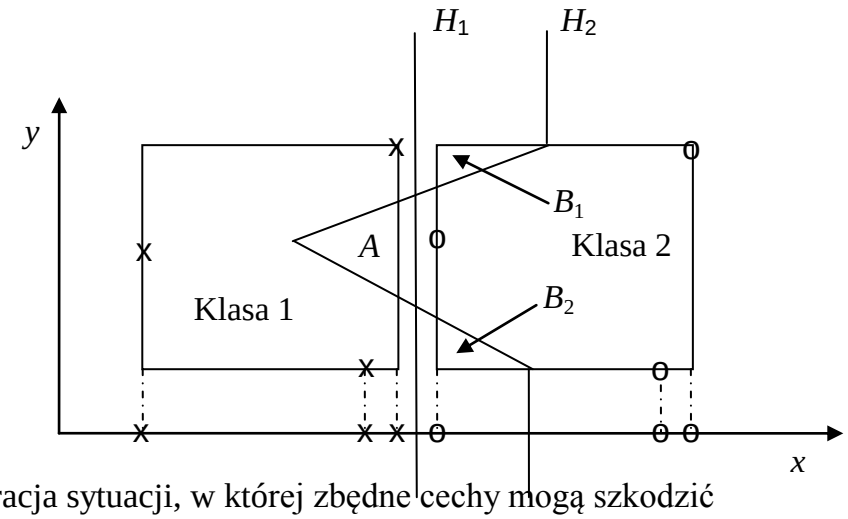
Można też w/w metodę oceny jakości klasyfikacji zastosować do wyznaczenia liczbowej macierzy przekłamań $R=\{r_{i,j}\}_{i,j=1}^{nc}$, w której element $r_{i,j}$ oznacza liczbę obiektów z klasy *i* zaklasyfikowanych do klasy *j*. Z definicji macierzy *R* wynika, że związek liczebności m_i klasy *i* w zbiorze uczącym z elementami $r_{i,j}$ jest następujący: $m_i=\sum_{j=1}^{nc} r_{i,j}$, zaś liczbę obiektów l_i zaliczonych do klasy *i* wyznaczyć można z wzoru: $l_i=\sum_{j=1}^{nc} r_{j,i}$. Na podstawie macierzy *R* łatwo oszacować można prawdopodobieństwo $p_{i,j}$, że obiekt z klasy *i* zostanie zaliczony do klasy *j* oraz prawdopodobieństwo $q_{i,j}$, że obiekt zaklasyfikowany do klasy *i* pochodzi faktycznie z klasy *j*. Związek macierzy $P=\{p_{i,j}\}_{i,j=1}^{nc}$ oraz macierzy $Q=\{q_{i,j}\}_{i,j=1}^{nc}$ z macierzą *R* jest następujący: $p_{i,j}=r_{i,j}/m_i$ oraz $q_{i,j}=r_{j,i}/l_i$.

Macierz prawdopodobieństw *Q* ma większe praktyczne znaczenie niż macierz *P*, gdyż mówi nam o wiarygodności $q_{i,j}$ z jaką następuje klasyfikacja, jeśli obiekt jest kwalifikowany do klasy *i*. Macierze *R*, *P* i *Q* będą określane jako macierze przekłamań, I, II i III rodzaju.

2.3 Frakcja błędów jako kryterium selekcji cech

Frakcja błędów jest najczęściej stosowanym kryterium podczas konstrukcji klasyfikatora, która powinna również obejmować selekcję cech. Część cech może nie mieć żadnego związku z rozważanymi klasami, a część może zawierać informacje nadmiarowe. Tego rodzaju cechy nie tylko podrażają syntezę klasyfikatora, ale ponadto mogą pogarszać jakość klasyfikacji. Ten fakt wyjaśnia prosty przykład podany na Rys. 2.1. W sytuacji jak na rysunku przyjęto, że każda z klas zajmuje obszar kwadratu i jest reprezentowana przez 3 punkty w przestrzeni cech. W tym przykładzie i w innych, w dalszej części pracy, krzyżykami będą reprezentowane obiekty z klasy 1, a kółkami z klasy 2.

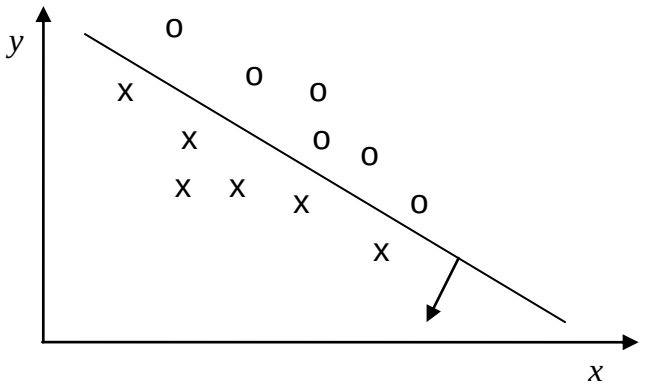
Aby zbudować klasyfikator, racjonalnie jest przyjąć, jako hiperpowierzchnię rozdzielającą, linię łamaną będącą miejscem geometrycznym punktów równo odległych od każdej z tych klas. Jeżeli zostaną użyte dwie cechy *x* i *y*, to otrzymana zostanie linia łamana H_2, rozdzielająca obie klasy. Wówczas punkty z klasy 1 (krzyżyki), znajdujące się w obszarze *A*, będą mylnie klasyfikowane do klasy 2 (kółka).



Rys. 2.1. Ilustracja sytuacji, w której zbędne cechy mogą szkodzić

Natomiast punkty z klasy 2, leżące w obszarach B_1 i B_2 będą nieprawidłowo zaliczane do klasy 1. Znacznie lepiej byłoby w tym przypadku użyć tylko jednej cechy *x*. Wówczas w przestrzeni jednowymiarowej, którą stanowi oś cechy *x*, hiperpowierzchnią rozdzielającą byłby punkt przecięcia się prostej H_1 z osią *x*. Obecność zbędnej cechy *y* nie pogarsza jakości klasyfikacji, jeśli tylko nie będzie ona wykorzystana do konstrukcji klasyfikatora. Ten niekorzystny efekt, polegający na tym, że zbędne cechy mogą szkodzić wynika z niedostatecznej liczności zbioru uczącego. Gdyby obszary A były wypełnione licznymi punktami z klasy 1, a obszary B_1 i B_2 dużą liczbą punktów z klasy 2, to zbędne cechy co najwyżej nie poprawiały by jakości klasyfikacji, ale też nie mogłyby jej znacząco szkodzić.

Możliwa jest też inna ciekawa sytuacja, polegająca na tym, że cechy, które użyte pojedynczo nie mają żadnego znaczenia, w zestawie mogą stanowić nawet idealną kombinację cech, co ilustruje Rys. 2.2.



Rys. 2.2. Cechy nieznaczące pojedynczo mogą tworzyć dobry zestaw

Po zrzutowaniu punktów zarówno na oś *x* jak i na oś *y* klasy są bardzo trudno rozdzielne i użycie pojedynczo którejkolwiek z tych cech nie miałoby sensu. Użycie obu takich cech jednocześnie oferuje bardzo dobrą jakość klasyfikacji.

Te proste przykłady potwierdzają celowość uwzględnienia selekcji cech podczas konstruowania klasyfikatora. Pomysłodawcą obu rysunków, Rys. 2.1 oraz Rys. 2.2 jest pan mgr Wojciech Siedlecki, były podopieczny autora niniejszej monografii, jako promotora pracy magisterskiej.

Idealna selekcja cech mogłaby polegać na przeglądzie wszystkich możliwych kombinacji spośród *n* cech, szacując dla każdej z nich prawdopodobieństwo mylnej decyzji i wyborze tej kombinacji, która oferuje najmniejszą wartość w/w prawdopodobieństwa.

Należy jednak uwzględnić fakt, że liczba możliwych kombinacji cech szybko rośnie wraz ze wzrostem liczby *n* dostępnych cech i wynosi ona 2^n-1. Z tego względu stosuje się bardziej oszczędne obliczeniowo procedury przeglądu tylko niektórych kombinacji cech [Devijver P.A., Kittler J., 1982]. Polegają one na kolejnym dołączaniu cech, kolejnym odrzucaniu cech lub na kombinacji dołączania i odrzucania.

W przypadku procedury kolejnego dołączania cech należy najpierw wyznaczyć cechę dla której frakcja błędów osiąga wartość minimalną. Potem do tej cechy na wszystkie możliwe sposoby dołączana jest druga cecha i spośród zweryfikowanych w ten sposób par cech wybrana zostaje para, oferująca najmniejszą frakcję błędów. W podobny sposób tworzone są trójki cech, czwórki i większe zestawy cech, kończąc na pełnym zestawie dostępnych *n* cech. Przykładowy przebieg selekcji cech z zastosowaniem procedury kolejnego dołączania cech pokazano w Tab. 2.1.

Tab. 2.1. Przykładowy przebieg procedury kolejnego dołączania cech

(1) 0,50	(4,1) 0,40	(4,3,1) 0,20	(4,3,5,1) 0,10*	(4,3,5,1,2) 0,20*
(2) 0,30	(4,2) 0,50	(4,3,2) 0,25	(4,3,5,2) 0,15	–
(3) 0,40	(4,3) 0,15*	(4,3,5) 0,10*	–	–
(4) 0,20*	(4,5) 0,30	–	–	–
(5) 0,30	–	–	–	–

Zestawy cech zostały podane w nawiasach, a obok nich frakcje błędów, które te zestawy oferują. Gwiazdkami zostały zaznaczone najlepsze zestawy cech w fazach kolejnych dołączeń. Spośród wszystkich kombinacji cech wybrana zostaje ta, dla której frakcja błędów jest najmniejsza. W podanym przykładzie będzie to zestaw złożony z cech 3, 4 i 5 oferujący frakcję błędów er=0,1. Taka sama frakcja błędów została

obliczona dla zestawu cech 1, 3, 4 oraz 5, ale lepiej jest wybrać zestaw z mniejszą liczbą cech.

Stosując procedurę kolejnego odrzucania cech, pierwszy weryfikowany zestaw cech zawiera wszystkie *n* cech. Następnie na wszystkie możliwe sposoby wyrzucana jest jedna z cech, tzn. oceniane są kombinacje złożone z *n*-1 cech i wybrana zostaje taka kombinacja dla której uzyskana została najmniejsza wartość frakcji błędów. W podobny sposób z wybranej kombinacji tworzone są kombinacje po *n*-2 cechy, itd., kończąc na kombinacji zawierającej tylko jedną cechę. Działanie tej procedury pokazano w Tab. 2.2. Tym razem wybranym zestawem cech okazała się kombinacja złożona z dwóch cech, 3 i 5, która oferuje frakcje błędów *er*=0,06. Przykłady obu przedstawionych procedur przeglądania zestawów cech pochodzą z pracy [Jóźwik A., 2005]

Tab. 2.2. Przykładowy przebieg procedury kolejnego odrzucania cech

(1,2,3,4,5) 0,20*	(2,3,4,5) 0,15	(3,4,5) 0,10*	(3,4) 0,15	(3) 0,40
–	(1,3,4,5) 0,10*	(1,4,5) 0,15	(3,5) 0,06*	(5) 0,30*
–	(1,2,4,5) 0,20	(1,3,5) 0,25	(4,5) 0,30	–
–	(1,2,3,5) 0,25	(1,3,4) 0,20	–	–
–	(1,2,3,4) 0,15	–	–	–

Możliwa jest również procedura stanowiąca kombinację obu w/w opisanych procedur przeglądu cech. Jej przebieg zaczyna się identycznie jak przebieg procedury kolejnego dołączania cech.

Począwszy od wyboru najlepszej trójki cech, zastosowana zostaje procedura odrzucania cech i kontynuowana jest tak długo jako długo uzyskiwane są zestawy cech, zawierające ta samą liczbę cech, co zestawy otrzymane dotychczas, ale oferujące mniejsze frakcje błędów. Ponieważ procedura ta przebiega początkowo w taki sam sposób, jak procedura kolejnego dołączania cech, to wystarczy zilustrować jej przebieg dopiero od momentu, gdy chwilowo zostanie wybrany zestaw złożony z trzech cech. Fragment tej procedury przedstawiony został w Tab. 2.3.

Tab. 2.3. Przykładowy przebieg procedury kombinowanej

(4,5) 0,30	(3,5,1) 0,25	(2,3) 0,08	(1,2,5) 0,15	(1,2,4,5) 0,20	(1,2,3,4,5) 0,20*
(3,5) 0,06*	(3,5,2) 0,05*	(2,5) 0,04*	(2,3,5) 0,20	(2,3,4,5) 0,15*	–
(4,3) 0,15	(3,4,5) 0,10	(3,5) 0,06	(2,4,5) 0,10*	–	–

Pierwsza kolumna Tab. 2.3 zawiera zestawy dwójek cech otrzymane po wyrzuceniu pojedynczej cechy z zestawu cech 3, 4 i 5 uzyskanego procedurą kolejnego dołączania cech, jak przedstawione to zostało w pierwszych trzech kolumnach Tab. 2.1. Najlepszą z tych dwójek cech jest para cech 3 i 5. Do tej pary na wszystkie możliwe sposoby dołączana jest jedna z pozostałych cech. Najlepsza z tak otrzymanych trójek cech składa się z cech 2, 3 i 5 (kolumna 2). Teraz ponownie przeprowadzane jest odrzucanie cech i w efekcie udaje się znaleźć zestaw cech 2 i 5, dla którego osiągnięta została, jak dotąd, najmniejsza frakcja błędów.

Dołączanie do tej dwójki cechy trzeciej nie przyniosło poprawy, bo otrzymana najlepsza z tych trójek, zawierająca cechy 2, 4 i 5 nie jest lepsza od dotychczas uzyskanej trójki złożonej z cech 3, 4 oraz 5, gdyż obie te trójki oferują frakcje błędów *er*=0,1. Gdyby było inaczej, to ponownie należałoby zastosować kolejne odrzucanie cech.

Nie ma ogólnych zasad wskazujących, którą z dwóch podstawowych procedur należy zastosować, dołączanie albo odrzucanie cech. W przypadkach, gdy liczba dostępnych *n* cech jest duża, porównywalna z liczebnością zbioru uczącego, zastosowanie procedury kolejnego odrzucania cech może być nie możliwe, gdyż mogą być otrzymywane liczne zestawy cech oferujące takie same wartości frakcji błędów. Może się dziać tak dlatego, że gdy liczba cech jest duża, to odrzucenie jednej, niezależnie której cechy, w ogóle może nie zmieniać wartości frakcji błędów. Wówczas bardziej racjonalne jest zastosowanie kolejnego dołączania cech lub procedury kombinowanej, chociaż ta druga jest znacznie bardziej złożona niż kolejne dołączanie cech. Sytuacje niejednoznaczne można rozstrzygać w różny sposób, np. na korzyść cechy, która pojedynczo oferuje mniejszą frakcje błędów lub której odrzucenie powodowałoby największy jego wzrost frakcji błędów.

3. WYZNACZANIE HIPERPŁASZCZYZNY ROZDZIELAJĄCEJ

Liniowa funkcja dyskryminacyjna może być wykorzystana w regule decyzyjnej odnoszącej się do dwóch klas, jak również może być bardzo użytecznym narzędziem do konstruowania klasyfikatorów wielo-decyzyjnych z wykorzystaniem struktury równoległej, którą podano na Rys. 1.2.

Prace dotyczące algorytmów wyznaczania hiperpłaszczyzn rozdzielających dwa zbiory X_1 i X_2 w przestrzeni cech prowadzone są od bardzo dawna [np. Nilsson N., 1965], w której to publikacji zaproponowany został jeden z najprostszych algorytmów Jest to algorytm heurystyczny, który, przy założeniu liniowej rozdzielności zbiorów, zapewnia uzyskanie hiperpłaszczyzny rozdzielającej w skończonej liczbie kroków.

Na uwagę szczególną uwagę zasługuje też algorytm zaprezentowany w pracy [Koziniec B.N., 1973], polegający na szukaniu pary najbliższych punktów, z których jeden należy do powłoki wypukłej zbioru X_1, a drugi do powłoki wypukłej zbioru X_2.

Jeszcze inne algorytmy mają charakter rekursywny i polegają na obrotach hiperpłaszczyzn w n wymiarowej przestrzeni cech wokół n-1 wymiarowej osi [Jóźwik A., 1983a; Jóźwik A., 1998a; Cendrowska D., 2005; Sturgulewski L., 2008]. Duża grupa algorytmów do wyznaczania hiperpłaszczyzny rozdzielającej ma charakter gradientowy jak np. przedstawione w pracach [Ho Y., C., Kashyap R. L., 1965; Duda R., O., Hart P.E., Stork D., G., 2001].

Algorytmy wyznaczania hiperpłaszczyzny rozdzielającej wykorzystują najczęściej podejście polegające na minimalizacji liniowej [Bobrowski L., Niemiro W., 1984], bądź kwadratowej funkcji celu [Mangasarian O.L., 2000; Vapnik W.N., 2000], z ograniczeniami liniowymi. Wśród tych metod szczególną popularnością cieszą się obecnie metody określane mianem maszyn wektorów podpierających (SVM) autorstwa Vapnika. We wcześniejszych jego pracach podobne podejście określane było metodą portretów uogólnionych [Vapnik W.N., Chervonenkis A. J.,1974]. Do znajdowania hiperpłaszczyzny rozdzielającej, poprzez minimalizację kwadratowej funkcji celu, w wymienionej pracy proponowana była metoda gradientów sprzężonych. Metoda portretów uogólnionych pozwala rozstrzygnąć, czy zbiory są, czy też nie, liniowo rozdzielne, a jeśli tak, to wyznaczyć hiperpłaszczyznę optymalną, w sensie jej odległości od najbliższego punktu z rozdzielonych zbiorów.

Obecnie zaś, w powiązaniu z metodą wektorów podpierających, stosowana jest częściej metoda mnożników Lagrange'a. W najtrudniejszym przypadku, gdy zbiory nie są separowalne liniowo, do funkcji celu wprowadza się współczynnik kary, który dobiera się eksperymentalnie. Innym rozwiązaniem, które może być zastosowane również z innymi metodami wyznaczania hiperpowierzchni rozdzielającej, jest przekształcenie rozdzielanych zbiorów do nowej przestrzeni cech, w której zbiory te mogą okazać się już liniowo rozdzielne. Podobny sposób był już sugerowany w pracy Nilssona [Nilsson N., 1965] z użyciem funkcji $\Phi_i(\underline{x})$, i=1,2,…,N. Wektory $\underline{x}$ opisujące obiekty w przestrzeni cech n wymiarowej, poprzez zastosowanie funkcji $\Phi_i(\underline{x})$ są przekształcane w wektory $\underline{\Phi}(\underline{x})=[\Phi_1(\underline{x}), \Phi_2(\underline{x}),\ldots, \Phi_N(\underline{x})]$ w przestrzeni N wymiarowej. Funkcje $\Phi_i(\underline{x})$ powinny być tak dobrane, aby w nowej przestrzeni cech liniowe rozdzielenie zbiorów było łatwiejsze.

Ideałem algorytmu pozwalającym skonstruować liniową funkcję dyskryminacyjną byłby taki algorytm, który by typował minimalny podzbiór punktów ze zbioru $X=X_1\cup X_2$, po usunięciu którego, pomniejszone w ten sposób zbiory X_1 i X_2 dałyby się już rozdzielić hiperpłaszczyzną, co jest równoważne skonstruowaniu takiej hiperpłaszczyzny, aby liczba punktów ze zbioru $X=X_1\cup X_2$ leżących po jej niewłaściwej

stronie była minimalna. Rozwiązanie tego zadania nie musi być oczywiście jednoznaczne.

W niniejszej monografii rozważania poświęcone będą wybranym, mniej skomplikowanym metodom wyznaczania hiperpłaszczyzny rozdzielającej, z których niektóre zostały zaproponowane przez autora niniejszej monografii.

3.1. Algorytm korekcji błędów

Liniowa rozdzielność podzbiorów zbioru uczącego, reprezentujących różne klasy, może bardzo uprościć konstrukcje klasyfikatora. Hiperpłaszczyzną można rozdzielić tylko dwa zbiory, ale ten fakt nie musi być przeszkodą do skonstruowania klasyfikatora wielo-decyzyjnego, co zostało już wyjaśnione wcześniej na Rys. 1.2.

Poza tym liniowo nierozdzielne zbiory X_1 i X_2 mogą być przekształcone w nową przestrzeń cech, zwykle o wyższej wymiarowości, w której już dadzą się rozdzielić hiperpłaszczyzną.

Liniowa rozdzielność (ścisła) dwóch zbiorów X_1 i X_2 w przestrzeni cech oznacza spełnienie niżej wymienionych warunków:

$$\begin{gathered} g(\underline{x})>0, \text{ gdy } \underline{x}\in X_1 \\ \text{oraz} \\ g(\underline{x})<0, \text{ gdy } \underline{x}\in X_2, \end{gathered} \tag{3.1}$$

gdzie $g(\underline{x})= \sum_{j=1}^{n} w_j \circ x_j + w_{n+1}$ i $\underline{x}=[x_1,x_2,\ldots,x_n]$. Funkcję dyskryminacyjną $g(\underline{x})$ można wyrazić zwięźlej przyjmując, że $\underline{w}=[w_1,w_2,w_3,\ldots,w_n]$. Wówczas $g(\underline{x})=\underline{w}*\underline{x}+w_{n+1}$, a warunki (3.1) zapisać można w postaci:

$$\begin{gathered} \underline{w}\circ\underline{x}+w_{n+1}>0, \text{ gdy } \underline{x}\in X_1 \\ \text{oraz} \\ \underline{w}\circ\underline{x}+w_{n+1}<0, \text{ gdy } \underline{x}\in X_2 \end{gathered} \tag{3.2}$$

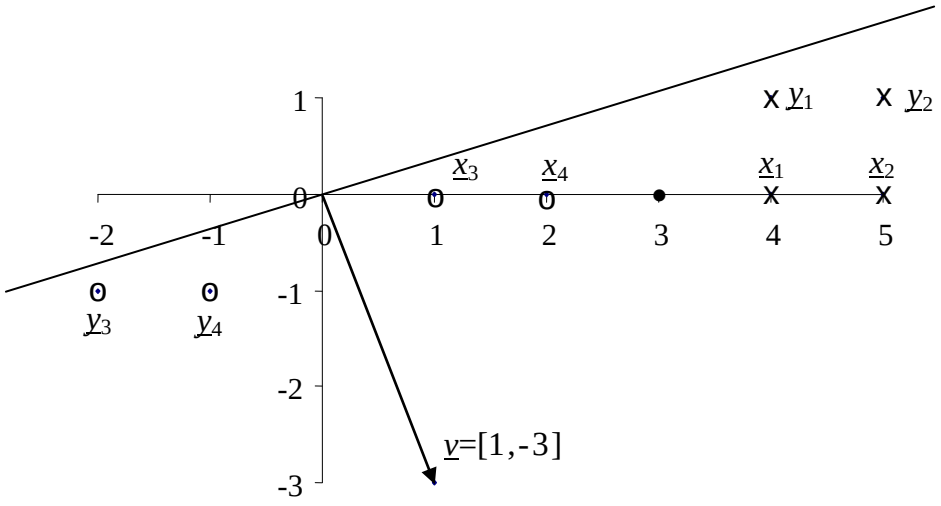
Jeżeli zbiór $X=X_1 \cup X_2$ z przestrzeni E^n przekształcony zostanie w zbiór Y w przestrzeni E^{n+1} zgodnie z formułą:

$$\begin{gathered} \underline{y}=h(\underline{x})=[x_1,x_2,\ldots,x_n,1], \text{ gdy } \underline{x}\in X_1 \\ \text{oraz} \\ \underline{y}=h(\underline{x})=[-x_1,-x_2,\ldots,-x_n,-1], \text{ gdy } \underline{x}\in X_2 \end{gathered} \tag{3.3}$$

i utworzony zostanie $(n+1)$-wymiarowy wektor $\underline{v}=[\underline{w},w_{n+1}]=[w_1,w_2,w_3,\ldots,w_n,w_{n+1}]$, to układ nierówności (3.2) przekształci się do postaci:

$$\underline{v}\circ\underline{y}>0, \text{ gdy } \underline{y}\in Y, \tag{3.4}$$

gdzie $Y=h(X)$. Działanie przekształcenia (3.3) zilustrowane zostało na Rys. 3.1.



Rys. 3.1. Ilustracja przekształcenia 3.3

Obiekty $\underline{x}_1$, $\underline{x}_2$, $\underline{x}_3$ i $\underline{x}_4$ leżące na prostej, czyli w przestrzeni jednowymiarowej, zostały przekształcone w przestrzeń dwuwymiarową. Liniowa rozdzielność zbiorów X_1 oraz X_2 oznacza, że zbiór $Y=\{\underline{y}_1,\underline{y}_2,\underline{y}_3,\underline{y}_4\}$ leży po jednej stronie pewnej hiperpłaszczyzny przechodzącej przez początek układu współrzędnych w przestrzeni dwuwymiarowej.

Wektor $\underline{v}$, dla którego zachodzi relacja (3.4), jednoznacznie określa liniową funkcję dyskryminacyjną $g(\underline{x})$ spełniającą warunki (3.1). W pracy Nilssona [Nilsson N., 1965] zostało podane w/w przekształcenie (3.3) zbioru $X=X_1 \cup X_2$ w zbiór Y oraz bardzo prosty, niżej przytoczony, algorytm znajdowania wektora $\underline{v}$ będącego rozwiązaniem układu nierówności (3.4).

Na podstawie zbioru Y utworzony zostaje nieskończony ciąg $(\underline{y}_k)_{k=1}^{\infty}$, w którym każdy element zbioru Y występuje nieskończoną liczbę razy. W przypadku, gdy $Y=\{\underline{y}_k\}_{k=1}^{m}$, to ciąg $(\underline{y}_k)_{k=1}^{\infty}$ może mieć postać: $(\underline{y}_1,\underline{y}_2,\ldots,\underline{y}_m, \underline{y}_{m+1},\underline{y}_{m+2},\ldots,\underline{y}_{2\cdot m},\ldots)$, przy czym $\underline{y}_{i\cdot m+k}=\underline{y}_k$, dla dowolnej liczby całkowitej i. Na podstawie ciągu $(\underline{y}_k)_{k=1}^{\infty}$ tworzony jest ciąg wektorów $(\underline{v}_k)_{k=1}^{\infty}$ wg niżej podanej reguły.

Definicja algorytmu:

$$\begin{aligned}
&\underline{v}_0=[0_1,0_2,\ldots,0_{n+1}], \text{ czyli jest to } (n+1) \text{ wymiarowy wektor zerowy;}\\
&\underline{v}_k=\underline{v}_{k-1}, \text{ gdy } \underline{v}_{k-1}\circ\underline{y}_k>0; \qquad (3.5)\\
&\underline{v}_k=\underline{v}_{k-1}+a\circ\underline{y}_k, \text{ gdy } \underline{v}_{k-1}\circ\underline{y}_k\leq 0,
\end{aligned}$$

gdzie a jest liczbą dodatnią, która nazywana jest współczynnikiem korekcji oraz jest dobierana eksperymentalnie tak, aby nie przekroczyć dopuszczalnego zakresu liczb, zazwyczaj jednak wystarczy przyjąć w definicji (3.5) algorytmu, że $a=1$.

Godną uwagi jest też wersja tego algorytmu ze współczynnikiem korekcji absolutnej, w której współczynnik korekcji zmienia się na każdym kroku iteracji wg reguły $a_k=-\underline{v}_{k-1}\circ\underline{y}_k/(\underline{y}_k)^2$. Może też być najmniejszą liczbą całkowitą większą niż liczba $-\underline{v}_{k-1}\circ\underline{y}_k/(\underline{y}_k)^2$. Współczynnik a_k jest dodatni, ponieważ $\underline{v}_{k-1}\circ\underline{y}_k$ jest ujemne. Współczynnik korekcji absolutnej gwarantuje, że po korekcji rozważany punkt $\underline{y}_k$ da nieujemny iloczyn skalarny $\underline{v}_k\circ\underline{y}_k$.

Zasada działania algorytmu korekcji błędów wynika ze spostrzeżenia, że $\underline{v}_k\circ\underline{y}_k=(\underline{v}_{k-1}+a\circ\underline{y}_k)\circ\underline{y}_k,\geq\underline{v}_{k-1}\circ\underline{y}_k$, to znaczy, że po korekcji wartość iloczynu skalarnego, odpowiadająca wektorowi $\underline{y}_k$, ulega powiększeniu. Taka zmiana idzie w pożądanym kierunku, gdyż celem zastosowania algorytmu jest znalezienie takiego wektora $\underline{v}$, który z każdym elementem zbioru Y da dodatni iloczyn skalarny, jak to określa warunek (3.4). Korekta powiększa wartość iloczynu z $\underline{v}_{k-1}\circ\underline{y}_k$ na $\underline{v}_k\circ\underline{y}_k$, ale wciąż nowa wartość tego iloczynu dla $\underline{y}_k$ może być ujemna, jeśli nie został zastosowany współczynnik korekcji absolutnej. Współczynnik korekcji absolutnej doprowadza ten iloczyn skalarny co najmniej do zera, a jeśli przed korektą był równy zeru, to do wartości dodatniej.

W wymienionej już wyżej książce podane zostało następujące twierdzenie.

Twierdzenie 3.1

Jeśli zbiory X_1 i X_2 są liniowo rozdzielne, to algorytm ten znajdzie rozwiązanie układu nierówności (3.4) w skończonej liczbie kroków korekcji.

Warto przytoczyć dowód tego twierdzenia, gdyż w dalszej części monografii zostanie opisana modyfikacja, której zbieżność będzie dowiedziona w podobny sposób.

Dowód

Dla uproszczenia rozważań warto przyjąć, że $a=1$. Ponadto, wygodnie jest założyć, że w ciągu $(\underline{y}_k)_{k=1}^{\infty}$ są tylko takie elementy, które wymagały korekcji. Przy tym założeniu: $\underline{v}_k=\underline{y}_1+\underline{y}_2+\ldots+\underline{y}_k$, a z pomnożenia obu stron przez wektor $\underline{v}$ będący rozwiązaniem układu (3.4), którego istnienie wynika z założonej liniowej rozdzielności zbiorów X_1 i X_2 wynika, że

$$\underline{v}\circ\underline{v}_k=\underline{v}\circ\underline{y}_1+\underline{v}\circ\underline{y}_2,\ldots,\underline{v}\circ\underline{y}_k\geq k\circ b, \tag{3.6}$$

gdzie $b=\min_{y\in Y}\underline{v}\circ\underline{y}>0$. Z nierówności Cauchego wynika zaś relacja $\underline{v}^2\circ\underline{v}_k^2\geq(\underline{v}\circ\underline{v}_k)^2$, a po uwzględnieniu (3.6) wynika prawdziwość następującej nierówności:

$$\underline{v}_k^2\geq(\underline{v}\circ\underline{v}_k)^2/\underline{v}^2\geq k^2\circ b^2/\underline{v}^2. \tag{3.7}$$

Z drugiej strony, wprost z definicji algorytmu wynika, że

$\underline{v}_k^2 = (\underline{v}_{k-1} + \underline{y}_k)^2 = \underline{v}_{k-1}^2 + 2 \circ \underline{v}_{k-1} \circ \underline{y}_k + \underline{y}_k^2$, ale $2 \circ \underline{v}_{k-1} \circ \underline{y}_k \leq 0$, gdyż na mocy założenia wektor $\underline{y}_k$ wymagał korekty, dlatego

$$\underline{v}_k^2 \leq \underline{v}_{k-1}^2 + \underline{y}_k^2 \leq \underline{v}_{k-2}^2 + \underline{y}_{k-1}^2 + \underline{y}_k^2 \leq \ldots \leq \sum_{j=1}^{k} \underline{y}_j^2 \leq k \circ c, \tag{3.8}$$

gdzie liczba $c = \max_{y \in Y} \underline{y}^2$.

Kojarząc z (3.7) i (3.8) otrzymuje się, że $k^2 \circ b^2 / \underline{v}^2 \leq \underline{v}_k^2 \leq k \circ c$, czyli $k^2 \circ b^2 / \underline{v}^2 \leq k \circ c$, a stąd już można uzyskać ograniczenie na k:

$$k \leq c \circ \underline{v}^2 / b^2, \tag{3.9}$$

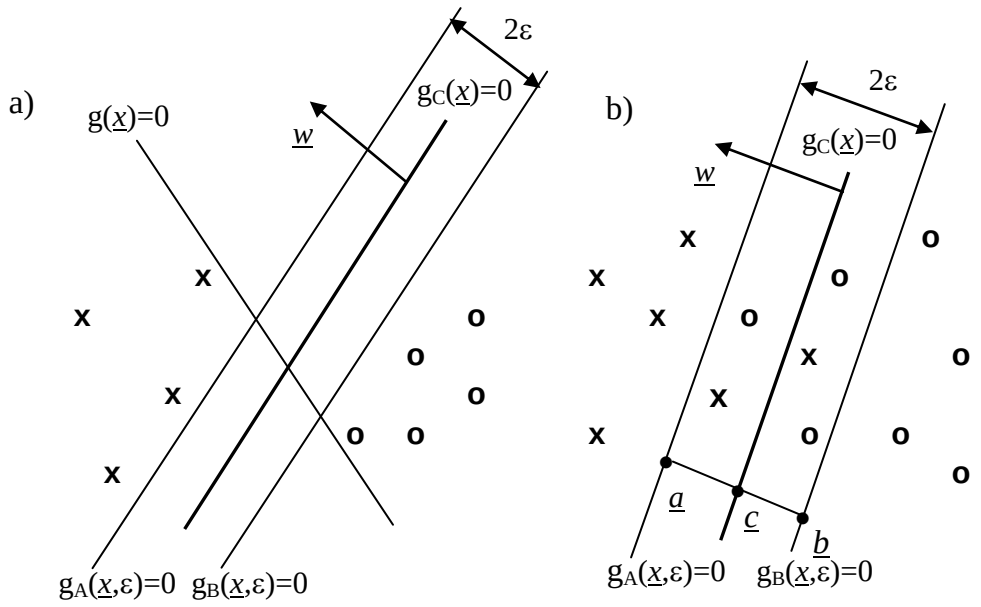
co kończy dowód twierdzenia.

A priori nie jest dana żadna informacja, czy zbiory X_1 i X_2 są liniowo rozdzielne, czy nie. Dlatego, żeby algorytm mógł się kiedyś na pewno zatrzymać należy ograniczyć liczbę iteracji. Iteracje tego algorytmu są bardzo mało kosztowne obliczeniowo, więc warto wykonać szereg eksperymentów przyjmując coraz to większe liczby ograniczające liczbę kroków, wyznaczając przy tym liczbę obiektów $\underline{y}$ ze zbioru Y, które spełniają warunek (3.4). Jako ograniczenia można przyjmować liczby będące wielokrotnością liczby m obiektów w zbiorze Y. Jeżeli, kilkukrotne zwiększenie liczby iteracji, w kolejnym eksperymencie, nie przyniesie pozytywnego wyniku, czyli uzyskania rozwiązania układu (3.4), to można uznać, że zbiory są liniowo nierozdzielne lub są trudno rozdzielne, tj. zbyt bliskie siebie, by osiągnąć sukces.

Przypadek liniowej nierozdzielności

Pomimo, że zbieżność tego algorytmu została wykazana tylko dla zbiorów X_1 i X_2 liniowo rozdzielnych, to może on być użyteczny również w przypadku ich liniowej nierozdzielności. Wystarczy po każdej korekcie w kroku k sprawdzać dla ilu elementów $\underline{y}_i$, $i=1,2,\ldots,m$, ze zbioru Y prawdziwa jest nierówność $\underline{v}_k \circ \underline{y}_i > 0$ i zapamiętywać ten wektor $\underline{v}^*$ spośród dotąd otrzymanych wektorów $\underline{v}_i$, $i=1,2,\ldots,k$, dla którego liczba takich wektorów jest największa. Dąży się w ten sposób do otrzymania hiperpłaszczyzny, określonej składowymi wektora $\underline{v}^*$, która poprawnie rozdziela możliwie największą liczbę obiektów ze zbiorów X_1 i X_2. Zastosowanie takiego postępowania do znanego zbioru Iris Data (http://archive.ics.uci.edu/ml/datasets.html, 3 klasy, 4 cechy i 150 obiektów), wraz ze strukturą równoległą jak na Rys. 1.2. z rozdziału 1, daje w wyniku klasyfikator, który na tym zbiorze myli się tylko 3 razy, co oznacza frakcję błędów $e=0{,}02$. Liczba iteracji dla par klas (1,2), (1,3) i (2,3) wyniosła odpowiednio 201, 201 i 8701. Przyjęte ograniczenie na liczbę iteracji, tj. prezentacji wektorów ze zbioru Y dla każdej z par klas wynosiło 10000.

Może się jednak zdarzyć, że wyznaczona hiperpłaszczyzna rozdzielająca, o równaniu $g(\underline{x})=0$, przebiegać ona będzie w bardzo małej odległości, jak to pokazano na Rys. 3.2a, od obiektów ze zbioru $X=X_1\cup X_2$, co nie jest wskazane.



Rys. 3.2. Ilustracja modyfikacji algorytmu korekcji błędów

Algorytm korekcji błędów można jednak zmodyfikować tak, by móc z jego pomocą znajdować parę równoległych hiperpłaszczyzn rozdzielających, $g_A(\underline{x})=0$ i $g_B(\underline{x})=0$, odległych od siebie o $2\circ\varepsilon$, jak zostało to zilustrowane na Rys. 3.2a. Uogólnienie takie, w inny sposób niż opisane poniżej, zostało zasugerowane w książce [Duda R., O., Hart P.E., Stork D., G., 2001].

Tego typu rozdzielność określana jest jako rozdzielność z prześwitem $ps=2\circ\varepsilon$ lub jako rozdzielność z marginesem ε. Intuicja podpowiada, że po klasyfikatorze dwu-decyzyjnym, zdefiniowanym funkcją dyskryminacyjną $g_C(\underline{x})=[g_A(\underline{x})+g_B(\underline{x})]/2$ (dzielenie przez 2 nie jest konieczne) można się spodziewać mniejszej frakcji mylnych klasyfikacji niż, gdy zostanie użyta funkcja dyskryminacyjna $g(\underline{x})$. Jeżeli przyjąć, że $g_C(\underline{x})=\underline{w}\circ\underline{x}+w_{n+1}=0$, to łatwo pokazać, że obie równoległe hiperpłaszczyzny, odległe od hiperpłaszczyzny $g_C(\underline{x})=0$ o ε, określone są równaniami: $g_A(\underline{x},\varepsilon)=\underline{w}\circ\underline{x}+w_{n+1}-\varepsilon\circ|\underline{w}|$ oraz $g_B(\underline{x},\varepsilon)=\underline{w}\circ\underline{x}+w_{n+1}+\varepsilon\circ|\underline{w}|$, gdzie $|\underline{w}|$ oznacza długość wektora $\underline{w}$.

Rzeczywiście, niech punktom $\underline{a}$, $\underline{b}$ na Rys. 3.2b, odpowiadają hiperpłaszczyzny przechodzące przez punkty $\underline{a}$ i $\underline{b}$ równolegle do hiperpłaszczyzny $g_C(\underline{x})=0$, przechodzącej przez punkt $\underline{c}$, wyrażają się odpowiednio równaniami postaci: $g_A(\underline{x})=\underline{w}\circ\underline{x}+w_{n+1}+a=0$ oraz $g_B(\underline{x})=\underline{w}\circ\underline{x}+w_{n+1}+b=0$. Wektor $\underline{a}=\underline{c}+\lambda\circ\underline{w}$, stąd $|a\text{-}c|=\lambda\circ|\underline{w}|=\varepsilon$, czyli $\lambda=\varepsilon/|\underline{w}|$. Podobnie $\underline{b}=\underline{c}-\lambda\circ\underline{w}$. Wektor $\underline{a}$ spełnia równanie $g_A(\underline{a})=\underline{w}\circ\underline{a}+w_{n+1}+a=0$, gdyż odpowiadający mu punkt $\underline{a}$ leży na hiperpłaszczyźnie $g_A(\underline{x})=0$, czyli

$g_A(\underline{a})=\underline{w}\circ(\underline{c}+\lambda\circ\underline{w})+w_{n+1}+a=\underline{w}\circ\underline{c}+w_{n+1}+a+\lambda\circ\underline{w}^2=a+\lambda\circ\underline{w}^2=0$, ponieważ pierwsze dwa składniki, to liczba $g_C(\underline{c})$, a ta jest równa zeru, z tej racji, że punkt $\underline{c}$ leży na hiperpłaszczyźnie $g_C(\underline{x})=0$. Wobec tego $a=-\lambda\circ\underline{w}^2=-(\varepsilon/|\underline{w}|)\circ\underline{w}^2=-\varepsilon\circ|\underline{w}|$.

Zatem również $g_B(\underline{b})=\underline{w}\circ(\underline{c}-\lambda\circ\underline{w})+w_{n+1}+b=\underline{w}\circ\underline{c}+w_{n+1}+b-\lambda\circ\underline{w}^2=b-\lambda\circ\underline{w}^2=0$,

stąd $b=\lambda\circ\underline{w}^2=(\varepsilon/|\underline{w}|)\circ\underline{w}^2=\varepsilon\circ|\underline{w}|$, gdyż $\underline{w}\circ\underline{c}+w_{n+1}=g_C(\underline{c})=0$.

Para funkcji $g_A(\underline{x},\varepsilon)$ oraz $g_B(\underline{x},\varepsilon)$ powinna czynić zadość następującym układom nierówności:

$$\begin{gathered} g_A(\underline{x},\varepsilon)=\underline{w}\circ\underline{x}+w_{n+1}-\varepsilon\circ|\underline{w}|>0,\ \text{gdy}\ \underline{x}\in X_1 \\ \text{oraz} \\ g_B(\underline{x},\varepsilon)=\underline{w}\circ\underline{x}+w_{n+1}+\varepsilon\circ|\underline{w}|<0,\ \text{gdy}\ \underline{x}\in X_2. \end{gathered} \tag{3.10}$$

Po pomnożeniu obustronnie drugich z w/w nierówności przez -1, układ (3.10) będzie miał postać:

$$\begin{gathered} \underline{w}\circ\underline{x}+w_{n+1}-\varepsilon\circ|\underline{w}|>0,\ \text{gdy}\ \underline{x}\in X_1 \\ \text{oraz} \\ \underline{w}\circ(-\underline{x})+w_{n+1}\circ(-1)-\varepsilon\circ|\underline{w}|>0,\ \text{gdy}\ \underline{x}\in X_2. \end{gathered} \tag{3.11}$$

Korzystając teraz z odwzorowania (3.3) zbioru $X=X_1\cup X_2$ w zbiór w Y i pamiętając, że $\underline{v}=[\underline{w},w_{n+1}]$, układ nierówności (3.11) przyjmie formę:

$$\underline{v}\circ\underline{y}-\varepsilon\circ|\underline{w}|>0,\ \text{gdy}\ \underline{y}\in Y. \tag{3.12}$$

Modyfikacja I

$$\begin{aligned} &\underline{v}_0=[0_1,0_2,\ldots,0_{n+1}],\ \text{czyli jest to}\ (n+1)\ \text{wymiarowy wektor zerowy;} \\ &\underline{v}_k=\underline{v}_{k-1},\ \text{gdy}\ \underline{v}_{k-1}\circ\underline{y}_k-\varepsilon\circ|\underline{w}_{k-1}|>0; \\ &\underline{v}_k=\underline{v}_{k-1}+a\circ\underline{y}_k,\ \text{gdy}\ \underline{v}_{k-1}\circ\underline{y}_k-\varepsilon\circ|\underline{w}_{k-1}|\le 0. \end{aligned} \tag{3.13}$$

Twierdzenie dotyczące tej modyfikacji, będące odpowiednikiem Twierdzenia 3.1 będzie teraz brzmiało:

Twierdzenie 3.2

Jeżeli zbiory X_1 i X_2 są rozdzielne z prześwitem $ps=2\circ\varepsilon$, czyli z marginesem ε, to algorytm działający wg modyfikacji I, znajdzie wektor $\underline{v}$ czyniący zadość układowi nierówności (3.12) w skończonej liczbie kroków, a składowe tego wektora jednoznacznie określą funkcje $g_A(\underline{x},\varepsilon)$ i $g_B(\underline{x},\varepsilon)$ spełniające warunki (3.10).

Dowód tego twierdzenia jest bardzo łatwy i podobny do przytoczonego już wyżej dowodu twierdzenia dotyczącego wersji oryginalnej. Jednak dla porównania tej modyfikacji z oryginalną wersją algorytmu zostanie niżej podany.

Dowód

W miejsce relacji (3.5) wektor $\underline{v}$, spełniający układ (3.12), na mocy założenia, że zbiory X_1 i X_2 są rozdzielne z prześwitem $ps=2\circ\varepsilon$, istnieje. Tym razem zamiast relacji (3.6) spełnia on następujący warunek:

$$\underline{v}\circ\underline{v}_k=\underline{v}\circ\underline{y}_1+\underline{v}\circ\underline{y}_2,\ldots,\underline{v}\circ\underline{y}_k\geq k\circ(b+\varepsilon\circ|\underline{w}|), \tag{3.14}$$

gdzie jak w wersji oryginalnej $b=\min_{y\in Y}\underline{v}\circ\underline{y}$. Jest oczywiste, że $b>\varepsilon\circ|\underline{w}|$, ponieważ z 3.12 wynika, że $\underline{v}\circ\underline{y}-\varepsilon\circ|\underline{w}|>0$, gdy $\underline{y}\in Y$. Wobec tego w relacji 3.9 w miejsce liczby b należy podstawić liczbę $(b+\varepsilon\circ|\underline{w}|)$, co daje w wyniku nowe ograniczenie na liczbę kroków k:

$$k\leq c\circ\underline{v}^2/(b+\varepsilon\circ|\underline{w}|)^2. \tag{3.15}$$

Można więc uznać, że liczba b jest to ta sama liczba, która występuje w relacji (3.9). Warto jednak zdać sobie sprawę z faktu, że w sensie ścisłym nie jest to prawda. W przypadku wersji oryginalnej przyjęte było założenie, że każdy iloczyn $\underline{v}\circ\underline{y}$ nieznanego wektora $\underline{v}$, którego istnienie wynika z założenia rozdzielności liniowej zbiorów X_1 i X_2, z każdym obiektem ze zbioru Y jest większy od zera. Zaś w tej modyfikacji progiem tym jest nie zero lecz liczba $\varepsilon\circ|\underline{w}|$. Inaczej mówiąc, przy założeniu tych samych odległości punktów zbioru X od hiperpłaszczyzny rozdzielającej dla oryginalnej wersji algorytmu, jak dla odległości tych punktów od najbliższej z dwóch hiperpłaszczyzn rozdzielających w wersji zmodyfikowanej, liczby b w relacji (3.9) oraz (3.15) byłyby rzeczywiście takie same. Wartość prawej strony relacji (3.15) byłaby, przy w/w założeniu mniejsza niż w przypadku relacji (3.9), ponieważ mianownik jest większy o $\varepsilon\circ|\underline{w}|$, co oznacza mniejszą liczbę korekcji algorytmu. To spostrzeżenie jest intuicyjnie zrozumiałe, gdyż większy prześwit pomiędzy zbiorami X_1 i X_2 implikuje szybsze wyznaczenie rozwiązania.

Niech $\underline{v}_0$ będzie rozwiązaniem układu (3.4) bądź (3.12), zależnie od tego, która z wersji algorytmu zostanie zastosowana, oryginalna, czy zmodyfikowana. Stosując wersję zmodyfikowaną nie ma gwarancji, że najmniejszy z iloczynów skalarnych $\underline{v}_0\circ\underline{y}$, gdzie $\underline{y}\in Y$ będzie większy niż dla wersji oryginalnej, ale przy założeniu liniowej rozdzielności zbiorów z prześwitem $ps=2\circ\varepsilon$ jest gwarancja, że będzie on większy niż $\varepsilon\circ|\underline{w}|$.

Po uzyskaniu rozwiązania 3.12, klasyfikator dwu-decyzyjny zdefiniowany zostanie ostatecznie przez hiperpłaszczyznę $g_C(\underline{x})=[g_A(\underline{x},\varepsilon)+g_B(\underline{x},\varepsilon)]/2=0$, zilustrowaną na Rys. 3.2a. Nie ma jednak przeszkód, by korzystać z obu hiperpłaszczyzn rozdzielających. Obiekty $\underline{x}$ dla których $g_A(\underline{x},\varepsilon)>0$ byłyby kwalifikowane do klasy 1, a obiekty dla których $g_B(\underline{x},\varepsilon)<0$ do klasy 2, a pozostałym znajdującym się pomiędzy hiperpłaszczyznami $g_A(\underline{x},\varepsilon)=0$ oraz $g_B(\underline{x},\varepsilon)=0$, czyli spełniającym jednocześnie warunki

$g_A(\underline{x},\varepsilon)\leq 0$ oraz $g_B(\underline{x},\varepsilon)\geq 0$ można by przypisać decyzję *nie wiem* lub decyzję rozmytą odpowiadającą udziałowi obiektów z obu klas w pasie zawartym pomiędzy hiperpłaszczyznami $g_A(\underline{x},\varepsilon)=0$ oraz $g_B(\underline{x},\varepsilon)=0$.

Podobnie, można rozważać rozdzielność liniową zbiorów z nakładką $2\circ\varepsilon$, jak to pokazane zostało na Rys. 3.2b. Tym razem funkcje $g_A(\underline{x},\varepsilon)$ oraz $g_B(\underline{x})$ powinny spełniać następujące układy nierówności:

$$\begin{gathered} g_A(\underline{x},\varepsilon)=\underline{w}\circ\underline{x}+w_{n+1}+\varepsilon\circ|\underline{w}|>0, \text{ gdy } \underline{x}\in X_1 \\ \text{oraz} \\ g_B(\underline{x},\varepsilon)=\underline{w}\circ\underline{x}+w_{n+1}-\varepsilon\circ|\underline{w}|<0, \text{ gdy } \underline{x}\in X_2. \end{gathered} \tag{3.16}$$

Wykorzystując zbiór Y, związek wektora $\underline{v}$ z wektorem $\underline{w}$, relację (3.16) da się przekształcić do bardziej zwartej formy:

$$\underline{v}\circ\underline{y}+\varepsilon\circ|\underline{w}|>0, \text{ gdy } \underline{y}\in Y. \tag{3.17}$$

Modyfikację algorytmu korekcji błędów, odnoszącą się do liniowej rozdzielności zbiorów X_1 i X_2 z nakładką $2\circ\varepsilon$, można zapisać następująco:

Modyfikacja II

$$\begin{gathered} \underline{v}_0=[0_1,0_2,\ldots,0_{n+1}], \text{ czyli jest to } (n+1) \text{ wymiarowy wektor zerowy;} \\ \underline{v}_k=\underline{v}_{k-1}, \text{ gdy } \underline{v}_{k-1}\circ\underline{y}_k+\varepsilon\circ|\underline{w}_{k-1}|>0; \\ \underline{v}_k=\underline{v}_{k-1}+a\circ\underline{y}_k, \text{ gdy } \underline{v}_{k-1}\circ\underline{y}_k+\varepsilon\circ|\underline{w}_{k-1}|\leq 0. \end{gathered} \tag{3.18}$$

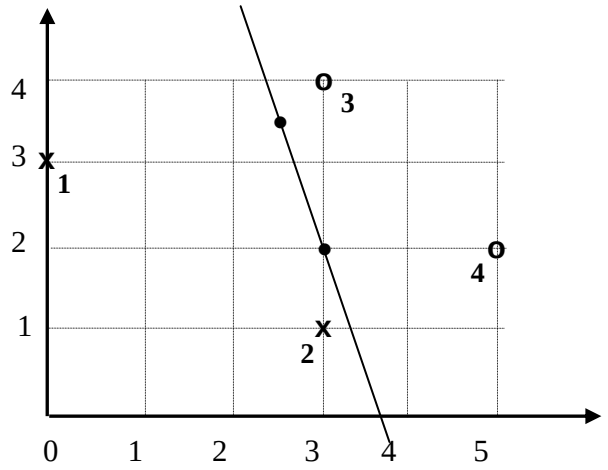
Tym razem, twierdzenie dotyczące zbieżności algorytmu mogłoby brzmieć: jeżeli zbiory X_1 i X_2 są liniowo rozdzielne z nakładką $2\circ\varepsilon$, to algorytm działający wg II modyfikacji znajdzie wektor $\underline{v}$ spełniający układ nierówności (3.17) w skończonej liczbie kroków. Jego składowe jednoznacznie określałyby hiperpłaszczyzny $g_A(\underline{x},\varepsilon)=0$ oraz $g_B(\underline{x},\varepsilon)=0$ czyniące zadość warunkom (3.16).

Niestety, stwierdzenie to na chwilę obecną pozostaje tylko hipotezą. Jednak nie ma przeszkód by tej modyfikacji nie stosować praktycznie. Iteracje tego algorytmu są bardzo proste, więc ograniczenia na ich liczbę mogą wyrażać się bardzo dużymi liczbami, rzędu kilkuset tysięcy i więcej, a w przypadku małych zbiorów nawet rzędu kilku milionów. Podobnie, jak to zostało zaproponowane odnośnie oryginalnej wersji algorytmu, po każdej korekcji można zapamiętywać najlepszy wynik, mierzony liczbą wektorów $\underline{y}$ spełniających układ (3.17).

Przykład

Poniżej zostanie zilustrowane działanie algorytmu korekcji błędów, w jego oryginalnej wersji. Niech $X_1=\{\underline{x}_1,\underline{x}_2\}$ i $X_2=\{\underline{x}_3,\underline{x}_4\}$, gdzie $\underline{x}_1=[1,3]$, $\underline{x}_2=[3,1]$, $\underline{x}_3=[3,4]$, $\underline{x}_4=[5,2]$. Zbiory te pokazane są na Rys. 3.3.

Po zastosowaniu przekształcenia (3.3) ze zbioru $X=X_1 \cup X_2$ otrzymamy następujący zbiór $Y=\{\underline{y}_1=[1,3,1], \underline{y}_2=[3,1,1], \underline{y}_3=[-3,-4,-1], \underline{y}_4=[-5,-2,-1]\}$. Na podstawie tego zbioru należy teraz utworzyć nieskończony ciąg, który spełnia założenie algorytmu, że każdy element tego zbioru występuje w ciągu nieskończoną liczbę razy.



Rys. 3.3. Zbiór danych dla ilustracji działania algorytmu korekcji błędów

Ciąg ten może mieć postać: $(\underline{y}_1,\underline{y}_2,\underline{y}_3,\underline{y}_4,\underline{y}_1,\underline{y}_2,\underline{y}_3,\underline{y}_4,\ldots)$, tzn. $\underline{y}_{k\circ 4+i}=\underline{y}_i$, gdzie k jest dowolną nieujemną liczbą całkowitą, a i numerem elementu ciągu. Przebieg algorytmu korekcji błędów dla współczynnika korekcji $a=1$ pokazuje Tab. 3.1.

Pokazany został cały przebieg algorytmu, pomimo dużej liczby kroków iteracji, po to by móc przekonać się, że nie charakteryzuje się on stopniowym zbliżaniem się do rozwiązania, czyli systematycznym wzrostem wartości iloczynów $\underline{v}_{k-1}\circ\underline{y}_k$ po wykonaniu kolejnej korekcji. W kolumnie oznaczonej jako liczba porządkowa, normalną czcionką podany został numer kolejnej korekcji, zaś czcionką pogrubioną numer kolejnej prezentacji obiektu ze zbioru Y. Rozwiązanie było osiągnięte po 44 korekcjach, czyli po 65 prezentacjach obiektów. Prezentacje od numerów 66 do 69 były konieczne dla weryfikacji, że uzyskany wektor $\underline{v}=\underline{v}_{69}$ jest rozwiązaniem układu (3.4).

Organizacja Tab. 3.1 jest następująca. Start algorytmu następuje począwszy od $k=1$. Wektor $\underline{v}_{k-1}$ z wiersza k-1 mnożony jest przez wektor $\underline{y}_k$ z wiersza k. Jeżeli iloczyn ten nie jest dodatni, to w wierszu k-tym tworzony jest nowy wektor $\underline{v}_k$. Jeśli natomiast $\underline{v}_{k-1}\circ\underline{y}_k$ jest dodatni, to wektor $\underline{v}$ nie ulega zmianie, czyli $\underline{v}_k=\underline{v}_{k-1}$. W ten sposób na podstawie ciągu $(\underline{y}_k)_{k=1}^{\infty}$ generowane są kolejne elementy ciągu $(\underline{v}_k)_{k=1}^{\infty}$.

Począwszy od kroku 66 cztery kolejne iloczyny $\underline{v}_{k-1}\circ\underline{y}_k$ są dodatnie, co oznacza, że uzyskany został wektor $\underline{v}=\underline{v}_{66}$, który z każdym elementem zbioru Y daje dodatni iloczyn skalarny. Wobec tego kontynuacja algorytmu już nie ma sensu, gdyż wektor $\underline{v}_k$ nie będzie już ulegał zmianie. Otrzymane zostało rozwiązanie układu nierówności (3.4),

którym jest wektor $\underline{v}$=[-3, -1, 11], a więc i funkcja dyskryminacyjna $g(\underline{x})=-3\circ x_1-x_2+11$ spełniająca warunki (3.1).

Tab. 3.1. Przebieg algorytmu korekcji błędów dla zbioru z Rys. 3.3.

L.p.	Wektor $\underline{y}$			Iloczyn	Wektor $\underline{v}$		
					v_1	v_2	v_3
0	y_1	y_2	y_3	$\underline{v}_{k-1}\circ\underline{y}_k$	0	0	0
1	1	3	1	0	1	3	1
2	-3	-4	-1	-16	-2	-1	0
3	1	3	1	-5	-1	2	1
4	3	1	1	0	2	3	2
5	-3	-4	-1	-20	-1	-1	1
6	1	3	1	-3	0	2	2
7	-3	-4	-1	-10	-3	-2	1
8	1	3	1	-8	-2	1	2
9	3	1	1	-3	1	2	3
10	-3	-4	-1	-14	-2	-2	2
11	1	3	1	-6	-1	1	3
12	-3	-4	-1	-4	-4	-3	2
13	-5	-2	-1	24	-4	-3	2
14	1	3	1	-11	-3	0	3
15	3	1	1	-6	0	1	4
16	-3	-4	-1	-8	-3	-3	3
17	1	3	1	-9	-2	0	4
18	3	1	1	-2	1	1	5
19	-3	-4	-1	-12	-2	-3	4
20	1	3	1	-7	-1	0	5
21	-3	-4	-1	-2	-4	-4	4
22	1	3	1	-12	-3	-1	5
23	3	1	1	-5	0	0	6
24	-3	-4	-1	-6	-3	-4	5

L.p.	Wektor $\underline{y}$			Iloczyn	Wektor $\underline{v}$		
	y_1	y_2	y_3	$\underline{v}_{k-1}\circ\underline{y}_k$	v_1	v_2	V_3
25	1	3	1	-10	-2	-1	6
26	3	1	1	-1	1	0	7
27	-3	-4	-1	-10	-2	-4	6
28	1	3	1	-8	-1	-1	7
29	-3	-4	-1	0	-4	-5	6
30	1	3	1	-13	-3	-2	7
31	3	1	1	-4	0	-1	8
32	-3	-4	-1	-4	-3	-5	7
33	1	3	1	-11	-2	-2	8
34	3	1	1	0	1	-1	9
35	-3	-4	-1	-8	-2	-5	8
36	1	3	1	-9	-1	-2	9
37	-5	-2	-1	0	-6	-4	8
38	1	3	1	-10	-5	-1	9
39	3	1	1	-7	-2	0	10
40	-3	-4	-1	-4	-5	-4	9
41	1	3	1	-8	-4	-1	10
42	3	1	1	-3	-1	0	11
43	-3	-4	-1	-8	-4	-4	10
44	1	3	1	-6	-3	-1	11
66	3	1	1	**1**	-3	-1	11
67	-3	-4	-1	**2**	-3	-1	11
68	-5	-2	-1	**6**	-3	-1	11
69	1	3	1	**5**	-3	-1	11

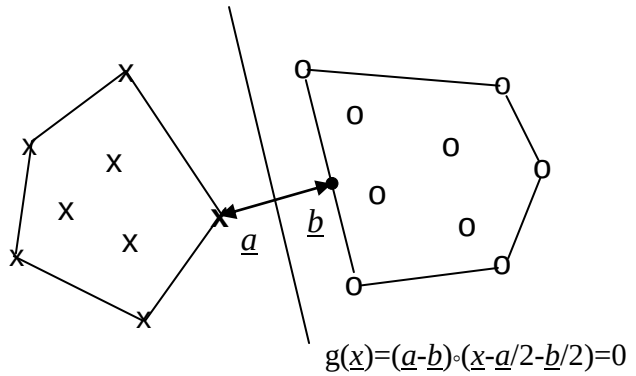
Dla obiektów $\underline{x}_1$ i $\underline{x}_2$ ze zbioru X_1 przyjmuje ona wartości $g(\underline{x}_1)=5$ i $g(\underline{x}_2)=1$, a dla obiektów $\underline{x}_3$ i $\underline{x}_4$ wartości $g(\underline{x}_3)=-2$ i $g(\underline{x}_4)=-6$. Prosta $g(\underline{x})=-3\circ x_1-x_2+11=0$ przechodzi przez punkty [3;2] oraz [2,5;3,5], które zostały zaznaczone na Rys. 3.3 i leżą w polu tego rysunku. Stąd łatwo można wykreślić hiperpłaszczyznę rozdzielającą, w tym przypadku linię prostą.

W pokazanym przykładzie, w ciągu $(\underline{y}_k)_{k=1}^{\infty}$ występowały najpierw wektory odpowiadające obiektom z klasy 1, a potem z klasy 2. Naprzemienna prezentacja wektorów $\underline{y}_k$ z różnych klas w tym przykładzie nie spowodowała zmniejszenia liczby kroków iteracji.

3.2. Iteracyjny algorytm wyznaczania hiperpłaszczyzny optymalnej

Mankamentem algorytmu korekcji błędów jest brak możliwości rozstrzygnięcia, czy analizowane zbiory X_1 i X_2 są liniowo rozdzielne. Ponadto, hiperpłaszczyzna rozdzielająca, która z jego użyciem zostanie ewentualnie wyznaczona, może być daleka od optymalnej i wówczas należy wielokrotnie eksperymentować stosując modyfikację I, zaproponowaną w poprzednim podrozdziale. Jak już zostało to zaznaczone w/w podrozdziale (3.1), wskazane jest, aby hiperpłaszczyzna rozdzielająca była możliwie najbardziej odległa od najbliższego obiektu ze zbioru $X=X_1\cup X_2$.

Przedstawiony teraz zostanie algorytm, który znajduje najbliższą sobie parę punktów $\underline{a}$ oraz $\underline{b}$, przy czym $\underline{a}\in\mathrm{Co}(X_1)$ i $\underline{b}\in\mathrm{Co}(X_2)$, gdzie symbol $\mathrm{Co}(X_i)$, i=1,2, oznacza powłokę wypukłą zbioru X_i. Na podstawie takiej pary jest już łatwo wyznaczyć równanie optymalnej hiperpłaszczyzny rozdzielającej. Idea tego algorytmu przedstawiona została na Rys. 3.4.



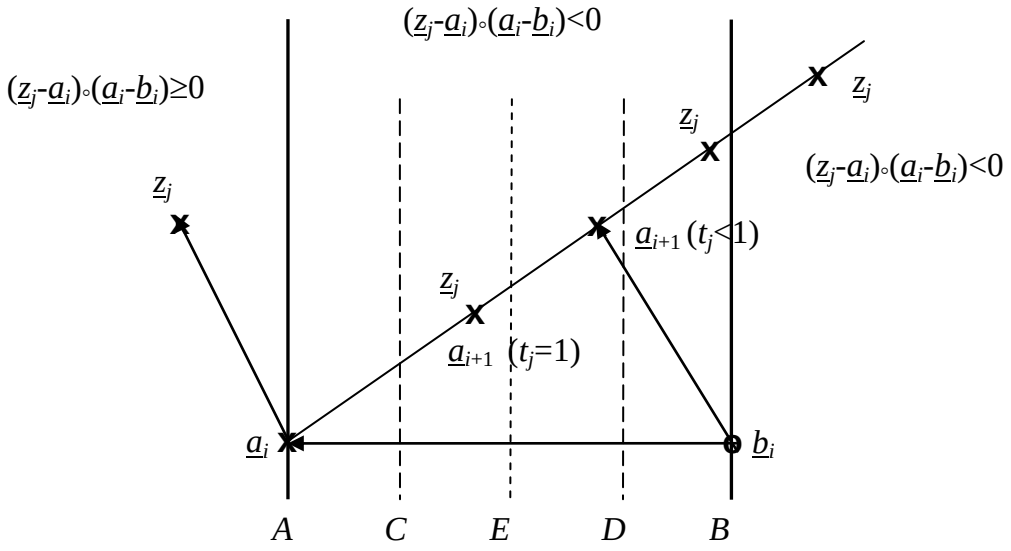
Rys. 3.4. Idea iteracyjnego algorytmu wyznaczania hiperpłaszczyzny rozdzielającej

Dla wygody, pod symbolami wektorów będą również rozumiane punkty w przestrzeni cech o tych samych współrzędnych, co składowe wektorów.

Podobnie jak w algorytmie korekcji błędów, z obiektów $\underline{x}_j$ należących do zbioru $X=X_1\cup X_2=\{\underline{x}_j\}_{j=1}^{m}$, należy utworzyć nieskończony ciąg $(\underline{z}_j)_{j=1}^{\infty}$ wektorów, tym razem n wymiarowych, taki, że każdy obiekt $\underline{x}$ zbioru X występuje w nim nieskończoną liczbę razy oraz wektory $\underline{z}_1$ i $\underline{z}_2$ pochodzą z różnych klas. Najwygodniej jest przyjąć, że $\underline{z}_j=\underline{x}_j$, gdy $j\le m$, $\underline{z}_{j-m}=\underline{z}_j$, gdy $j>m$, $\underline{z}_1\in X_1$ oraz $\underline{z}_2\in Z_2$, tzn. ciąg ten jest następujący: $(\underline{z}_j)_{j=1}^{\infty}=(\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_m,\underline{x}_{m+1},\underline{x}_{m+2},\ldots,\underline{x}_{2\cdot m},\ldots)$, przy czym $\underline{x}_1\in X_1$ oraz $\underline{x}_2\in X_2$.

Jako pierwsze przybliżenie szukanych wektorów $\underline{a}$ i $\underline{b}$ przyjęte zostaną wektory $\underline{z}_1$ oraz $\underline{z}_2$, tzn. $\underline{a}_1=\underline{z}_1$ i $\underline{b}_2=\underline{z}_2$. Wystarczy teraz pokazać, jak znając i-te przybliżenie w/w wektorów wyznaczyć kolejne, z indeksem i+1. A zatem, niech para $\underline{a}_i$ oraz $\underline{b}_i$ stanowi

i-te przybliżenie szukanych wektorów $\underline{a}$ i $\underline{b}$, a prezentowany wektor $\underline{z}_j$ niech pochodzi z klasy 1. Może on zajmować różne położenia, jak to zostało pokazane na Rys. 3.5.



Rys. 3.5. Ilustracja idei algorytmu iteracyjnego

Sytuacji, w której pochodziłby on z przeciwnej klasy nie warto analizować, gdyż rozważania byłyby symetryczne, tzn. miejsce wektora $\underline{a}_i$ zajęłby wektor $\underline{b}_i$ i odwrotnie.

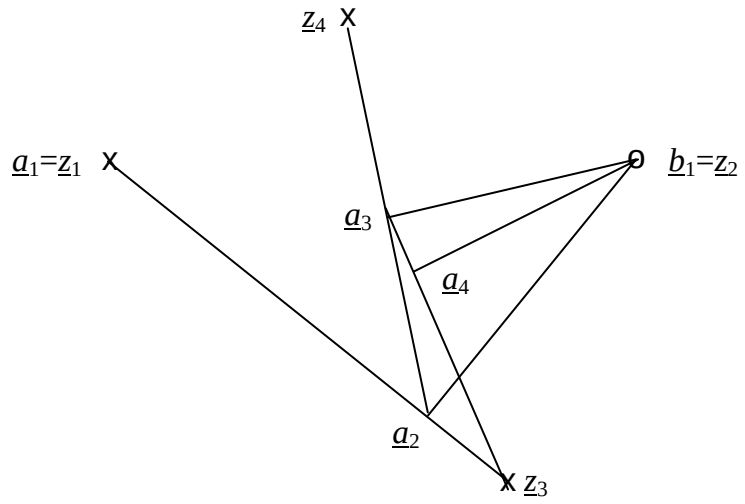
Wektory $\underline{a}_i$ oraz $\underline{b}_i$ wyznaczają parę równoległych hiperpłaszczyzn A i B ortogonalnych do wektora $(\underline{a}_i-\underline{b}_i)$ i przechodzących odpowiednio przez punkty $\underline{a}$ i $\underline{b}$. Prezentowany wektor $\underline{z}_j$ może leżeć na lewo od hiperpłaszczyzny A lub na niej, czyli po stronie wskazanej przez wektor $(\underline{a}_i-\underline{b}_i)$, pomiędzy hiperpłaszczyznami A i B oraz na prawo od hiperpłaszczyzny B lub na niej, tj. po przeciwnej stronie niż ta, jaką wskazuje wektor $(\underline{a}_i-\underline{b}_i)$. Poszukiwane są zaś punkty $\underline{a}$ oraz $\underline{b}$, które wyznaczałyby taką parę hiperpłaszczyzn A i B, aby wewnątrz pasa pomiędzy tymi hiperpłaszczyznami nie znajdował się żaden z wektorów ciągu $(\underline{z}_j)_{j=1}^{\infty}$, czyli żaden obiekt zbioru $X=X_1\cup X_2$.

Punkt $\underline{z}_j$ pochodzi z klasy 1, jak zostało przyjęte, więc powinien leżeć z lewej strony hiperpłaszczyzny A, a ściślej z przeciwnej strony niż hiperpłaszczyzna B. Jeśli tak nie jest, czyli znajduje się on na prawo od hiperpłaszczyzny A, w pasie pomiędzy hiperpłaszczyznami A i B, na hiperpłaszczyźnie B lub na prawo od niej, to para wektorów $\underline{a}_i$ oraz $\underline{b}_i$ wymaga uaktualnienia, czyli wyznaczenia nowej pary wektorów, czyli $\underline{a}_{i+1}$ oraz $\underline{b}_{i+1}$. Przez określenie *na prawo* od hiperpłaszczyzny B należy rozumieć przeciwną stronę w stosunku do hiperpłaszczyzny A. Rozpoznania, po której stronie hiperpłaszczyzny *A* znajduje się prezentowany aktualnie punkt $\underline{z}_j$, można dokonać badając wartość iloczynu skalarnego $(\underline{z}_j-\underline{a}_i)\circ(\underline{a}_i-\underline{b}_i)$. Jeśli jest on dodatni lub równy zeru,

to nie ma potrzeby korekcji, czyli $\underline{a}_{i+1}=\underline{a}_i$. Gdy zaś jest on ujemny, to $\underline{a}_{i+1}$ ulega zmianie.

Odwołując się do Rys. 3.5 łatwo zauważyć, że cały odcinek, łączący punkty odpowiadające wektorom $\underline{z}_j$ i $\underline{a}_i$ leży wewnątrz uwypuklenia $\mathrm{Co}(X_1)$. Na tym odcinku należy znaleźć punkt $\underline{a}_{i+1}$, który będzie najbliższy punktowi $\underline{b}_i$. Jeżeli rzut punktu $\underline{b}_i$ na prostą przechodzącą przez punkty $\underline{z}_j$ i $\underline{a}_i$ znajdzie się poza odcinkiem łączącym te punkty, to $\underline{a}_{i+1}=\underline{z}_j$, bo rzeczywiście, wtedy punkt $\underline{z}_j$ jest z całego w/w odcinka jest najbliższy punktowi $\underline{b}_i$. Jeżeli zaś rzut punktu $\underline{b}_i$ znajdzie się na odcinku łączącym $\underline{z}_j$ z $\underline{a}_i$, to najbliższym punktem dla $\underline{b}_i$ na tym odcinku będzie właśnie ten rzut, to oznacza, że on będzie wektorem $\underline{a}_{i+1}$. Tak więc, obie w/w sytuacje można objąć jedną relacją: $(\underline{a}_{i+1}-\underline{a}_i)=t_j\circ(\underline{z}_j-\underline{a}_i)$, gdzie $t_j\in(0,1]$, tzn., że $\underline{a}_{i+1}=\underline{a}_i+t_j\circ(\underline{z}_j-\underline{a}_i)$. Przypadek, gdy $t_j=0$, nie wymaga korekcji, bo wtedy wektor $\underline{z}_j$ znajdowałby się na hiperpłaszczyźnie A. Gdy $t_j=1$, to $\underline{a}_{i+1}=\underline{z}_j$, a jeśli $0<t_j<1$, to $\underline{a}_{i+1}$ jest rzutem $\underline{b}_i$ znajdującym się odcinku, który łączy punkt $\underline{z}_j$ z punktem $\underline{a}_i$. Żeby obliczyć $\underline{a}_{i+1}$, należy zatem wyznaczyć ułamek t_j.

Wektory $(\underline{z}_j-\underline{a}_i)$ oraz $(\underline{a}_{i+1}-\underline{b}_i)$ są ortogonalne, zatem $(\underline{z}_j-\underline{a}_i)\circ(\underline{a}_{i+1}-\underline{b}_i)=0$, a uwzględnienie, że $\underline{a}_{i+1}=\underline{a}_i+t_j\circ(\underline{z}_j-\underline{a}_i)$ da w wyniku równanie $(\underline{z}_j-\underline{a}_i)\circ[(\underline{a}_i+t_j\circ(\underline{z}_j-\underline{a}_i)-\underline{b}_i]=0$, skąd $t_j=(\underline{z}_j-\underline{a}_i)\circ(\underline{b}_i-\underline{a}_i)/(\underline{z}_j-\underline{a}_i)^2$. Algorytm działający wg przedstawionej wyżej zasady kolejnych przybliżeń wektorów $\underline{a}$ i $\underline{b}$ może wymagać nieskończonej liczby kroków iteracji, co ilustruje przykład pokazany na Rys. 3.6.



Rys. 3.6. Przykład ilustrujący konieczność wykonania nieskończonej liczby kroków

Zbiór X_1 zawiera cztery obiekty, czyli $X_1=\{\underline{z}_1,\underline{z}_3,\underline{z}_4\}$, a zbiór X_2 tylko jeden obiekt, tzn. $X_2=\{\underline{z}_2\}$. Ciąg $(\underline{z}_j)_{j=1}^{\infty}$ ma następującą postać: $(\underline{z}_1,\underline{z}_2,\underline{z}_3,\underline{z}_4,\underline{z}_1,\underline{z}_2,\underline{z}_3,\underline{z}_4,\ldots)$. W pierwszej iteracji $\underline{a}_1=\underline{z}_1$ oraz $\underline{b}_1=\underline{z}_2$. Punkt $\underline{a}_2$ jest rzutem punktu $\underline{b}_1$ na odcinek łączący punkt $\underline{a}_1$ z punktem $\underline{z}_3$. Z kolei punkt $\underline{a}_3$ jest rzutem punktu $\underline{b}_1$ na odcinek łączący punkt

$\underline{a}_2$ z punktem $\underline{z}_4$. Punkt $\underline{a}_4$ jest rzutem punktu $\underline{b}_1$ na odcinek, który łączy punkt $\underline{a}_3$ z punktem $\underline{z}_3$. Nie pokazany już na Rys. 3.6 punkt $\underline{a}_5$ byłby rzutem punktu $\underline{b}_1$ na odcinek łączący punkt $\underline{a}_4$ z punktem $\underline{z}_4$. Kontynuacja skutkowałaby korekcjami przybliżeń tylko punktu $\underline{a}$ i korekcje te byłyby wykonywane wyłącznie po prezentacji obiektów $\underline{z}_3$ i $\underline{z}_4$.

Jednym z naturalnych warunków zatrzymania algorytmu jest, gdy $d(\underline{a},\underline{b})=0$, gdzie $d(\circ,\circ)$ jest funkcją odległości, zwykle euklidesowej. Wtedy zbiory X_1 i X_2 uznane by były za liniowo nierozdzielne. Ze względów numerycznych warunek $d(\underline{a},\underline{b})=0$ należy zastąpić warunkiem $d(\underline{a},\underline{b})\leq\delta$, gdzie δ jest małą dodatnią liczbą rzeczywistą. Natomiast, w przypadku liniowej rozdzielności zbiorów warunek $(\underline{z}_j-\underline{a}_i)\circ(\underline{a}_i-\underline{b}_i)\geq 0$, dla wszystkich wektorów ze zbioru Z, może nie być spełniony w skończonym czasie, co zostało pokazane wyżej i zilustrowane na Rys. 3.6. Rozwiązaniem zagadnienia może być zastąpienie warunku $(\underline{z}_j-\underline{a}_i)\circ(\underline{a}_i-\underline{b}_i)\geq 0$ warunkiem $(\underline{z}_j-\underline{a}_i)\circ(\underline{a}_i-\underline{b}_i)\geq -\varepsilon\circ(\underline{a}_i-\underline{b}_i)\circ(\underline{a}_i-\underline{b}_i)$, gdzie liczba ε jest z zakresu $[0,1/2]$.

Oznacza to, że korekta przybliżenia punktu $\underline{a}$ następowałaby dopiero w przypadku, gdy dla $\underline{z}_j$ zajdzie warunek $(\underline{z}_j-\underline{a}_i)\circ(\underline{a}_i-\underline{b}_i)<-\varepsilon\circ(\underline{a}_i-\underline{b}_i)\circ(\underline{a}_i-\underline{b}_i)$, czyli gdy punkt $\underline{z}_j$ wtargnie w obszar na prawo od hiperpłaszczyzny A dostatecznie głęboko, głębiej niż $|\varepsilon\circ(\underline{a}_i-\underline{b}_i)|$. Koziniec [Koziniec, 1973] zaleca stosowanie parametru ε przyjmującego wartości od 0,1 do 0,25. W wymienionej pracy brak jest formalnego dowodu zbieżności algorytmu. Aczkolwiek zostało wyraźnie stwierdzone, że dla $\varepsilon>0$ i $\delta>0$ algorytm zatrzyma się po skończonej liczbie kroków korekcji przybliżeń wektorów $\underline{a}$ i $\underline{b}$. Zbieżność algorytmu jest oczywista, gdyż generowanemu ciągowi par wektorów $(\underline{a}_i,\underline{b}_i)$ odpowiada ciąg odległości $d(\underline{a}_i,\underline{b}_i)$, który jest nierosnący, nieskończony i ograniczony od dołu.

Interpretacja geometryczna w/w warunku $(\underline{z}_j-\underline{a}_i)\circ(\underline{a}_i-\underline{b}_i)<-\varepsilon\circ(\underline{a}_i-\underline{b}_i)\circ(\underline{a}_i-\underline{b}_i)$ będzie oczywista, jeśli warunek ten przedstawi się w innej formie: $|\underline{z}_j-\underline{a}_i|\circ|\underline{a}_i-\underline{b}_i|\circ\cos(\alpha)<-\varepsilon\circ|\underline{a}_i-\underline{b}_i|\circ|\underline{a}_i-\underline{b}_i|$, przy czym α jest kątem pomiędzy wektorami $(\underline{z}_j-\underline{a}_i)$ oraz $(\underline{a}_i-\underline{b}_i)$. Stąd po uproszczeniu wyżej wymieniony warunek przyjmie postać: $|\underline{z}_j-\underline{a}_i|\circ\cos(\alpha)<-\varepsilon\circ|\underline{a}_i-\underline{b}_i|$, tzn. rzut wektora $\underline{z}_j-\underline{a}_i$ na wektor $\underline{a}_i-\underline{b}_i$ jest fragmentem wektora $\underline{b}_i-\underline{a}_i$, maksymalnie jego połową, bo $0\leq\varepsilon\leq 1/2$ i $\cos(\alpha)$ jest ujemny, co można zauważyć analizując sytuację pokazaną na Rys. 3.5. Jeżeli $\varepsilon=1/2$, to nowy wektor $\underline{a}_{i+1}$, różny od poprzednika $\underline{a}_i$, będzie wyznaczany dopiero, gdy punkt $\underline{z}_j$, należący do X_1, znajdzie się na prawo od hiperpłaszczyzny E pokazanej na Rys. 3.5.

Podobne rozważania można by przeprowadzić dla sytuacji, gdy punkt $\underline{z}_j$, będzie ze zbioru X_2. Role punktów $\underline{a}_i$ oraz $\underline{b}_i$ uległyby wówczas zamianie. A nowy wektor $\underline{b}_{i+1}$, różny od swojego poprzednika $\underline{b}_i$, będzie wyznaczany dopiero, gdy punkt $\underline{z}_j$, należący do X_2 znajdzie się na lewo od hiperpłaszczyzny E pokazanej na Rys. 3.5. Dla $\varepsilon=1/4$ korekcja pary $(\underline{a}_i,\underline{b}_i)$ nastąpi dopiero, gdy punkt $\underline{z}_j$ będzie należał do X_1 i znajdzie się na prawo od hiperpłaszczyzny C lub, gdy będzie on ze zbioru X_2 i będzie leżał na lewo od hiperpłaszczyzny D.

W ogólnym przypadku, gdy zbiory X_1 i X_2 są liniowo rozdzielne, to omawiany algorytm działający z parametrem $\varepsilon>0$ znajdzie parę wektorów $\underline{a}$ i $\underline{b}$, a określona przez nie hiperpłaszczyzna o równaniu $(\underline{a}-\underline{b})\circ\underline{x}-0{,}5\circ(\underline{a}-\underline{b})\circ(\underline{a}+\underline{b})=0$ będzie rozdzielała zbiory X_1 i X_2 z prześwitem $ps=(1-2\circ\varepsilon)\circ|\underline{a}-\underline{b}|$. Dla sytuacji jak na Rys 3.5 i $\varepsilon=1/4$, gwarantowany prześwit byłby zawarty pomiędzy hiperpłaszczyznami C i D. Gdy $\varepsilon=0$, to prześwit byłby zawarty pomiędzy hiperpłaszczyznami A i B, a jeśli $\varepsilon=1/2$, to algorytm znajdzie hiperpłaszczyznę rozdzielającą, która nie gwarantuje żadnego prześwitu. Zaprezentowane wyżej wyjaśnienia zasady działania algorytmu powinny uczynić niżej podany formalny opis algorytmu zrozumiałym. Symbol „:=" oznacza, że wartość wyrażenia występującego po prawej stronie tego symbolu należy podstawić pod zmienną znajdującą się z jego lewej strony.

Definicja algorytmu iteracyjnego

1. $\underline{a}_1=\underline{z}_1$; $\underline{b}_1=\underline{z}_2$; i:=1; j:=2; lcz:=2; m:=5;
2. Jeżeli $d(\underline{a}_i,\underline{b}_i)\le\delta$, to {pisz: *Zbiory X_1 oraz X_2 nie są liniowo rozdzielne*, skocz do 7};
3. Jeżeli $j<m$, to j:=j+1 w przeciwnym przypadku j:=1; lcz:=lcz+1;
4. Jeżeli $\underline{z}_j\in X_1$ oraz $(\underline{a}_i-\underline{b}_i)\circ(\underline{z}_j-\underline{a}_i)<-\varepsilon\circ(\underline{a}_i-\underline{b}_i)^2$, to {$b_{i+1}$:=$b_i$; $t_j:=(\underline{z}_j-\underline{a}_i)\circ(\underline{b}_i-\underline{a}_i)/(\underline{z}_j-\underline{a}_i)^2$; jeżeli $t_j<1$, to $\underline{a}_{i+1}:=\underline{a}_i+t_j\circ(\underline{z}_j-\underline{a}_i)$; w przeciwnym przypadku $\underline{a}_{i+1}:=\underline{z}_j$; $i:=i+1$; $lcz:=0$; skocz do 2};
5. Jeżeli $\underline{z}_j\in X_2$ oraz $(\underline{b}_i-\underline{a}_i)\circ(\underline{z}_j-\underline{b}_i)<-\varepsilon\circ(\underline{b}_i-\underline{a}_i)^2$, to {$\underline{a}_{i+1}:=\underline{a}_i$; $t_j:=(\underline{z}_j-\underline{b}_i)\circ(\underline{a}_i-\underline{b}_i)/(\underline{z}_j-\underline{b}_i)^2$; jeżeli $t_j<1$, to $\underline{b}_{i+1}:=\underline{b}_i+t_j\circ(\underline{z}_j-\underline{b}_i)$; w przeciwnym przypadki $\underline{b}_{i+1}:=\underline{z}_j$; $i:=i+1$; $lcz:=0$; skocz do 2};
6. Jeżeli $lcz=m$, to {pisz: *Zbiory X i Y są liniowo rozdzielne, wektory $\underline{a}_i$ i $\underline{b}_i$ wyznaczają optymalną hiperpłaszczyznę rozdzielającą.*; skocz do 7}, w przeciwnym przypadku skocz do 3;
7. Koniec.

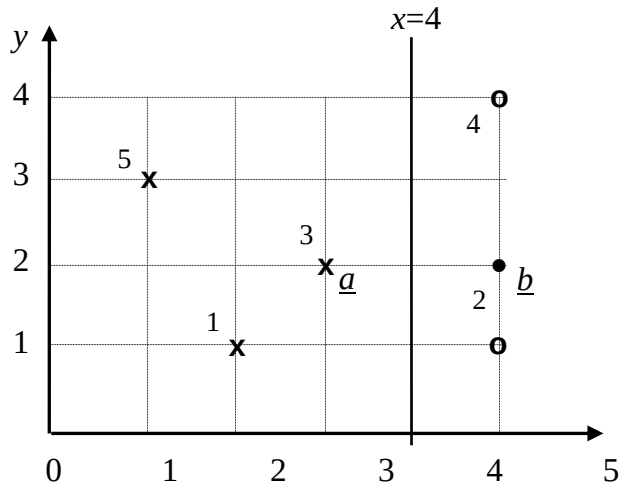
Przykład

Działanie algorytmu zostanie zilustrowane na prostym przykładzie złożonym z pięciu obiektów pokazanych na Rys. 3.7.

Rozwiązanie

$X_1=\{\underline{x}_1,\underline{x}_3,\underline{x}_5\}=\{[2,1],[3,2],[1,3]\}$. $X_2=\{\underline{x}_2,\underline{x}_4\}=\{[5,1],[5,4]\}$. Zbiór Z utworzony na podstawie zbiorów X_1 i X_2 zawiera 5 ponumerowanych elementów jak na Rys.3.7, Obiekty pochodzące ze zbioru X_1 mają indeksy nieparzyste, a indeksy ze zbioru X_2 parzyste, czyli $Z=\{\underline{z}_j\}_{j=1}^{5}=\{[2,1],[5,1],[3,2],[5,4],[1,3]\}$, przy czym $X_1=\{\underline{z}_1,\underline{z}_3,\underline{z}_5\}$ i $X_2=\{\underline{z}_2,\underline{z}_4\}$.

Nieskończony ciąg $(\underline{z}_j)_{j=1}^{\infty}$ utworzony ze zbioru Z ma postać: $\{\underline{z}_1,\underline{z}_2,\ldots,\underline{z}_5,\underline{z}_{5+1},\ldots,\underline{z}_{2\bullet 5},\ldots\}$, przy czym $z_{k\bullet 5+j}=\underline{z}_j$, dla dowolnej dodatniej liczby całkowitej i.



Rys. 3.7. Zbiory X i Y do przykładu liczbowego dla algorytmu iteracyjnego

Uporządkowanie zbioru Z nie musi być naprzemiennie, ważne jest tylko to, aby dwa pierwsze wyrazy tego ciągu zawierały obiekty pochodzące z różnych klas. Na podstawie ciągu $(\underline{z}_j)_{j=1}^{\infty}$ generowany jest ciąg $(\underline{a}_i,\underline{b}_i)_{i=1}^{\infty}$ przybliżeń pary wektorów $(\underline{a}_i,\underline{b}_i)$, przy czym nie ma odpowiedniości pomiędzy indeksami elementów $\underline{z}_j$ oraz $\underline{a}_i$.

Przebieg algorytmu dla w/w danych jest następujący:

1. $\underline{a}_1=\underline{z}_1=[2,1]$; $\underline{b}_1=\underline{z}2=[5,1]$; i:=1; j:=2, lcz=2; {Wskaźnik i jest numerem kolejnego przybliżenia szukanych wektorów $\underline{a}$ i $\underline{b}$, wskaźnik j jest numerem aktualnie podawanego obiektu, a lcz oznacza gwarantowaną liczbę obiektów znajdujących się po prawidłowych stronach względem pasa wyznaczonego przez hiperpłaszczyzny rozdzielające}
2. $d(\underline{a}_i,\underline{b}_i)=d(\underline{a}_1,\underline{b}_1)=d([2,1],[5,1])=3>0$; {funkcja $d(\underline{a}_i,\underline{b}_i)$ jest odległością miejską pomiędzy punktami $\underline{a}_i$ i $\underline{b}_i$}
3. j:=3, lcz:=3;
4. $\underline{z}_3=[3,2]\in X$, $(\underline{a}_1-\underline{b}_1)\circ(\underline{z}_3-\underline{a}_1)=([2,1]-[5,2])\circ(3,2]-[2,1])=[-3,0]\circ[1,1]=-3<0$;
   $\underline{b}_2=\underline{b}_1=[5,1]$;
   $t_j=t_3=(\underline{z}_3-\underline{a}_1)\circ(\underline{b}_1-\underline{a}_1)/(\underline{z}_3-\underline{a}_1)^2=([3,2]-[2,1])\circ([5,1]-[2,1])/([3,2]-2,1])^2=$
   $=([1,1][3,0]/[1,1]^2=3/2>1$; stąd wynika, że $\underline{a}_2=\underline{z}_3=[3,2]$;
   i:=i+1=2; lcz=0; skok do 2;
2. $d(\underline{a}_i,\underline{b}_i)=d(\underline{a}_2,\underline{b}_2)=d([3,2],[5,1])=\sqrt{5}>0$;
3. j:=j+1=4; lcz=0;

4. $\underline{z}_4=[5,4]\notin X_1$; {pozostałe instrukcje tego kroku nie są wykonywane}
5. $\underline{z}_4=[5,4]\in X_2$; $(\underline{b}_2-\underline{a}_2)\circ(\underline{z}_4-\underline{b}_2)=([5,1]-[3,2])\circ[5,4]-[5,1])=[2,-1]\circ[0,3]=-3<0$;
   $\underline{a}_3=\underline{a}_2=[3,2]$;
   $t_j=t_4=(\underline{z}_4-\underline{b}_2)\circ(\underline{a}_2-\underline{b}_2)/(\underline{z}_4-\underline{b}_2)^2=([5,4]-[5,1])\circ([3,2]-[5,1])/([5,4]-[5,1])^2=$
   $=([0,3]\circ[-2,1]/[0,3]^2=1/3<1$;
   $\underline{b}_3=\underline{b}_2+t_4\circ(\underline{z}_4-\underline{b}_2)=[5,1]+\frac{1}{3}\circ([5,4]-[5,1])=[5,1]+\frac{1}{3}\circ[0,3]=[5,2]$;
   $i:=i+1=3$; $lcz=0$; skok do 2;
2. $d(\underline{a}_i,\underline{b}_i)=d(\underline{a}_3,\underline{b}_3)=d([3,2],[5,2])=2>0$;
3. $j:=j+1=5$; $lcz:=1$;
4. $\underline{z}_5=[1,3]\in X_1$, $(\underline{a}_3-\underline{b}_3)\circ(\underline{z}_5-\underline{a}_3)=([3,2]-[5,2])\circ(1,3]-[3,2])=[-2,0]\circ[-2,1]=4>0$;
   {pozostałe instrukcje tego kroku nie są wykonywane}
5. $\underline{z}_5=[1,3]\notin X_2$; {pozostałe instrukcje tego kroku nie są wykonywane}
6. $lcz<m$; skok do 3;
3. $j:=1$; $lcz:=lcz+1=2$;
4. if $\underline{z}_1\in X_1$ and $(\underline{a}_3-\underline{b}_3)\circ(\underline{z}_1-\underline{a}_3)=([3,2]-[5,2])\circ([2,1]-[3,2])=[-2,0]\circ[-1,-1]=2>0$;
5. $\underline{z}_1=[2,1]\notin X_2$; {pozostałe instrukcje tego kroku nie są wykonywane}
6. $lcz<m$ {bo $lcz=1$, a $m=5$}; skok do 3;
3. $j:=j+1=2$; $lcz:=lcz+1=3$;
4. $\underline{z}_2\notin X_1$; {pozostałe instrukcje tego kroku nie są wykonywane}
5. if $\underline{z}_2\in X_2$; $(\underline{b}_3-\underline{a}_3)\circ(\underline{z}_2-\underline{b}_3)=([5,2]-[3,2])\circ([5,1]-[5,2])=[2,0]\circ[0,-1]=0$;
   {pozostałe instrukcje tego kroku nie są wykonywane}
6. $lcz<m$ {bo $lcz=3$, a $m=5$}; skok do 3;
3. $j:=j+1=3$; $lcz:=lcz+1=4$;
4. $\underline{z}_3=[3,2]\in X_1$; $(\underline{a}_3-\underline{b}_3)\circ(\underline{z}_3-\underline{a}_3)=([3,2]-[5,2])\circ([3,2]-[3,2])=[-2,0]\circ[0,0]=0$;
   {pozostałe instrukcje tego kroku nie są wykonywane}
5. $\underline{z}_3=[3,2]\notin X_2$; {pozostałe instrukcje tego kroku nie są wykonywane}
6. $lcz<m$ {bo $lcz=4$, a $m=5$}; skok do 3;
3. $j:=j+1=4$; $lcz:=lcz+1=5$;
4. $\underline{z}_4=[5,4]\notin X_1$; {pozostałe instrukcje tego kroku nie są wykonywane}
5. $\underline{z}_4=[5,4]\in X_2$; $(\underline{b}_3-\underline{a}_3)\circ(\underline{z}_4-\underline{b}_3)=([5,2]-[3,2])\circ([5,4]-[5,2])=[2,0]\circ[0,2]=0$;
   {pozostałe instrukcje tego kroku nie są wykonywane}
6. $lcz=m=5$, „Zbiory X_1 i X_{12} są liniowo rozdzielne”, wektory $\underline{a}=\underline{a}_3=[3,2]$ i $\underline{b}=\underline{b}_3=[5,2]$ wyznaczają hiperpłaszczyznę rozdzielającą, skok do 7;

7. Koniec.

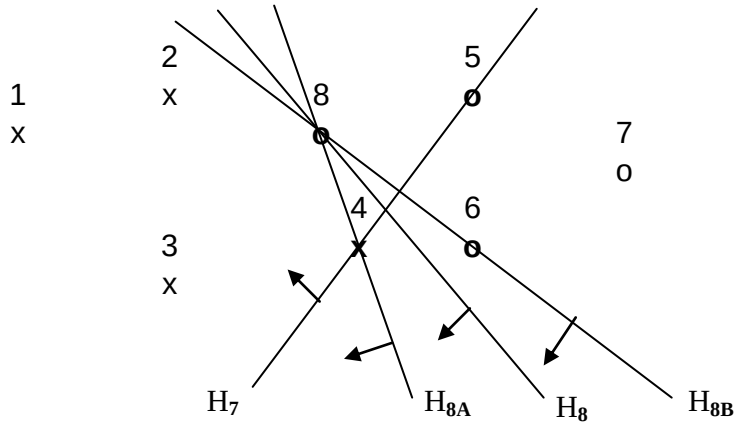
3.3.Rekursywny algorytm badania rozdzielności liniowej

Przedmiotem rozważań niniejszego podrozdziału będzie słaba rozdzielność liniowa zbiorów X_1 i X_2. Zamiast układu nierówności 3.1, tym razem wystarczy, że będzie spełniony układ:

$$g(\underline{x})\geq 0,\ \text{gdy}\ \underline{x}\in X_1$$
$$\text{oraz} \tag{3.19}$$
$$g(\underline{x})\geq 0,\ \text{gdy}\ \underline{x}\in X_2,$$

gdzie $g(\underline{x})=\sum_{j=1}^{n} w_j \circ x_j + w_{n+1}$ i $\underline{x}=[x_1,x_2,\ldots,x_n]$.

Idea proponowanego algorytmu została zilustrowana na Rys. 3.8.



Rys. 3.8. Idea działania algorytmu rekursywnego

Hiperpłaszczyzna H_7, na rysunku reprezentuje ją prosta, rozdziela prawidłowo punkty o numerach od 1 do 7. Strzałki wskazują dodatnie strony hiperpłaszczyzn. Krzyżykami, jak zwykle, zostały zaznaczone obiekty z klasy 1, a kółkami obiekty z klasy 2. Nie jest to rozdzielność ścisła, gdyż hiperpłaszczyzna H_7 przechodzi przez 2 punkty, jeden z klasy 1 i drugi z klasy 2. Kolejny prezentowany obiekt o numerze 8 znajduje się po niewłaściwej stronie. Można udowodnić, że przez punkt 8 da się przeprowadzić hiperpłaszczyznę H_8 rozdzielającą prawidłowo, w sensie rozdzielności słabej, wszystkie 8 punktów, jeśli tylko istnieje hiperpłaszczyzna prawidłowo je rozdzielająca. Takich hiperpłaszczyzn może być wiele. W sytuacji dwuwymiarowej jak na Rys. 3.8 będzie to cały pęk prostych zawartych pomiędzy prostymi H_{8A} i H_{8B}.

Podobne prawo jest słuszne w ogólnym n wymiarowym przypadku dla zbioru $X=X_1\cup X_2=\{\underline{x}_j\}_{j=1}^{m}$. Jeśli zbiory X_1 i X_2 są liniowo rozdzielne i pewna hiperpłaszczyzna

H_i rozdziela prawidłowo punkty ze zbioru $\{\underline{x}_j\}_{j=1}^{i}$, a punkt $\underline{x}_{i+1}$ leży po niewłaściwej stronie hiperpłaszczyzny H_i, to przez punkt $\underline{x}_{i+1}$ przechodzi hiperpłaszczyzna H_{i+1} poprawnie rozdzielająca wszystkie punkty zbioru $\{\underline{x}_j\}_{j=1}^{i+1}$. Prawo to zostanie w dalszej części pracy sformułowane w innej formie i dowiedzione. Pozwala ono obniżyć o jeden wymiar przestrzeni, w której szukany jest wektor wag, czyli współczynniki hiperpłaszczyzny H_{i+1}. Fakt, że hiperpłaszczyzna H_{i+1} będzie przechodzić przez punkt $\underline{x}_{i+1}$ pozwala utworzyć równanie, z którego można obliczyć jedną z n+1 wag i dzięki temu pozostanie do znalezienia n wag.

Dalsze rozważania odnoszące się do tego algorytmu wygodniej będzie prowadzić korzystając, jak w przypadku algorytmu korekcji błędów, z odwzorowania $h(\circ,\circ)$ określonego wzorami (3.3). Stosując odwzorowanie $h(X)$ zbioru $X=X_1\cup X_2$ w zbiór Y, układ (3.19) zostanie przekształcony do następującej postaci:

$$\underline{v}\circ\underline{y}\geq 0, \text{ gdy } \underline{y}\in Y, \tag{3.20}$$

gdzie $\underline{v}=[w_1,w_2,w_3,\ldots,w_n,w_{n+1}]$ oraz $Y=h(X)=\{y_j\}_{j=1}^{m}$.

Wektor $\underline{v}$, który spełnia układ nierówności (3.12) będzie określany jako rozwiązanie dla zbioru Y. Algorytm poszukiwania rozwiązania $\underline{v}$ układu (3.20) dla zbioru Y zostanie zdefiniowany rekursywnie. Jego podstawą jest następujące twierdzenie.

Twierdzenie 3.3

Niech $\underline{v}_i$ będzie rozwiązaniem układu postaci 3.20 dla zbioru $Y_i=\{\underline{y}_j\}_{j=1}^{i}\subset Y\subset E^{n+1}$ w pewnej (k+1) wymiarowej podprzestrzeni P^{k+1} przestrzeni cech E^{n+1} i niech nie będzie nim dla zbioru $Y_{i+1}=\{\underline{y}_j\}_{j=1}^{i+1}\subset Y$. Jeśli w podprzestrzeni P^{k+1} istnieje rozwiązanie $\underline{v}_0$ dla zbioru Y, to w podprzestrzeni $P^{k}=P^{k+1}\cap\{\underline{y}_{i+1}\}^{\perp}$ istnieje rozwiązanie $\underline{v}_{i+1}$ dla zbioru Y_{i+1}.

Dowód

Rozwiązania dla zbioru Y_{i+1} można poszukiwać wśród kombinacji wypukłych wektorów $\underline{v}_i$ i $\underline{v}_0$, tzn. $\underline{v}_{i+1}=t\circ\underline{v}_i+(1-t)\circ\underline{v}_0$. Ponieważ $\underline{v}_{i+1}\circ\underline{y}_{i+1}=0$, zatem po podstawieniu za $\underline{v}_{i+1}$ wyrażenia $t\circ\underline{v}_i+(1-t)\circ\underline{v}_0$ powstanie da równanie, z którego da się obliczyć t: $[t\circ\underline{v}_i+(1-t)\circ\underline{v}_0]\circ\underline{y}_{i+1}=0$, czyli $t\circ(\underline{v}_i\circ\underline{y}_{i+1}-\underline{v}_0\circ\underline{y}_{i+1})=-\underline{v}_0\circ\underline{y}_{i+1}$, stąd wynika, że $t=-\underline{v}_0\circ\underline{y}_{i+1}/(\underline{v}_i\circ\underline{y}_{i+1}-\underline{v}_0\circ\underline{y}_{i+1})$. Iloczyn $\underline{v}_0\circ\underline{y}_{i+1}$ jest nieujemny, zaś iloczyn $\underline{v}_i\circ\underline{y}_{i+1}$ ujemny. Zatem liczba t przyjmuje wartości z przedziału [0,1). Wartość 1 jest wykluczona, bo wówczas nie byłoby korekty wektora $\underline{v}_i$, gdyż zachodziłaby relacja $\underline{v}_{i+1}=\underline{v}_i$.

Łatwo sprawdzić, że dla każdej wartości t z przedziału [0,1) wektor $\underline{v}_{i+1}=t\circ\underline{v}_i+(1-t)\circ\underline{v}_0$ daje nieujemny iloczyn skalarny z każdym elementem zbioru Y_i, a więc i dla tej wyżej wyliczonej, ponieważ ona również mieści się w wymienionym przedziale.

Rzeczywiście: $\underline{v}_{i+1} \circ \underline{y}_j = [t \circ \underline{v}_i + (1\text{-}t) \circ \underline{v}_0] \circ \underline{y}_j = t \circ (\underline{v}_i \circ \underline{y}_j) + (1\text{-}t) \circ (\underline{v}_0 \circ \underline{y}_j) \geq 0$, dla $j = 1, 2, \ldots i$, ponieważ $\underline{v}_i \circ \underline{y}_j \geq 0$ oraz $\underline{v}_0 \circ \underline{y}_j \geq 0$, na mocy założeń twierdzenia. Zaś Natomiast dla wyliczonej wartości t iloczyn skalarny $\underline{v}_{i+1} \circ \underline{y}_{i+1} = 0$, ponieważ z tego właśnie równania współczynnik t został obliczony. Zerowa wartość tego iloczynu skalarnego oznacza, że wektor $\underline{v}_{i+1}$ należy do podprzestrzeni $P^k = P^{k+1} \cap \{\underline{y}_{i+1}\}^{\perp}$, co kończy dowód twierdzenia.

Zbiór Y dla którego poszukiwane jest rozwiązanie $\underline{v}$ powstał jako wynik przekształcenia $h(\circ,\circ)$ zbioru $X = X_1 \cup X_2$. Dla opisu algorytmu nie jest już konieczna informacja, że Y zawiera punkty, które powstały z przekształcenia obiektów, pochodzących ze zbiorów X_1 i X_2, o czym informują ostatnie składowe elementów zbioru Y , czyli składowe $\underline{y}_{n+1} = 1$, gdyż powstały one z przekształcenia obiektów ze zbioru X_1 i $\underline{y}_{n+1} = -1$, jeśli są one wynikiem odwzorowania obiektów ze zbioru X_2. Zatem proponowany algorytm będzie przeznaczony do ogólniej sformułowanego zadania, tj. do poszukiwania rozwiązania $\underline{v}$ dla pewnego zbioru $Z = \{\underline{z}_j\}_{j=1}^{m} \subset E^{n+1}$ w podprzestrzeni $P^{k+1} \subset E^{n+1}$, gdzie m jest liczebnością zbioru Z. Jeśli w podprzestrzeni P^{k+1} rozwiązanie dla zbioru Z istnieje, to algorytm powinien je móc wyznaczyć tak dla zbioru Z jak i dla dowolnego jego podzbioru. W dalszych rozważaniach występować będą również podzbiory zbioru Z oznaczane indeksami nie przewyższającymi liczby $n+1$. Z tego powodu wygodnie będzie przyjąć jeszcze jeden symbol na jego oznaczenie, tzn., że $Z_{k+1} = Z$.

Biorąc pod uwagę, że algorytm ma prowadzić ostatecznie do wyznaczenia rozwiązania dla zbioru Y, to wystarczy potem podstawić $Z = Y$ i $P^{n+1} = E^{n+1}$. Twierdzenie 3.3 jest prawdziwe również dla zbioru Z, którego punkty nie mają współrzędnych z_{n+1} równych +1 lub -1, jak to ma miejsce w przypadku zbioru Y.

Zanim podana zostanie definicja algorytmu, konieczne jest ustalenie znaczenia używanych symboli.

Symbolem $(\underline{v}, lg) := A(Z_{k+1}, P^{k+1})$ oznaczony zostanie algorytm znajdowania rozwiązania dla zbioru Z_{k+1} w podprzestrzeni P^{k+1}. Indeks dolny przy zbiorze Z_{k+1} nie jest konieczny, ale to podwójne zaznaczenie wymiarowości przestrzeni, w której poszukiwane jest rowiązanie poszukiwania rozwiązania znacznie ułatwi śledzenie algorytmu na ilustrujących jego działanie przykładach, podanych w dalszej części niniejszej monografii, po definicji algorytmu. Wynikiem algorytmu jest para złożona z wektora $\underline{v}$ oraz ze skalara lg przyjmującego wartość 0, jeśli w wyniku zastosowania algorytmu okaże się, że otrzymany wektor $\underline{v}$ jest rozwiązaniem dla zbioru Z_{k+1} w podprzestrzeni P^{k+1} albo skalara lg przyjmującego wartość -1, gdy okaże się, że uzyskany wektor $\underline{v}$ nie jest szukanym rozwiązaniem. Chcąc znaleźć rozwiązanie układu równań (3.20) należy wywołać zaproponowany algorytm z danymi: $Z_{n+1} = Y$ i $P^{n+1} = E^{n+1}$, czyli $(\underline{v}, lg) := A(Y, E^{n+1})$. Po wprowadzeniu w/w oznaczeń algorytm może już być zdefiniowany.

Analizę instrukcji algorytmu wygodniej będzie przeprowadzić śledząc jednocześnie odpowiednie kroki algorytmu w przytoczonych podanych dwóch przykładach ilustrujących jego działanie. Pierwszy z podanych przykładów zawiera dane w przestrzeni dwu-wymiarowej i odnosi się do pary zbiorów X_1 *i* X_2 liniowo rozdzielnych. Zbiór danych pokazany został na Rys. 3.9. Drugi przykład dotyczy zbiorów liniowo nierozdzielnych i został zilustrowany z wykorzystaniem danych w przestrzeni jednowymiarowej.

Definicja algorytmu rekursywnego

Wołaj $(\underline{v}, lg) := A(Z_{k+1}, P^{k+1})$;

k.1 $\underline{v}$:=dowolny niezerowy i ortogonalny rzut wektora ze zbioru Z_{k+1} na podprzestrzeń P^{k+1}, jeśli taki rzut istnieje;

k.2 Jeśli w Z_{k+1} nie istnieje żaden wektor, którego ortogonalny rzut na P^{k+1} byłby niezerowy, to przyjmij $lg := -1$ i skocz do k.12;

k.3 $DZ_{k+1} := \{\underline{z} \in Z_{k+1} : \underline{v} \circ \underline{z} \le 0\}$;

k.4 Jeśli DZ_{k+1} jest pusty, to podstaw $lg := 0$ i skocz do k.12;

k.5 jeśli k=0 i $DZ_{k+1} \neq \varnothing$, to $lg := -1$ i skocz do k.12;

k.6 $Z_k := Z_{k+1} - DZ_{k+1}$;

k.7 $J := n-k+1$; b_j:=dowolny wektor z DZ_{k+1};

k.8 Wołaj $(\underline{v}, lg) := A(Z_k, P^k)$, gdzie $P^k := P^{k+1} \cap \{\underline{b}_j\}^{\perp}$;

k.9 Jeśli $lg = -1$, to skocz do k.12;

k.10 $DZ_{k+1} := \{\underline{z} \in Z_{k+1} : \underline{v} \circ \underline{z} \le 0\}$;

k.11 Jeśli $DZ_{k+1} \neq \varnothing$, to skocz do k.6;

k.12 Jeśli $lg = -1$, to pisz: *Rozwiązanie nie istnieje, zbiory* X_1 *i* X_2 *nie są liniowo rozdzielne* w przeciwnym przypadku, gdy $lg = 0$ i $k = n$, to pisz: *Zbiory* X_1 i X_2 *są liniowo rozdzielne, a składowe wektora* $\underline{v}$ *są współczynnikami hiperpłaszczyzny rozdzielającej.*

Przykład 1 – zbiory liniowo rozdzielne

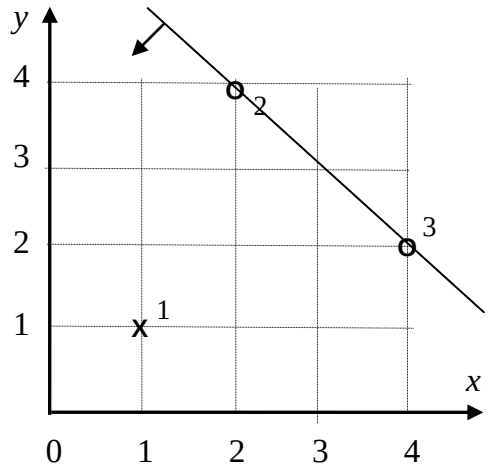
$X_1 = \{[1,1]\}$, $X_2 = \{[4,2],[2,4]\}$, $n=2$, $Y = \{\underline{y}_1 = [1,1,1], \underline{y}_2 = [-4,-2,-1], \underline{y}_3 = [-2,-4,-1]\}$.

Rozwiązanie: $k := n = 2$; $Z_{k+1} = Z_3 = Y$; $P^{k+1} = P^3 = E^3$;

Wołaj $(\underline{v}, lg) := A(Z_3, P^3)$;

2.01 $\underline{v} := \underline{y}_1 = [1,1,1]$; ponieważ $\underline{y}_1$ jest z Z_3 i jest już w P^3, to jest też i rzutem na podprzestrzeń P^3;

2.02 Warunek jest niespełniony;



Rys. 3.9. Ilustracja do przykładu zastosowania algorytmu rekursywnego

2.03 DZ_3:={$\underline{y}_2,\underline{y}_3$}; bo $\underline{v}\circ\underline{y}_1$=3, $\underline{v}\circ\underline{y}_2$=-7 i $\underline{v}\circ\underline{y}_3$=-7;

2.04 Warunek jest niespełniony;

2.05 Warunek jest niespełniony;

2.06 $Z_2=Z_3-DZ_3$={$\underline{y}_1$}; krok 2.05 jest nieaktywny

2.07 j:=n-k+1=1; $\underline{b}_1=\underline{y}_2$;

2.08 Wołaj ($\underline{v},lg$):=A(Z_2, P^2); $P^2= P^3\cap\{\underline{b}_1\}^\perp = \{\underline{b}_1\}^\perp = \{\underline{y}_2\}^\perp$;

1.01 $\underline{v}$:=y_1-($\underline{b}_1\circ\underline{y}_1$)/($\underline{b}_1\circ\underline{b}_1$)$\circ\underline{b}_1$=

=[1,1,1]-([-4,-2,-1]$\circ$[1,1,1])/([-4,-2,-1]^2)$\circ$[-4,-2,-1]=

=[-1,1,2]/3$\rightarrow$[-1,1,2]; {stały mnożnik można dla wygody opuścić}

1.02 Warunek jest niespełniony;

1.03 DZ_2:={pusty}; krok 1.1 jest nieaktywny;

1.04 Lg:=0; skocz do 1.12

1.12 Nic nie rób;

2.09 Warunek jest niespełniony;

2.10 DZ_3:={$\underline{y}_3$}; bo $\underline{v}\circ\underline{y}_1$=2, $\underline{v}\circ\underline{y}_2$=0 i $\underline{v}\circ\underline{y}_3$=-4;

2.11 Skocz do 2.06;

2.06 Z_2:=Z_3-DZ_3={$\underline{y}_1,\underline{y}_2$};

2.07 j:=n-k+1=1; $\underline{b}_1=\underline{y}_3$;

2.08 Wołaj ($\underline{v},lg$):=A(Z_2, P^2); P^2:= $P^3\cap\{\underline{b}_1\}^\perp = \{\underline{b}_1\}^\perp =\{\underline{y}_3\}^\perp$;

1.01. $\underline{v}$:=$\underline{y}_1$-($\underline{b}_1\circ\underline{y}_1$)/($\underline{b}_1\circ\underline{b}_1$)$\circ\underline{b}_1$=

=[1,1,1]-([-2,-4,-1]$\circ$[1,1,1])/([-2,-4,-1]^2)$\circ$[-2,-4,-1]=

=[1,-1,2]/3→[1,-1,2]; stały mnożnik można dla wygody opuścić

1.02 Warunek jest niespełniony;

1.03 DZ_2:={$\underline{y}_2$}, bo $\underline{v}\circ\underline{y}_1$= 2, $\underline{v}\circ\underline{y}_2$=-4 i $\underline{v}\circ\underline{y}_3$=0;

1.04 Warunek jest niespełniony;

1.05 Warunek jest niespełniony;

1.06 Z_1:=Z_2-DZ_2={$\underline{y}_1$};

1.07 j:=n-k+1=2; $\underline{b}_2$=$\underline{y}_2$;

1.08 Wołaj ($\underline{v}$, lg):=$A(Z_1,P^1)$;
P^1=$P^2\cap\{\underline{b}_2\}^\perp$=$P^2\cap\{\underline{y}_2\}^\perp$ = $\{\underline{y}_3\}^\perp\cap\{\underline{y}_2\}^\perp$ = $\{\underline{y}_3, \underline{y}_2\}^\perp$;

0.01 $\underline{b}_2$:=$\underline{b}_2$-($\underline{b}_1\circ\underline{b}_2$)/($\underline{b}_1\circ\underline{b}_1$)$\circ\underline{b}_1$=

=[-4,-2,-1]-([-2,-4,-1]$\circ$ [-4,-2,-1])/[-2,-4,-1]$^2\circ$ [-2,-4,-1]=

=[-4,-2,-1]-(17/21)*[-2,-4,-1]=[-50,26,-4]/21→[-50,26,-4];

$\underline{v}$:=$\underline{y}_1$-($\underline{b}_1\circ\underline{y}_1$)/($\underline{b}_1\circ\underline{b}_1$)$\circ\underline{b}_1$-($\underline{b}_2\circ\underline{y}_1$)/($\underline{b}_2\circ\underline{b}_2$)$\circ\underline{b}_2$=

=[1,1,1]+(1/3)$\circ$ [-2,-4,-1]-(-1/114)$\circ$ [-50,26,-4]=

=[-12,-12,72]/114→[-1,-1,6]; stały czynnik można opuścić

0.02 Warunek jest niespełniony;

0.03 DZ_0:=∅; bo $\underline{v}\circ\underline{y}_1$=4;

0.04 lg:=0; skocz do 0.12;

0.12 nic nie rób;

1.09 Warunek jest niespełniony;

1.10 DZ_2:=∅; bo $\underline{v}\circ\underline{y}_2$=0;

1.11 Warunek jest niespełniony;

1.12 Nic nie rób;

2.09 Warunek nie spełniony;

2.10 DZ_3=∅; bo $\underline{v}\circ\underline{y}_2$= 0 i $\underline{v}\circ\underline{y}_3$=0;

2.12 *Zbiory X_1 i X_2 są liniowo rozdzielne, a składowe wektora $\underline{v}$ są współczynnikami hiperpłaszczyzny rozdzielającej*, bo lg=0 oraz k=2=n; *wektor* [-1,-1,6] *wyznacza hiperpłaszczyznę rozdzielającą*.

Zaprezentowana została wersja algorytmu z zastosowaniem ortogonalizacji. Zamiast operacji kroku *k*.01 można zastosować metodę eliminacji zmiennych. Wektor $\underline{v}$ ma być ortogonalny jednocześnie do $\underline{b}_1$ i do $\underline{b}_2$ i jest on rzutem $\underline{y}_1$ na podprzestrzeń $\{\underline{y}_3,\underline{y}_2\}^\perp$. A zatem $\underline{v}$=$\underline{y}_1$+$\alpha\circ\underline{b}_1$+$\beta\circ\underline{b}_2$, a z układu równań: $\underline{v}\circ\underline{b}_1$=0, $\underline{v}\circ\underline{b}_2$=0, można

wyliczyć, że α=7/38 oraz β=7/38 i stąd $\underline{v}$=[1,1,1]+(7/38)$\circ$[-4,-2,-1]+(7/38)$\circ$[-2,-4,-1]=[-4,-4,24]/38=(2/19)$\circ$[-1,-1,6] $\rightarrow$[-1,-1,6].

Przykład 2 – zbiory liniowo nierozdzielne
X_1={[2],[5]} i X_2={[4]},
Y={[2,1],[5,1],[-4,-1]},
$\underline{y}_1$=[2,1], $\underline{y}_2$=[5,1] i $\underline{y}_3$=[-4,-1]. Podobnie, aby znaleźć rozwiązanie dla Y w E^1 należy podstawić $Z_2=Y$ oraz $P^2=E^2$ oraz wywołać $(\underline{v},lg):=\mathrm{A}(Z_{k+1},P^{k+1})$, przyjmując k=1 oraz n=1. Przebieg algorytmu będzie następujący:

Wołaj $(\underline{v},lg):=A(Z_2,P^2)$,
1.1 $\underline{v}:=\underline{y}_1$;
1.2 Warunek nie spełniony;
1.3 $DZ_2:=\{\underline{y}_3\}$;
1.4 Warunek nie spełniony;
1.5 Warunek nie spełniony;
1.6 $Z_1=\{\underline{y}_1,\underline{y}_2\}$;
1.7 j:=1; $\underline{b}_1:=\underline{y}_3$;
1.8 Wołaj $(\underline{v},lg):=\mathrm{A}(Z_1,\{b_1\}^{\perp})$;
0.1 $\underline{V}:=y_1-(\underline{b}_1\circ\underline{y}_1/\underline{b}_1\circ\underline{b}_1)\circ b_1$=[2,1]-(-9/17)$\circ$[-4,-1]=(2/17)$\circ$[-1,4];
0.2 Warunek nie spełniony;
0.3 $DZ_1:=\{\underline{y}_2\}$;
0.4 Nic nie rób;
0.5 Podstaw lg:= -1 i idź do 0.12;
0.12 Nic nie rób;
1.9 Idź do 1.12;
1.12 *Rozwiązanie nie istnieje, zbiory X_1 i X_2 nie są liniowo rozdzielne.*

3.4. Uzupełnienie algorytmu rekursywnego

Wynik zastosowania algorytmu rekursywnego nie jest jednoznaczny, gdyż jego przebieg zależy od tego jaki wektor zostanie wybrany jako startowy w kroku *k*.1 jego definicji. Stosując ten algorytm wielokrotnie, z różnym wektorem startowym, można otrzymać wiele rozwiązań układu nierówności (3.20), różnych od siebie bądź nie. Rozwiązania $\underline{v}_1$ i $\underline{v}_2$ są różne, jeśli nie zachodzi relacja $\underline{v}_1=\lambda\circ\underline{v}_2$, dla żadnej dodatniej liczby λ. W praktyce, otrzymanie kilku różnych rozwiązań może być korzystne. Wynika to z następującego twierdzenia.

Twierdzenie 3.4

Niech wektory $\underline{v}_i$, $i=1,2,\ldots,k$, będą rozwiązaniami układu (3.20), a Z_i niech będą podzbiorami zbioru Y, których elementy dają z wektorami $\underline{v}_i$ zerowe iloczyny skalarne, tzn. $Z_i=\{\underline{y}\in Y: \underline{v}_i\circ\underline{y}=0\}$. Wtedy dowolna dodatnia kombinacja $\underline{v}^*$ wektorów $\underline{v}_i$ jest rozwiązaniem układu (3.20) i daje zerowe iloczyny skalarne $\underline{v}_i\circ\underline{y}$ tylko z tymi elementami zbioru Y, które należą do zbioru $Z^*=Z_1\cap Z_2\cap\ldots\cap Z_k$.

Dowód

Skoro wektor $\underline{v}^*$ jest kombinacją dodatnią wektorów $\underline{v}_i$, to znaczy, że $\underline{v}^*=\sum_{i=1}^{k}\lambda_i\circ\underline{v}_i$. Po pomnożeniu obustronnym tej relacji przez wektor $\underline{y}$, który nie należy do zbioru Z^* powstanie relacja: $\underline{v}^*\circ\underline{y}=\sum_{i=1}^{k}\lambda_i\circ\underline{v}_i\circ\underline{y}$, w której przynajmniej jeden ze składników $\lambda_i\circ\underline{v}_i\circ\underline{y}$, jest dodatni, co wynika z definicji zbioru Z^*, a to oznacza, że $\underline{v}^*\circ\underline{y}>0$. Jeżeli zaś $\underline{v}$ jest elementem zbioru Z^*, to każdy ze składników $\lambda_i\circ\underline{v}_i\circ\underline{y}$ jest zerowy, a zatem i $\underline{v}^*\circ\underline{y}=0$, co kończy dowód.

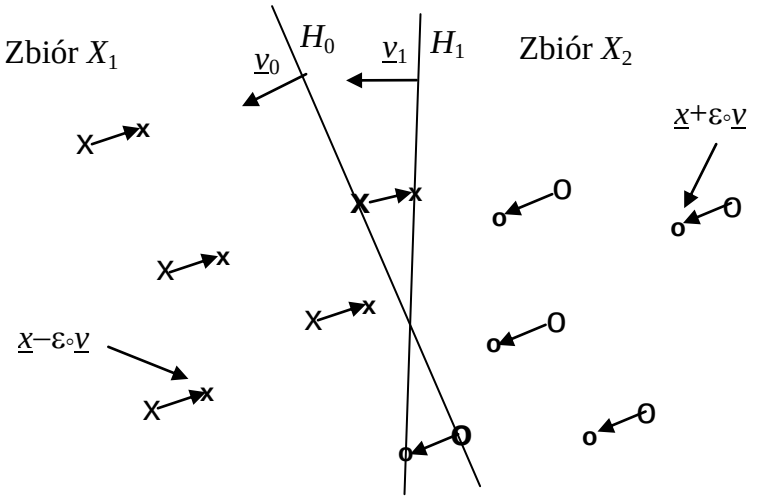
Korzyść z tego twierdzenia jest taka, że daje ono możliwość zredukowania liczby punktów, dla których $\underline{v}^*\circ\underline{y}=0$, co oznacza, że odpowiadająca wektorowi $\underline{v}^*$ hiperpłaszczyzna rozdzielająca zbiory X_1 i X_2 zawiera mniej punktów ze zbiorów X_1 i X_2. Gdyby takich punktów nie było wcale, czyli $Z^*=\varnothing$, to hiperpłaszczyzna ta ściśle rozdzielałaby zbiory X_1 i X_2.

Oczywiście, możliwość wykorzystania tego twierdzenia w wymienionym celu jest realna, tylko wtedy, gdy zbiory X_1 i X_2 są ściśle liniowo rozdzielne. Idea tego twierdzenia [Jóźwik A., 1981] została wykorzystana w rozprawie doktorskiej [Sturgulewski Ł. 2008] do wyznaczania hiperplaszczyzny ściśle rozdzielającej. Badaniu różnych typów rozdzielności liniowej dwóch zbiorów, z wykorzystaniem algorytmu rekursywnego, poświęcona została także inna praca doktorska [Cendrowska D., 2007]. Prezentacje rozwiązań, zawartych w obu w/w pracach zajęłyby zbyt dużo miejsca i z tego powodu nie będą zamieszczone w niniejszej monografii.

Innym rozwiązaniem pozwalającym na otrzymanie hiperpłaszczyzny ściśle rozdzielającej badane zbiory X_1 i X_2, jeśli ona istnieje, jest powiększenie tych zbiorów w taki sposób, aby ze słabej liniowej rozdzielności zbiorów powiększonych wynikała ścisła liniowa rozdzielność zbiorów oryginalnych. Ideę tą ilustruje Rys. 3.10.

Korzyść z takiego podejścia może być odniesiona, jeśli w dyspozycji badacza znajduje się algorytm badania słabej rozdzielności liniowej.

W wyniku zastosowania algorytmu rekursywnego do zbiorów X_1 i X_2 otrzymana zostanie hiperpłaszczyzna H_0. Zbiór X_1 zostaje powiększony w ten sposób, że z każdego punktu $\underline{x}$ tego zbioru tworzy się dodatkowy punkt $\underline{x}-\varepsilon\circ\underline{v}_0$, gdzie $\underline{v}_0$ jest wektorem normalnym hiperpłaszczyzny H_0 słabo rozdzielającej zbiory X_1 i X_2.



Rys. 3.10. Metoda powiększania zbiorów oryginalnych

Podobnie, z każdego punktu $\underline{x}$ ze zbioru X_2 generuje się nowy punkt $\underline{x}+\varepsilon\circ\underline{v}_0$. Jeżeli X_1 i X_2 zawierają m punktów, to taka jedna sesja, tj. operacja, powiększania zbiorów powoduje wzrost liczby punktów w powiększonych zbiorach o m. Na Rys.3.10 punkty oryginalnych zbiorów X_1 i X_2 zostały przedstawione większymi krzyżykami i kółkami, a punkty dodane mniejszymi. Do tak powiększonych zbiorów można ponownie zastosować algorytm rekursywny uzyskując nową hiperpłaszczyznę rozdzielającą H_1 z wektorem normalnym $\underline{v}_1$. Opisaną operację dodawania nowych punktów można powtórzyć dołączając do już powiększonych zbiorów nowe punkty postaci: $\underline{x}-\varepsilon\circ\underline{v}_1$ do zbioru X_1 i punkty postaci $\underline{x}+\varepsilon\circ\underline{v}_1$ do X_2. Podobnie można wygenerować kolejną hiperpłaszczyznę rozdzielającą, której odległość od najbliższego obiektu ze zbioru $X=X_1\cup X_2$ będzie większa niż to miało miejsce w przypadku poprzedniej hiperpłaszczyzny. Z wyznaczania kolejnych hiperpłaszczyzn można zrezygnować, gdy różnice odległości kolejnych hiperpłaszczyzn od najbliższego obiektu ze zbioru X będą już dostatecznie małe.

W pokazanym przykładzie na Rys. 3.10, już po jednej operacji dodania nowych punktów uzyskana została ścisła rozdzielność liniowa badanych zbiorów. Hiperpłaszczyzna H_1 słabo rozdzielająca powiększone zbiory rozdziela ściśle zbiory oryginalne, czyli X_1 i X_2.

Poniżej zostanie sformułowany algorytm wykorzystujący przedstawione wyżej podejście do badania rozdzielności liniowej dwóch zbiorów skończonych w przestrzeni cech.

Algorytm badania rozdzielności liniowej z zadanym prześwitem

Badanie rozdzielności liniowej z zadanym prześwitem polega na wielokrotnym zastosowaniu jednego z algorytmów przedstawionych w artykułach autora niniejszej

monografii [Jóźwik A., 1983a, Jóźwik A., 1998a], który dla wygody obecnych rozważań zostanie zapisany w postaci:

$$(\underline{w}, w_{n+1}, lg) = A(X_1, X_2), \tag{3.21}$$

gdzie $\underline{w}=[w_1, w_2, \ldots, w_n]$ oraz $|\underline{w}|=1$.

Zwraca on wektor $\underline{v}=[w_1, w_2, \ldots, w_n, w_{n+1}]$ o składowych rzeczywistych oraz liczbę *lg* przyjmującą wartości -1 lub 0. Jeżeli $lg=-1$, to oznacza, że zbiory X_1 i X_2 okazały się liniowo nierozdzielne, a gdy $lg=0$, to zbiory te są liniowo rozdzielne oraz równanie $g(x)=\underline{w}\circ\underline{x}+w_{n+1}=0$ opisuje równanie hiperpłaszczyzny rozdzielającej *H*.

Definicja algorytmu

1. Zadaj prześwit $ps=2\circ\varepsilon$, gdzie ε jest liczbą rzeczywistą oraz parametr stopu δ zależny od dokładności obliczeń;
2. Wywołaj procedurę $(\underline{w}_0, w_{0,n+1}, lg) := A(X_1, X_2)$;
3. Jeżeli $lg=-1$ to skocz do 10;
4. Podstaw $Y_1 := X_1$ oraz $Y_2 := X_2$**;**
5. Utwórz zbiór $Z_1=\{\underline{x}-\varepsilon\circ\underline{w}: x\in X_1\}$ oraz $Z_2=\{x+\varepsilon\circ\underline{w}: x\in X_2\}$;
6. Utwórz $Y_1 := Y_1\cup Z_1$ oraz $Y_2 := Y_2\cup Z_2$;
7. Wywołaj procedurę $(\underline{w}, w_{n+1}, lg) := A(Y_1, Y_2)$;
8. Jeżeli $lg=-1$, to skocz do 10;
9. Jeśli $|d(H, X)-\varepsilon|<\delta$, to skocz do 10, w przeciwnym przypadku skocz do 5;
10. Jeżeli $lg=-1$, to zbiory X_1 i X_2 nie mogą być rozdzielone z zadanym prześwitem *ps*, w przeciwnym przypadku, gdy $lg=0$, to należy przyjąć, że hiperpłaszczyzna zdefiniowana równaniem $g(x)=\underline{w}\circ\underline{x}+w_{n+1}=0$ rozdziela zbiory X_1 oraz X_2 z prześwitem *ps*, przy czym $g(\underline{x})>0$, gdy $\underline{x}\in X_1$ oraz $g(\underline{x})<0$, gdy $\underline{x}\in X_2$.

Warto zauważyć, że algorytm $(\underline{w}, w_{n+1}, lg) := A(Y_1, Y_2)$ jest wywoływany dla coraz to większych zbiorów Y_1 i Y_2.

Dowód zbieżności algorytmu

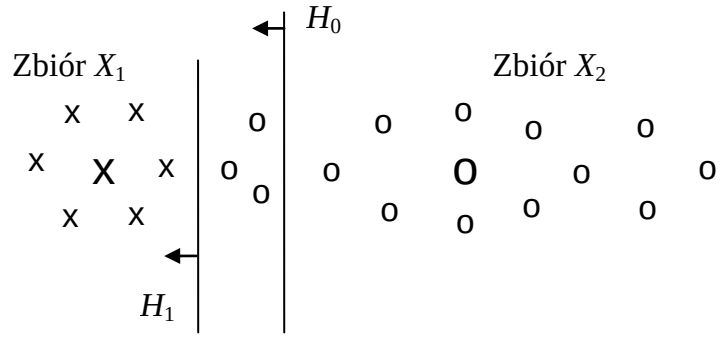
Niech zbiory X_1 i X_2 będą ściśle liniowo rozdzielne oraz niech H_0 oznacza hiperpłaszczyznę zdefiniowaną równaniem $g_0(\underline{x})=\underline{w}_0\circ\underline{x}+w_{0,n+1}=0$, tj. uzyskaną w kroku 2 algorytmu. Kolejne hiperpłaszczyzny H_i, $i=1,2,3,...$, otrzymywane są jako wynik wywołań w kroku 7 i tworzą nieskończony ciąg $(H_i)_{i=1}^{\infty}$. Hiperpłaszczyzny H_i powstają jako wynik rozdzielania zbiorów $Y_{i,1}$ i $Y_{i,2}$, tzn. z wywołań $(\underline{w}_i, w_{i,n+1}, lg) := A(Y_{i,1}, Y_{i,2})$ i są określane równaniami $g_i(\underline{x})=\underline{w}_i\circ\underline{x}+w_{i,n+1}=0$. Nieskończoność ciągu (H_i) została przyjęta tylko dla celu niniejszego dowodu. Oznacza to, że bezpośrednio po kroku 9 następuje skok do kroku 5, tzn. algorytm nigdy się nie kończy. Dla każdej pary zbiorów $Y_{i,1}$ i $Y_{i,2}$ istnieje zbiór S_i hiperpłaszczyzn rozdzielających te zbiory. Łatwo

zauważyć, że $Y_{i,1} \subset Y_{i+1,1}$ oraz $Y_{i+1,2} \subset Y_{i,2}$, a stąd wynika, że $S_{i+1} \subset S_i$. Niech G_i będzie hiperpłaszczyzną, ze zbioru hiperpłaszczyzn S_i, której odległość od zbioru $X=X_1 \cup X_2$ jest minimalna. Hiperpłaszczyzna H_i również należy do zbioru S_i. Nietrudno zauważyć, że dla każdego i, $i=1,2,3,\ldots$, zachodzi nierówność: $d(H_i,X) \geq d(G_i,X)$. Ponieważ $S_{i+1} \subset S_i$, to $d(G_{i+1},X) \geq d(G_i,X)$ (minimum na większym zbiorze jest nie większe). Ciąg $d(G_i,X)$, $i=1,2,3,\ldots$, jest niemalejącym, nieskończonym i ograniczonym od góry, tzn. $d(G_i,X) \leq \varepsilon$. A zatem musi być zbieżny. Metodą nie wprost łatwo wykazać, że nie może być zbieżny do mniejszej granicy niż ε. A więc $d(G_i,X) \rightarrow \varepsilon$, a ponieważ $d(G_i,X) \leq d(H_i,X) \leq \varepsilon$, to również $d(H_i,X) \rightarrow \varepsilon$, co było do wykazania.

3.5. Edytowanie zbioru uczącego dla liniowej rozdzielności zbiorów

Jednym z prostszych sposobów konstruowania liniowej funkcji dyskryminacyjnej dla przypadku dwóch zbiorów jest wyznaczenie hiperpłaszczyzny ortogonalnej do odcinka łączącego środki ciężkości klas i przechodzącej przez jego środek. Klasyfikator wykorzystujący funkcję dyskryminacyjną określoną tą hiperpłaszczyzną jest ekwiwalentny klasyfikatorowi minimalno-odległościowemu, w którym każda z klas reprezentowana jest przez środek ciężkości. Obiekt przyporządkowywany jest do klasy, której reprezentant znajduje się w mniejszej odległości.

Klasyfikator minimalno-odległościowy może dawać bardzo złe wyniki. Zależy to od rozkładu obiektów, co zostało pokazane na Rys. 3.11.

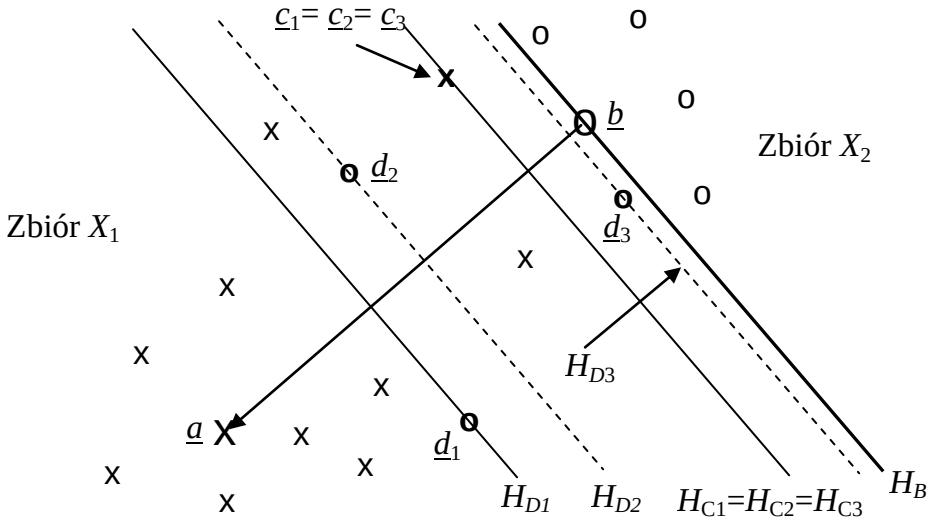


Rys. 3.11. Ilustracja wady klasyfikatora minimalno-odległościowego

Znakami większych rozmiarów zostały zaznaczone środki ciężkości klas, a mniejszymi obiekty zbioru uczącego. Hiperpłaszczyzna H_0 wyznacza rozdzielenie obiektów według klasyfikatora minimalno-odległościowego. Obiekty ze zbioru X_1 powinny leżeć po jej lewej stronie, jak wskazuje strzałka, czyli wektor normalny, a zatem trzy obiekty z klasy 2 znajdują się złej stronie tej hiperpłaszczyzny.

Wśród hiperpłaszczyzn równoległych do hiperpłaszczyzny H_0 istnieje bardziej korzystna hiperpłaszczyzna H_1, która idealnie rozdziela zbiory X_1 i X_2, a jej wyznaczenie jest zadaniem bardzo prostym, gdyż jej równanie różni się od równania hiperpłaszczyzny H_0 tylko wyrazem wolnym.

Przedstawioną ideę można wykorzystać również w przypadku zbiorów liniowo nierozdzielnych. By wyjaśnić sposób jej wykorzystania wygodnie będzie posłużyć ilustrującym przykładem pokazanym na Rys. 3.12.



Rys. 3.12. Ilustracja do metody edycji zbioru uczącego dla liniowej rozdzielności

Pierwszą operacją obliczeniową jest wyznaczenie środków ciężkości klas. Na Rys.3.12 są nimi punkty $\underline{a}$ i $\underline{b}$. Przez punkt $\underline{b}$ można poprowadzić hiperpłaszczyznę przechodząca przez ten punkt i ortogonalną do odcinka łączącego punkty $\underline{a}$ i $\underline{b}$, czyli ortogonalną do wektora $\underline{a}$-$\underline{b}$. Równanie tej hiperpłaszczyzny jest następujące:
$g_B(\underline{x})=(\underline{a}-\underline{b})\circ(\underline{x}-\underline{b})=0$. Wyznaczany jest teraz punkt $\underline{c}_1$ ze zbioru X_1 dla którego funkcja $g_B(\underline{x})$ osiąga minimum oraz punkt $\underline{d}_1$ ze zbioru X_2 dla którego funkcja $g_B(\underline{x})$ osiąga maksimum.

Przez równoległe przesunięcie hiperpłaszczyzny H_B, pierwszy raz do punktu $\underline{c}_1$ i drugi raz do punktu $\underline{d}_1$ otrzymana zostanie para hiperpłaszczyzn H_{C1} oraz H_{D1}. W pasie pomiędzy tymi hiperpłaszczyznami znajdują się trzy obiekty z klasy 1 oraz 2 obiekty z klasy 2.

Skoro w tym pasie, łącznie z punktami znajdującymi się na ograniczających go hiperpłaszczyznach leży więcej obiektów z klasy 1, to przyjmuje się, że z dwóch skrajnych punktów $\underline{c}_1$ oraz $\underline{d}_1$ w obszar klasy 1 *wtargnął* punkt $\underline{d}_1$, dlatego zostaje on wyrzucony ze zbioru X_2. Ponownie wyznaczana jest nowa para punktów skrajnych $\underline{c}_2$ i

$\underline{d}_2$. Punkt $\underline{c}_2$ będzie taki sam jak punkt $\underline{c}_1$, gdyż wyznaczany jest jako obiekt ze zbioru X_1 dla którego funkcja $g_B(\underline{x})$ osiąga minimum, a zbiór X_1 nie uległ zmianie. Uaktualnienia wymaga tylko punkt $\underline{d}_1$, tzn. wyznaczany jest nowy punkt $\underline{d}_2$ dla którego funkcja $g_B(\underline{x})$ osiąga maksimum na zbiorze X_2 pomniejszonym o punkt $\underline{d}_1$, tzn. na zbiorze X_2-$\{\underline{d}_1\}$.

W nowym pasie zawartym pomiędzy hiperpłaszczyznami H_{C2} oraz H_{D2}, o równaniach $g_{C2}(\underline{x})=(\underline{a}\text{-}\underline{b})\circ(\underline{x}\text{-}\underline{c}_2)=0$ i $g_{D2}(\underline{x})=(\underline{a}\text{-}\underline{b})\circ(\underline{x}\text{-}\underline{d}_2)=0$ odpowiednio, znajdują się dwa obiekty z klasy 1 i jeden z klasy 2, a więc usunięciu podlega punkt $\underline{d}_2$. Następnie musi być teraz wyznaczone uaktualnienie punktu $\underline{d}_2$, tzn. należy znaleźć punkt $\underline{d}_3$, a $\underline{c}_3=\underline{c}_2$, ponieważ zbiór X_1 nie uległ zmianie.

Po wykonaniu tej operacji, w pasie pomiędzy hiperpłaszczyznami H_{C3} oraz H_{D3}, nie ma żadnego obiektu poza obiektami znajdującymi się na tych hiperpłaszczyznach. Można to rozpoznać po tym, że $g_B(\underline{c}_3)>g_B(\underline{d}_3)$. Spełnienie tej nierówności jest warunkiem stopu algorytmu. Podczas operacji podobnych do wyżej opisanych może się zdarzyć, że w pasie pomiędzy hiperpłaszczyznami H_{Ci} oraz H_{Di} znajdować się będzie jednakowa liczba obiektów z każdej z klas. Wówczas, oba punkty $\underline{c}_i$ i $\underline{d}_i$ są wyrzucane oraz oba uaktualnienia $\underline{c}_{i+1}$ i $\underline{d}_{i+1}$ muszą być obliczane. Zasadę usuwania obiektów, w tym również w sytuacjach remisowych, da się zilustrować na przykładzie jednowymiarowym, jak na Rys. 3.13.

Wartość cechy	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0. Klasa obiektu	x	x	x	–o	x x	x o	o	–x	o		o	o		o
1. Klasa obiektu	x	x	x		x x	x –o	o	–x	o		o	o		o
2. Klasa obiektu	x	x	x		x x	x	o		o		o	o		o

Rys. 3.13. Przykład zbioru uczącego dla ilustracji zasady usuwania obiektów

Punktami skrajnymi, czyli definiującymi pas nakładania się klas, są punkty o wartościach cech $\underline{d}_1$=[4] i $\underline{c}_1$=[8], co zostało zaznaczone znakiem „–” przed obiektem (wiersz z numerem 0). Pomiędzy nimi znajdują się 4 krzyżyki i trzy kółka, łącznie z punktami o współrzędnych $\underline{c}_1$ i $\underline{d}_1$. Wyrzucony zatem musi być punkt $\underline{d}_1$. W kolejnym podejściu skrajnymi punktami będą: kółko $\underline{d}_2$=[6] i krzyżyk $\underline{c}_2$=[8] (wiersz z numerem 1). W pasie pomiędzy punktami ograniczającymi, wliczając punkty skrajne tego pasa $\underline{c}_2$ i $\underline{d}_2$, znajdują się teraz dwa krzyżyki i dwa kółka. Zatem krzyżyk $\underline{c}_2$=[8] oraz kółko $\underline{d}_2$=[6] zostają wyrzucone. Wreszcie pozostanie pas zawarty pomiędzy krzyżykiem $\underline{c}_3$=[6] oraz kółkiem $\underline{c}_3$=[7]. Hiperpłaszczyzna rozdzielająca będzie opisana równaniem $h(x)=g_{C3}(\underline{x})+g_{D3}(\underline{x})=(6\text{-}x)+(7\text{-}x)=13\text{-}2\circ x=0$. Obiekty dla których $h(x)\geq 0$ będą zaliczane do klasy 1, a obiekty dla których $h(x)<0$ do klasy 2.

Usuwanie obiektów ze zbioru uczącego w celu konstruowania klasyfikatora określane będzie jako edytowanie zbioru.

Definicja algorytmu

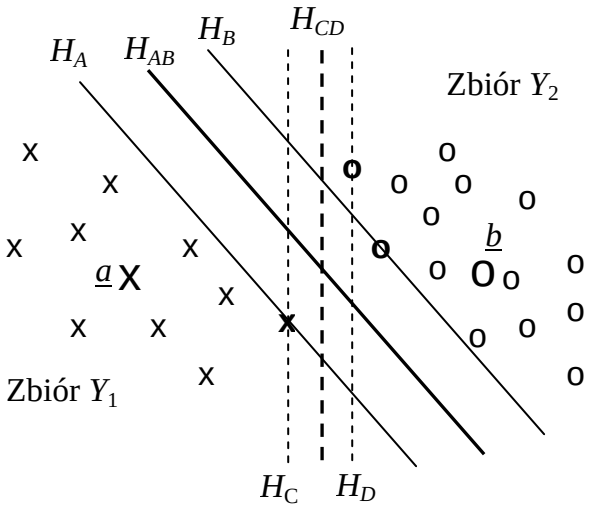
1. Wyznacz środki ciężkości $\underline{a}$ oraz $\underline{b}$ klasy 1 oraz klasy 2 odpowiednio;
2. Wyznacz hiperpłaszczyznę $g_B(\underline{x})=0$ przechodzącą przez punkt $\underline{b}$ i prostopadłą do wektora o początku $\underline{b}$ i końcu $\underline{a}$, tzn. $\underline{a}$-$\underline{b}$ jest jej wektorem normalnym.
3. Wśród punktów z klasy 1 znajdź punkt $\underline{c}$ dla którego funkcja $g_B(\underline{x})$ osiąga minimum, a wśród punktów z klasy 2 punkt $\underline{d}$ dla którego funkcja $g_B(\underline{x})$ osiąga maksimum, a następnie wyznacz dwie hiperpłaszczyzny $g_C(\underline{x})=0$ oraz $g_D(\underline{x})=0$ przechodzące odpowiednio przez punkty $\underline{c}$ i $\underline{d}$, równoległe do hiperpłaszczyzny $g_B(\underline{x})=0$, tzn. mające wektor $\underline{a}$-$\underline{b}$ jako wektor normalny.
4. Jeżeli punkt $\underline{c}$ leży po stronie punktu $\underline{a}$, a punkt $\underline{d}$ po stronie punktu $\underline{b}$, tj. $g_B(\underline{c})>g_B(\underline{d})$, to skocz do punktu 8.
5. Jeśli $g_B(\underline{c})\leq g_B(\underline{d})$, to znajdź liczbę l_1 takich punktów $\underline{x}$ z klasy 1 oraz liczbę l_2 takich punktów $\underline{x}$ z klasy 2, że $g_C(\underline{x})\geq 0$ i $g_D(\underline{x})\leq 0$.
6. Jeżeli l1>l2, to usuń punkt $\underline{d}$ i wyznacz punkt $\underline{b}$ na nowo. Gdy zaś l2>l1, to usuń punkt $\underline{c}$ i wyznacz punkt $\underline{a}$ na nowo. W przypadku, gdy l1=l2, to usuń zarówno punkt $\underline{c}$ jak i $\underline{d}$.
7. Skocz do podpunktu 3.
8. Wyznacz funkcję $h(\underline{x})=g_C(\underline{x})+g_D(\underline{x})$.

Funkcja dyskryminacyjna $h(\underline{x})$ definiuje klasyfikator dla pary klas 1 i 2. Jeżeli $h(\underline{x})\geq 0$, to punkt $\underline{x}$ kwalifikowany jest do klasy 1, w przeciwnym przypadku do klasy 2.

W wyniku operacji edytowania zbioru uczącego X, czyli usunięcia części jego obiektów, powstaną dwa nowe, liniowo rozdzielne, zbiory Y_1 oraz Y_2, przy czym $Y_1 \subset X_1$ i $Y_2 \subset X_2$. Można więc do tych zbiorów zastosować, algorytm korekcji błędów, algorytm iteracyjny lub algorytm rekursywny uzupełniony o możliwość badania liniowej rozdzielności z marginesem. Standardowo, krok 8 w definicji algorytmu rozsądnie byłoby zastąpić algorytmem iteracyjnym.

W przypadku, gdyby jego wynik wskazywał na brak rozdzielności, co może być skutkiem błędu numerycznego, można uciec się do algorytmu rekursywnego w uzupełnionej wersji dla liniowej rozdzielności z marginesem. Korzyści wynikające z takiej kombinacji zaproponowanych algorytmów zostaną wyjaśnione na przykładzie podanym na Rys. 3.14.

Stosując edytowanie zbioru uczącego dla liniowej rozdzielności uzyskana zostanie para hiperpłaszczyzn H_C i H_D oraz wynikowa hiperpłaszczyzna H_{CD}. Wszystkie te trzy hiperpłaszczyzny są prostopadłe do wektora łączącego punkty $\underline{a}$ i $\underline{b}$,



Rys. 3.14. Przykład przemawiający za użyciem w kroku 8 algorytmu iteracyjnego

czyli do wektora *a*-*b*. Zbiory Y_1 oraz Y_2 są z pewnością liniowo rozdzielne, ale uzyskana hiperpłaszczyzna wynikowa H_{CD} rozdziela te zbiory z niewielkim marginesem. Zastosowanie algorytmu iteracyjnego pozwoliłoby uzyskać nie mniejszy margines, ponieważ byłby to margines optymalny, tj. największy z możliwych. Hiperpłaszczyzna wynikowa H_{AB} przebiegałaby w środku między hiperpłaszczyznami H_A i H_B.

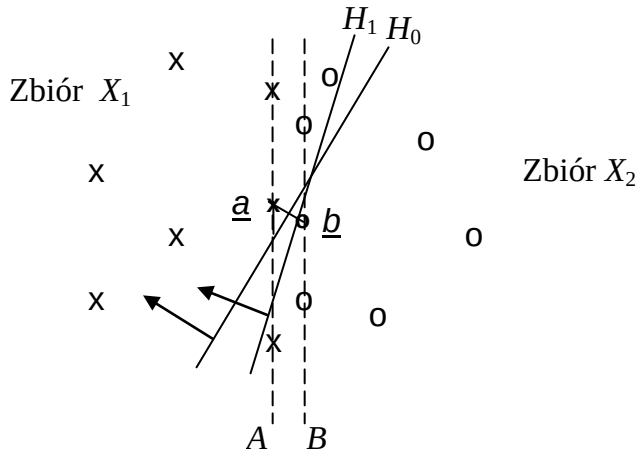
Algorytm ten, od kilku już lat jest prezentowany, przez autora niniejszej monografii, na studiach pierwszego i drugiego stopnia, a w zastosowaniu do zadania dwu-decyzyjnego, był przedmiotem pracy magisterskiej [Rychlik T., 2012]. Jego poprawność została zweryfikowana na wymienionym już w rozdziale 3 zbiorze Iris Data, oddzielnie dla każdej 3 możliwych par klas. Ponadto, w innej pracy, tym razem inżynierskiej [Tomaszewski W.P., 2013], w zastosowaniu do problemu wielo-decyzyjnego, został on porównany z klasyfikatorem 1-NS na zbiorze Iris Data oraz na innych zbiorach dostępnych w Internecie (http://archive.ics.uci.edu/ml/datasets.html). Zastosowany klasyfikator miał strukturę równoległą z Rys. 1.2. W kilku przypadkach, zaprezentowany algorytm z edycją dla liniowej rozdzielności oferował mniejsze frakcje błędów niż klasyfikator typu 1-NS.

Podsumowanie rozdziału

Przedstawione zostały cztery typy algorytmów wyznaczania hiperpłaszczyzn rozdzielających. Nie można jednak stwierdzić, że któryś z tych algorytmów mógłby wyeliminować z zastosowań pozostałe. Najprostszy z nich, algorytm korekcji błędów, może być zastosowany niezależnie od liniowej nierozdzielności zbiorów. Podobna jest sytuacja z algorytmem edytowania zbiorów dla liniowej rozdzielności. Są to więc algorytmy bardziej uniwersalne niż pozostałe dwa algorytmy. Algorytmy: iteracyjny i

rekursywny dotyczą badania liniowej rozdzielności zbiorów i w przypadku, gdy okaże się, że nie ma ona miejsca, nie umożliwiają skonstruowania akceptowalnej liniowej funkcji dyskryminacyjnej. Jednak warto zbadać, czy nie zachodzi szczególny przypadek, jakim jest ścisła rozdzielność liniowa, bo wówczas można ten fakt wykorzystać do poprawy jakości klasyfikacji, mierzonej odsetkiem poprawnych decyzji.

Algorytm iteracyjny pozwala znaleźć hiperpłaszczyznę optymalną. Ale w przypadku, gdy zbiory są liniowo rozdzielne, ale bardzo bliskie siebie, może dać mylną odpowiedź wskutek błędu numerycznego. Hiperpłaszczyzna jako miejsce geometryczne punktów równo odległych od punktów $\underline{a}$ i $\underline{b}$ otrzymanych przez zastosowanie algorytmu iteracyjnego może nie być wyznaczona wystarczająco dokładnie, co zostało pokazane na Rys. 3.15.



Rys. 3.15. Zbiory trudno rozdzielne dla algorytmu iteracyjnego

Na rysunku tym, większymi znakami, krzyżykami albo kółkami, zostały zaznaczone obiekty ze zbioru uczącego, zaś mniejszymi punkty znalezione przez algorytm iteracyjny, tzn. punkty $\underline{a}$ i $\underline{b}$. Powłoki wypukłe $\mathrm{Co}(X_1)$ i $\mathrm{Co}(X_2)$ zbiorów X_1 i X_2 są ściśle rozdzielne liniowo, ponieważ pomiędzy hiperpłaszczyznami A i B nie ma żadnych punktów ze zbioru uczącego $X=X_1\cup X_2$. Algorytm powinien wyznaczyć dwa najbliższe sobie punkty należące do przeciwnych powłok, czyli punkty $\underline{a}$ i $\underline{b}$. Jednakże wskutek ograniczonej dokładności obliczeń punkty te faktycznie nie będą sobie najbliższe.

Hiperpłaszczyzna H_0 przechodząca przez środek odcinka łączącego $\underline{a}$ i $\underline{b}$ i ortogonalna do niego może nie rozdzielić poprawnie zbiorów X_1 i X_2 i algorytm da odpowiedź, że badane zbiory nie są liniowo rozdzielne. Natomiast, algorytm rekursywny może dać odpowiedź, że zbiory te są liniowo rozdzielne w sensie słabej

rozdzielności, a jeśli zostanie zastosowane powiększenie zbiorów X_1 i X_2, zaproponowane w podrozdziale 3.4, to zostanie wykryta ścisła rozdzielność liniowa tych zbiorów. Algorytm ten w porównaniu z algorytmem iteracyjnym jest jednak bardziej złożony i znacznie trudniejszy do oprogramowania. Możliwym jest, że również algorytm korekcji błędów znajdzie hiperpłaszczyznę ściśle rozdzielającą, jednak ta cecha nie może być zagwarantowana, ponieważ nie jest znana liczba kroków, jaką trzeba wykonać, a informacja, że jest ona skończona ma praktycznie małe znaczenie.

4. METODY MINIMALNO-ODLEGŁOŚCIOWE

Poprzedni rozdział poświęcony był klasyfikatorom dwu-decyzyjnym, w których funkcja dyskryminacyjna miała charakter liniowy. W interpretacji geometrycznej oznaczało to rozdzielanie hiperpłaszczyzną dwóch zbiorów, w taki sposób, aby możliwie najwięcej obiektów leżało po jej właściwej stronie, obiekty z klasy 1 po dodatniej stronie lub na tej hiperpłaszczyźnie, a obiekty z klasy 2 po ujemnej. Ograniczenie dwu-decyzyjnością nie jest problemem, gdyż z klasyfikatorów dwu-decyzyjnych mogą być zbudowane klasyfikatory wielo-decyzyjne, jak sugeruje to struktura pokazana na Rys. 1.2 zamieszczonym w rozdziale 1. Istotą takiego podejścia było różnicowanie klas. Obiekty znajdujące się po jednej stronie hiperpłaszczyzny rozdzielającej traktowane były jako pochodzące z innej klasy niż obiekty leżące po drugiej stronie.

W obecnym następnym podrozdziale rozważane będą metody bazujące raczej na podobieństwie obiektów niż różnicowaniu ich na podstawie położenia w odniesieniu do hiperpłaszczyzn. Jako funkcje podobieństwa obiektów do rozważanych klas mogą być przyjęte funkcje dyskryminacyjne realizowane w strukturze pokazanej na Rys. 1.3 z rozdziału 1. W przypadku dwóch klas można dwóch funkcji dyskryminacyjnych $g_1(\underline{x})$ i $g_2(\underline{x})$, których wartości są odpowiednio miarami podobieństwa obiektów $\underline{x}$ do klasy 1 i 2, odpowiednio. Znacznie wygodniej jest jednak użyć jednej różnicującej funkcji dyskryminacyjnej o postaci $g_{12}(\underline{x})=g_1(\underline{x})-g_2(\underline{x})$.

4.1. Klasyfikator minimalno-odległościowy

Podobieństwo obiektu do klasy można mierzyć funkcją odległości, wystarczy by każda z klas i była reprezentowana przez jeden punkt $\underline{p}_i$ $i=1,2,\ldots,nc$, w przestrzeni cech, przy czym mniejszej odległości klasyfikowanego obiektu $\underline{x}$ od punktu $\underline{p}_i$ odpowiada większe podobieństwo tego obiektu do klasy i. Zastosowanie funkcji odległości mieści się też w ogólnym schemacie klasyfikatora z Rys.1.3, co staje się zrozumiałe, jeśli przyjąć, że funkcja $g_i(\underline{x})=-d(\underline{x},\underline{p}_i)$ albo $g_i(\underline{x})=1/[1+d(\underline{x},\underline{p}_i)]$. Punkty $\underline{p}_i$,

i=1,2,…,nc, mogą być środkami ciężkości klas albo punktem, którego współrzędnymi są mediany cech albo jeszcze innym punktem wyznaczonym na podstawie zbioru uczącego. Ten typ klasyfikatora znany jest jako klasyfikator minimalno-odległościowy. Najczęściej stosowaną funkcją odległości jest miara euklidesowa $d(\underline{x},\underline{y})=(\sum_{j=1}^{n}(x_j-y_j)^2)^{1/2}$ lub odległość miejska $d(\underline{x},\underline{y})=\sum_{j=1}^{n}|x_j-y_j|$, gdzie $\underline{x}=[x_1,x_2,\ldots,x_n]$. Listę innych stosowanych miar odległości można znaleźć w pracy [Jajuga K., 1990].

W przypadku zastosowania odległości euklidesowej klasyfikator minimalno-odległościowy da się sprowadzić do struktury przedstawionej na Rys. 1.3, z liniowymi funkcjami dyskryminacyjnymi, co niżej zostanie pokazane. Zamiast odległości $d(\underline{x},\underline{p}_i)$ można korzystać z jej kwadratu:

$$d^2(\underline{x},\underline{p}_i)=(\underline{x}-\underline{p}_i)^2=\underline{x}^2-2\circ\underline{p}_i\circ\underline{x}+(\underline{p}_i)^2=\underline{x}^2-[2\circ\underline{p}_i\circ\underline{x}-(\underline{p}_i)^2]=\underline{x}^2-g_i(\underline{x}), \qquad (4.1)$$

gdzie $g_i(\underline{x})=2\circ\underline{p}_i\circ\underline{x}-(\underline{p}_i)^2$.

Odległość $d(\underline{x},\underline{p}_i)$ osiąga minimum dla tych samych wartości wskaźnika i dla których funkcja $g_i(\underline{x})$ przyjmuje maksimum. Funkcje $g_i(\underline{x})$ mogą więc służyć jako funkcje dyskryminacyjne i być realizowanymi w blokach struktury pokazanej na Rys. 1.3. Są to funkcje liniowe. Podstawiając $\underline{x}=[x_1,x_2,\ldots,x_n]$ oraz $\underline{p}_i=[p_{i,1},p_{i,2},\ldots,p_{i,n}]$ można ją rozpisać w postaci:

$g_i(\underline{x})=2\circ[p_{i,1},p_{i,2},\ldots,p_{i,n}]\circ[x_1,x_2,\ldots,x_n]-[p_{i,1},p_{i,2},\ldots,p_{i,n}]^2$, skąd wynika, że

$g_i(\underline{x})=\sum_{j=1}^{n}2\circ p_{i,j}\circ x_j-\sum_{j=1}^{n}p_{i,j}^2$. Funkcja dyskryminacyjna dla klasy i ma zatem postać:

$$g_i(\underline{x})=\sum_{j=1}^{n}w_{i,j}\circ x_j+w_{i,n+1}, \qquad (4.2)$$

gdzie $w_{i,j}=2\circ p_{i,j}$ oraz $w_{i,n+1}=-\sum_{j=1}^{n}p_{i,j}^2$.

Z przeprowadzonych wyżej rozważań wynika, ze klasyfikator minimalno-odległościowy może być zbudowany jako maszyna liniowa [Nilsson. N., 1965].

W przypadku dwóch klas maszyna liniowa może działać z jedną funkcją dyskryminacyjną i wymaga wówczas tylko n mnożeń, podczas gdy oryginalna wersja klasyfikatora minimalno-odległościowego wymagałaby $2\circ n$ mnożeń. Jeśli klas jest więcej niż dwie, to już ta korzyść nie ma miejsca, gdyż należy użyć wtedy tyle funkcji dyskryminacyjnych ile jest klas, co było już przedmiotem rozważań w podrozdziale 1.3.

W cytowanej wyżej pracy nie została uwzględniona inna zaleta maszyny liniowej, a związana jest ona z problemem standaryzacji cech. Uwzględniając standaryzację cech w klasyfikatorze minimalno-odległościowym, nie ma potrzeby standaryzacji całego zbioru uczącego. Wystarczy, że będą standaryzowane tylko punkty $\underline{p}_i$,

$i=1,2,\ldots,nc$, będące reprezentantami klas. Jest to operacja jednorazowa, ale jednak każdy nowo klasyfikowany obiekt też musi być standaryzowany. Maszyna liniowa zaś daje możliwość włączenia standaryzacji do wag funkcji dyskryminacyjnych i wtedy klasyfikowanych obiektów nie należy standaryzować.

Klasyfikator minimalno-odległościowy, jeśli uwzględniona zostanie standaryzacja cech, przyporządkowuje obiektowi $\underline{x}$ klasę i, jeśli $d(\underline{x}^s, p_i{}^s)=\min_j d(\underline{x}^s, p_j^s)$, gdzie $p_i{}^s$ oraz p_j^s są standaryzowanymi środkami ciężkości klas, a $\underline{x}^s=[x_1^s, x_2^s,\ldots, x_n^s]$ jest klasyfikowanym obiektem po standaryzacji. Funkcja dyskryminacyjna (4.2), w przypadku uwzględnienia standaryzacji cech przyjmie postać:

$$g_i(\underline{x}^s)=\sum_{j=1}^{n} w_{i,j}\circ x_j^s + w_{i,n+1}, \tag{4.3}$$

gdzie $w_{i,j}=2\circ p_{i,j}^s$ oraz $w_{i,n+1}=-\sum_{j=1}^{n} (p_{i,j}^s)^2$.

We wzorze (4.3) za zmienne x_j^s można podstawić $x_j^s=(x_j-mv_j)/sd_j$, gdzie mv_j i sd_j są odpowiednio wartością średnią i odchyleniem standardowym cechy j lub medianą i odchyleniem medianowym tej cechy. Wtedy otrzymana zostanie funkcja dyskryminacyjna $g_i(\underline{x}^s)=\sum_{j=1}^{n} \frac{w_{i,j}}{sd_j}\circ x_j + w_{i,n+1}-\sum_{j=1}^{n} \frac{w_{i,j}}{sd_j}\circ mv_j$, w której już nie występują standaryzowane wartości cech. Można ją zapisać w następującej formie:

$$g_i^s(\underline{x})=\sum_{j=1}^{n} w_{i,j}^s\circ x_j + w_{i,n+1}^s, \tag{4.4}$$

gdzie $w_{i,j}^s=\frac{w_{i,j}}{sd_j}$, a $w_{i,n+1}^s=w_{i,n+1}-\sum_{j=1}^{n} \frac{w_{i,j}}{sd_j}\circ mv_j$.

Funkcja (4.4) nie zawiera już standaryzowanych wartości cech, gdyż standaryzacja zawarta została w jej w wagach.

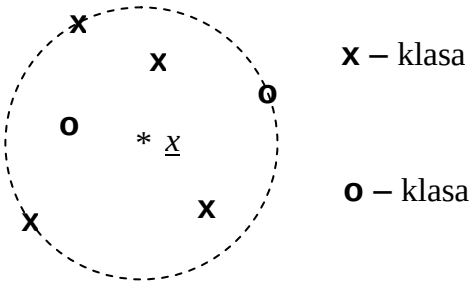
Klasyfikator minimalno-odległościowy nie daje w ogólnym przypadku dobrych wyników klasyfikacji, co zilustrowane zostało w rozdziale 3 na Rys. 3.11, ale w szczególnych zadaniach może on oferować akceptowalną jakość klasyfikacji, a przy tym jest bardzo szybki. Zajmuje on także mało pamięci, ponieważ w fazie klasyfikacji wystarczy pamiętać nc środków ciężkości lub innych nc punktów reprezentujących rozpatrywane klasy. Takie punkty stanowią zbiór odniesienia dla klasyfikacji. Musi on być cały czas przechowywany w pamięci komputera w fazie klasyfikacji. Ze względu na prostotę klasyfikatora minimalno-odległościowego, korzystne jest zbadanie, czy w konkretnym zadaniu nie jest on wystarczający, gdyż taki eksperyment nie wymaga dużego nakładu pracy.

Oczywiście, jak w przypadku każdego innego klasyfikatora warto przy konstruowaniu uwzględnić selekcję cech. Ponadto, ze względu na małą jego

elastyczność, tj. małą zdolność dopasowywania się do zbioru uczącego, frakcję błędów można oceniać na podstawie zbioru uczącego, bez konieczności użycia zbioru testującego, czy też uciekania się do metody minus jednego elementu.

4.2. Reguła *k* najbliższych sąsiadów

Reguła *k* najbliższych sąsiadów (*k*-NS) została zaproponowana ponad 60 lat temu w pracy [Fix E., Hodges J.L., 1952]. Klasyfikator działający wg tej reguły przyporządkowuje obiektowi klasę z której pochodzi większość spośród *k* najbliższych mu obiektów w zbiorze uczącym. Jeżeli wyznaczenie *k*-tego najbliższego sąsiada nie jest jednoznaczne, czyli w tej samej odległości od klasyfikowanego obiektu co *k*-ty najbliższy sąsiad są jeszcze inne obiekty, to one wszystkie powinny wziąć udział w głosowaniu. Inaczej mówiąc, *k*-ty najbliższy sąsiad definiuje najmniejszą hiperkulę, która zawiera co najmniej *k* obiektów ze zbioru uczącego, najbliższych klasyfikowanemu obiektowi. W głosowaniu zaś biorą udział wszystkie obiekty, które w tej hiperkuli się znajdą. Zasadę tą zilustrowano na Rys.4.1, który pokazuje, że w tej samej odległości, co czwarty najbliższy sąsiad, bez względu na to, który obiekt nim będzie, znajdują się jeszcze dwa inne obiekty.



Rys. 4.1. Ilustracja do wyjaśnienia zasady działania reguły 6-NS.

Nie ma powodów by wszystkie obiekty, które znalazły się w hiperkuli o promieniu równym odległości do czwartego najbliższego sąsiada nie miały brać udziału w głosowaniu.

Pomimo swej prostoty klasyfikator działający na podstawie tej reguły oferuje wysoką jakość klasyfikacji, w sensie prawdopodobieństwa poprawnej decyzji, w porównaniu do wielu innych znanych algorytmów [Carpenter G. A., Grossberg S.,1996]. Przeprowadzone zostały eksperymenty, które dotyczyły 6 klas, 36 liczbowych cech, zbioru uczącego o liczności 4435 obiektów i zbioru testującego zawierającego 2000 obiektów. Klasyfikatory, znane jako *k*-NN, Fuzzy Artmap, RBF, sieci neuronowe z metodą propagacji wstecznej i regresja logistyczna oferowały odpowiednio 91, 89, 88, 86 i 83 procent błędów.

Reguła *k*-NS bezpośrednio aproksymuje warunkowe prawdopodobieństwa klas $p(j/\underline{x})$, o których była mowa w rozdziale 1 we wzorze 1.1. Oszacowaniem prawdopodobieństwa $p(j/\underline{x})$ jest proporcja k_j/k, gdzie k_j jest liczbą najbliższych sąsiadów klasyfikowanego obiektu $\underline{x}$ z klasy j wśród k najbliższych sąsiadów.

Klasyfikator *k*-NS może być traktowany jako aproksymacja klasyfikatora bayesowskiego i wtedy prawdopodobieństwa $p(j/\underline{x})$ nie są szacowane bezpośrednio lecz z wykorzystaniem wzoru Bayes'a, tzn. są one wyliczane na podstawie prawdopodobieństw $p(j)$, gęstości rozkładu prawdopodobieństwa $f(\underline{x}/j)$ oraz gęstości rozkładu prawdopodobieństwa $f(\underline{x})$. Jeżeli istnieje potrzeba tylko rozstrzygnięcia, które z prawdopodobieństw $p(j/\underline{x})$ osiąga wartość maksymalną, to, jak już to było zauważone w rozdziale 1, szacowanie $f(\underline{x})$ nie jest konieczne.

Otoczenie klasyfikowanego obiektu $\underline{x}$, zawierające k najbliższych sąsiadów może być wykorzystane do aproksymacji funkcji gęstości $f(\underline{x}/j)$ oraz $f(\underline{x})$ rozkładu prawdopodobieństw potrzebnych do oszacowania prawdopodobieństw $p(j/\underline{x})$. Zaś prawdopodobieństwo $p(j)$ pojawienia się obiektu z klasy j może być oszacowane wprost ze zbioru uczącego, jeśli tylko został on zebrany zgodnie z częstościami występowania klas. Wykorzystując sąsiedztwo klasyfikowanego obiektu zawierające k obiektów, oszacowania funkcji występujących we wzorze Bayes'a są następujące: $p(j)=m_j/m$, $f(\underline{x}/j)= k_j/m_j$, $f(\underline{x})=k/m$. Podstawienie ich do wzoru Bayes'a daje wynik:

$$p(j/\underline{x})=p(j)\circ f(\underline{x}/j)/f(\underline{x})=\frac{m_j}{m}\circ\frac{k_j}{k}\circ\frac{m}{k}=\frac{k_j}{k}, \tag{4.5}$$

tzn. uzyskane zostało takie samo oszacowanie jak poprzednio.

Przedstawione podejście nie jest jedynym dla oszacowania funkcji gęstości $f(\underline{x}/j)$ oraz $f(\underline{x})$ rozkładu prawdopodobieństw. Z nieparametrycznych metod, wartymi uwagi są tzw. okna Parzena [Duda R., O., Hart P.E., Stork D., G., 2001]. Szczegółowo są one też przedstawione w publikacjach książkowych [Koronacki J., Ćwik J., 2005; Stąpor K., 2005].

W przeciwieństwie do klasyfikatorów, w których wykorzystywane są hiperpłaszczyzny rozdzielające, klasyfikatory działające wg reguły *k*-NS mogą być stosowane dla dowolnej liczby klas. Nie ma więc konieczności, uciekania się do równoległej sieci klasyfikatorów dwu-decyzyjnych, która zilustrowana została na Rys. 1.2 w rozdziale 1. Jednak, warto rozważyć, czy taka struktura nie oferuje wyższej jakości klasyfikacji w przypadku, gdy klasyfikatory składowe są dwu-decyzyjnymi klasyfikatorami typu *k*-NS. Dwa czynniki mogą wpłynąć na wyższą jakość klasyfikacji oferowaną przez strukturę równoległą w porównaniu z jakością klasyfikacji uzyskiwaną dla standardowego klasyfikatora *k*-NS. Pierwszym jest wyznaczanie oddzielnej liczby najbliższych sąsiadów dla każdego klasyfikatora składowego, a drugim przeprowadzanie oddzielnej selekcji cech.

Równoległy klasyfikator *k*-NS

Klasyfikator, definiuje w przestrzeni cech obszary decyzyjne. Punkt w przestrzeni cech należy do obszaru decyzyjnego klasy *i*, jeśli zostałby zaliczony do tej klasy przez klasyfikator. W przypadku standardowego klasyfikatora *k*-NS na przebieg granicy pomiędzy dwoma obszarami decyzyjnymi mają wpływ także obiekty z klas trzecich, jeśli zadanie dotyczy większej liczby klas niż dwie. Dzieje się tak poprzez zależność liczby *k* oraz wyselekcjonowanego zestawu cech od obiektów wszystkich klas reprezentowanych w zbiorze uczącym. W przypadku struktury równoległej, każdy z klasyfikatorów składowych jest konstruowany dla pary klas (*i*,*j*), *i*,=1,2,…,*nc*-1, *j*=*i*+1,*i*+2,…,*nc*, niezależnie od obiektów z pozostałych klas. Obiekty z tych pozostałych klas mogą wpływać na granicę pomiędzy klasami *i* oraz *j* tylko w fazie głosowania klasyfikatorów składowych, a nie głosowania obiektów, tj. najbliższych sąsiadów obiektu aktualnie klasyfikowanego. Warto więc zweryfikować czy klasyfikator o strukturze równoległej nie oferuje niższego prawdopodobieństwa mylnej klasyfikacji. Klasyfikator o strukturze równoległej złożony z dwudecyzyjnych klasyfikatorów *k*-NS został, prawdopodobnie po raz pierwszy, przedstawiony w publikacji [Jóźwik A., Vernazza G., 1988]. Kilka lat później zadanie porównania klasyfikatora równoległego ze standardowym zostało podjęte w pracy [Jóźwik A., 1994], z zastosowaniem sztucznie wygenerowanych zbiorów uczących w dwuwymiarowej przestrzeni cech, zawierających trzy klasy.

W pracy tej przeprowadzone zostały dwie serie eksperymentów dla różnych liczności zbiorów uczących, ale niżej zostaną przytoczone wyniki tylko serii eksperymentów, z liczniejszymi zbiorami uczącymi.

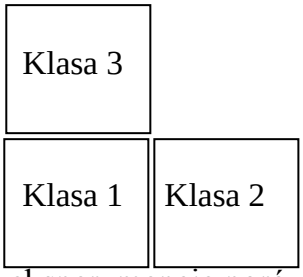
Klasy zajmowały obszar kwadratów o wierzchołkach, których składowe były dodatnimi liczbami całkowitymi :

klasa 1: $\underline{a}_1$=[0, 0], $\underline{b}_1$=[100,0], $\underline{c}_1$=[100,100], $\underline{d}_1$=[0,100],

klasa 2: $\underline{a}_2$=[100, 0], $\underline{b}_2$=[200,0], $\underline{c}_2$=[200,100], $\underline{d}_2$=[100,100].

klasa 3: $\underline{a}_3$=[0, 100], $\underline{b}_3$=[100,100], $\underline{c}_3$=[100,200], $\underline{d}_3$=[0,200],

W każdym z tych kwadratów wygenerowanych było po 1000 punktów o rozkładzie równomiernym. Współrzędne *x* i *y* punktów z klasy 1 przyjmowały wartości liczb rzeczywistych z przedziału [0,100]. Punkty z klasy 2 przyjmowały wartości liczb rzeczywistych o współrzędnych *x* z przedziału [100,200], a współrzędnych *y* z przedziału [0,100]. Natomiast, współrzędna *x* punktów z klasy 3 przyjmowała wartości liczb rzeczywistych z zakresu [0,100], a współrzędna *y* wartości liczb rzeczywistych z przedziału [100,200]. Współrzędne punktów losowane były z dokładnością do 0,01. Obszary klas zostały zilustrowane na Rys. 4.2.



Rys. 4.2. Ułożenie klas w eksperymencie porównawczym

Liczby *k* najbliższych sąsiadów były wyznaczone eksperymentalnie, z zastosowaniem jako kryterium frakcji błędów obliczonej metodą minus jednego elementu. Jest oczywiste, że standardowy klasyfikator *k*-NS musi wykorzystywać obie cechy *x* oraz *y* i równoległa sieć dwu-decyzyjnych klasyfikatorów *k*-NS również. Ale każdy klasyfikator składowy może używać tylko jednej cechy, która mogłaby być wybrana jako wynik selekcji cech. I tak dla klasyfikatora skonstruowanego dla pary klas 1 i 2 wystarczy cecha *x*, dla klasyfikatora rozstrzygającego pomiędzy klasami 1 i 3 wystarczającą cechą będzie cecha *y*. Natomiast, dla pary klas 2 i 3 mogłaby być zarówno cecha *x* jak i *y*, ale w eksperymentach wybrana została tylko cecha *x*. Na jedną serię eksperymentów składało się wygenerowanie po 1000 punktów dla każdej z klas oraz obliczenie frakcji błędów dla trzech klasyfikatorów typu *k*-NS: standardowego klasyfikatora, równoległej sieci bez selekcji cech oraz równoległej sieci z przeprowadzeniem selekcji cech. Takich serii było 10. Frakcje błędów liczone były dla wszystkich możliwych wartości *k*. Wybrane zostały najmniejsze wartości z tych frakcji i odpowiadające im wartości liczb *k*. Wyniki tych eksperymentów zostały pokazane w Tab. 4.1.

Tab. 4.1. Frakcje błędów dla trzech różnych klasyfikatorów

Numer eksperymentu	Klasyfikator standardowy	Równoległa sieć bez selekcji cech	Równoległa sieć z selekcją cech
1	0,0067	0,0057	0,0000
2	0,0030	0,0027	0,0000
3	0,0047	0,0043	0,0000
4	0.0017	0,0000	0,0000
5	0.0053	0,0027	0,0000
6	0,0043	0,0000	0,0000
7	0,0057	0,0000	0,0000
8	0,0030	0,0000	0,0000
9	0,0027	0,0007	0,0003
10	0,0043	0,0000	0,0000

Wyznaczone liczby najbliższych sąsiadów, dla klasyfikatora standardowego, wahały się od 1900 do 2100, przy czym należy zaznaczyć, że w przypadkach niejednoznacznych wybierane były największe liczby *k*. Liczby *k* ustalone dla równoległej sieci klasyfikatorów były już znacznie mniejsze, bo przyjmowały wartości od 1 do 230.

W eksperymentach tych wystąpił efekt *dostrajania* się klasyfikatora pod konkretny zbiór danych, jednak odbywało się ono tylko z użyciem jednego parametru, czyli liczby *k*. Autor niniejszej monografii zamierza w przyszłości powtórzyć podobne eksperymenty z bardziej liczebnymi zbiorami i z oceną frakcji błędów zarówno metodą minus jeden element jak i z zastosowaniem zbiorów testujących.

Wyniki eksperymentów przemawiają za równoległą siecią klasyfikatorów dwu-decyzyjnych. Omawiana wyżej równoległa sieć klasyfikatorów dwu-decyzyjnych była przedmiotem rozważań w pracy [Siedlecki W., 1994], gdzie autor wykazał na drodze analitycznej przewagę klasyfikatora równoległego nad standardowym, z punktu widzenia jakości klasyfikacji.

4.3. Konstrukcja klasyfikatora *k*-NS w przypadku braków wartości cech

W badaniach biomedycznych często występują niekompletne zbiory danych, w tym również zbiory danych wykorzystywane do konstruowania reguł diagnostycznych, tzn. klasyfikatorów. Najprostszym sposobem postępowania w takich przypadkach jest usunięcie bądź obiektów z brakującymi wartościami cech, bądź usunięcie cech z brakującymi wartościami cech. Rozważania obecnego rozdziału ograniczone będą do pierwszego z wyżej wymienionych sposobów. Liczba odrzuconych obiektów ze zbioru uczącego zależy od wybranych cech. Przyjęto, że odrzucane będą tylko te obiekty dla których występują braki wartości, ale tylko w cechach, które zostaną ostatecznie wybrane.

W przeprowadzonych poniżej rozważaniach przyjęte zostało założenie, że prawdopodobieństwo mylnej decyzji szacowane jest z zastosowaniem metody minus jednego elementu, omawianej już w podrozdziale 2.1.

Podczas selekcji cech stosowane są zwykle procedury kolejnego dołączania cech albo kolejnego odrzucania cech lub ewentualnie kombinacja tych procedur. Dla każdej z przeglądanych kombinacji cech szacowany jest błąd klasyfikacji i ostatecznie wybierana jest kombinacja cech, która oferuje najmniejszy błąd klasyfikacji. Liczba obiektów, które należy usunąć, aby pozostały zbiór uczący był wolny od cech z brakującymi wartościami zależy od kombinacji użytych cech.

Jeżeli zastosowana zostanie procedura kolejnego odrzucania cech, to początkowej jej fazie zbiory uczące po usunięciu obiektów z brakującymi wartościami cech będą

mniejsze, a w miarę odrzucania kolejnych cech, zbiory te będą miały większą liczebność. Natomiast, w przypadku procedury kolejnego dołączania cech, na początku procedura ta będzie działać z większymi zbiorami uczącymi, które w miarę dołączania kolejnych cech będą ulegać zmniejszeniu. Intuicyjnie wydaje się, że procedura kolejnego dołączania cech powinna być bardziej efektywna niż procedura kolejnego odrzucania cech.

Niezależnie od zastosowanej procedury selekcji cech, liczba odrzuconych obiektów ze zbioru uczącego może się zmieniać, a więc zmianie będą także ulegać proporcje klas w uzyskiwanych zmniejszonych zbiorach uczących. Te zmiany uniemożliwiają rzetelne porównywanie kombinacji cech pod kątem oferowanych przez nie frakcji mylnych decyzji. Zanim jednak jakiekolwiek obiekty zostaną usunięte można ocenić prawdopodobieństwa występowania klas, przy założeniu, że zbiór uczący został zebrany zgodnie z częstościami występowania klas, o czym już była mowa w poprzednim rozdziale.

Jeżeli liczności klas wynoszą $m_1,m_2,\ldots, m_{nc}$, gdzie nc jest liczbą klas, a m jest liczebnością zbioru uczącego, to można oszacować prawdopodobieństwa p_k występowania klas przyjmując $p_k=m_k/m$. Zmiany proporcji liczności klas w zbiorze uczącym zakłócają: standaryzację cech, liczenie frakcji błędów, o czym była już mowa wyżej, obliczanie macierzy przekłamań, a także regułę decyzyjną, według której ma działać klasyfikator. Jednak znając faktyczne proporcje p_k liczności klas można dokonywać korekcji każdego z w/w etapów konstrukcji i oceny klasyfikatora.

Standaryzacja cech

Rozważania tego podrozdziału poświęcone zostaną standaryzacji klasycznej, która wymaga obliczenia wartości średnich cech oraz ich odchyleń standardowych, jak to wynika z relacji 1.5. Wpływ naruszenia właściwej proporcji liczności klas na wartości mv_j oraz sd_j, tj. średnich i odchyleń standardowych cech, można skorygować. W tym celu parametry standaryzacji wyrazić jako funkcje $mv_{k,j}$, $sd_{k,j}$ oraz p_k, czyli średnich i odchyleń standardowych cech obliczonych dla każdej klasy k oddzielnie, a także prawdopodobieństw p_k, znanych a priori lub wyznaczonych ze wzoru:

$$p_k=m_k/m. \tag{4.6}$$

Niech obiekty rozważanych klas mają indeksy $I(k)$, $k=1,2,3,\ldots,nc$. Zatem indeksy wszystkich obiektów zawierają się w zbiorze:

$$I=\sum_{k=1}^{nc} I(k). \tag{4.7}$$

Wartości $mv_{k,j}$ liczy się ze wzoru:

$$mv_{k,j}=(\sum_{i\in I(k)} x_{i,j})/m_k, \tag{4.8}$$

stąd

$$\sum_{i\in I(k)} x_{i,j} = m_k \circ mv_{k,j} . \tag{4.9}$$

Uwzględniając (4.9), policzenie średniej wartości *j*-tej cechy umożliwia formuła:

$$mv_j = (\sum_{i\in I} x_{i,j})/m = \sum_{k=1}^{nc} (\sum_{i\in I(k)} x_{i,j}/)m = \sum_{k=1}^{nc} (m_k/m) \circ mv_{k,j} ,$$

a więc

$$mv_j = \sum_{k=1}^{nc} p_k \circ mv_{k,j} .. \tag{4.10}$$

Nieco bardziej złożone jest obliczenie sd_j jako funkcji $sd_{k,j}$ oraz p_k, ponieważ formuła ma postać

$$sd_j^2 = \sum_{i\in I} (x_{i,j} - mv_j)^2)/m = \sum_{i\in I} (x_{i,j}^2 - 2 \circ x_{i,j} \circ mv_j + mv_j^2)/m =$$

$$= (\sum_{i\in I} x_{i,j}^2 - 2 \circ mv_j \circ \sum_{i\in I} x_{i,j} + \sum_{i\in I} mv_j^2)/m = (\sum_{i\in I} x_{i,j}^2 - 2 \circ m \circ mv_j^2 + m \circ mv_j^2)/m ,$$ czyli

$$sd_j^2 = \frac{1}{m} \circ \sum_{i\in I} x_{i,j}^2 - mv_j^2 . \tag{4.11}$$

Sumę $\sum_{i\in I} x_{i,j}^2$ we wzorze (4.11) należy wyrazić jako funkcję odchyleń standardowych liczonych dla każdej klasy oddzielnie oraz prawdopodobieństw p_k. Przez podobieństwo do wzoru (4.11) odchylenia standardowe dla poszczególnych klas wyrażają się wzorem:

$$sd_{k,j}^2 = \frac{1}{m_k} \circ \sum_{i\in I(k)} x_{i,j}^2 - mv_{k,j}^2 , \tag{4.12}$$

z którego wynika, że

$$\sum_{i\in I(k)} x_{i,j}^2 = m_k \circ sd_{k,j}^2 + m_k \circ mv_{k,j}^2 . \tag{4.13}$$

Sumę $\sum_{i\in I} x_{i,j}^2$, potrzebną by skorzystać ze wzoru (4.11) można wyrazić jako sumę sum cząstkowych określonych relacjami (4.13) uwzględniając jednocześnie relację (4.6):

$$\sum_{i\in I} x_{i,j}^2 = \sum_{k=1}^{nc} \sum_{i\in I(k)} x_{i,j}^2 = \sum_{k=1}^{nc} (m_k \circ sd_{k,j}^2 + m_k \circ mv_{k,j}^2) = \sum_{k=1}^{nc} (p_k \circ m \circ sd_{k,j}^2 + p_k \circ m \circ mv_{k,j}^2) =$$

$$= m \circ \sum_{k=1}^{nc} (p_k \circ sd_{k,j}^2 + p_k \circ mv_{k,j}^2),$$ tzn.

$$\sum_{i\in I} x_{i,j}^2 = m \circ \sum_{k=1}^{nc} p_k \circ (sd_{k,j}^2 + mv_{k,j}^2), \tag{4.14}$$

Z podstawienia (4.14) do (4.11) wynika, że

$$sd_j^2 = \frac{1}{m} \circ \sum_{i \in I} x_{i,j}^2 - mv_j^2 = \frac{1}{m} \circ m \circ \sum_{k=1}^{nc} p_k \circ (sd_{k,j}^2 + mv_{k,j}^2) - mv_j^2 = \sum_{k=1}^{nc} p_k \circ (sd_{k,j}^2 + mv_{k,j}^2) - mv_j^2 .$$

Ostatecznie,

$$sd_j = (\sum_{k=1}^{nc} p_k \circ (sd_{k,j}^2 + mv_{k,j}^2) - mv_j^2)^{1/2} . \quad (4.15)$$

Obliczanie frakcji pomyłek i macierzy przekłamań

Niech liczba błędów w wyniku klasyfikacji m_k obiektów z klasy k wynosi e_k, wtedy frakcja błędów dla klasy k wyniesie $b_k=e_k/m_k$. Łączna liczba błędów e wyniesie zatem $e=\sum_{k=1}^{nc} e_k$, a frakcja błędów dla wszystkich klas

$b=e/m=(\sum_{k=1}^{nc} e_k)/m=(b_k \circ m_k)/m$, czyli na podstawie (4.6):

$$b = \sum_{k=1}^{nc} p_k \circ b_k . \quad (4.16)$$

Z relacji (4.16) wynika, że znając frakcje błędów klasyfikacji dla poszczególnych klas można, wyznaczyć łączną frakcję błędów klasyfikacji.

Kolejnym zadaniem jest policzenie macierzy $P=\{p_{k,j}\}_{k,j=1}^{nc}$ prawdopodobieństw, że obiekt pochodzący z klasy k-tej zostanie zaliczony do klasy j-tej. Macierz tą należy wyznaczyć na podstawie macierzy $R=\{r_{k,j}\}_{k,j=1}^{nc}$, gdzie przez $r_{k,j}$ oznacza liczbę obiektów z klasy k zaliczonych klasy j. Gdyby liczebność klasy k była zgodna z założonymi udziałami tej klasy w zbiorze uczącym, to liczba obiektów $s_{k,j}$ z klasy k zaliczonych do klasy j wyniosłaby

$$s_{k,j}=(r_{k,j}/m_k) \circ l_k, \text{ gdzie } l_k = p_k \circ m. \quad (4.17)$$

Liczba $l_k=p_k \circ m$ jest spodziewaną liczbą obiektów z klasy k, gdyby udział tej klasy był zgodny ze statystyką występowania tej klasy w zbiorze uczącym. Prawdopodobieństwo $p_{k,j}$ można obliczyć jako: $p_{k,j}=s_{k,j}/l_k$, co po podstawieniu za $s_{k,j}$ oraz za l_k daje:

$$p_{k,j}=r_{k,j}/m_k. \quad (4.18)$$

Relacja (4.18) oznacza, że prawdopodobieństwa $p_{k,j}$ są obliczane tak jak gdyby częstości klas w zbiorze uczącym były zgodne z prawdopodobieństwami p_k.

Zdecydowanie bardziej złożone jest obliczanie macierzy $q_{k,j}$ prawdopodobieństw, że obiekt zaliczony do klasy k pochodzi faktycznie z klasy j. Niech s_k oznacza liczbę obiektów, jaka byłaby zaliczona do klasy k, gdyby klasy były reprezentowane w zbiorze uczącym zgodnie ze statystyką ich występowania, jak to definiują udziały p_k, k=1,2,3,…,nc, tych klas w zbiorze uczącym. Prawdopodobieństwa $q_{k,j}$ oblicza się ze

wzoru $q_{k,j}=s_{j,k}/s_k$. Liczby $s_{j,k}$ oblicza się ze wzoru (4.17), a liczbę s_k korzystając z relacji: $s_k=\sum_{j=1}^{nc} s_{j,k}=\sum_{j=1}^{nc}(r_{j,k}/m_j)\circ p_{,j}\circ m$, stąd $q_{k,j}=(r_{j,k}/m_j)\circ p_j\circ m/\sum_{j=1}^{nc}(r_{j,k}/m_j)\circ p_j\circ m$. Ostatecznie po skróceniu przez m otrzymuje się:

$$q_{k,j}=(r_{j,k}/m_j)\circ p_j/\sum_{j=1}^{nc}(r_{j,k}/m_j)\circ p_j. \tag{4.19}$$

Skorygowana reguła *k*-NS

Przedstawione dotąd sposoby rozwiązywania zadań korygowania standaryzacji cech, liczenia frakcji błędów oraz macierzy przekłamań odnoszą się do dowolnego typu klasyfikatora. Jednak szczególnie wygodnie jest je stosować w klasyfikatorach działających wg reguły *k*-NS, gdyż dla tej reguły bardzo wygodnie stosuje się metodę minus jednego elementu. Ponadto, reguła *k*-NS daje się łatwo skorygować na okoliczność zaburzenia właściwych proporcji pomiędzy częstościami klas w zbiorze uczącym.

W najmniejszej hiperkuli H_k zawierającej k najbliższych sąsiadów znajduje się pewna liczba k_i obiektów z klasy i. Frakcja k_i/m_i oznacza frakcję obiektów z klasy i, które znalazły się wśród k najbliższych sąsiadów. Gdyby udział klasy i w zbiorze uczącym był p_i, to przy liczebności zbioru uczącego równej m w klasie i byłoby nie m_i lecz $p_i \cdot m$ obiektów. Stąd, liczba obiektów w hiperkuli H_k z klasy i wyniosłaby: $q_i=(k_i/m_i)\cdot p_i \cdot m$.

Łatwo spostrzec, że mnożnik m nie ma wpływu na wybór klasy i, odpowiadającej maksymalnej wartości q_i. Zatem klasyfikowanemu obiektowi powinna być przydzielona klasa i dla której liczba

$$q_i=(k_i/m_i)\cdot p_i \tag{4.20}$$

osiąga maksimum. Tak zdefiniowana zasada działania klasyfikatora stanowi skorygowaną regułę *k*-NS.

Wymagana forma danych wejściowych

Pierwszy wiersz pliku ze zbiorem uczącym powinien zawierać 3 liczby: liczbę klas nc, liczbę cech n oraz liczbę obiektów m. Każdy z pozostałych wierszy zawiera w pierwszej kolumnie numer klasy, a w pozostałych kolumnach tego wiersza są wartości cech obiektu, jak to zostało zilustrowane w Tab. 4.2.

W miejsce brakujących wartości cech może być wpisana dowolna liczba rzeczywista. Informacja o brakujących wartościach cech zawarta jest oddzielnym zbiorze o m wierszach i n kolumnach i wartościach 0 albo 1. Liczba zero w i-tym wierszu i j-tej kolumnie oznacza brak wartości j-tej cechy dla i-tego obiektu. Taka notacja pozwala na łatwe oznaczenie, które z obiektów zbioru uczącego są dla danej kombinacji cech aktywne. Wystarczy np. wymnożyć wartości binarne odpowiedniego

wiersza z tablicy braków. Wartość zero tego iloczynu będzie oznaczać, że dany wiersz nie jest aktywny, tj. odpowiedni obiekt został chwilowo usunięty ze zbioru uczącego.

Tab. 4.2. Forma przygotowania danych (bez pierwszego wiersza zbioru uczącego)

Lp.	Klasa	Cecha 1	Cecha 2	Cecha 3	Tablica braków			Aktywność
1	1	3,4	2,1	5,3	1	1	1	1
2	1	2,3	3,1	2,5	1	1	1	1
3	1	1,0	4,4	3,2	1	1	1	1
4	2	~~5,2~~	4,5	7,4	0	1	1	0
5	2	4,8	5,5	6,7	1	1	1	1
6	1	2,7	~~4,1~~	4,0	1	0	1	0
7	1	1,9	3,9	2,7	1	1	1	1
8	2	6,3	4,7	6,0	1	1	1	1
9	2	5,0	6,4	6,5	1	1	1	1
10	2	5,8	~~6,1~~	~~6,9~~	1	0	0	0

Na Rys. 4.3. został pokazany prosty przykład dla zilustrowania sposobu liczenia frakcji błędów i macierzy przekłamań. Wprost z Rys.4.3a można ustalić, że p_1=4/8=1/2 oraz p_2=4/8=1/2.

(a)

x_1		x_2		x_3	x_4		x_5	x_6			x_7		x_8
x		**x**		**x**	**o**		**o**	**x**			**o**		**o**

(b)

Zbiór uczący	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Tablica braków	1	1	1	0	1	1	1	1

Rys. 4.3. Przykład dla ilustracji działania skorygowanej reguły 1-NS

Natomiast, Rys.4.3b informuje, że brak jest wartości cechy dla obiektu x_4. Przebieg metody minus jednego elementu jest następujący. Po usunięciu obiektu x_4 każdy z obiektów, z wyjątkiem x_5 i x_6, ma jako najbliższego sąsiada obiekt z tej samej co on klasy. To znaczy, że tylko x_5 i x_6 są mylnie klasyfikowane.

W klasie pierwszej 3 obiekty były poprawnie klasyfikowane podczas przebiegu procedury minus jednego elementu i jeden obiekt mylnie, zaś w klasie drugiej dwa obiekty były poprawnie przyporządkowane, a jeden mylnie.

Wyniki zastosowania metody minus jednego elementu zostały pokazane w Tab.4.3. Zawiera ona macierze przekłamań I, II i III rodzaju, zdefiniowane w podrozdziale 2.2.

Frakcja pomyłek dla klasy pierwszej wynosi b_1=0,25 a dla drugiej b_2=0,33, a zatem frakcja błędów dla obu klas wynosi $b=p_1 \cdot b_1+p_2 \cdot b_2=0{,}5 \cdot 0{,}25+0{,}5 \cdot 0{,}33=0{,}290$.

Tab.4.3. Macierze przekłamań dla danych z Rys.5.1 i skorygowanej reguły 1-NS

Macierz $R=\{r_{i,j}\}_{i,j}^{nc}$		
	1	2
1	3	1
2	1	2

Macierz $P=\{p_{i,j}\}_{i,j}^{nc}$		
	1	2
1	0,75	0,25
2	0,33	0,67

Macierz $Q=\{q_{i,j}\}_{i,j}^{nc}$		
	1	2
1	0,69	0,31
2	0,27	0,73

Bez przeprowadzenia korekty ze względu na brak danej, frakcja błędów wyniosłaby b=2/7=0,286 (dwie pomyłki na siedem obiektów, gdyż jeden zostałby usunięty).

Różnica ta jest bardzo mała, dlatego, że zbiór uczący jest bardzo mały i brak jest tylko jednej danej. Gdyby zrezygnowano z korekty na brak danej macierz *P* pozostałaby bez zmian. Natomiast macierz *Q*, w przypadku rezygnacji z korekty na brak danej byłaby w tym przypadku identyczna z macierzą *P*. Oznacza to, że zastosowanie skorygowanej reguły 1-NS zmienia w tym przypadku macierz *Q* znacząco.

4.4. Rozmyta reguła *k*-NS

Przynależność obiektu $\underline{x}$ do klasy *j* można zapisać jako wektor przynależności $\underline{v}_o=[0_1, 0_2,\ldots,1_j,\ldots,0_{nc}]$, gdzie *nc* jest liczbą klas, a indeks „*o*" oznacza decyzję ostrą, jako przeciwieństwo decyzji rozmytej. Przy takim zapisie już nie ma konieczności zliczania liczb *k* najbliższych sąsiadów, wystarczy obliczyć średni wektor z wektorów przynależności najbliższych sąsiadów, a następnie składowej wektora wynikowego przyjmującej największą wartość nadać wartość 1, a pozostałym składowym przypisać wartości równe zeru. Jeżeli trzem najbliższym sąsiadom $\underline{x}_{1NS}$, $\underline{x}_{2NS}$ i $\underline{x}_{3NS}$, w przypadku dwóch klas, odpowiadają wektory przynależności $\underline{v}_{1NS}=[1,0]$, $\underline{v}_{2NS}=[0,1]$ i $\underline{v}_{3NS}=[0,1]$, to średni z tych wektorów będzie już rozmyty i wyniesie $\underline{v}_R=[1/3,2/3]$, a po zmianie 2/3 na 1 i 1/3 na 0 powstanie ostry wektor przynależności $\underline{v}_O=[0,1]$. Wynik głosowania najbliższych sąsiadów mierzony jest, w takim podejściu, nie liczbami, ale frakcjami głosów oddanych na poszczególne klasy. W podanym przykładzie te frakcje wyniosły 1/3 oraz 2/3. Taka realizacja reguły *k*-NS jest ogólniejsza, bo obejmuje również przypadek, gdy przynależność obiektów w zbiorze uczącym jest opisana rozmytymi wektorami przynależności. Składowe wektorów przynależności przyjmują wartości z zakresu [0,1] i sumują się do jedności.

Jeżeli klasyfikacja dotyczy wyłącznie decyzji ostrych, to punkt przestrzeni cech, w którym wystąpił *i*-ty najbliższy sąsiad spośród *k* najbliższych sąsiadów, należy do każdej z klas *j* z nieznanym prawdopodobieństwem $p_{j,iNS}=p(j/\underline{x}_{iNS})$, $j=1,2,\ldots,nc$. Jego

przynależność można opisać rozmytym wektorem $\underline{v}_{i\mathrm{NS}}=[p_{1,i\mathrm{NS}},p_{2,i\mathrm{NS}},\ldots,p_{j,i\mathrm{NS}},\ldots,p_{nc,i\mathrm{NS}}]$. Tak więc, *i*-ty najbliższy sąsiad $\underline{v}_{i\mathrm{NS}}$ *nie powinien* całego swojego głosu oddawać na jedną z klas, ale rozdzielić swój głos pomiędzy wszystkie klasy wg udziałów wskazanych przez składowe $p_{j,iNS}$ wektora $\underline{v}_{i\mathrm{NS}}$. Składowe te nie są znane, jednak można je oszacować.

Najbliższym sąsiadem klasyfikowanego obiektu może być każdy obiekt zbioru uczącego, a zatem każdemu obiektowi $\underline{x}_i$ zbioru uczącego należy przypisać nowy rozmyty wektor przynależności $\underline{v}_i=[p_{1,i},p_{2,i},\ldots,p_{j,i},\ldots,p_{nc,i}]$. Jego składowe można oszacować wg następującej zasady.

Reguła I rozmywania przynależności

Jeżeli obiekt $\underline{x}_i$ pochodzi z klasy *l*,

to $p_{j,i}=p(j/\underline{x}_i)=k_j/(k+1)$, gdy $j\neq l$, (4.21)

oraz $p_{j,i}=p(j/\underline{x}_i)=(k_j+1)/(k+1)$, gdy $j=l$, dla $j=1,2,\ldots,nc$,

gdzie *k* jest liczbą najbliższych sąsiadów obiektu $\underline{x}_i$, z których k_j należy do klasy *j*.

Taka reguła rozmywania przynależności, wynika ze spostrzeżenia, że obiekt $\underline{x}_i$, który jest z klasy *l* oddaje głos na swoją klasę, czyli na klasę *l*. Odnosi się ona do problemu klasyfikacji ostrej ze zbiorem uczącym obiektów, których przynależność określona jest numerami klas. Reguła (4.21) może być użyta do zbioru uczącego tylko jeden raz, gdyż po jej jednokrotnym użyciu każdy obiekt $\underline{x}_i$ $i=1,2,\ldots,m$, zbioru uczącego ma przypisany nowy wektor przynależności $\underline{v}_i=[p_{1,i},p_{2,i},\ldots,p_{j,i},\ldots,p_{nc,i}]$, który na ogół jest już wektorem przynależności rozmytej. Tylko w przypadku, gdyby wszyscy jego najbliżsi sąsiedzi pochodzili z tej samej klasy, np. *j*, to miałby on postać: $\underline{v}_i=[0_1, 0_2,\ldots,1_j,\ldots,0_{nc}]$.

Jeżeli przynależność obiektów w zbiorze uczącym oznaczona jest nie poprzez numery klas, lecz wektorami przynależności postaci $\underline{v}_i^r=[p^r_{1,i},p^r_{2,i},\ldots,p^r_{j,i},\ldots,p^r_{nc,i}]$, to należy rozważyć inną podaną niżej regułę II, którą można już stosować wielokrotnie dla zbioru uczącego, stąd górny indeks wskazujący, że wektor przynależności powstał w wyniku *r* krotnego zastosowania reguły rozmywania przynależności. W przypadku klasyfikacji ostrej, dla $r=0$, wszystkie składowe wektorów przynależności są binarne, czyli dla każdego *i* wektor $\underline{v}_i=[0_1,0_2,\ldots,1_j,\ldots,0_{nc}]$, jeśli $\underline{x}_i$ jest z klasy *j*. Inaczej mówiąc, $r=0$ oznacza, że wektory $\underline{x}_i$ mają przypisaną przynależność oryginalną, jeszcze ani raz nie rozmywaną.

Reguła II rozmywania przynależności

$$\underline{v}_i^{r+1}=[p^{r+1}_{1,i},p^{r+1}_{2,i},\ldots,p^{r+1}_{j,i},\ldots,p^{r+1}_{nc,i}]=(\sum_{h=1}^{k}\underline{v}_{i,h\mathrm{NS}}+\underline{v}_i^r)/(k+1), \quad (4.22)$$

gdzie dolny indeks „*i*,*h*NS" oznacza, chodzi o wektor przynależności *h*-tego z kolei najbliższego sąsiada obiektu $\underline{x}_i$. Dzielenie przez *k*+1 w formule (4.22) wynika stąd, że wynikowy wektor przynależności $\underline{v}_i^{r+1}$ powstaje jako średni wektor przynależności z wektorów przynależności swoich *k* sąsiadów oraz jego samego.

Problemami, które należy rozwiązać są: wybór liczby *k* najbliższych sąsiadów oraz kryterium jej wyboru. Reguła *k*-NS ma działać ze zbiorem uczącym, którego obiekty będą miały nowe wektory przynależności. Naturalną wydaje się być hipoteza, że składowe wektorów $\underline{v}_i$ powinny przyjąć takie wartości, aby uzyskiwane na ich podstawie frakcje błędów były jak najmniejsze. Stąd prosty wniosek, że jako *k* należy przyjąć taką wartość, dla której frakcja błędów, wyznaczona, np. metodą minus jednego elementu jest najmniejsza. Tak wybrana liczba *k* określi już jednoznacznie składowe rozmytych wektorów przynależności.

Skoro wymieniony wyżej wektor $\underline{v}_i$ przynależności *i*-tego obiektu zbioru uczącego ma *nc* składowych, a zbiór uczący ma *m* obiektów, to z wektorów tych można utworzyć macierz przynależności zbioru uczącego W_0 zawierającą *m* wierszy i *nc* kolumn. Na zbiór uczący mogą składać się obiekty z przypisanymi im rozmytymi wektorami przynależności, których składowe mogą być mniejsze niż 1. Wówczas błąd pojedynczej klasyfikacji liczy się wg wzoru 1.4 podanego w podrozdziale 1. Warto zaznaczyć, że składowe rozmytych wektorów przynależności nie muszą mieć interpretacji prawdopodobieństw przynależności do klas. Mogą tymi składowymi być ogólnie rozumiane stopnie przynależności do klas, np. udziały różnych metali w stopie, gdyby celem klasyfikacji było rozpoznawanie zawartości stopów. W dalszych rozważaniach domyślnie będzie przyjmowane, że składowymi wektora przynależności będą stopnie przynależności obiektu do poszczególnych klas.

Jeżeli przedmiotem zainteresowania jest klasyfikacja z decyzjami ostrymi, obiekty w zbiorze uczącym mają przynależności określone numerami klas, to po opisaniu ich wektorami przynależności postaci $\underline{v}_o=[0_1,0_2,\ldots,0_{nc}]$ powstanie w wyniku binarna macierz przynależności W_0. Na podstawie II reguły rozmywania przynależności określonej relacją (4.22) i metody minus jednego elementu (bądź innej metody obliczania frakcji błędnych decyzji) można utworzyć nieskończony ciąg trójek:

$$(W_0,k_0,e_0),(W_1,k_1,e_1),(W_2,k_2,e_2),\ldots,(W_r,k_r,e_r),(W_{r+1},k_{r+1},e_{r+1})\ldots \qquad (4.23)$$

W ciągu tym k_r jest liczbą najbliższych sąsiadów dla której klasyfikator działający wg reguły *k*-NS z macierzą W_r osiąga najmniejszą frakcję e_r błędów klasyfikacji. Macierz przynależności W_1, w przypadku, gdy klasy obiektów zbioru uczącego są oznaczone numerami klas, należy wyznaczyć stosując I regułę (4.21) rozmywania przynależności. Natomiast, w pozostałych przypadkach, do wyznaczana macierzy W_{r+1} używana jest II reguła (4.22) rozmywania przynależności. Stosowania reguły I można uniknąć, jeśli

oznaczenie przynależności obiektów zbioru uczącego numerami klas zostanie przekształcone w binarne wektory przynależności, a z nich utworzona zostanie macierz W_0.

Tworzenie ciągu (4.23) kontynuowane jest tak długo jak długo $e_{r+1} < e_r$. Z chwilą, gdy $e_{r+1} \geq e_r$, generowanie ciągu $(W_r, k_r, e_r)_{r=0}^{\infty}$ zostaje przerwane. Ostatecznie klasyfikator działać będzie z macierzą przynależności W_r oraz regułą decyzyjną k_r-NS. Relacja (4.23) definiuje schemat uczenia rozmytej wersji reguły k-NS [Jóźwik A., 1983b]. Algorytm ten był szczegółowo analizowany przez Bezdeka i współautorów [Bezdek J.C., Chuah S.K., Leep D. ,1986].

Sposób wyznaczania macierzy przynależności W_1, zostanie zilustrowany na jednowymiarowym zbiorze dotyczącym dwóch klas, pokazanym na Rys.4.4.

Przykład 1

Podany niżej przykład dotyczy 10 obiektów, z których 5 oznaczonych krzyżykami należy do klasy 1, a drugie 5 oznaczonych kółeczkami do klasy 2.

Wartość cechy	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Klasa obiektu	x	x	x			x	o		x		O	o	o	o
Obiekt	$\underline{x}_1$	$\underline{x}_2$	$\underline{x}_3$			$\underline{x}_4$	$\underline{x}_5$		$\underline{x}_6$		$\underline{X}_7$	$\underline{x}_8$	$\underline{x}_9$	$\underline{x}_{10}$

Rys. 4.4. Dane dla ilustracji działania I reguły rozmywania przynależności.

Zastosowanie metody minus jednego elementu dla klasyfikatora działającego wg reguły 1-NS daje w wyniku trzy pomyłki. Mylnie rozpoznanymi będą obiekty $\underline{x}_4$, $\underline{x}_5$ oraz $\underline{x}_6$, bo każdy z tych obiektów ma w charakterze swojego najbliższego sąsiada obiekt z przeciwnej klasy. Zaś dla reguły 2-NS oraz 3-NS obiekt $\underline{x}_4$ będzie już poprawnie klasyfikowany. W klasyfikacji tego obiektu nie ma żadnej różnicy pomiędzy regułami 2-NS i 3-NS, gdyż w głosowaniu biorą udział te same obiekty jako najbliżsi sąsiedzi. Drugi i trzeci najbliższy sąsiad tego obiektu znajdują się od $\underline{x}_4$ w tej samej odległości, więc i tak w regule 2-NS bierze lokalnie udział trzech najbliższych sąsiadów.

Za optymalną liczbę najbliższych sąsiadów można przyjąć k=3. Nieparzysta liczba k w zadaniach dwu-decyzyjnych jest niezawodnym zabezpieczeniem przed niejednoznacznymi wynikami głosowań k sąsiadów. Po ustaleniu liczby najbliższych sąsiadów można zastosować I lub II regułę rozmywania przynależności.

Obiekt $\underline{x}_4$ ma w swoim otoczeniu trzech najbliższych sąsiadów, obiekty $\underline{x}_3$, $\underline{x}_5$ i $\underline{x}_6$, stąd k_1=2, k_2=1 i należy do klasy l=1. Zatem zgodnie z regułą I, tzn. relacją (4.21)

$p_{1,4}=p(1/\underline{x}_4)=(k_1+1)/(k+1)=3/4$, a $p_{2,4}=p(2/\underline{x}_4)=k_2/(k+1)=1/4$.

Jednak skorzystanie z reguły II wydaje się być wygodniejsze, bo zgodnie z (4.22):

$\underline{v}_4^1=[p_{1,4}^1,p_{2,4}^1]=\underline{v}_{4,1NS}+\underline{v}_{4,2NS}+\underline{v}_{4,3NS}+\underline{v}_4^0=([1,0]+[0,1]+[1,0]+[1,0])/4=[3/4,1/4]$.

Indeks górny przy wektorach, w przypadku II reguły wprowadzono dlatego, że reguła ta może być użyta wielokrotnie, jak podano poprzednio.

W podobny sposób można wyznaczyć nowe wektory przynależności dla pozostałych 9 obiektów, które pokazane zostały w Tab.4.4.

Tab.4.4. Ilustracja rozmywania ostrej macierzy przynależności

Macierz przynależności W_0	
01: [1, 0]	06: [1, 0]
02: [1, 0]	07: [0, 1]
03: [1, 0]	08: [0, 1]
04: [1, 0]	09: [0, 1]
05: [0, 1]	10: [0, 1]

Macierz przynależności W_1	
01: [4/4, 0/4]	06: [2/5, 3/5]
02: [4/4, 0/4]	07: [1/4, 3/4]
03: [4/4, 0/4]	08: [0/4, 4/4]
04: [3/4, 1/4]	09: [0/4, 4/4]
05: [3/5, 2/5]	10: [0/4, 4/4]

W przedstawionym przykładzie w głosowaniu brały udział najczęściej 4 obiekty, tzn. obiekt, dla którego nowy wektor przynależności był tworzony oraz trzech jego najbliższych sąsiadów. Wyjątek wystąpił dla obiektów $\underline{x}_5$ i $\underline{x}_6$, dla których trzeci najbliższy sąsiad znajdował się w tej samej odległości, co i czwarty.

W roku 1985 pojawiły się w piśmiennictwie propozycje 3 innych rozmytych reguł *k*-NS zaproponowane przez Kellera i wsp. [Keller J.M., Gray M.R., Givens J.A., 1985], z których druga z opisanych w w/w pracy, odnosi się wyłącznie do dwu-decyzyjnego zadania klasyfikacji i dlatego zostanie pominięta w niniejsze monografii. Pozostałe dwie metody różnią się od siebie sposobami określenia inicjujących stopni przynależności $p_{j,i}$ obiektów $\underline{x}_i$ ze zbioru uczącego do każdej z klas j.

Pierwsza reguł korzysta wprost z macierzy binarnej W_0, czyli $p_{j,i}=1$, gdy obiekt $\underline{x}_i$ należy do klasy j oraz $p_{j,i}=0$, gdy do niej nie należy. Druga z reguł, opisana relacją (4.24), inicjuje już rozmytą macierz przynależności obiektów $\underline{x}_i$ ze zbioru uczącego. Składowe wektorów składających się na macierz W_0, tym razem nie binarnej, liczone są wg formuły:

$$p_{j,l}=\frac{k_j}{k}\circ 0{,}49+\delta_{j,l}\circ 0{,}51, \tag{4.24}$$

przy czym $\delta_{j,l}$ przyjmuje wartość 1, gdy $j=l$ i wartość 0, gdy $j\neq l$, gdzie l jest klasą obiektu $\underline{x}_i$. Stopnie przynależności nowo klasyfikowanych obiektów wyznaczane są wg dość złożonej reguły:

$$p_{j,\underline{x}}=[\sum_{h=1}^{k} p_{j,h\mathrm{NS}}\circ(1/d(\underline{x},\underline{x}_{h\mathrm{NS}})^{2/(l-1)}]/[\sum_{h=1}^{k}(1/d(\underline{x},\underline{x}_{h\mathrm{NS}})^{2/(l-1)}], \tag{4.25}$$

gdzie $p_{j,h\mathrm{NS}}$ jest stopniem przynależności h-tego najbliższego sąsiada klasyfikowanego obiektu $\underline{x}$ do klasy j, $d(\underline{x},\underline{x}_{h\mathrm{NS}})$ jest odległością pomiędzy obiektem $\underline{x}$ a jego h-tym najbliższym sąsiadem, k jest zadaną liczbą najbliższych sąsiadów oraz l jest parametrem reguły (4.24) większym od jedności.

Wartości stopni przynależności $p_{j,h\mathrm{NS}}$ najbliższych sąsiadów, występujące we wzorze (4.25), są elementami macierzy W_0, niezależnie od tego czy jej elementy są binarne, czy też określone zostały relacją (4.24).

Przykład 2

Dla lepszego zrozumienia wzoru (4.24) stopnie przynależności obiektu $\underline{x}_5$, z przykładu 1, zostaną teraz obliczone z tego wzoru, zakładając k=3. W tej samej odległości od obiektu $\underline{x}_5$, co jego trzeci najbliższy sąsiad, znajdują się dwa obiekty i to z przeciwnych klas, tzn. obiekt $\underline{x}_3$ oraz obiekt $\underline{x}_7$, oznacza to, że należy zastosować regułę 4-NS. Wśród tych 4 najbliższych sąsiadów k_1=3 jest z klasy 1 (krzyżyki) i k_2=1 z klasy 2 (kółka). Obiekt $\underline{x}_5$ jest z klasy 2, a zatem l=2. Według formuły (4.24) obiekt $\underline{x}_5$ otrzymuje nowe stopnie przynależności:

$p_{1,2}=\frac{3}{4}\circ 0{,}49+0\circ 0{,}51=0{,}3675$, $p_{2,2}=\frac{1}{4}\circ 0{,}49+1\circ 0{,}51=0{,}6325$, tzn. $\underline{\nu}_5=[0{,}3675;0{,}6325]$,

co oznacza, że przeważa przynależność do klasy 2. W przykładzie 1, jak wynika to z Tab. 4.4. obiektowi $\underline{x}_5$ przypisany został inny rozmyty wektor przynależność, czyli $\underline{\nu}_5=[0{,}6;0{,}4]$, tzn. większy jest stopień przynależności do klasy 1, chociaż obiekt pochodzi faktycznie z klasy 2. Formuła 4.24 zawsze nada obiektowi większy stopień przynależności do tej klasy, z której on faktycznie pochodzi, natomiast formuły (4.21) i (4.22), odnoszące się do rozmytej reguły k-NS zaproponowanej przez autora niniejszej monografii [Jóźwik A., 1983b], mogą nadać obiektowi najwyższy stopień przynależności do klasy przeciwnej niż klasa, z której on pochodzi. Zdaniem autora niniejszej monografii jest to poważna wada rozmytej reguły k-NS opisanej w pracy Kellera i wsp. [Keller J.M., Gray M.R., Givens J.A., 1985], gdyż pojedynczy punkt reprezentujący dowolną jedną klasę, nawet otoczony bardzo gęsto punktami reprezentującymi inną klasę, będzie miał zawsze przypisany wyższy stopień przynależności do klasy, z której pochodzi.

Ponieważ we wzorze (4.25) funkcje odległości występują jako swoje odwrotności, to bliższy sąsiad ma większy wpływ na wartość $p_{j,\underline{x}}$. Wraz ze wzrostem parametru l odległości od najbliższych sąsiadów mają mniejszy wpływ. W miarę jak parametr l będzie zbliżał się do 1, bliżsi sąsiedzi będą mieli coraz to większy wpływ na wartość $p_{j,h\mathrm{NS}}$. Niestety, w wymienionej publikacji Kellera i wsp. nie rozważono sytuacji, gdy klasyfikowany obiekt pokryje się z którymś z najbliższych sąsiadów. Odległość $d(\underline{x},\underline{x}_{h\mathrm{NS}})$ przyjmie wówczas wartość zero i zastosowanie formuły (4.24) będzie

niemożliwe. Wykluczenie takiego najbliższego sąsiada byłoby zafałszowaniem sytuacji. W skrajnym przypadku, wszyscy najbliżsi sąsiedzi mogą pokryć się z klasyfikowanym obiektem.

4.5. Klasyfikacja z wykorzystaniem obszarów klas

W niektórych zastosowaniach oszacowane frakcje błędów, bez względu na typ proponowanego klasyfikatora, mogą być na tyle duże, że konstrukcja klasyfikatora traci sens. Wyjściem z tej sytuacji może być wyznaczenie z przestrzeni cech obszarów klas, a co za tym idzie i obszarów pokrywania się klas. Granic tych obszarów nie trzeba opisywać analitycznie. Wystarczy posiadać algorytm, który pozwoli stwierdzić, czy dany obiekt należy do obszaru wskazanej klasy. Jeżeli będzie należał jednocześnie do dwóch lub większej liczby klas, to znaczy, że należy on do obszaru pokrywania się tych klas. Klasyfikator może podejmować decyzje tylko wówczas, gdy obiekt znajdzie się dokładnie w obszarze tylko jednej klasy. Dla potrzeb tego rozdziału wygodniej będzie zmienić dotychczasowe symbole oznaczające zbiór uczący i jego podzbiory reprezentujące poszczególne klasy.

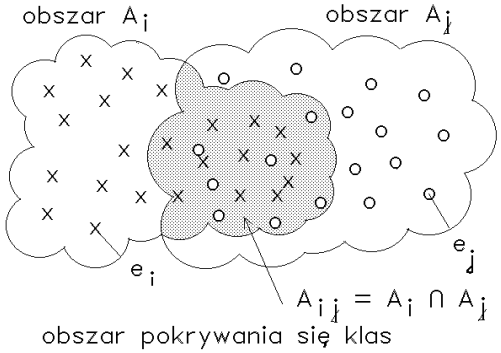
Niech rozłączne zbiory $U_1,U_2,\ldots,U_{nc}$ stanowią podział zbioru uczącego, a ich suma mnogościowa daje w wyniku zbiór uczący U. Dla podzbiorów tych można zdefiniować w następujący sposób pewne liczby e_i, i obszary A_i :

$$e_i = \max_{u_j \in U_i} d(U_i - \{u_j\}, \{u_j\}), \tag{4.26}$$

$$A_i = \{x: d(U_i, \{x\}) \leq e_i\}. \tag{4.27}$$

gdzie $d(\circ,\circ)$ jest funkcją odległości pomiędzy dwoma zbiorami, rozumianą jako odległość pomiędzy najbliższymi punktami, z których jeden należy do jednego, a drugi do drugiego zbioru. Łatwo zauważyć, że e_i jest największą odległością pomiędzy obiektem z klasy i a najbliższym mu sąsiadem z tej samej klasy. Zaś obszar A_i to miejsce geometryczne punktów odległych od U_i co najwyżej o e_i. Przykład ilustrujący obszary klas przedstawia Rys. 4.5.

Klasyfikator zdefiniowany za pomocą w/w zaproponowanych obszarów klas można scharakteryzować podając dwie frakcje: frakcję mylnych decyzji i frakcję braku decyzji (odpowiedź: *nie wiem*). Decyzja poprawna będzie miała miejsce tylko wtedy, gdy obiekt będzie z klasy i oraz znajdzie się w obszarze tej klasy. Mylna zaś zdarzy się tylko wówczas, gdy obiekt będzie z klasy i a wpadnie do obszaru tylko jednej ale innej klasy j, $i \neq j$. Jeżeli obiekt znajdzie się w obszarze $A_{i,j}$ pokrywania się klas lub poza obszarem każdej z klas, to odpowiedzią klasyfikatora będzie decyzja *nie wiem*.

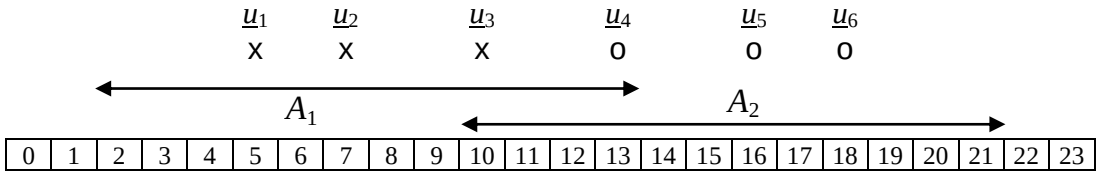


Rys. 4.5. Ilustracja obszarów klas dla metryki euklidesowej i dwóch klas

Istnieje kilka rozwiązań umożliwiających zmniejszenie liczby decyzji typu *nie wiem*. Jednym z nich jest zastosowanie, dla obiektów z obszaru pokrywania się klas, reguły *k*-NS. Jednak lepszym rozwiązaniem jest przeprowadzenie dwu etapowej selekcji cech. W pierwszym etapie jest wykonywana selekcja cech wykorzystująca, jako kryterium, frakcję pokrywania się klas. Następnie, dla obiektów z uzyskanego w ten sposób obszaru pokrywania się klas, przeprowadzana jest druga selekcja cech z frakcją błędów jako kryterium dla *k*-NS. Takie podejście zostało zastosowane do rozpoznawania zdjęć lotniczych [Jóźwik A., Sernico S., Roli F. 1998)].

Poniżej zostanie przytoczony prosty przykład, pokazany na Rys. 4.6, ilustrujący wyznaczanie obszarów klas oraz zastosowanie metody minus jednego elementu.

Przykład



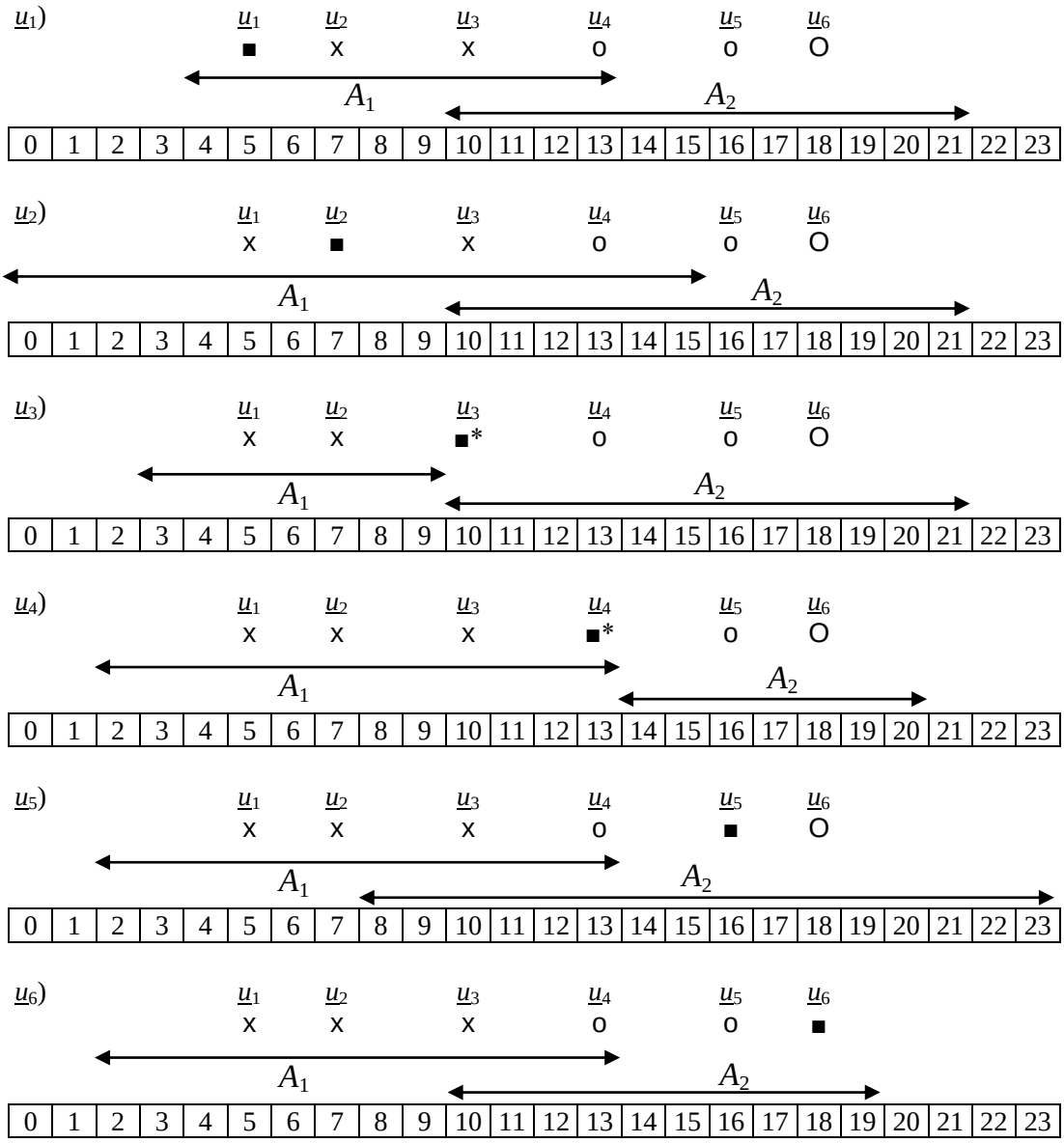
Rys 4.6. Przykład dwóch klas z zaznaczonymi obszarami A_1 i A_2.

Obiekty pokazane na Rys. 4.6 dla wygody opisane są cechami, których wartości są liczbami całkowitymi. W klasie 1 (krzyżyki) największą odległość do najbliższego sąsiada ma obiekt $\underline{u}_3$ i wynosi ona $e_1=d(\underline{u}_3,\underline{u}_2)=|10-7|=3$.

Podobnie, dla klasy 2 (kółeczka) największa odległość od obiektu do jego najbliższego sąsiada równa się $e_2=d(\underline{u}_4,\underline{u}_5)=|16-13|=3$. Liczby e_1 i e_2 spełniają warunek (4.26). Łatwo teraz wyznaczyć $A_1=[2,13]$, $A_2=[10,21]$ oraz obszar $A_{1,2}=[10,13]$.

Frakcja mylnych decyzji oraz frakcja odpowiedzi typu *nie wiem* oszacowana jest metodą minus jednego elementu, a sposób szacowania wyjaśnia Rys. 4.7. Natomiast, szczegółowe wyniki uzyskane metodą minus jednego elementu zawiera Tab. 4.5.

W metodzie minus jednego elementu kolejno zabierany jest ze zbioru uczącego obiekt $\underline{u}_i$ i klasyfikowany na podstawie obszarów klas wyznaczonych dla zbioru pozostałych obiektów, czyli dla zbioru U-$\{\underline{u}_i\}$.



Rys. 4.7. Sposób obliczania frakcji mylnych decyzji i frakcji odpowiedzi *nie wiem* metodą minus jednego elementu

W podanym przykładzie, mylnie zaklasyfikowanymi są dwa obiekty zaznaczone gwiazdką, tzn. obiekty $\underline{u}_3$ oraz $\underline{u}_4$. Obiekty te znalazły się blisko granicy pomiędzy klasami i każdy z nich wpadł w obszar klasy przeciwnej. Żaden z sześciu obiektów nie znalazł się poza obszarami klas, ani też żaden obiekt nie wpadł w obszar pokrywania się klas.

Liczby e_i, $i=1,2\ldots,nc$, nie muszą być w trakcie wykonywania metody przebiegu metody minus jednego elementu wyznaczane każdorazowo przed każdą klasyfikacją kolejnego obiektu wg formuły (4.26). Mogą być ustalone jednorazowo na podstawie

całego zbioru uczącego. Oczywiście interpretacja uzyskanych wyników byłaby wtedy inna.

Tab. 4.5. Wyniki przebiegu metody minus jednego elementu dla przykładu z Rys. 4.7

Obiekt	Klasa	E_1	e_2	A_1	A_2	$A_{1,2}$	Klasyfikacja
$\underline{u}_1$=[5]	1	3	3	[4,13]	[10,21]	[10,13]	1
$\underline{u}_2$=[7]	1	5	3	[0,15]	[10,21]	[10,15]	1
$\underline{u}_3$=[10]	1	2	3	[3,9]	[10,21]	∅	2
$\underline{u}_4$=[13]	2	3	2	[2,13]	[14,20]	∅	1
$\underline{u}_5$=[16]	2	3	5	[2,13]	[8,23]	[8,13]	2
$\underline{u}_6$=[18]	2	3	3	[2,13]	[10,19]	[10,13]	2

Należy się spodziewać, że w miarę wzrostu liczności zbioru uczącego, wartości liczb e_i będą raczej zmniejszać się. Wystąpienie obiektów odstających (szumów) może spowodować, że zastosowanie relacji 4.26 dla wyznaczenia liczb e_i straci sens, gdyż obszary pokrywania się klas mogą zawierać zbyt dużą liczbę obiektów ze zbioru uczącego. W takim przypadku korzystniejszy może być eksperymentalny dobór tych liczb.

Warto zauważyć, że podczas przebiegu metody minus jednego elementu zmianie ulega obszar tylko tej klasy, z której zabierany jest obiekt przeznaczony do klasyfikacji.

Idea obszarów klas może być wykorzystana do konstrukcji klasyfikatora wieloetapowego. Obiekty z obszaru pokrywania się klas w etapie z indeksem i mogą stanowić zbiór uczący dla etapu z numerem (i+1).

Sposób działania takiego klasyfikatora został wyjaśniony na przykładzie, w którym użyty został zbiór Iris Data, a wyniki podane zostały w Tab. 4.6.

W pierwszym etapie, kolumna I, obszar pokrywania się klas zawierał 46 obiektów, poza wszystkimi obszarami znalazły się 3 obiekty, żaden obiekt nie został mylnie zaklasyfikowany, a 101 zostało poprawnie zaklasyfikowanych. Obiekty z obszaru pokrywania się klas w pierwszym etapie, których było 46, stanowiły zbiór uczący drugiego etapu. W drugim etapie, kolumna II, obszar pokrywania się klas zawierał 22 obiekty, jeden obiekt znalazł się poza obszarami obu klas, żaden nie był mylnie zaklasyfikowany, a 23 obiekty zostały poprawnie zaklasyfikowane. Zawartości pozostałych kolumn, których numery są zarazem numerami etapów, interpretuje się podobnie.

Etapów było tyle ile tylko było to możliwe obliczeniowo, choć z praktycznego punktu widzenia sensowność zastosowania więcej niż trzech etapów w tym zadaniu jest co najmniej wątpliwa.

Tab. 4.6. Wyniki klasyfikacji wieloetapowej dla zbioru Iris Data.

Opis sytuacji – wiersze/numer etapu – kolumny	I	II	III	IV	V	Razem
Liczba punktów w zbiorze uczącym	150	46	22	12	6	-
Liczba punktów w obszarach pokrywania klas razem	46	22	12	6	0	-
Liczba punktów z decyzją *nie wiem*	3	1	0	2	0	6
Liczba punktów źle zaklasyfikowanych	0	0	1	1	1	3
Liczba punktów poprawnie zaklasyfikowanych	101	23	9	3	5	141

Klasyfikator wielostopniowy

Klasyfikowany obiekt może znaleźć się w różnym położeniu w odniesieniu do obszarów klas wyznaczanych na poszczególnych etapach. W zależności od tego położenia podejmowane są etapowe decyzje, wg następującego schematu.

1. Klasyfikowany obiekt $\underline{x}$ nie leży w żadnym z obszarów A_i. Wtedy decyzja brzmi: *nie wiem*. Obszarów A_i może być mniej niż liczba *nc* rozpatrywanych klas. Niektóre z nich lub wszystkie mogą być w szczególności miary zero, tj. zawierać jeden punkt lub kilka punktów pokrywających się. Z matematycznego punktu widzenia nie byłyby one obszarami.
2. Klasyfikowany obiekt leży dokładnie w jednym z obszarów A_i. Punkt zostaje przyporządkowany do klasy *i*.
3. Na aktualnym etapie obszar pokrywania się klas, w którym leży klasyfikowany obiekt, zawiera dokładnie te same obiekty, co odpowiedni obszar pokrywania się klas z poprzedniego etapu. Obiektowi przyporządkowana zostaje wtedy do wyboru:
   a) Klasa najliczniej reprezentowana w aktualnie rozpatrywanym obszarze pokrywania się klas, a gdy ten krok nie rozstrzygnie, to według tej samej zasady dla obszaru pokrywania się z etapu o dwa stopnie wcześniejszego (bo etap poprzedni zawiera te same obiekty) i ewentualnie bierzemy odpowiedni obszar z etapu o trzy stopnie wcześniejszego, itd. kończąc ewentualnie na całym pierwotnym zbiorze uczącym. W razie nierozstrzygnięcia podejmowana jest decyzja losowa.
   b) Klasa wytypowana z użyciem reguły *k*-NS.
4. Obiekt należy do obszaru pokrywania się *ncp* klas i zawartość tego obszaru różni się od zawartości odpowiedniego obszaru z poprzedniego etapu. Punkty tego obszaru tworzą wówczas zbiór uczący dla następnego etapu. Konstruowane są nowe obszary A_i , jest ich tym razem *ncp*. Liczba *ncp* może być mniejsza niż liczba *nc* rozważanych klas. W obszarze pokrywania się klas, na aktualnym etapie, mogą nie być reprezentowane wszystkie klasy. Dla klasyfikowanego obiektu będą więc kolejno rozpatrywane, od początku, przypadki 1, 2, 3 i obecny 4.

Definicja algorytmu (nieznacznie zmieniona) i przykłady ilustrujące jego działanie pochodzą z publikacji [Jóźwik A. i Stawska Z., 2000].

4.6. Redukcja zbiorów odniesienia

Zbiór obiektów, który musi być pamiętany w fazie klasyfikacji nazywany jest zbiorem odniesienia. Może nim być podzbiór zbioru uczącego lub nowy zbiór złożony z punktów sztucznie wygenerowanych ze zbioru uczącego. W przypadku klasyfikatora minimalno-odległościowego zbiorem odniesienia jest najczęściej zbiór *nc* środków ciężkości lub środków medianowych rozważanych klas, o czym była już mowa. Klasyfikatory wykorzystujące regułę *k*-NS wymagają przechowywania w pamięci komputera całego zbioru uczącego jako zbioru odniesienia. Wielkość zbioru odniesienia ma istotny wpływ na szybkość klasyfikacji.

Klasyfikator może być na przykład zastosowany do analizy obrazów optycznych, w których rozpoznawanymi obiektami są piksele, opisane cechami wydzielonymi na przykład na podstawie stopni szarości pikseli z sąsiedztwa [Jóźwik A., Kieś P., 2005]. Liczebność zbiorów uczących może wówczas wynosić rzędu kilkadziesiąt i więcej tysięcy. Szybkość klasyfikacji, z użyciem reguły *k*-NS, mogłaby wtedy nie być akceptowalna. Wyjściem z sytuacji mogłoby być w takim przypadku zastosowanie klasyfikatora minimalno-odległościowego lub klasyfikatora zaproponowanego w podrozdziale 2.8, skonstruowanego na podstawie zbioru uczącego edytowanego dla liniowej rozdzielności i wykorzystaniu struktury równoległej z Rys. 1.2. Jakość klasyfikacji w tych rozwiązaniach może być jednak nie wystarczająca.

Innym rozwiązaniem może być zastosowanie klasyfikatora najbliższego sąsiada, jako szczególnego przypadku użycia reguły *k*-NS dla $k=1$. Klasyfikator ten też wymaga standardowo pamiętania całego zbioru uczącego w charakterze zbioru odniesienia, ale działa znacznie szybciej niż klasyfikator *k*-NS dla $k>1$, jednak i tak nie będzie on wystarczająco szybko działał dla zbiorów zawierających wiele dziesiątek czy setek tysięcy obiektów. Dlatego dla reguły 1-NS opracowano liczne algorytmy redukcji i kondensacji zbioru odniesienia. Ponadto, regułę *k*-NS można aproksymować regułą 1-NS. Aby dokonać tej aproksymacji należy binarną macierz przynależności W_0, stosując jedną z formuł 4.21 lub 4.22, przekształcić w rozmytą macierz przynależności W_1, a następnie każdy ze składających się na nią rozmytych wektorów przynależności przekształcić w numer klasy odpowiadający jego największej składowej. Na przykład, obiekt $\underline{x}_5$ z Rys. 4.4 zmieniłby wówczas przynależność z klasy 2 do klasy 1, jak wynika z Tab. 4.4.

Innym rozwiązaniem mogłoby być opracowanie algorytmu szybkiego wyznaczania najbliższego sąsiada, co zostało zaproponowane, np. w rozprawie [Grabowski S., 2003].

Przedmiotem dalszych rozważań będą zatem algorytmy redukcji i kondensacji zbioru odniesienia dla reguły 1-NS.

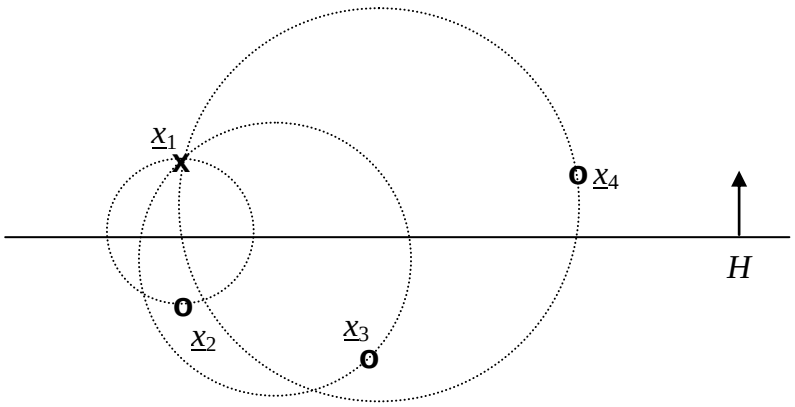
Algorytm Harta

Metoda Harta [Hart P.E., 1968] działa w następujący sposób. Pierwszy obiekt $\underline{u}_1$ z oryginalnego zbioru odniesienia O, tj. zbioru uczącego U, $O=U=\{\underline{u}_i\}_{i=1}^{m}$, kwalifikowany jest do zredukowanego zbioru odniesienia Z. Kolejny obiekt $\underline{u}_i$ klasyfikowany jest zgodnie z regułą 1-NS działającą z aktualnym zbiorem zredukowanym Z. Jeżeli zostanie on mylnie zaklasyfikowany, to zostaje on dołączony do aktualnego zbioru zredukowanego, czyli $Z:=Z\cup\{\underline{u}_i\}$. Po klasyfikacji ostatniego z obiektów zbioru O ponownie kolejno klasyfikowane są w ten sposób obiekty zbioru O, od obiektu $\underline{u}_1$ do obiektu $\underline{u}_m$. Taki cykl klasyfikacji powtarzany jest tak długo, aż kolejnych m tego typu klasyfikacji będzie poprawnych. Uzyskany w ten sposób zbiór Z jest wynikowym zredukowanym zbiorem odniesienia.

Algorytm ten jest określany jako zgodny (z oryginalnym zbiorem odniesienia), gdyż reguła 1-NS działająca z nim jako zbiorem odniesienia prawidłowo przydziela klasy obiektom ze zbioru O. Wadą tego algorytmu jest rekrutowanie do zbioru zredukowanego, w początkowej jego fazie, obiektów leżących daleko od granic klas. Obiekty takie prawdopodobnie są nadmiarowymi. Dlatego [Gates G. W., 1972] zaproponowano usuwanie takich obiektów. Jeśli usunięcie obiektu ze zredukowanego zbioru odniesienia nie naruszy zgodności tego zbioru ze zbiorem oryginalnym, to taki obiekt jest trwale z niego usunięty. Innym rozwiązaniem, sugerowanym przez autora niniejszej monografii jest powtórzenie procedury Harta, ale tym razem z modyfikacją polegająca na tym, że najpierw prezentowane są obiekty ze zbioru Z w kolejności odwrotnej do tej, w której były one do niego rekrutowane, a następnie jest dokonywana prezentacja pozostałych obiektów, czyli obiektów ze zbioru O-Z.

Algorytm Tomeka

W publikacji [Tomek I., 1976] zostały zaproponowane dwa algorytmy redukcji zbiorów odniesienia, jednak tylko jeden z nich zyskał zainteresowanie czytelników. Algorytm ten działa w następujący sposób. Dla każdej pary obiektów $\underline{x}$ i $\underline{y}$ należących do różnych klas konstruowana jest hiperkula o środku w punkcie $(\underline{x}+\underline{y})/2$ i promieniu $r=d(\underline{x},\underline{y})/2$, gdzie $d(\circ,\circ)$ jest metryką euklidesową. Jest to hiperkula rozpięta na punktach $\underline{x}$ oraz $\underline{y}$, której średnicą jest odcinek łączący $\underline{x}$ z $\underline{y}$. Jeżeli wnętrze tej hiperkuli nie zawiera żadnego obiektu ze zbioru odniesienia O, to oba obiekty $\underline{x}$ oraz $\underline{y}$ wchodzą do zredukowanego zbioru odniesienia Z. Autor algorytmu podał twierdzenie mówiące o tym, że tak otrzymany zbiór zredukowany jest zgodny, ale okazało się one fałszywe [Toussaint G.T., 1994], co pokazuje Rys 4.8.



Rys. 4.8. Kontrprzykład dla twierdzenia Tomeka

Obiekty $\underline{x}_1$ i $\underline{x}_2$ tworzą parę tomekową, bo wnętrze rozpiętej na nich kuli jest nie zawiera żadnego obiektu. Zaś kula rozpięta na obiektach $\underline{x}_1$ i $\underline{x}_3$ ma w swym wnętrzu obiekt $\underline{x}_2$, a kula rozpięta na obiektach $\underline{x}_1$ i $\underline{x}_4$ zawiera w swym wnętrzu obiekt $\underline{x}_3$. Zbiór zredukowany zawiera więc tylko obiekty $\underline{x}_1$ i $\underline{x}_2$. Obiektom znajdującym się powyżej prostej H bliżej jest do obiektu $\underline{x}_1$ niż do obiektu $\underline{x}_2$. Obiekt $\underline{x}_4$ będzie więc mylnie klasyfikowany i psuje zgodność zbioru zredukowanego. Dołączenie do zbioru zredukowanego obiektu $\underline{x}_4$ uczyni go zbiorem zgodnym.

Wymienioną wadę algorytmu łatwo jest naprawić. Wystarczy dołączyć dodatkowe obiekty posługując się algorytmem Harta. Niektóre pary obiektów w zbiorze zredukowanym mogą okazać się zbędne.

Modyfikacja Gowdy i Krishny

Ciekawa propozycja modyfikacji algorytmu Harta została zaproponowana kilka lat później w publikacji [Gowda K.C. and Krishna G., 1979]. Modyfikację tą można znacznie prościej opisać niż uczynili to autorzy wyżej wymienionej pracy. W tym celu warto wprowadzić pojęcie miary pozycyjnej $mp(\underline{x})$ obiektu $\underline{x}$. Niech $\underline{y}$ będzie najbliższym sąsiadem obiektu $\underline{x}$ pochodzącym z przeciwnej. Liczba obiektów z tej samej klasy co $\underline{x}$ znajdujących się nie dalej od obiektu $\underline{y}$ niż obiekt $\underline{x}$ jest wartością miary pozycyjnej obiektu $\underline{x}$. Inaczej mówiąc, jeżeli $\underline{x}$ jest k-tym najbliższym sąsiadem obiektu $\underline{y}$, spośród obiektów tej klasy, z której pochodzi $\underline{x}$, to $mp(\underline{x})=k$. Dla każdego obiektu $\underline{x}$ redukowanego zbioru odniesienia O liczona jest miara $mp(\underline{x})$ oraz odległość $d(\underline{x},\underline{y})$ do najbliższego obiektu z klasy przeciwnej. Obiekty zbioru O porządkowane są wg rosnących wartości miar $mp(\underline{x})$, a obiekty z jednakowymi wartościami tych miar wg odległości i również rosnąco. Dla tak uporządkowanego zbioru O należy zastosować algorytm Harta.

Jest oczywiste, że obiekty z mniejszymi wartościami miar $mp(\underline{x})$ i odległościami $d(\underline{x},\underline{y})$ leżą bliżej granic klas.

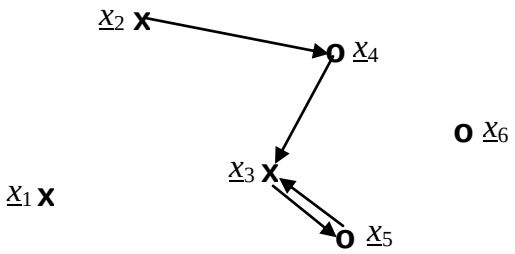
Modyfikacje z użyciem miary reprezentatywności obiektu

Inny sposób wykorzystania algorytmu Harta, po wcześniejszym odpowiednim uporządkowaniu oryginalnego zbioru odniesienia został zaproponowany w rozprawie doktorskiej [Raniszewski M., 2009]. Autor wprowadził miarę reprezentatywności $rm(\underline{x})$ obiektu $\underline{x}$ mierzoną liczbą obiektów ze zbioru uczącego dla których ten obiekt jest najbliższym sąsiadem. Obiekty zbioru uczącego są sortowane najpierw wg malejących wartości miary $rm(\underline{x})$, a następnie obiekty z tymi samymi wartościami $rm(\underline{x})$ sortowane są rosnąco wg miary $mp(\underline{x})$ zaproponowanej przez Gowdę i Krishnę. Obu tych miar można użyć w odwrotnej kolejności.

Oprócz opisu w/w podejścia wymieniona wyżej rozprawa zawiera opis wielu znanych algorytmów redukcji, w tym kilku innych, niż już wyżej wymieniony, zaproponowanych przez jej autora i stanowi opis stanu wiedzy w tej tematyce.

Metoda obiektów wzajemnie najbliższych.

Jest to drugi algorytm [Jóźwik A., Kieś P., 2005], poza już opisanym algorytmem Tomeka, w którym do zbioru zredukowanego rekrutowane są pary obiektów z różnych klas, a otrzymany w ten sposób zbiór jest uzupełniany z użyciem procedury Harta. Jednak w tym algorytmie pary kwalifikowanych do zbioru zredukowanego obiektów wyznaczane są w inny sposób, który został zilustrowany na Rys. 4.9.



Rys. 4.9. Sposób wyznaczania obiektów wzajemnie najbliższych

Każdemu obiektowi z oryginalnego zbioru odniesienia przyporządkowana zostaje para obiektów wzajemnie najbliższych, pochodzących z dwóch klas, klasy 1 i klasy 2. Procedura musi być przeprowadzona dla wszystkich obiektów zbioru, który ma być zredukowany. Dla dowolnego obiektu $\underline{x}$, np. z klasy 1 zaczyna się ona od wyznaczenia najbliższego sąsiada $\underline{s}_1$ z przeciwnej klasy, czyli z klasy 2. Następnie, dla obiektu $\underline{s}_1$ szuka się najbliższego mu sąsiada $\underline{s}_2$ z klasy 1. Potem, dla $\underline{s}_2$ wyznaczany jest najbliższy sąsiad z klasy 2, itd., aż w końcu otrzymamy pętlę, czyli począwszy od pewnego l zajdzie relacja $\underline{s}_{l+2}=\underline{s}_l$. Żeby uniknąć pętli obejmującej więcej niż dwa

obiekty, należy przyjąć, że w przypadkach niejednoznacznych wybierany jest obiekt z niższym indeksem.

W sytuacji pokazanej na Rys. 4.9 dla obiektu $\underline{x}=\underline{x}_2$ z klasy 1 znaleziony został najbliższy mu obiekt $\underline{s}_1=\underline{x}_4$ z klasy 2. Dla obiektu $\underline{s}_1$ najbliższym obiektem z przeciwnej klasy jest obiekt $\underline{s}_2=\underline{x}_3$ z klasy 1, a najbliższym dla $\underline{s}_2$ jest obiekt $\underline{s}_3=\underline{x}_5$ z klasy 2. Obiekt $\underline{s}_3$ ma w charakterze swojego najbliższego sąsiada obiekt $\underline{s}_5=\underline{x}_3$. Warunek końca zadania wyznaczenia pary obiektów wzajemnie najbliższych dla obiektu $\underline{x}_2$ został spełniony. Obiekty $\underline{x}_3$ i $\underline{x}_5$ stanowią parę obiektów wzajemnie najbliższych skojarzoną z obiektem $\underline{x}_2$.

Przedstawiony algorytm odnosi się wyłącznie do dwóch klas, a w porównaniu do podobnego mu algorytmu Tomeka, par obiektów wzajemnie najbliższych jest znacznie mniej niż par *tomekowych*. Tak otrzymany zbiór należy uzupełnić stosując procedurę Harta.

Modyfikacja algorytmu Tomeka

Algorytm Tomeka wyznacza pary obiektów $\underline{a}$ i $\underline{b}$ z różnych klas. Na podstawie takiej pary można skonstruować hiperpłaszczyznę przechodzącą przez środek odcinka łączącego punkty $\underline{a}$ i $\underline{b}$ oraz ortogonalną do niego. Z tą hiperpłaszczyzną można związać nową sztuczną cechę obiektu $\underline{x}$, która będzie przyjmować tylko 3 wartości: +1, gdy obiekt $\underline{x}$ znajdzie się po stronie punktu $\underline{a}$, 0, gdy $\underline{x}$ będzie leżał na tej hiperpłaszczyźnie oraz -1, gdy znajdzie się po stronie punktu $\underline{b}$. Zero można wyeliminować, jeśli obiektom leżącym na hiperpłaszczyźnie przyporządkuje się wartość +1, zamiast zera. Taką konwersję można wykonać dla każdego obiektu oryginalnego zbioru odniesienia O.

Tak więc każdej tomekowej parze obiektów odpowiada wzajemnie jednoznacznie nowa sztuczna cecha. Zastosowanie selekcji sztucznych cech wiązać się więc będzie z redukcją par tomekowych. Obiekty, które powinny znaleźć się w zbiorze zredukowanym, będą zatem wskazane przez cechy, bo każda z cech wytypuje parę tomekową, z której obiekty wejdą do zbioru zredukowanego. Otrzymany zostanie zbiór zredukowany w przestrzeni sztucznych cech. W nowej przestrzeni cech, w regule 1-NS, można stosować zarówno kwadrat odległości euklidesowej jak i odległość miejską. Nowo klasyfikowane obiekty muszą być przekonwertowane do nowej przestrzeni cech z użyciem tylko tych hiperpłaszczyzn, które odpowiadają wyselekcjonowanym cechom. Selekcję nowych cech przeprowadza się poprzez zastosowanie procedur kolejnego dołączania, kolejnego odrzucania lub procedury kombinowanej, omawianych już w podrozdziale 2.3. Jako kryterium można przyjąć frakcje błędów wyznaczoną metodą minus jednego elementu.

Hiperpłaszczyzny, które były podstawą dla zdefiniowania sztucznych cech dzielą przestrzeń oryginalnych cech na wypukłe obszary. Ale nie w każdym z nich znajduje się obiekt zbioru uczącego wchodzący w skład zbioru zredukowanego wyznaczonego algorytmem Tomeka. Część z takich obszarów może być pusta. W wyniku selekcji sztucznych cech, liczba par obiektów odpowiadająca tym cechom i liczba hiperpłaszczyzn powiązanych z tymi cechami ulegną zwykle zmniejszeniu. Liczba obszarów w przestrzeni oryginalnej zmniejszy się również. Jednak obszary puste wciąż mogą występować. Jeżeli klasyfikowany obiekt $\underline{x}$ znajdzie się w niepustym obszarze oryginalnej przestrzeni cech, to jego odległość do najbliższego sąsiada w nowej przestrzeni będzie równa zeru, gdyż wartości sztucznych cech obu tych obiektów będą identyczne. Idea w/w sztucznych cech, choć w inny sposób wykorzystana, została zaproponowana w publikacji [Jóźwik A., Chmielewski L., Skłodowski M., Cudny W., 2001].

Przykład

Sposób zastosowania metody sztucznych cech został zilustrowany na zbiorze danych z Tab.4.7, opisanych dwoma cechami c_1 i c_2.

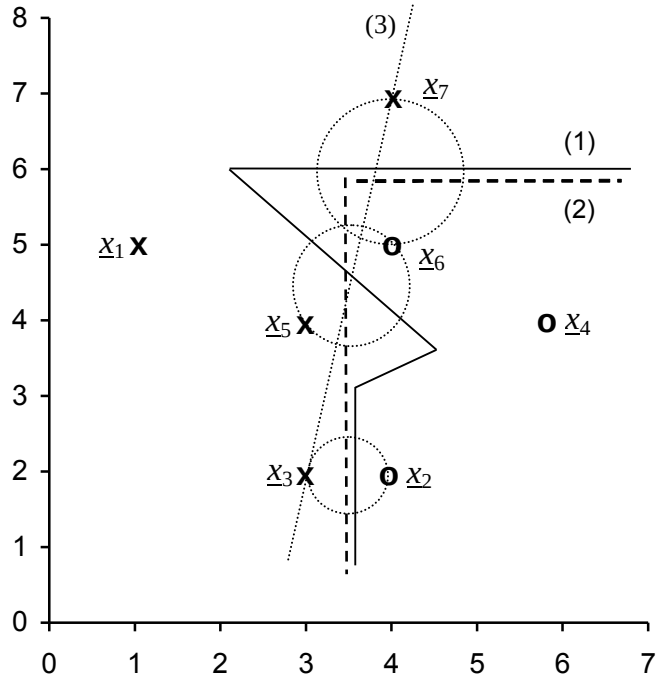
Tab. 4.7. Przykładowy zbiór dla ilustracji sposobu zastosowania sztucznych cech

Obiekt	Klasa	Cecha c_1	Cecha c_2	Cecha $c_{3,2}$	Cecha $c_{5,6}$	Cecha $c_{7,6}$	Cecha $c_{1,4}$
1	1	1,0	5,0	1	1	-1	1
2	2	4,0	2,0	-1	1	-1	-1
3	1	3,0	2,0	1	1	-1	0
4	2	6,0	4,0	-1	-1	-1	-1
5	1	3,0	4,0	1	1	-1	1
6	2	4,0	5,0	-1	-1	-1	-1
7	1	4,0	7,0	-1	-1	1	0

Pary punktów ($\underline{x}_3$,$\underline{x}_2$), ($\underline{x}_5$,$\underline{x}_6$) i ($\underline{x}_7$,$\underline{x}_6$) są parami tomekowymi, gdyż rozpięte na nich kule nie zawierają w swych wnętrzach żadnych obiektów. Zatem wszystkie obiekty z tych par wchodzą do zbioru zredukowanego. Linia łamana (1) wyznaczona przez regułę 1-NS będzie się jednak składać z fragmentów czterech linii prostych, gdyż oprócz wymienionych trzech par obiektów, para ($\underline{x}_5$,$\underline{x}_2$) też wprowadzi swój odcinek jako miejsce geometryczne punktów równoodległych od obiektów $\underline{x}_2$ i $\underline{x}_5$.

Zbiór ten pokazany został na Rys. 4.10.

Na podstawie wybranych algorytmem Tomeka trzech par obiektów ($\underline{x}_3$,$\underline{x}_2$), ($\underline{x}_5$,$\underline{x}_6$) i ($\underline{x}_7$,$\underline{x}_6$) można wyznaczyć hiperpłaszczyzny i na ich podstawie trzy sztuczne cechy: $c_{3,2}$, $c_{5,6}$ i $c_{5,6}$.



Rys. 4.10. Ilustracja do przykładu wyjaśniającego metodę sztucznych cech.

Hiperpłaszczyzny te zostały zorientowane w taki sposób aby, w odpowiadających im parach obiektów, obiekt z klasy 1 (krzyżyk) znajdował się po jej dodatniej stronie, a obiekt z klasy 2 (kółko) po ujemnej. Wartości nowych sztucznych cech $c_{3,2}$, $c_{5,6}$ i $c_{7,6}$ podane zostały w Tab. 4.7.

Stosując metodę minus jednego elementu dla zbioru uczącego złożonego z 7 obiektów i 3 sztucznych cech, można policzyć frakcję błędów, która wynosi ona e_1=2/7, gdyż mylnie klasyfikowanymi obiektami będą tylko obiekty $\underline{x}_2$ i $\underline{x}_7$ (dla 1-NS i metryki miejskiej). Jeżeli przeprowadzona zostanie selekcja utworzonych sztucznych cech, to w jej wyniku wybrane zostaną wybranymi cechami okażą się dwie cechy: $c_{3,2}$ i $c_{7,6}$, co oznacza, że wystarczy przechowywać w pamięci komputera wyłącznie cztery obiekty: $\underline{x}_2$, $\underline{x}_3$, $\underline{x}_6$ i $\underline{x}_7$. Frakcja błędów dla tych wybranych cech wyniesie e_2=1/7, ponieważ jedynym mylnie klasyfikowanym obiektem jest obiekt $\underline{x}_7$. Do klasy 2 (kółka) będą zaliczane tylko obiekty $\underline{x}$, dla których $c_{3,2}$=-1 i $c_{7,6}$=-1, czyli $\underline{x}$= [-1,-1], a do klasy 1 (krzyżyki) obiekty opisane pozostałymi kombinacjami wartości cech, czyli, gdy $\underline{x}$= [-1,1] albo $\underline{x}$= [1,-1]. Linia przerywana na Rys. 4.10 dzieli przestrzeń cech na dwa regiony decyzyjne, na lewo lub powyżej linii przerywanej (2) region klasy 1, a na prawo i poniżej tej linii region klasy 2.

Warto zauważyć, że zastosowanie metody minus jednego elementu dla zbioru uczącego z pierwotnymi cechami c_1 i c_2 daje w wyniku frakcję błędów e_0=5/7, czyli

znacznie wyższą, niż obie frakcje e_1 i e_2 wyżej wymienione i policzone dla sztucznych cech. Frakcje e_1 i e_2 nie mogą stanowić miar dla rzetelnej oceny klasyfikatora 1-NS działającego w nowej przestrzeni cech.

W przypadku małego zbioru można by zrezygnować z algorytmu Tomeka i stworzyć sztuczne cechy dla wszystkich możliwych par obiektów z przeciwnych klas. Takich par w rozważanym przykładzie jest 12. Po przeprowadzeniu selekcji sztucznych cech wybrana zostałaby tylko 1 cecha $c_{1,4}$, której wartości podane zostały w Tab. 4.7. Odpowiada ona hiperpłaszczyźnie (3) na Rys.4.10. Jak już było wyżej wspomniane, zerowe wartości sztucznych cech można zmienić na wartości równe 1. Tak więc, w omawianym przykładzie, wystarczyłaby tylko jedna sztuczna cecha, a zgodny zbiór zredukowany dla reguły 1-NS, w oryginalnej przestrzeni cech, mógłby zawierać tylko parę obiektów, obiekty $\underline{x}_1$ i $\underline{x}_4$, przy czym nowe obiekty znajdujące się w jednakowych odległościach od każdego z tych dwóch obiektów powinny być kwalifikowane do klasy 1 (krzyżyki).

Bąbelkowy algorytm redukcji zbioru odniesienia

Istota przyspieszenia klasyfikacji metodami redukcji zbioru odniesienia polega na tym, że najbliższy sąsiad obiektu klasyfikowanego wyznaczany jest z mniejszego zbioru obiektów, czyli ze zbioru zredukowanego. Zaprezentowane teraz zostanie inne podejście, które wymaga zmiany samej reguły decyzyjnej. Oryginalny zbiór odniesienia O można pokryć jednorodnymi rozłącznymi, w sensie rozłączności słabej, hiperkulami, które zawierają obiekty tylko jednej klasy. Tak więc hiperkule te mają przydzieloną przynależność. Wspólnymi punktami tych hiperkul mogą być jedynie punkty leżące na ich sferach. Jeżeli klasyfikowany obiekt $\underline{x}$ znajdzie się wewnątrz pewnej hiperkuli lub tylko na jej sferze, to przypisana mu zostanie klasa tej hiperkuli. Gdy zaś będzie leżał jednocześnie na sferach kilku hiperkul, to przyjmie klasę hiperkuli wcześniej utworzonej, czyli z mniejszym indeksem. Każda z utworzonych hiperkul może zawierać wiele lub tylko jeden obiekt. W przypadku, gdy klasyfikowany obiekt znajdzie się poza każdą z hiperkul, przypisana mu będzie klasa najbliższej hiperkuli.

Tworzenie hiperkul rozpoczyna się od losowo wybranego obiektu $\underline{x}_1$, dla którego znajdowana jest odległość r_1 do najbliższego sąsiada $\underline{y}_1$ z innej klasy. Pierwsza z hiperkul $K(\underline{x}_1,r_1,l_1)$, gdzie liczba l_1 oznacza liczbę obiektów zawartych wewnątrz niej lub na jej sferze, została zatem już zdefiniowana. Sposób tworzenia kolejnych hiperkul określić można indukcyjnie. Należy teraz pokazać jak, mając już utworzone hiperkule $K(\underline{x}_j,r_j,l_j)$, j=1,2,…,i, utworzyć hiperkulę $K(\underline{x}_{i+1},r_{i+1},l_{i+1})$. Spoza obiektów niepokrytych jeszcze przez dotychczas utworzone hiperkule wybierany jest losowo obiekt $\underline{x}_{i+1}$, liczona jest odległość d_{i+1} do najbliższego sąsiada z innej klasy niż $\underline{x}_{i+1}$ oraz wyznaczana jest odległość q_{i+1} do najbliższej hiperkuli. Promień r_{i+1} tworzonej

hiperkuli przyjmuje mniejszą z dwóch wartości: d_{i+1} i q_{i+1}. Zatem ograniczeniem przy tworzeniu kolejnej hiperkuli jest najbliższy obiekt z innej klasy lub sfera innej hiperkuli.

Opisany wyżej algorytm, jako jeden z kilku, został zaimplementowany w ramach rozprawy doktorskiej [Sierszeń A.,2009]. Nie dawał on zadowalających wyników. Były one zbliżone do wyników oryginalnej reguły 1-NS, tylko w przypadku bardzo dużych zbiorów uczących, rzędu kilkadziesiąt tysięcy obiektów (2 klasy+13 cech). Jednak pod względem szybkości klasyfikacji to podejście okazało się bardzo efektywne, również dzięki i temu, że otrzymane hiperkule były malejąco sortowane wg wielkości ich promieni i w takiej kolejności były one weryfikowane, czy nie zawierają klasyfikowanego obiektu. Nasuwa się zatem sugestia, że lepiej byłoby sortować uzyskane hiperkule wg liczebności zawartych w nich obiektów.

Powodem niezadowalających efektów jest w znacznej części fakt, że algorytm ten nie daje w wyniku zgodnego zbioru hiperkul. Część obiektów ze zbioru uczącego może znajdować się na sferach hiperkul reprezentujących przeciwne klasy. Poza tym nie jest jednoznaczny, czyli jego wyniki nie są powtarzalne.

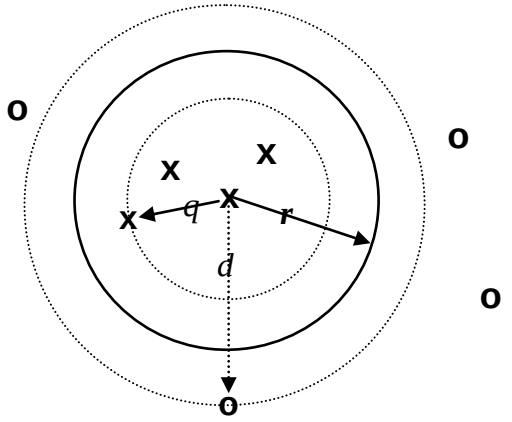
Reguła hiperkul

Opisany algorytm można łatwo zmodyfikować, by usunąć zarówno jej niejednoznaczność, wyposażyć w zgodność z oryginalnym zbiorem odniesienia O oraz by ciąg wytworzonych hiperkul był uporządkowany tak, by kolejna w ciągu hiperkula zawierała możliwie najwięcej obiektów spoza zbioru obiektów już pokrytych przez poprzednio wyznaczone hiperkule.

Dla każdego obiektu $\underline{x} \in O$ należy wyznaczyć odległość d do najbliższego sąsiada z innej klasy oraz odległość q do najdalszego obiektu z tej samej klasy co $\underline{x}$, ale w odległości nie przekraczającej d. Tworzone są teraz hiperkule $K(\underline{x},r,Z)$, z których każda jest określona przez środek $\underline{x}$, promień $r=(d+q)/2$ oraz zbiór obiektów Z, które znalazły się wewnątrz hiperkuli lub na jej sferze. Sposób tworzenia takiej hiperkuli przestawiony został na Rys. 4.11.

Obiekty z klasy 1 (krzyżyki) znajdują się na lub wewnątrz kuli o środku w punkcie $\underline{x}$ i promieniu q. Zaś obiekty z klasy 2 (kołka) na lub wewnątrz kuli o tym samym środku i promieniu d. Spośród tych hiperkul, jako pierwsza, wybierana jest hiperkula $K(\underline{x}_1,r_1,Z_1)$, w której, łącznie z jej sferą, znajduje się najwięcej obiektów ze zbioru O, tzn. hiperkula z najliczniejszym zbiorem Z.

Tworzenie kolejnej hiperkuli, podobnie jak w przypadku bąbelkowego algorytmu redukcji zbioru odniesienia, da się zdefiniować rekurencyjnie, przyjmując, że hiperkule $K(\underline{x}_j,r_j,Z_j)$, $j=1,2,\ldots,i$, zostały już utworzone. Jako hiperkula $K(\underline{x}_{i+1},r_{i+1},Z_{i+1})$ wybrana zostaje ta z pozostałych hiperkul $K(\underline{x},r,Z)$, dla której zbiór Z zawiera



Rys. 4.11. Ilustracja sposobu wyznaczania promienia hiperkuli

maksymalną liczbę obiektów spoza zbioru S_i=$Z_1 \cup Z_2 \cup,\ldots,\cup\ Z_i$. Jeżeli przy okazji tego wybierania zostanie wykryte, że któraś z tych pozostałych hiperkul $K(\underline{x},r,Z)$ nie ma w swoim zbiorze Z obiektów spoza zbioru S_i, to zostaje trwale usunięta. Liczba n_k hiperkul $K(\underline{x}_j,r_j,Z_j)$ pokrywających zbiór O będzie z oczywistych powodów znacznie mniejsza niż liczba m obiektów zbioru O. Warto zauważyć, że hiperkule zawierające obiekty tej samej klasy mogą się nakładać na siebie. Otrzymany ciąg hiperkul jest tak uporządkowany, by maksymalizować szansę znalezienia się klasyfikowanego obiektu w hiperkuli z najmniejszym indeksem. Jeśli obiekt nie znajdzie się w żadnej z wyznaczonych hiperkul, to przypisana mu zostaje klasa hiperkuli najbliższej. Odległość punktu od hiperkuli rozumiana jest jako odległość tego punktu do środka hiperkuli pomniejszona o jej promień.

4.7. Kondensacja zbiorów odniesienia

Algorytmy przedstawione poprzednim podrozdziale pozwalały wyznaczać zbiory odniesienia o mniejszych liczebnościach niż zbiory uczące i były ich podzbiorami. Zredukowane zbiory odniesienia nie muszą być podzbiorami zbioru uczącego, ale mogą być punktami w przestrzeni cech utworzonymi na podstawie zbioru uczącego, czyli obiektami sztucznymi. Przykładem zbioru odniesienia może być zbiór środków ciężkości lub środków medianowych wyznaczanych dla klasyfikatora minimalno-odległościowego. Jest to jednocześnie przykład najmocniejszej kondensacji. Kondensacja tym się różni od redukcji, że skondensowany zbiór odniesienia nie jest podzbiorem kondensowanego zbioru uczącego. Jednak klasyfikator minimalno-odległościowy, jak już było to wyjaśnione w podrozdziale 3.5, może nie oferować satysfakcjonującej jakości klasyfikacji. Klasyfikator 1-NS może mieć nad nim dużą przewagę, jeśli chodzi o jakość klasyfikacji i jest bardziej uniwersalny, bo tworzy

bardziej złożone hiperpowierzchnię rozdzielające. Stąd, w dalszym ciągu będzie on przedmiotem rozważań. Rozróżnienie redukcji i kondensacji zostało wprowadzone przez autora niniejszej monografii. Inni autorzy rozróżniają te dwa sposoby uzyskiwania zbiorów odniesienia o zredukowanych liczebnościach, ale tych pojęć używają zamiennie, traktując je jako synonimy [Ainslie M.C. and Sanchez J.S., 2002], [Hart P.E. (1968)].

Algorytm Changa

Pierwszy algorytm kondensacji został zaproponowany w pracy [Chang C.L., 1974] i polegał na łączeniu obiektów z tej samej klasy i zastępowaniu ich jednym nowym sztucznym obiektem. Zbiór skondensowany *S* początkowo równy jest całemu zbiorowi uczącemu, czyli oryginalnemu zbiorowi odniesienia *O*. Oba te zbiory muszą być pamiętane, bo *S* będzie stopniowo ulegał zmniejszeniu, a *O* będzie służył do weryfikacji zgodności. W *S* mogą być tylko pamiętane indeksy obiektów z *O*, które wchodzą w skład *S*. Oczywiście zbiór *S* jest zgodny, bo nie może nie być zgodny z samym sobą czyli ze zbiorem *O*. Następnie, wyznacza się parę obiektów najbliższych $\underline{a}$ oraz $\underline{b}$, pochodzących z tej samej klasy i chwilowo tworzy się z nich nowy sztuczny obiekt $\underline{c}=(\underline{a}+\underline{b})/2$. Jeżeli wyrzucenie obiektów $\underline{a}$ oraz $\underline{b}$ i zastąpienie ich obiektem $\underline{c}$ nie likwiduje zgodności zbioru *S*, to obiekty $\underline{a}$ oraz $\underline{b}$ zostają trwale usunięte ze zbioru *S*. Podobną operację przeprowadza się dla zbioru $S:= S\cup\underline{c}-\{\underline{a},\underline{b}\}$. Następnie w aktualnym zbiorze *S* ponownie szukana jest nowa para $\underline{a}$ oraz $\underline{b}$ i sprawdzana jest możliwość zastąpienia tej pary kolejny nowym obiektem $\underline{c}$, który z niej został utworzony. Taka operacja jest powtarzana tak długo, aż w aktualnym zbiorze *S* nie znajdzie się żadnej pary obiektów, którą dałoby się zastąpić średnim obiektem, z niej wyznaczonym.

Algorytm hiperpłaszczyzn tnących

Inna idea kondensacji zbioru uczącego została zaproponowana w materiałach konferencyjnych [Jóźwik A., Serpico S. B. and Roli F., 1995] i później opisana też publikacji [Chen C. H. and Jóźwik A., 1996]. Polega ona na podziale zbioru uczącego na żądaną liczbę podzbiorów i zastąpieniu każdego z tych podzbiorów środkiem ciężkości i przyporządkowaniu mu klasy (tj. etykiety) takiej jak klasa, do której należy większość jego obiektów. Sytuacje niejednoznaczne można rozstrzygać na korzyść klasy liczniejszej w całym zbiorze uczącym, a ostatecznie klasy z mniejszym indeksem. Dla zdefiniowania algorytmu wprowadzone zostało pojęcie średnicy zbioru, rozumianej jako największa odległość pomiędzy dwoma obiektami zbioru.

Niech *nd* będzie wymaganą liczebnością zbioru skondensowanego *S*. Oryginalny zbiór odniesienia *O*, tzn. zbiór uczący, będzie sukcesywnie dzielony na coraz to większą liczbę podzbiorów O_i, aż w końcu będzie ich *nd*, tzn. $i=1,2,\ldots,nd$.

Warunkiem początkowym jest: O_1=O, co oznacza, że zbiór O nie został jeszcze podzielony. Środek ciężkości $\underline{s}_1$ zbioru O_1 stanowi jednoelementowy zbiór skondensowany S_1. Temu środkowi ciężkości przypisana zostaje klasa, jaką ma większość obiektów w O_1.

Sposób tworzenia następnych podziałów zbioru O na podzbiory można opisać rekurencyjnie, zakładając, że w pewnym kroku jest nc podzbiorów zbioru O, a należy ich liczbę powiększyć do nc+1. Skoro jest ich nc, to $O=O_1\cup O_2\cup\ldots\cup O_{nc}$, a aktualny zbiór skondensowany S składa się ze środków ciężkości $\underline{s}_j$ odpowiadających podzbiorom O_j, j=1,2,…,nc, których przynależność jest taka jak większości obiektów odpowiedniego podzbioru O_j. Z tych podzbiorów wybierany jest podzbiór O_i, zawierający obiekty co najmniej z dwóch klas i którego średnica jest największa. Jeśli nie ma już podzbiorów O_j zawierających obiekty co najmniej z dwóch klas, to wybierany jest tylko podzbiór O_i o największej średnicy. Niejednoznaczność może być rozstrzygnięta na korzyść podzbioru z mniejszym indeksem.

Podzbiór O_i zostanie podzielony na dwa podzbiory i w ten sposób liczba podzbiorów O_j, składających się na zbiór O, powiększy się z nc na nc+1. W jaki Sposób wykonania podziału zostanie opisany w następnym akapicie. Dla dalszych rozważań, zbiór O_i, przeznaczony do podziału, wygodnie jest oznaczyć dodatkowym symbolem, czyli przyjąć, że $D=O_i$. Zatem zbiór D zostanie podzielony na dwa podzbiory D_1 oraz D_2. Z ciągu podzbiorów O_j, j=1,2,…,nc, zostaje usunięty dotychczasowy podzbiór O_i, a jego miejsce zajmuje nowy podzbiór $O_i=D_1$. Ponadto, ciąg ten zostaje powiększony o nowy element $O_{nc+1}=D_2$. Zbiór skondensowany S również zostaje uaktualniony. Dotychczasowy punkt $\underline{s}_i$ jest usuwany, a na jego miejsce wchodzi nowy punkt $\underline{s}_i$ będący środkiem ciężkości zbioru D_1. Oprócz tej zamiany, do zbioru skondensowanego S dołączony zostaje środek ciężkości zbioru D_2, jako punkt $\underline{s}_{nc+1}$.

W opisanej wyżej rekurencyjnej procedurze, liczebność zbioru skondensowanego S powiększa się jednocześnie z liczebnością podzbiorów O_j, które składają się na zbiór O. Możliwe jest więc kontrolowanie jakości klasyfikacji po każdym uaktualnieniu zbioru skondensowanego i zatrzymanie procedury zanim wielkość zbioru S osiągnie zadaną liczbę nd elementów, jeśli frakcja mylnych decyzji okaże się akceptowalna. W przypadku rezygnacji z kontroli jakości klasyfikacji na bieżąco, wystarczy elementy zbioru S wyznaczyć dopiero po uzyskaniu nd podzbiorów O_j, j=1,2,…,nd, jako środki ciężkości tych podzbiorów. Kontrola jakości może odbywać się na podstawie klasyfikacji obiektów ze zbioru O lub oddzielnego zbioru walidacyjnego.

Nie został jeszcze wyjaśniony sposób podziału zbioru D na podzbiory D_1 i D_2. Otóż, w zbiorze D należy znaleźć parę obiektów $\underline{p}_1$ i $\underline{p}_2$ maksymalnie odległych od siebie. Na podstawie tych obiektów konstruowana jest hiperpłaszczyzna przechodząca

przez środek odcinka łączącego punkty $\underline{p}_1$ i $\underline{p}_2$ oraz ortogonalna do niego. Z punktów zbioru D, znajdujących się na tej hiperpłaszczyźnie lub bliżej punktu $\underline{p}_1$ utworzony zostaje zbiór D_1, a z punktów bliższych obiektowi $\underline{p}_2$ zbiór D_2.

Opis formalny zaproponowanego algorytmu jest następujący:

Definicja algorytmu

1. Ustal pożądaną liczbę *nd* obiektów (punktów) w zbiorze skondensowanym S;
2. Przyjmij $S=\varnothing$ oraz $nc:=1$, nc – aktualna liczba podzbiorów zbioru O;
3. Przyjmij $O_1:=O$;
4. Oblicz $\underline{s}_1$= środek ciężkości zbioru O_1, przypisz mu klasę, dołącz go do S oraz przyjmij $i=1$;
5. W zbiorze $D=O_i$ znajdź dwa najbardziej odległe od siebie punkty $\underline{p}_1$ i $\underline{p}_2$;
6. Podziel zbiór D na dwa podzbiory *D1* i *D2*, gdzie $d(\circ,\circ)$ jest funkcją odległości:
   $D_1:=\{\underline{p}\in D: d(\underline{p},\underline{p}_1)\le d(\underline{p},\underline{p}_1)\}$,
   $D_2:=\{\underline{p}\in D: d(\underline{p},\underline{p}_1)> d(\underline{p},\underline{p}_1)\}$;
7. $nc:=nc+1$, $O_i:=D_1$, $O_{nc}:=D_2$, usuń $\underline{s}_i$ z aktualnego zbioru skondensowanego S;
8. Oblicz ciężkości $\underline{s}_i$ i $\underline{s}_{nc}$ zbiorów O_i i O_{nc}, przypisz im klasy i dołącz do S;
9. Jeżeli $nc=nd$, to skocz do 14;
10. Przyjmij $J_1:=\{j: j\le nc$ i O_j zawiera obiekty co najmniej z dwóch klas$\}$, $J_2:=\{j: j\le nc\}-J_1$;
11. Przyjmij $J:=J_1$ jeśli J_1 nie jest pusty, w przeciwnym przypadku $J:=J_2$;
12. Wyznacz parę punktów $\underline{p}_1$ i $\underline{p}_2$ najbardziej odległych od siebie w każdym z podzbiorów O_j dla $j\in J$ i zapamiętaj indeks $i\in J$ zbioru O_i o największej średnicy;
13. Skocz do 5;
14. Koniec.

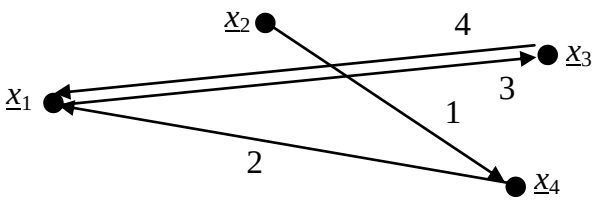
Przedstawiony algorytm można modyfikować na wiele sposobów, chociażby ze względu na kryterium typowania podzbioru O_i przeznaczonego do podziału (krok 12). Zatem można wybierać zbiór o największej średnicy albo najliczniejszy zbiór O_i.

Liczba *nd* podzbiorów, czyli wielkość zbioru skondensowanego nie musi być zadawana. Procedurę podziału kolejnego ze zbiorów można kontynuować, aż wyczerpane zostaną podzbiory zawierające reprezentantów przynajmniej dwóch klas. W takim przypadku liczebność zbioru skondensowanego zostanie wyznaczona automatycznie.

Następne modyfikacje mogą polegać na rekrutowaniu do zbioru skondensowanego S nie pojedynczych środków ciężkości O_i, lecz jednocześnie środków ciężkości każdej z klas reprezentowanej w zbiorze O_i. Wszystkie wymienione wyżej możliwości modyfikacji algorytmu hiperpłaszczyzn tnących zostały rozpatrzone i porównane

eksperymentalnie ze sobą i z oryginalną wersją reguły 1-NS na dziewięciu dostępnych w Internecie zbiorach danych [Ainslie M.C. and Sanchez J.S.,2002; Sanchez J.S., 2004]. Okazało się, że najlepszy wynik uzyskano, gdy do podziału wybierany jest podzbiór najliczniejszy zawierający obiekty co najmniej z dwóch klas, a algorytm jest kontynuowany tylko do momentu wyczerpania podzbiorów spełniających te warunki.

Jeszcze inna modyfikacja [Jóźwik A., Kieś P., 2005], mająca na celu przyspieszenie algorytmu, polega na zastąpieniu średnicy dzielonego zbioru odległością pomiędzy parą obiektów wzajemnie najdalszych. Obiekty $\underline{x}$ oraz $\underline{y}$ są wzajemnie najdalsze, jeżeli $\underline{y}$ jest najdalej położonym obiektem od $\underline{x}$ i jednocześnie $\underline{x}$ jest najdalszym obiektem względem $\underline{y}$. Sposób wyznaczania obiektów wzajemnie najdalszych zilustrowano na Rys 4.12.



Rys. 4.12. Ilustracja sposobu wyznaczania obiektów wzajemnie najdalszych

Parę punktów wzajemnie najdalszych wyznacza się dla każdego obiektu odpowiedniego podzbioru. Wyznaczając taką parę dla obiektu $\underline{x}_2$ z Rys. 4.12 znaleźć należy najdalszy mu obiekt, jest nim obiekt $\underline{x}_4$, podobnie najdalszym obiektem dla obiektu $\underline{x}_4$ jest obiekt $\underline{x}_1$, dalej dla $\underline{x}_1$ najdalszym obiektem jest $\underline{x}_3$, a dla $\underline{x}_3$ najdalszym jest $\underline{x}_1$. Powstała pętla, która obejmuje obiekty wzajemnie najdalsze.

Przebieg algorytmu w jego pierwotnej wersji został zilustrowany w Tab. 4.8, z tym, że wielkość zbioru wynikowego nie była zadana i procedura kończyła się po uzyskaniu podzbiorów, z których każdy zawierał obiekty tylko jednej klasy.

Tab. 4.8. Przykładowy przebieg kondensacji zbioru odniesienia

I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
II	**x**			**x**			**x**						**x**				**o**						**x**				**o**								**o**		**o**
III									**1**										**1**																		
IV								**2**																						**3**							
V															**4**										**5**												
VI				**x**									**x**				**o**						**x**				**o**									**o**	

Gdyby zadana liczebność zbioru skondensowanego była *nd*=6, to przebieg algorytmu byłby identyczny. W wierszu I podana została skala osi współrzędnej, by można było odczytywać wartości cech. Zbiór uczący, pokazany w wierszu II, który stanowi pierwotny, czyli nie skondensowany zbiór odniesienia *O*, składa się z pięciu obiektów z klasy 1 (krzyżyki) i z czterech obiektów klasy 2 (kółka).

W pierwszym kroku algorytmu na zbiór *O* składa się tylko jeden jego podzbiór $O_1=O$, a zbiór skondensowany *S* zawiera tylko jeden element, czyli środek ciężkości $\underline{s}_1$=(1+4+7+13+ +17+23+27+35+37)/9=18,2 zbioru O_1. Przypisana mu zostanie klasa 1, gdyż większość obiektów tego zbioru pochodzi z klasy 1. Teraz należy przyjąć $D=O_1$ i dokonać podziału zbioru *D* na dwa podzbiory D_1 i D_2. Maksymalna odległość pomiędzy obiektami zbioru *D* wynosi 37-0=37, co oznacza, że podział wystąpi w medianie liczb od 1 do 37 (współrzędne skrajnych punktów), a więc na wartości 19. Symbolizuje go liczba 1 umieszczona w kolumnie 3. Podzbiór D_1, tworzą obiekty na lewo od punktu podziału, czyli wartości 19 na osi współrzędnej, a na podzbiór D_2 składają się obiekty leżące na prawo od punktu podziału.

Położenie cyfry 1 w III wierszu wskazuje jaka wartość cechy wyznacza podział 1. Zbiór odniesienia *O* składa się teraz z dwóch podzbiorów $O_1=D_1$ oraz $O_2=D_2$. Aktualny zbiór skondensowany *S* ma teraz dwa nowe elementy: $\underline{s}_1$=(1+4+7+13+17)/5=8,4 oraz $\underline{s}_2$=(23+27+35+37/4=30,5, które są środkami ciężkości zbiorów O_1 oraz O_2. Punkt $\underline{s}_1$ ma przypisaną klasę 1, a punkt $\underline{s}_2$ klasę 2. W podzbiorze znajdującym się na lewo odległość pomiędzy dwoma najdalszymi obiektami wynosi 17-1=16, a dla zbioru prawego wynosi ona 37-23=14. Zatem podziałowi należy poddać podzbiór lewy, jako aktualny zbiór *D*. Wartości cech jego obiektów rozciągają się od 1 do 17, podział więc wystąpi w medianie liczb od 1 do 17, czyli w 9. Położenie punktu podziału wskazuje liczba 2 w IV wierszu Tab.4.8. Aktualny zbiór D_1 tworzą obiekty na lewo od wartości 9, a na zbiór D_2 składają się obiekty leżące pomiędzy wartościami 9 i 19. Zbiór O_1 ulega teraz uaktualnieniu i równy jest zbiorowi D_1 (obiekty na lewo od wartości 9), zbiór O_2 pozostaje niezmieniony (obiekty na prawo od wartości 19), a zbiór O_3 równa się teraz zbiorowi D_2.

Na zbiór skondensowany składają się obecnie punkty: nowy $\underline{s}_1$=4 (środek ciężkości zbioru D_1) z przypisana mu klasą 1, niezmieniony $\underline{s}_2$=30,5 z klasy 2 oraz nowy $\underline{s}_3$=(13+17)/2=15 z klasy 1, ponieważ ta klasa przeważała w nadzbiorze *D*.

W podobny sposób można kontynuować podział zbioru *O* na coraz to mniejsze podzbiory. Kolejny 3 podział (wiersz IV Tab.4.8) dotyczyłby zbioru obiektów o współrzędnych na prawo od wartości 19, a wartością dzielącą byłaby liczba 30. Podział 4 (wiersz V tabeli), zrealizowany z wartością 15, odnosiłby się tylko do podzbioru złożonego z dwóch obiektów o współrzędnych 13 i 17. Ostatni podział 5 (zaznaczony też w wierszu V), dokonany byłby wartością cechy równą 25. Zawartość wynikowego

zbioru skondensowanego pokazana została w VI wierszu Tab.4.8.

Porównanie obu typów algorytmów zmniejszania zbioru odniesienia, czyli redukcji i kondensacji sugeruje rozwiązanie będące kombinacją obu tych podejść. Kombinacja ta daje lepsze wyniki niż samodzielne użycie kondensacji, gdy wymagana jest wielkość zbioru odniesienia niewiększa niż z góry zadana liczba obiektów.

Wyniki eksperymentów zawarte w pracach [Jóźwik A., Kieś P., 2005] oraz [Jóźwik A., 2006] pokazują, że korzystnie jest stosować kombinację, w której pierwszym etapem jest kondensacja, a drugim redukcja. Kombinacja ta w przeprowadzonych eksperymentach dawała od 2 do 3 razy mniejsze frakcje błędów niż zastosowanie tylko samej kondensacji. Odwrotna kolejność daje znacznie większe frakcje mylnych decyzji.

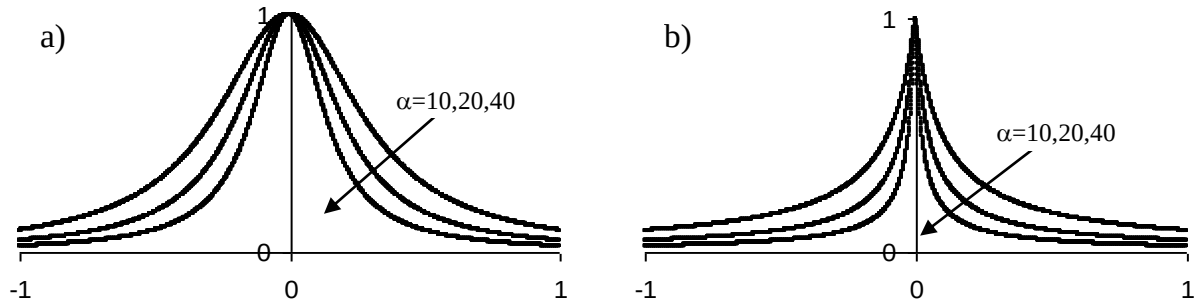
W eksperymentach wykorzystany został zbiór danych liczący 80800 obiektów, 2 klasy i 13 cech, z czego 40000 zostało wylosowanych do zbioru uczącego a pozostałe 40800 stanowiło zbiór testujący. Kondensacja była przeprowadzana z zastosowaniem algorytmu hiperpłaszczyzn tnących, a redukcja w pierwszej w/w prac dwoma algorytmami: Harta oraz Gowdy i Krishny. W drugiej z wymienionych prac zrezygnowano z algorytmu Gowdy i Krishny.

4.8. Metoda funkcji potencjałowych

Miara odległości może być użyta do zdefiniowania funkcji $K(\underline{x},\underline{u}_j)$ określanej jako potencjał generowany przez punkt $\underline{u}_j$ w punkcie $\underline{x}$. Termin *potencjał* został zapożyczony z fizyki poprzez skojarzenie z potencjałem elektrostatycznym ładunku elektrycznego. Funkcja $K(\underline{x},\underline{u}_j)$ powinna przyjmować wartość maksymalną w punkcie $\underline{x}=\underline{x}_j$, a jej wartość powinna spadać asymptotycznie do zera w miarę jak $\underline{x}$ oddala się od punktu $\underline{x}_j$. Najprostsza postać takiej funkcji ma formę: $K_a(\underline{x},\underline{u}_j)=1/[1+\alpha\circ d^2(\underline{x},\underline{u}_i)]$ lub $K_b(\underline{x},\underline{u}_j)=1/[1+\alpha\circ d(\underline{x},\underline{u}_i)]$, przy czym α jest parametrem dobieranym eksperymentalnie podczas dostrajania i testowania klasyfikatora wykorzystującego funkcje $K(\underline{x},\underline{u}_j)$.

Funkcje potencjałowe $K(\underline{x},\underline{u}_j)$, tak $K_a(\underline{x},\underline{u}_j)$ jak i $K_b(\underline{x},\underline{u}_j)$, pokazane zostały na Rys. 4.13.

Można je wykorzystać do zdefiniowania funkcji dyskryminacyjnych $g_i(\underline{x})$, $i=1,2,\ldots,nc$, gdzie nc jest liczbą klas, by zgodnie ze schematem z Rys.1.3, skonstruować klasyfikator. Jako funkcje dyskryminacyjne można przyjąć potencjały sumacyjne, czyli $g_i(\underline{x})=K_i(\underline{x})=\sum_{j\in I(i)} \lambda_j \circ K(\underline{x},\underline{u}_j)$, gdzie $I(i)$ jest zbiorem indeksów obiektów z klasy i, $1\leq i\leq nc$, a λ_j jest liczbą całkowitą wskazującą ile potencjałów $K(\underline{x},\underline{u}_j)$ obiektu $\underline{u}_i$ składa się na potencjał sumacyjny.



Rys. 4.13. Ilustracja kształtu funkcji potencjałowych:
a) $K_a(\underline{x},\underline{u}_j)=1/[1+\alpha_\circ d^2(\underline{x},\underline{u}_i)]$, b) $K_b(\underline{x},\underline{u}_j)=1/[1+\alpha_\circ d^2(\underline{x},\underline{u}_i)]$,

Klasyfikowany obiekt zaliczony będzie do klasy, której odpowiada największa wartość $g_i(\underline{x})$. Jest to wersja klasyfikatora bez uczenia.

Tego typu klasyfikator może być uczony, poprzez korygowanie potencjałów sumacyjnych $g_i(\underline{x})$. Początkowy potencjał sumacyjny może być tożsamościowo równy zeru. Jeżeli weryfikując, czy taki klasyfikator poprawnie przydziela klasę każdemu obiektowi ze zbioru uczącego okaże się, że dla pewnego obiektu $\underline{u}$ z klasy i potencjał sumacyjny $g_i(\underline{u})$ nie uzyskuje największej wartości, to można go powiększyć o $K(\underline{x},\underline{u})$, tzn. dokonać korekty polegającej na jego powiększeniu: $g_i(\underline{x}):=g_i(\underline{x})+K(\underline{x},\underline{u})$. Jeżeli w zbiorze uczącym nie ma obiektów z różnych klas, które byłyby w przestrzeni cech reprezentowane przez te same punkty, to korekcje te doprowadzą do całkowitego rozdzielenia obiektów ze zbioru uczącego. Jednak nie zawsze całkowite rozdzielenie punktów ze zbioru uczącego jest pożądane, gdyż może ono doprowadzić do przeuczenia klasyfikatora. Oznacza to, że wyniki oceny klasyfikatora na oddzielnym zbiorze testującym mogą być gorsze niż gdyby w/w uczenie było przerwane wcześniej.

Wyznaczone przez klasyfikator powierzchnie rozdzielające mogą być zbyt *skomplikowane*, np. pojedynczy obiekt, nawet gęsto otoczony obiektami z innej klasy, spowoduje, że wszystkie obiekty z jego bliskiego otoczenia będą zaliczane mylnie do klasy, z której on sam pochodzi. Na ten aspekt, który dotyczy także i reguły 1-NS, zwrócona została uwaga w książce [Tadeusiewicz R., Flasiński M.,1991].

Dobrym rozwiązaniem tego problemu mogło by być podzielenie zbioru danych, czyli zbioru X obiektów o znanej przynależności do klas, na trzy rozłączne części: część uczącą U, część walidacyjną W oraz część testującą T. Po każdej korekcji potencjałów sumacyjnych $g_i(\underline{x})$ $i=1,2,\ldots,nc$, należy obliczyć frakcję błędów na zbiorze W i stale trzymać w pamięci komputera komplet tych funkcji $g_i(\underline{x})$, które oferują najmniejszą frakcję pomyłek er_0. Jeżeli kilka kolejnych prezentacji całego zbioru U nie prowadzi już do uzyskania mniejszych frakcji błędów obliczonej na zbiorze W niż er_0, to uczenie klasyfikatora można przerwać. Na koniec należy wyznaczyć frakcję błędów

z wykorzystaniem zbioru *T*. Zbiór *W* byłby używany przy takim podejściu wielokrotnie, a funkcja potencjału sumacyjnego byłaby podbierana tak, aby frakcja błędów uzyskana dla obiektów z tego zbioru była minimalna. Ocena klasyfikatora, jaka odpowiada najmniejszej wartości frakcji błędów na zbiorze *W* nie byłaby rzetelna. Rzetelna ocena klasyfikatora wymaga oddzielnego zbioru testującego *T*, który zostałby użyty jednokrotnie.

W przypadku, gdy klasyfikacja odnosi się tylko do dwóch klas, to podobnie, jak dla liniowych funkcji dyskryminacyjnych opisanych w podrozdziale 1.3, dwie funkcje dyskryminacyjne $g_i(\underline{x})$, i=1,2, będące teraz potencjałami sumacyjnymi, mogą być zastąpione jednym potencjałem sumacyjnym:

$$g(\underline{x})=g_{1,2}(\underline{x})=K(\underline{x})=K_1(\underline{x})-K(\underline{x})_2=\sum_{j\in I(1)}\lambda_j\circ K(\underline{x},\underline{u}_j)-\sum_{j\in I(2)}\lambda_j\circ K(\underline{x},\underline{u}_j). \qquad (4.28)$$

Przypadek dwu-decyzyjny, określony relacją (4.28) wart jest oddzielnej uwagi. Niech $(\underline{u}_j)_{j=1}^{\infty}$ będzie nieskończonym ciągiem obiektów ze zbioru uczącego *U*, w którym każdy obiekt występuje nieskończoną liczbę razy. Temu ciągowi będzie odpowiadał ciąg potencjałów sumacyjnych $K_j(\underline{x})_{j=1}^{\infty}$. Uczenie reguły decyzyjnej klasyfikatora można zdefiniować następująco:

$$K_{j+1}(\underline{x})=K_j(\underline{x})+r\circ K(\underline{x},\underline{u}_{j+1}), \qquad (4.29)$$

gdzie r=0, gdy $K_j(\underline{u}_{j+1})\geq 0$ i $\underline{u}_{j+1}\in$ klasy 1 albo $K_j(\underline{u}_{j+1})<0$ i $\underline{u}_{j+1}\in$ klasy 2,

r=1, gdy $K_j(\underline{u}_{j+1})<0$ i $\underline{u}_{j+1}\in$ klasy 1,

r=-1, gdy $K_j(\underline{u}_{j+1})\geq 0$, a $\underline{u}_{j+1}\in$ klasy 2

oraz $K_0(\underline{x})\equiv 0$.

Przykładowe obliczenia zostaną wykonane dla potencjałów $K(\underline{x},\underline{u}_j)=K_a(\underline{x},\underline{u}_j)=1/[1+\alpha\circ d^2(\underline{x},\underline{u}_i)]$, a wartość współczynnika α=1.

Przykład

Działanie algorytmu uczenia wg reguły (4.29) można zilustrować na dla danych jednowymiarowych pokazanych na Rys. 4.14.

Wartość cechy	0	1	2	3	4	5	6	7	8	9	10	11
Klasa			x			x	o		x		o	o
Obiekt			$\underline{u}_1$			$\underline{u}_2$	$\underline{u}_4$		$\underline{u}_3$		$\underline{u}_5$	$\underline{u}_6$

Rys. 4.14. Zbiór danych dla ilustracji uczenia metodą funkcji potencjałowych

Potencjały te dla poszczególnych obiektów są następujące: $K(\underline{x},\underline{u}_1)=1/[1+(x-2)^2]$, $K(\underline{x},\underline{u}_2)=1/[1+(x-5)^2]$, $K(\underline{x},\underline{u}_3)=1/[1+(x-8)^2]$, $K(\underline{x},\underline{u}_4)=1/[1+(x-6)^2]$, $K(\underline{x},\underline{u}_5)=1/[1+(x-10)^2]$ i $K(\underline{x},\underline{u}_6)=1/[1+(x-11)^2]$. Prezentacja obiektów $\underline{u}_1,\underline{u}_2,\underline{u}_3$ nie zmienia wartości

potencjału sumacyjnego, tzn. $K_3(\underline{x})=K_2(\underline{x})=K_1(\underline{x})=K_0(\underline{x})=0$. Ale potencjał $K_3(\underline{u}_4)=0$, a powinien być ujemny, bo $\underline{u}_4$ jest z klasy 2. Zatem do potencjału $K_3(\underline{x})$ należy dodać potencjał $K(\underline{x},\underline{u}_4)=1/[1+(x\text{-}6)^2]$, a więc $K_4(\underline{x})=K_3(\underline{x})\text{-}[1/(x\text{-}6)^2]=-1/[1+(x\text{-}6)^2]$. Dla kolejnych obiektów $\underline{u}_5$ oraz $\underline{u}_6$, które są z klasy 2, potencjały $K_4(\underline{u}_5)=$-0,059, $K_4(\underline{u}_6)=$-0,38 są ujemne i korekcja nie jest potrzebna, czyli $K_6(\underline{x})=K_5(\underline{x})=K_4(\underline{x})$. Dopiero druga tura prezentacji obiektów ze zbioru uczącego będzie wymagać korekcji:

$K_6(\underline{u}_1)=$-0,059=<0$\rightarrow K_7(\underline{x})=K_6(\underline{x})\text{-}K(\underline{x},\underline{u}_1)=\text{-}1/[1+(x\text{-}6)^2]+1/[1+(x\text{-}2)^2]$,

$K_7(\underline{u}_2)=$-0,400<0$\rightarrow K_8(\underline{x})=K_7(\underline{x})+K(\underline{x},\underline{u}_2)=\text{-}1/[1+(x\text{-}6)^2]+1/[1+(x\text{-}2)^2]+1/[1+(x\text{-}5)^2]$,

$K_8(\underline{u}_3)=$-0,073<0$\rightarrow K_9(\underline{x})=K_8(\underline{x})+K(\underline{x},\underline{u}_3)=\text{-}1/[1+(x\text{-}6)^2]+1/[1+(x\text{-}2)^2]+1/[1+(x\text{-}5)^2]+$
$+1/[1+(x\text{-}8)^2]$, $K_9(\underline{u}_4)=-0{,}241<0\rightarrow K_{10}(\underline{x})=K_9(\underline{x})$, $K_{10}(\underline{u}_5)=$ 0,195 >0$\rightarrow K_{10}(\underline{x})=K_9(\underline{x})\text{-}$
$\text{-}K(\underline{x},\underline{u}_3)=\text{-}1/[1+(x\text{-}6)^2]+1/[1+(x\text{-}2)^2]+1/[1+(x\text{-}5)^2]+1/[1+(x\text{-}8)^2]-1/[1+(x\text{-}10)^2]$.

Potencjał sumacyjny $K(\underline{x})=K_{10}(\underline{x})$ jest już nieujemny dla obiektów z klasy 1 i ujemny dla obiektów z klasy 2: $K(\underline{u}_1)=$1,053, $K(\underline{u}_2)=$0,662, $K(\underline{u}_3)=$0,727, $K(\underline{u}_4)=$-0,300, $K(\underline{u}_5)=$0,805, $K(\underline{u}_6)=$-0,399. Potencjał obiektu $\underline{u}_6$ okazał się niepotrzebny.

Zastosowanie w/w sposobu uczenia dla funkcji $K_b(\underline{x},\underline{u}_j)=1/[1+\alpha\circ d(\underline{x},\underline{u}_i)]$ może wymagać korekcji dla innych obiektów niż użycie funkcji $K_a(\underline{x},\underline{u}_j)=1/[1+\alpha\circ d^2(\underline{x},\underline{u}_i)]$. I tak też się dzieje, co ilustruje Rys. 4.15.

I	$\underline{u}_1$	$\underline{u}_2$	$\underline{u}_3$	$\underline{u}_4$	$\underline{u}_5$	$\underline{u}_6$	$\underline{u}_1$	$\underline{u}_2$	$\underline{u}_3$	$\underline{u}_4$	$\underline{u}_5$	$\underline{u}_6$	$\underline{u}_1$	$\underline{u}_2$	$\underline{u}_3$	$\underline{u}_4$	$\underline{u}_5$	$\underline{u}_6$	$\underline{u}_1$	$\underline{u}_2$	$\underline{u}_3$
IIa				*			*	*	*		*						17				
IIb				*			*	*			*				*						21

Rys. 4.15. Zbiór danych dla ilustracji uczenia metodą funkcji potencjałowych

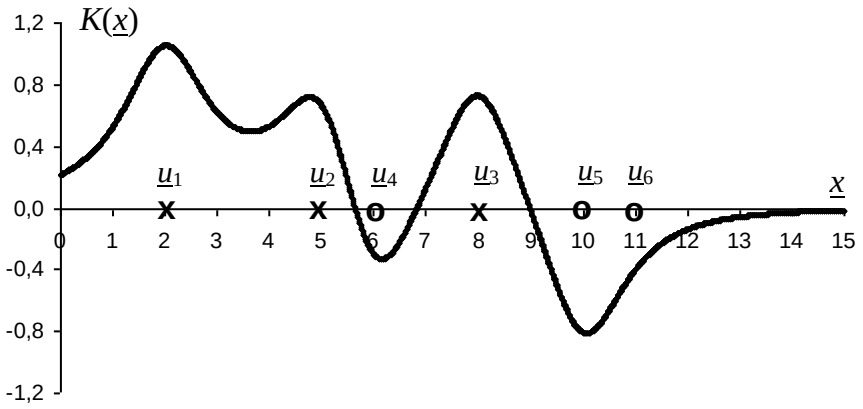
W wierszu nr I zamieszczony został ciąg obiektów ze zbioru uczącego. Zawartość wiersza nr IIa wskazuje, że uczenie, w tym przypadku trwające aż do prawidłowego rozdzielenia całego zbioru uczącego, z użyciem potencjałów $K_a(\underline{x},\underline{u}_j)$, zakończyło się po pięciu korekcjach. Były one konieczne kolejno dla obiektów $\underline{u}_4$, $\underline{u}_1$, $\underline{u}_2$, $\underline{u}_3$ i $\underline{u}_5$, co na rysunku zostało zaznaczone gwiazdką.

Uczenie zakończyło się dopiero po 17 prezentacjach obiektów ze zbioru uczącego, gdyż po uzyskaniu rozdzielenia zbiorów algorytm musiał wykonać jeszcze sześć prezentacji, nie wymagających korekcji, by móc stwierdzić sukces.

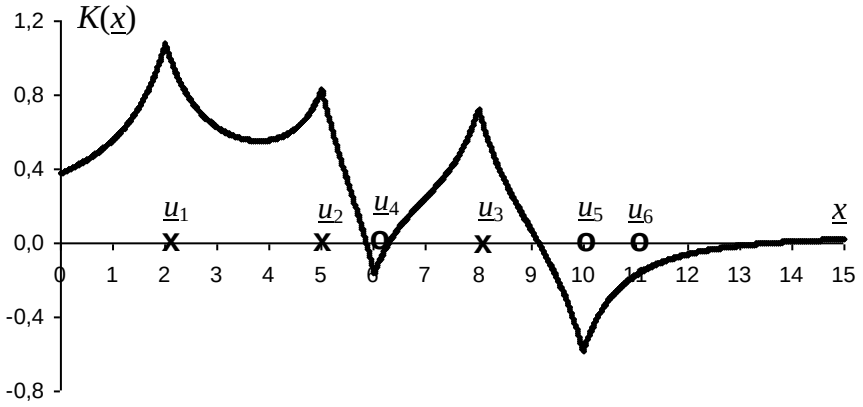
Użycie potencjałów $K_b(\underline{x},\underline{u}_j)$ również wymagało pięciu korekcji, ale nieco innych, co widać z zawartości wiersza IIb. Liczba koniecznych prezentacji obiektów ciągu uczącego wyniosła teraz 21. Liczby 17 i 21 zostały na Rys. 4.15 pokazane pod znaczeniami obiektów na których zakończone zostało uczenie.

Kształt potencjałów $K(\underline{x})$ utworzonych raz z funkcji potencjałowych $K_a(\underline{x},\underline{u}_j)$ i drugi raz z funkcji $K_b(\underline{x},\underline{u}_j)$, został pokazany na Rys. 4.16.

a)



b)



Rys. 4.16. Ilustracja potencjałów sumacyjnych dla funkcji: a) $K_a(\underline{x},\underline{u}_j)$ i b) $K_b(\underline{x},\underline{u}_j)$

W obu przypadkach na potencjał sumacyjny złożyły się potencjały tych samych pięciu obiektów, choć przebieg uczenia był różny, ale nie musi to być regułą dla innych danych.

Przedstawiona wersja metody funkcji potencjałowych określana jest przez jej twórców [Ajzierman M.A., Bravierman E.M., Rozonoer A.I.,1970] jako realizacja maszynowa. Druga z dwóch opisanych w w/w pracy realizacji metody nazwana została perceptronową. Jest ona bardzo podobna do algorytmu korekcji błędów, opisanego w podrozdziale 3.1, tyle, że wykonywanego w nowej przestrzeni cech. Realizacja ta nie ma już charakteru minimalno-odległościowego.

Cenną zaletą metody funkcji potencjałowych, w odniesieniu do reguły *k*-NS jest to, że nie potrzebują wyboru najbliższych sąsiadów. Najbliżsi sąsiedzi mają i tak

największy głos w sposób bardziej naturalny, bo ich potencjał będzie miał najsilniejszy wpływ na wartość potencjału sumacyjnego.

W regule *k*-NS, odległości pomiędzy obiektem klasyfikowanym, a wszystkimi obiektami zbioru uczącego muszą być obliczane. Natomiast, w metodzie funkcji potencjałowych nie jest to niekonieczne. Liczone muszą być odległości tylko do tych obiektów, których prezentacja spowodowała konieczność korekty potencjału sumacyjnego. Większość obiektów zbioru uczącego może nie mieć udziału w potencjałach sumacyjnych stanowiących funkcje dyskryminacyjne klasyfikatora. Poważną wadą metody funkcji potencjałowych jest zależność uzyskiwanego potencjału sumacyjnego od uporządkowania zbioru uczącego.

W przeciwieństwie do reguły *k*-NS dla metody funkcji potencjałowych z uczeniem dość kosztowna byłaby implementacja metody minus jednego elementu celem oceny jakości klasyfikacji. Natomiast, nie ma problemu z metodą minus jednego elementu, jeśli zastosowana zostanie metoda funkcji potencjałowych bez uczenia.

Są liczne dziedziny zastosowań metod rozpoznawania, w których zebranie dostatecznie licznego zbioru obiektów o znanej przynależności jest trudne, tak ze względu na koszty jak i na czas potrzebny do jego zgromadzenia. Ma to miejsce głównie w zastosowaniach biomedycznych oraz w ekologii. Dlatego warto zbiory te wykorzystywać w możliwie najefektywniejszy sposób. Propozycję nowego podejścia do konstrukcji i oceny jakości klasyfikatora zawiera następny podrozdział.

4.9. Nowy schemat oceny jakości klasyfikacji

W podrozdziale 2.1 omawiane były metody oceny jakości klasyfikacji. Przy okazji omawiania warunku zatrzymania uczenia klasyfikatora, działającego na podstawie reguły decyzyjnej wykorzystującej potencjały sumacyjne, zostało zasugerowane użycie dodatkowego zbioru *W* zwanego walidacyjnym. Kolejne potencjały sumacyjne podczas uczenia konstruowane były na podstawie obiektów ze zbioru uczącego *U*, zaś wybór jednego z tych potencjałów odbywał się z użyciem zbioru walidacyjnego *W*. Ostateczna ocena klasyfikatora dokonywana była z zastosowaniem zbioru testującego *T*, użytego dokładnie jeden raz. Taki schemat konstrukcji i oceny klasyfikatora zamieszczony został w Tab. 4.9, w wierszu nr I . Zastosowanie metody minus jednego elementu, byłoby bardzo kosztowne. Wymagałoby tyle sesji uczenia, czyli wyznaczania potencjałów sumacyjnych $K_i(x)$ dla każdej z klas i, i=1,2,…,*nc*, jaka jest liczebność zbioru uczącego.

Wada ta nie występuje w przypadku reguły *k*-NS, ponieważ dość łatwo implementuje się metodę minus jednego elementu. W związku z tym zbiór obiektów ze znaną przynależnością wystarczy podzielić na dwie części, część uczącą $U=\{\underline{u}_j\}_{j=1}^{m}$

i część testującą T. Żeby wyznaczyć liczbę k najbliższych sąsiadów oferującą najmniejszą frakcję błędów, należy takie frakcje policzyć dla wszystkich możliwych wartości k, k=1,2,…,m-1.

Tab. 4.9. Zestawienie wybranych schematów konstrukcji i oceny klasyfikatorów

Wiersz tabeli	Metoda oceny jakości klasyfikacji	Zbiór konstrukcyjny	Zbiór do oceny w fazie uczenia	Zbiór oceny końcowej
I	Met. zbioru walidacyjnego	U	W	T
II	Met. minus jednego elementu	U	U	T
III	Met. minus jednego elementu	X	U	T

Standardowo w fazie wyznaczania k, czyli uczenia klasyfikatora k-NS, każdy obiekt $\underline{u}_i$ klasyfikowany jest regułą k-NS ze zbiorem odniesienia $O_i=U-\{\underline{u}_i\}$, tak jak wymaga tego metoda minus jednego elementu. Najbliżsi sąsiedzi wyszukiwani są w zbiorze O_i. Po ustaleniu wartości k, frakcję błędów klasyfikatora można ocenić klasyfikując każdy obiekt ze zbioru testującego T regułą k-NS ze zbiorem odniesienia $O=U$, tzn. najbliżsi sąsiedzi wyznaczani są ze zbioru U. Takie postępowanie zostało w Tab.4.9 oznaczone jako schemat nr II.

Kolejny schemat, proponowany przez autora niniejszej monografii, zaznaczony w Tab. 4.9, jako schemat nr III, wygodniej będzie opisać z użyciem zbioru $X=U\cup T$. Tym razem frakcja błędów oceniana jest też na zbiorze U, ale w procedurze minus jednego elementu każdy obiekt $\underline{u}_i\in U$ klasyfikowany jest regułą k-NS ze zbiorem odniesienia $O_i=X-\{\underline{u}_i\}$. Końcowe testowanie klasyfikatora odbywa się obecnie z zastosowaniem metody minus jednego elementu, co nie miało miejsca w schematach I i II. Inaczej mówiąc, jeżeli $T=\{\underline{t}_j\}_{j=1}^{mt}$, to każdy obiekt $\underline{t}_i\in T$ klasyfikowany jest wg reguły k-NS ze zbiorem odniesienia $O_i=X-\{\underline{t}_i\}$. Jest to więc oszacowanie frakcji błędów dla klasyfikatora, który będzie stosowany ze zbiorem odniesienia X, a nie z jego podzbiorem U. Powinno się to przełożyć na wyższą jakość klasyfikacji.

5. Podsumowanie i perspektywy

Preferowane właściwości metod klasyfikacji

Życzeniem każdego autora jest, aby jego praca przyniosła korzyści możliwie największej liczbie osób. A jeśli zawiera ona opis opracowanych przez niego metod, to również dużą satysfakcją będzie, gdy okaże się, że są one implementowane, a następnie stosowane. Wymienione przesłanki miały decydujący wpływ na kierunki badań prowadzonych przez autora, a w konsekwencji i na wybór zamieszczonych w monografii metod. Aby wymienione życzenie mogło się spełnić, to oprócz prostoty,

najważniejsza jest efektywność proponowanych metod. Obie te właściwości nie muszą się wzajemnie wykluczać. Prostota metod ułatwia autorowi nawiązanie współpracy ze specjalistami z innych dziedzin, w których one mogą znaleźć zastosowanie. Zaś powiązana z nią łatwość implementacji ułatwia napisanie programu, który będzie wolny od poważnych błędów, a jeśli się taki przy opracowaniu programu zdarzy, to będzie łatwiejszy do zlokalizowania. Autor ma nadzieję, że zaproponowane w monografii metody, będące wynikiem jego badań, w znacznym stopniu charakteryzują się wyżej wymienionymi właściwościami. Poniżej, opisane zostaną najważniejsze cechy niektórych z zamieszczonych w pracy metod.

Metody już opublikowane przez autora

Z już opublikowanych przez autora metod i zamieszczonych w niniejszej monografii, najwięcej zastosowań znalazła rozmyta reguła *k* najbliższych sąsiadów [Jóźwik A.,1983b], jako wynik jego współpracy z osobami zajmującymi się chemią fizyczną [Lesiak B., Jabłoński A., Zagórska M. and Jóźwik A., 1988; Lesiak B. and Jóźwik A., 2004; Lesiak B., Biliński A., Jóźwik A., 2005]. Jedną z jej zalet jest jakość klasyfikacji, która często jest wyższa niż dla standardowej reguły *k*-NS. Rozmyta wersja reguły *k*-NS stanowi uogólnienie standardowej reguły *k*-NS. Sama idea tej reguły powstała dzięki współpracy z osobami zajmującymi się spektroskopią elektronową. Pojawiły się zadania, w których nie sposób było zastosować standardową regułę *k*-NS, ani żadną inną metodę klasyfikacji ostrej, czyli nie rozmytej, jak np. rozpoznawanie udziałów w stopach metali, o czym już było wspomniane w rozdziale czwartym.

Rozmyta wersja reguły *k*-NS była też często wykorzystywana w zagadnieniach biomedycznych w ramach współpracy autora z biologami i lekarzami [Sokołowska B., Jóźwik A., Pokorski M., 2003]. Motywacją dla jej zastosowania, zamiast wersji standardowej, była wyższa jakość klasyfikacji. Ponadto, we wszystkich zastosowaniach, zwłaszcza biomedycznych bardzo ważny był problem selekcji cech. Była ona przeprowadzana z użyciem frakcji błędów, jako kryterium, liczonej metodą minus jednego elementu. Metodę tą bardzo wygodnie jest stosować w połączeniu z regułą *k*-NS, wyznaczając dla każdej z ocenianych kombinacji cech optymalną liczbę *k*. Rozmyta reguła była szczegółowo analizowana przez innych autorów w publikacji [Bezdek J.C., Chuah S.K., Leep D. ,1986], którzy również zauważyli jej użyteczność, szczególnie w przypadkach małych zbiorów uczących. Warto podkreślić, że wachlarz zadań, w jakim obie wersje reguły *k*-NS, standardowa i rozmyta, mogą być użyteczne jest bardzo szeroki, a założenia ich stosowalności są raczej słabe.

Następną opublikowaną metodą zaproponowaną przez autora, mającą charakter uniwersalny, czyli możliwą do zastosowań w wielu różnych zadaniach klasyfikacji, jest

metoda konstrukcji klasyfikatorów polegająca na wyznaczaniu obszarów klas i obszarów pokrywania się klas [Jóźwik A., Serpico S. and Roli F.,1998]. Podobnie jak reguła k-NS, klasyfikator wykorzystujący obszary klas może mieć duże znaczenie praktyczne, głównie w sytuacjach, gdy inne klasyfikatory, w tym klasyfikator *k*-NS, oferują wysokie prawdopodobieństwo mylnej decyzji, czyniąc się bezużytecznymi. Wydzielenie obszaru pokrywania się klas daje w tej sytuacji możliwość zbudowania klasyfikatora użytecznego, gdyż część obiektów spoza obszaru pokrywania się klas zostanie sklasyfikowana z akceptowalną wiarygodnością [Jóźwik A., Stawska Z. Grabowski M., Filipiak K., Rudowski R., Opolski G., 2003].

Klasyfikator wielostopniowy [Jóźwik A. i Stawska Z., 2000], zbudowany z wykorzystaniem obszarów klas, pozwala też przyspieszyć klasyfikację. Znaczna część klasyfikowanych obiektów, która znajdzie się w obszarze tylko jednej klasy, zostanie szybko sklasyfikowana, a nieco bardziej złożona reguła *k*-NS lub jej wersja rozmyta, będzie zastosowana tylko dla obiektów *trudniejszych,* w końcowym etapie klasyfikacji. Przyspieszenie, w porównaniu ze standardową regułą *k*-NS, nastąpi również dlatego, że wyznaczanie optymalnego *k* ograniczone zostanie tylko do obszaru pokrywania się klas. Ponadto, może zaoferować wyższą jakość klasyfikacji, ponieważ parametr *k*, czyli liczba najbliższych sąsiadów będzie dobrana tylko dla części zbioru uczącego, zawartego w obszarze pokrywania się klas.

Kolejne już opublikowane metody i zamieszczone w niniejszej monografii, opracowane przez jej autora, odnosiły się do redukcji i kondensacji zbiorów odniesienia dla reguły 1-NS. Wśród algorytmów redukcji był to algorytm obiektów wzajemnie najbliższych [Jóźwik A., Kieś P., 2005]. Natomiast do kondensacji zbiorów odniesienia zaproponowany został algorytm hiperpłaszczyzn tnących [Jóźwik A., Serpico S. B. and Roli F. 1995; Chen C. H. and Jóźwik A., 1996]. Zarówno redukcja jak i kondensacja są szczególnie użyteczne w przypadku dużych zbiorów danych i były już przez autora stosowane w przypadku zbiorów uczących o liczebnościach rzędu kilku tysięcy, a niektóre z nich nawet dla zbiorów zawierających kilkadziesiąt tysięcy obiektów.

Niniejsza monografia zawiera również propozycję, już publikowaną przez autora [Jóźwik A., 2004], bardziej efektywnego wykorzystania zbioru obiektów o znanej przynależności do klas. Jest to nowy schemat oceny jakości klasyfikacji, który może być szczególnie użyteczny w przypadku małych zbiorów uczących. Wyliczane wg niego frakcje błędów powinny być dokładniejsze, ponieważ są wyznaczane na podstawie liczniejszych zbiorów odniesienia, obejmujących zbiór uczący i testujący, tzn. będących ich mnogościową sumą.

Wymienione dotąd metody, mają charakter uniwersalny, czyli nie wymagają spełnienia rygorystycznych założeń przez zbiory uczące, jak to ma miejsce w

przypadku algorytmów wykorzystujących hiperpłaszczyzny rozdzielające. Wyznaczanie hiperpłaszczyzn rozdzielających nie będzie użyteczne, jeśli rozkład obiektów ze zbioru uczącego będzie zbyt złożony, np. gdy obiekty jednej klasy będą znajdować się wewnątrz pewnej hiperkuli, a obiekty innej klasy równomiernie rozłożone na zewnątrz tej hiperkuli. Ale pomimo to, wyznaczanie hiperpłaszczyzn rozdzielających warte jest zainteresowania, gdyż nierzadko zdarzają się zbiory uczące, dla których takie podejście może okazać się efektywne.

Do metod już opublikowanych przez autora niniejszej monografii należy też rekursywny algorytm badania rozdzielności liniowej dwóch zbiorów zaproponowany w dwóch różnych wersjach, raz z zastosowaniem rozwiązywania układu równań liniowych metodą eliminacji zmiennych, a drugi raz z użyciem ortogonalizacji. Obie te wersje [Jóźwik A., 1983a; 1998a] różnią się sposobami wyznaczania pierwszego przybliżenia poszukiwanego rozwiązania, czyli hiperpłaszczyzny startowej.

Metody niepublikowane wcześniej przez autora

Monografia zawiera też nowe metody, jeszcze niepublikowane. Do nich należy skorygowana reguła *k*-NS, która może mieć duże znaczenie praktyczne, zwłaszcza w przypadku jednoczesnego zastosowania korekcji standaryzacji cech, liczenia błędu klasyfikacji i macierzy przekłamań. Głównym zastosowaniem tej reguły mogą być zadania konstrukcji klasyfikatora, nie tylko w przypadku zbiorów uczących z brakującymi wartościami cech, ale także zadania, w których proporcje liczebności klas w zbiorze uczącym nie odpowiadają rzeczywistości.

Nową ideą jest reguła hiperkul, jako metoda redukcji zbioru odniesienia wraz z jednoczesną zmianą reguły decyzyjnej, określona w niniejszej pracy jako reguła hiperkul. Polega ona na pokryciu zbioru uczącego hiperkulami, które mogą nachodzić na siebie, jeśli dotyczą tej samej klasy. Wszystkie kule są jednorodne, czyli każda z nich zawiera obiekty tylko jednej klasy. Obszar przestrzeni cech pokryty przez hiperkule jednej klasy jest rozłączny z obszarem przestrzeni cech pokrytym przez hiperkule dowolnej innej klasy. Jej skuteczność jest bardzo zależna od rozkładu klas w zbiorze odniesienia i z pewnością jest bardzo wrażliwa na obiekty odstające, czyli pojedyncze obiekty otoczone obiektami z innej klasy. Można temu zaradzić edytując zbiór uczący, np. usuwając z niego wszystkie obiekty, które w charakterze najbliższego sąsiada mają obiekt z innej klasy. Operację ta można powtórzyć wielokrotnie. Zbiór uczący zostanie w ten sposób oczyszczony z obiektów odstających.

Opracowany wcześniej rekursywny algorytm badania rozdzielności liniowej dwóch zbiorów został uzupełniony o nowy etap. On sam stanowi pierwszy etap i pozwala znaleźć hiperpłaszczyznę odpowiadającą słabej rozdzielności lub wykryć, że taka hiperpłaszczyzna nie istnieje. Jeśli pierwszy etap da odpowiedź twierdzącą, to dodany

etap umożliwi zbadanie, czy badane zbiory są rozdzielne z zadanym prześwitem. Przyjmując bardzo małą wartość prześwitu, można rozstrzygnąć, czy badane zbiory są ściśle liniowo rozdzielne. Natomiast, stosując tylko drugi etap kilkakrotnie i dobierając różne wartości prześwitu można wyznaczyć hiperpłaszczyznę rozdzielającą bliską optymalnej.

Nowym wynikiem jest też zaproponowanie algorytmu konstruowania klasyfikatora na podstawie zbioru uczącego edytowanego dla liniowej rozdzielności.

W niniejszej monografii zaprezentowana została zmodyfikowana idea wykorzystania sztucznych cech w zastosowaniu do redukcji wielkości zbioru odniesienia dla reguły najbliższego sąsiada. W poprzedniej wersji stosowano selekcję sztucznych cech do redukcji zbioru odniesienia wykorzystywanego w oryginalnej przestrzeni cech. W nowej wersji klasyfikator działa w przestrzeni sztucznych cech, czyli każdy nowo klasyfikowany obiekt musi być przekonwertowany do nowej przestrzeni cech. Korzyść z takiego podejścia to wyższa jakość klasyfikacji przy nieznacznym koszcie obliczeniowym, wynikającym z konwersji klasyfikowanych obiektów do nowej przestrzeni cech.

Planowane dalsze badania

1. Problem konstrukcji klasyfikatora, w przypadku zbioru uczącego z brakującymi wartościami cech może być jeszcze lepiej rozwiązany, jeżeli zastosowana zostanie struktura równoległa złożona z klasyfikatorów dwudecyzyjnych, z których każdy będzie działał wg skorygowanej reguły *k*-NS. Należy spodziewać się mniejszej liczby obiektów ze zbioru uczącego, które nie brałyby udziału w konstruowaniu klasyfikatora. Poprawa jakości klasyfikacji wydaje się być intuicyjnie oczywista, ale warto zaplanować serie eksperymentów, na kilku dostępnych w Internecie i zróżnicowanych zbiorach danych. Eksperymenty te mogłyby być przeprowadzone dla różnych proporcji liczb obiektów, w których wystąpiły braki wartości cech, w odniesieniu do wielkości zbioru uczącego.

2. W strukturze równoległej z Rys. 1.2, głosowanie klasyfikatorów składowych może polegać na zliczaniu głosów jako wyników klasyfikacji ostrej albo rozmytej, w której głosy najbliższych sąsiadów w klasyfikatorach składowych mogą być rozproszone pomiędzy wszystkie rozważane klasy. Nie zostało jeszcze rozstrzygnięte, które z tych dwóch rodzajów głosowań, przy założeniu, że ostatecznie chodzi o klasyfikator produkujący decyzje ostre, oferuje mniejsze frakcje błędów.

3. Sposób wyznaczania obszarów klas na podstawie relacji 4.26 i 4.27 z pewnością może być ulepszony. Warto przemyśleć inne, choć podobnego rodzaju, reguły określania obszarów klas. Nie wykluczone, że zamiast obliczania wartości e_i wg wzoru

4.26, czyli równej największej odległości do najbliższego sąsiada z tej samej klasy, lepiej byłoby stosować średnią z takich wartości policzonych dla k najbardziej odległych najbliższych sąsiadów, co spowodowałoby zmniejszenie wielkości obszarów klas i wielkości obszarów pokrywania się klas, a w konsekwencji i liczb zawartych w nich obiektów. Poza tym taki sposób określania obszarów klas byłby mniej wrażliwy na obecność obiektów nietypowych. Oczywiście, także i relacja 4.27 powinna ulec podobnej modyfikacji. Weryfikacja efektywności wymienionej możliwych modyfikacji wzorów 4.26 i 4.27 mogłaby także być przeprowadzona eksperymentalnie na zbiorach internetowych, aby można było porównać uzyskane wyniki z już istniejącymi dla tych zbiorów w literaturze.

4. Jeśli chodzi o redukcję zbioru odniesienia z wykorzystaniem sztucznych cech, to warto rozważyć inne procedury, niż algorytm Tomeka, do ekstrakcji sztucznych cech. Należałoby także rozważyć, czy w przypadku, gdy nowo klasyfikowany obiekt znajdzie się w pustym obszarze, co da się stwierdzić, jeśli odległość do jego najbliższego sąsiada będzie większa o zera, nie warto przydzielać mu tej samej klasy, jaką ma najbliższy niepusty obszar.

5. Kolejnym tematem badawczym może być wykorzystanie sztucznie utworzonych cech w algorytmie hiperpłaszczyzn tnących, w przypadku, gdy algorytm ten będzie użyty z opcją automatycznego wyznaczania wynikowego zbioru skondensowanego. Każdy z uzyskanych wtedy podzbiorów byłby jednorodny, bo zawierałby obiekty tylko jednej klasy. Wyznaczane w tym algorytmie hiperpłaszczyzny można wykorzystać do zdefiniowania sztucznych cech w identyczny sposób, jak to było zaproponowane w podrozdziale 4.6. Hiperpłaszczyzny tnące dzielą przestrzeń cech na jednorodne lub puste obszary w przestrzeni cech. Obiektowi przypisywana byłaby wówczas taka klasa, z jakiej pochodzą obiekty obszaru, w którym się on znajdzie. Jeśli zaś znajdzie się w obszarze pustym, to powinien zostać przypisany klasie najbliższego obszaru. W istniejącej wersji algorytmu o klasie decyduje najbliższy środek ciężkości odpowiedniego podzbioru.

6. Inna modyfikacja algorytmu hiperpłaszczyzn tnących, która powinna być zbadana, może polegać na zastosowaniu innego sposobu podziału tworzonych podzbiorów w wyniku podziałów hiperpłaszczyznami tnącymi. W przypadku dwóch klas hiperplaszczyzny tnące mogłyby być konstruowane nie w oparciu o pary obiektów najdalszych lub wzajemnie najdalszych (co nie jest tym samym), ale o pary środków ciężkości klas obliczanych dla obiektów znajdujących się wewnątrz przeznaczonego do podziału kolejnego podzbioru. W zadaniach z większą liczbą klas, hiperpłaszczyzny tnące mogłyby oddzielać najliczniejszą klasę podzbioru od pozostałych klas dzielonego podzbioru, w taki sam sposób jak w sytuacji dwóch klas.

7. Reguła hiperkul, pierwszy raz opisana w niniejszej monografii, jak już wyżej zostało to zaznaczone, może służyć jednocześnie redukcji zbioru odniesienia i selekcji cech. Jej kryterium byłaby liczba wyznaczonych hiperkul pokrywających cały zbiór uczący, po odrzuceniu hiperkul zbędnych, w stosunku do liczebności zbioru uczącego. Godnym eksperymentów jest rozważenie zależności liczby otrzymywanych hiperkul od rodzaju użytej miary odległości.

8. Istnieją też możliwości modyfikacji rekursywnego algorytmu badania rozdzielności liniowej dwóch zbiorów w taki sposób, aby w przypadku ścisłej rozdzielności liniowej badanych zbiorów wyznaczona była para hiperpłaszczyzn rozdzielających maksymalnie od siebie odległych, czyli ostatecznie hiperpłaszczyzna rozdzielająca z maksymalnym marginesem. Podobnie, warto przeprowadzić badania nad możliwością stworzenia takiej modyfikacji, aby można było badać rozdzielność liniową z nakładką, jak to miało miejsce w przypadku algorytmu korekcji błędów omawianego w podrozdziale 3.1. O ile algorytm korekcji mógł być racjonalnie stosowany niezależnie od tego czy badane zbiory są czy też nie liniowo rozdzielne, to algorytm rekursywny by mógł spełnić swoje zadanie, musi zapewnić uzyskanie odpowiedniego rozwiązania, czyli hiperpłaszczyzny rozdzielającej, jeśli takie istnieje. Tak więc, nie są problemem definicje obu wyżej sugerowanych modyfikacji, ale przeprowadzenie dowodu ich zbieżności, gdy faktyczna rozdzielność liniowa z prześwitem i odpowiednio z nakładką ma miejsce.

9. Algorytm edycji zbioru uczącego dla liniowej rozdzielności z podrozdziału 3.5 typuje kolejno obiekty zbioru uczącego przeznaczone do usunięcia uzyskując w końcu możliwość wyznaczenia hiperpłaszczyzny rozdzielającej, ale wśród hiperpłaszczyzn prostopadłych do odcinka łączącego środki ciężkości klas, obliczonych już dla pomniejszonych zbiorów. Nie można wykluczyć, że lepiej byłoby badać rozdzielność liniową już po każdym zmniejszeniu, któregokolwiek z badanych zbiorów, nawet o jeden obiekt. Takie postępowanie prowadziłoby do zmniejszenia liczby usuwanych obiektów.

10. Frakcja błędów wyznaczona metodą minus jednego elementu może służyć do wyznaczenia optymalnej liczby k w regule k-NS. W procedurze tej każdy obiekt u zbioru uczącego U klasyfikowany jest na podstawie zbioru odniesienia $O=U-\{u\}$ złożonego z pozostałych obiektów. Zwiększa to szansę na to, aby najbliższym w sąsiedztwie był obiekt z jednej z przeciwnych klas. Frakcje błędów mogą więc być zawyżone. Eksperymenty opisane w publikacji [Jóźwik A., 2002] pokazały, że optymalne wartości k w regule k-NS, ustalane na podstawie frakcji błędów wyznaczanych metodą zbioru walidacyjnego były wyższe niż w przypadku frakcji błędów określanych metodą minus jednego elementu. Jednym z powodów takiego

zjawiska może być w/w fakt przeszacowywania frakcji błędów, co sprawia, że w metodzie minus jednego elementu mniejsze wartości frakcji błędów były uzyskiwane dla mniejszych wartości *k*. Rzeczywiście, frakcje błędów otrzymywane dla metody minus jednego elementu były wyższe. Eksperymenty były przeprowadzane na małych sztucznie generowanych zbiorach uczących, dwuwymiarowych i zawierających w jednej serii po 10 obiektów, a w drugiej po 40 obiektów w klasach. Zbiory uczące i walidacyjne były równoliczne. Wątpliwości budzi fakt, że proporcje pomiędzy średnimi wielkościami liczb *k* wyznaczanymi metodą zbioru walidacyjnego i metodą minus jednego elementu były prawie takie same dla obu w/w serii eksperymentów. Ta kwestia wymaga jeszcze przemyśleń i ponownego eksperymentalnego zweryfikowania z użyciem dostępnych zbiorów internetowych.

6. Cytowana literatura

Ainslie M.C. and Sanchez J.S. (2002), *Space partitioning for instance reduction in lazy learning algorithms*, In 2nd Workshop on Integration and Collaboration Aspects of Data Mining, Decision Suport and Meta-Learning, pp. 13-18

Ajzierman M.A., Bravierman E.M., Rozonoer A.I. (1970), *Mietod potiencjalnych funkcji w tieorii obuczienia maszin*, Nauka

Bayes T. (1763), *An essay towards solving a problem in the doctrine of chances*, Philosophical Transactions of the Royal Society, vol. 53, pp. 370-418

Bezdek J.C., Chuah S.K., Leep D. (1986), *Generalized k-NN rule.* Fuzzy sets and Systems, vol. 18, pp. 237-256

Bobrowski L., Niemiro W. (1984), *A method of synthesis of linear discriminant function in the case of nonseparability*, Patter Recognition vol. 17, no. 2, pp. 205-210

Carpenter G. A., Grossberg S. (1996), *Learning, categorization, rule formation, and prediction by fuzzy neural networks*, in the book "Fuzzy logic and neural network handbook, edited by C.H. Chen, McGraw-Hill Series on Computer Engineering, New York, pp. 1.3-1.45

Cendrowska D. (2005), *Konstrukcja klasyfikatora obiektów z wykorzystaniem algorytmu badania rozdzielności liniowej dwóch zbiorów*, Praca doktorska obroniona w Instytucie Podstawowych Problemów Techniki PAN, promotor: W. Kosiński

Chang C.L. (1974), *Finding prototype for nearest neighbor classifiers*, IEEE Transactions on Computers, vol. 23 (Corresp.), pp.1179-1184

Devijver P.A., Kittler J. (1982), *Pattern recognition*: *A statistical approach*, Prentice Hall, London.

Duda R.O., Hart P.E., Stork D.G. (2001), *Pattern classification*, Wiley Interscience, New York

Fix E., Hodges J.L. (1952), *Discriminatory analysis: nonparametric discrimination small sample performance*, Project 21-49-004, report number 11, USAF school of aviation medicine, Randolph Field, Texas, pp. 280-322

Gates G. W. (1972), The reduced nearest neighbor rule, IEEE Transactions on Information Theory, vol. 18 (Corresp.), pp. 431-433.

Chen C. H. and Jóźwik A. (1996), *A sample set condensation algorithm for the class sensitive artificial neural network*, Pattern Recognition Letters, vol. 17, pp. 819-823

Gowda K. C. and Krishna G. (1979), *The condensed nearest neighbor rule using the concept of mutual nearest neighborhood*, IEEE Transactions on Information Theory, vol. 25 (Corresp.), pp. 488-490.

Grabowski S. (2003), *Konstrukcja klasyfikatorów minimalno-odległościowych o strukturze sieciowej*, rozprawa doktorska obroniona na AGH, promotor: D. Sankowski

Hart P.E. (1968), *The condensed nearest neighbor rule*, IEEE Trans. on Information Theory, vol. 14 (Corresp.), pp. 515-516

Ho Y.C., Kashyap R.L. (1965), *An algorithm for linear inequalities and its application*, IEEE Transactions on Electronic Computers, v. 14, pp. 683-688

Jajuga K. (1990), *Statystyczna teoria rozpoznawania obrazów*, PWN, Warszawa.

Jóźwik A.(1981), *A double stage algorithm for the investigation of linear separability of two sets in pattern classifying problems*, IV Polish-Italian Symposium , Ischia, 1978, Quaderni de La Richerca Scientifica, v. 108, pp. 41-45

Jóźwik A. (1983a), *A recursive method for the investigation of the linear separability of two sets*, Pattern Recognition, vol. 16, no. 4, pp. 429-431

Jóźwik A. (1983b), *A learning scheme for a fuzzy k-NN rule*, Pattern Recognition Letters 1, pp. 287-289

Jóźwik A. (1994), *Pattern recognition method based on k nearest neighbor rule*, Journal of Communications, vol. XLV, July-August, pp. 27-29

Jóźwik A. (1998a), *Algorytm badania liniowej rozdzielności dwóch zbiorów i perspektywy jego wykorzystania do konstrukcji klasyfikatorów*, VI Konferencja "Sieci i Systemy Informatyczne - teoria, projekty, wdrożenia", Łódź, październik 1998, materiały konferencyjne, str. 311-316

Jóźwik A. (2002), *Badanie właściwości dwóch metod oceny jakości klasyfikatorów typu k-najblizszych sąsiadów*", Materiały X Konferencji *Sieci i Systemy Informatyczne*, str., 537-548

Jóźwik A. (2004), *Nowy schemat testowania klasyfikatorów*, Materiały XII Konferencji *Sieci i Systemy Informatyczne*, Łódź ,str., 425-430

Jóźwik A. (2005), *Minimalno-odległościowe i inne metody konstrukcji klasyfikatorów odcinkowo-liniowych,* Prace Inst. Biocybernetyki i Inżynierii Biomedycznej nr 64.

Jóźwik A. (2006), *Combining reference set condensation and reduction algorithms for controlling the compromise between the reference set size and classification quality*, Materiały XIV Konferencji Sieci i Systemy Informatyczne, Łódź, str., 213-215

Jóźwik A., Chmielewski L., Skłodowski M., Cudny W. (2001), *A proposition of the new feature space and its use to construction of a fast minimum distance classifier*, w książce Komputerowe Systemy Rozpoznawania (referaty II Konferencji KOSYR2001), Wrocław, pp. 381-386

Jóźwik A., Janecki J. and Demczuk M. (1998), *A multistage NN type classifier based on class overlap rate minimization and its application to cardio-circulatory events prediction*, Proceedings of the IV National Conference on Biocybernetics and Biomedical Engineering , Zwierzyniec, September, pp. 47-51

Jóźwik A., Kieś P. (2005), *Reference set reduction for 1-NN rule based on finding mutually nearest and mutually furthest pairs of points*, Advances in Soft Computing, Computer Recognition Systems, Springer-Verlag, Berlin-Heidelberg, pp. 195-202

Jóźwik A., Serpico S., Roli F. (1998), *A parallel network of modified 1-NN and k-NN classifiers – application to remote-sensing image classification*, Pattern Recognition Letters 19, pp. 57-62

Jóźwik A., Serpico S. B. and Roli F. (1995), *Condensed Version of the k-NN rule remote sensing image classification*, Image and Signal Processing for Remote Sensing II, EUROPTO Proceedings, SPIE, vol. 2579, pp. 196-198

Jóźwik A., Serpico S. and Roli F. (1998), *A parallel network of modified 1-NN and k-NN classifiers -application to remote-sensing image classification*, Pattern Recognition Letters 19, pp. 57-62

Jóźwik A. i Stawska Z. (2000), *Wielostopniowy klasyfikator typu najbliższy sąsiad z każdej klasy*, Materiały VIII Konferencji „Sieci i Systemy Informatyczne", Łódź, str. 339-346

Jóźwik A., Stawska Z. Grabowski M., Filipiak K., Rudowski R., Opolski G. (2003), *Distance based classifiers and their use to analysis of data concerned acute coronary syndromes*, w książce Komputerowe Systemy Rozpoznawania (referaty III Konferencji KOSYR2003), pp. 369-375

Jóźwik A., Vernazza G. (1988), Recognition of leucocytes by a parallel *k*-NN classifier, Lecture Notes of the `ICB Seminar, pp. 138-153

Keller J.M., Gray M.R., Givens J.A., (1985), *A fuzzy k-nearest neighbour algorithm*, IEEE Trans. on Systems Man and Cybernetics, vol. SMC-15, pp 580-585.

Kohavi R., (1995), *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*, http://robotics.stanford.edu/%7Eronnyk/accEst.pdf

Koronacki J., Ćwik J. (2005), *Statystyczne systemy uczące się*, WNT.

Koziniec B.N. (1973): Riekurientnyj algoritm razdzielenia dwoch wypukłych obołocziek, w Algoritmy obuczienia w razpoznawanii obrazow, (pod redakcją W. N. Vapnika), Sowieckoje Radio, Moskwa, str. 43-50 (w jęz. rosyjskim)

Kurzyński M. (1997), *Rozpoznawanie obiektów. Metody statystyczne*. Oficyna Wydawnicza Politechniki Wrocławskiej.

Lachenbruch P. A.(1965), *Estimation of Error Rates in Discriminant Analysis*, Ph.D. dissertation, University of California, Los Angeles, Chapter 5.

Lesiak B., Biliński A., Jóźwik A. (2005), *Segregation in CuPd alloys studied by x-ray photoelectron spectroscopy using lineshape analysis and the fuzzy k-nearest neighbour rule*, Polish J. Chem. 79, p.1365

Lesiak B., Jabłoński A., Zagórska M. and Jóźwik A. (1988), *Identification of synthetic metals from the shape of the carbon KLL spectra by pattern recognition method*, Surface and Interface Analysis, vol. 12, pp. 461-467

Lesiak B. and Jóźwik A. (2004), *Quantitative analysis of AuPd alloys from the shape of XPS spectra by the fuzzy rule*, Surface and Interface Analysis, vol. 36, pp. 793-797

Mangasarian O.L. (2000): *Generalized Support VectorMachines, Advances in Large Margin classifiers*, MIT Press, available at ftp://ftp.cs.wisc.edu/math-prog/tech-reports/98-14.ps, pp. 135–146

Nilsson N. (1965), *Learning machines*, McGraw-Hill, New York.

Raniszewski M. (2009), *Metody silnej redukcji i edycji zbioru odniesienia dla reguły typu najbliższy sąsiad*, Praca doktorska, obroniona na Politechnice Łódzkiej (KIS), promotor: A. Jóźwik

Rychlik T. (2012), *Zmodyfikowany dwu-decyzyjny klasyfikator ze standaryzacją*, Praca magisterska, obroniona na Uniwersytecie Łódzkim (WFiIS), promotor A. Jóźwik

Sanchez J.S. (2004), *High training set size reduction by space partitioning and prototype abstraction*, Pattern Recognition 37, pp. 1561-1564

Siedlecki W. (1994), *A formula for multi-class distributed classifiers*, Pattern Recognition Letters, Volume 15, Issue 8, August 1994, Pages 739-742

Sierszeń A. (2009), *Metody kondensacji zbioru odniesienia dla reguł decyzyjnych opartych na funkcji odległości*, Praca doktorska, Politechnika Łódzka (KIS), promotor: A. Jóźwik

Sokołowska B., Jóźwik A., Pokorski M. (2003), *A fuzzy-classifier system to distinguish respiratory pattern envolving after diaphragm paralysis in cat*, Japanese Journal of Physiology, vol. 53, pp. 301-307

Stąpor K. (2005), *Automatyczna klasyfikacja obiektów*, EXIT, Warszawa

Sturgulewski Ł.(2008), *Algorytmy badania ścisłej rozdzielności liniowej dwóch zbiorów*, Praca doktorska, Politechnika Łódzka (KIS), promotor: A. Jóźwik

Tomaszewski W.P.(2013), *System rozpoznawania twarzy działający w oparciu o zmodyfikowany wielodecyzyjny klasyfikator minimalno-odległościowy*, Praca inżynierska, wykonana w WSKSiM, promotor: G. Osiński

Tadeusiewicz R., Flasiński M. (1991), *Rozpoznawanie obrazów*, PWN, Warszawa

Tomek I. (1977), *Two modifications of CNN*, IEEE Trans. Systems, Man, and Cybernetics, vol. 7, no. 2, pp. 92-94

Toussaint G. T. (1994), *A counterexample to Tomek's consistency theorem for a condensed nearest neighbor decision rule*, Pattern Recognition Letters, vol.15, pp.797-801

Vapnik V.N. (2000), *The Nature of Statistical Learning Theory*. Springer, New York.

Vapnik V.N., Chervonenkis A., J. (1974), *Teoria rozpoznawania obrazow*, Nauka, Moskwa (w jęz. rosyjskim)