

Magdalena TOPCZEWSKA<sup>1</sup>, Kurt FRISCHMUTH<sup>2</sup><sup>1</sup> POLITECHNIKA BIAŁOSTOCKA, WYDZIAŁ INFORMATYKI, ul. Wiejska 45a, 15-351 Białystok<sup>2</sup> POLITECHNIKA KOSZALIŃSKA, WYDZIAŁ MECHANICZNY, ul. Raclawicka 15-17, 75-620 Koszalin

## Strategie klasyfikacji oparte na dipolach

Dr inż. Magdalena TOPCZEWSKA

Ukończyła studia w Instytucie Informatyki Politechniki Białostockiej w Białymstoku w roku 1997. Obroniła pracę doktorską w roku 2009. Zatrudniona jest na stanowisku adiunkta na Wydziale Informatyki Politechniki Białostockiej. Jej zainteresowania naukowe obejmują eksplorację oraz statystyczną analizę danych, a także metody klasyfikacji i wizualizacji danych.



e-mail: m.topczewska@pb.edu.pl

Prof. nadzw. dr hab. inż. Kurt FRISCHMUTH

Ukończył studia na Wydziale Matematyki, Informatyki i Mechaniki Stosowanej Uniwersytetu Warszawskiego w roku 1980. Obronił pracę doktorską w roku 1983 w IPPT PAN. Uzyskał stopień doktora habilitowanego na Uniwersytecie w Rostocku w roku 1990. Wykłada na Politechnice Koszalińskiej od roku 2001. Jego zainteresowania naukowe to metody numeryczne w zastosowaniach technicznych, w szczególności symulacja procesów nieliniowych. Zajmował się modelowaniem tarcia, zużycia i zjawisk falowych.



e-mail: kurt.frischmuth@uni-rostock.de

## Streszczenie

W pracy przedstawiona została metoda klasyfikacji oparta na kryterium dipolowym. Przeprowadzone testy numeryczne skłoniły autorów do modyfikacji początkowego kryterium prezentowanego w pracy [1] w taki sposób, by znaleziona hiperpłaszczyzna separowała jak największą liczbę obiektów należących do różnych klas, nie rozdzielając obiektów z tej samej klasy. Nowe podejście zostało przetestowane na wygenerowanych zbiorach danych, a otrzymane wyniki pozwoliły na implementację w postaci wywołań rekurencyjnych.

Słowa kluczowe: klasyfikacja, dipol, kryterium dipolowe.

## Dipole-based classification strategies

## Abstract

Finding a separating hyperplane for two classes by minimizing an error functional summing contributions for each poorly classified data point is by now common practice. Effective implementations in the framework of SVM or in terms of CPL functions are available and shown to work well for quite large sets of data. In this article a classification method based on dipoles is presented, which is a modification of a criterion given in [1]. We search for a hyperplane cutting between classes and not through their middle. The crux is to define a suitable functional, which is small on lines with good separation power and little damage, easy to calculate and to minimize. We perform numerical tests and modify the criterion in a way that preserves the intention of finding cuts between classes (and not through their middle), which separate as many data points as possible. However, we do not *count* (in integers), but *weigh* by taking the distance to the wrong side. The approach was tested on some synthetic data sets using a recursive implementation.

Keywords: classification, dipole, dipole criterion.

## 1. Wstęp

Dysponując odpowiednim zestawem danych pomiarowych można budować systemy wspomaganie decyzji mające zastosowanie w wielu dziedzinach gospodarki i przemysłu. Przykładami mogą być: diagnozowanie stanu technicznego maszyn i przydzielanie ich do jednej z klas akceptacji, automatyczne rozpoznawanie celów na podstawie danych odczytanych z wielu czujników, systemy sterowania robotami i pojazdami bezzałogowymi, automatyczna detekcja zagrożeń osób i obiektów czy też wspomaganie diagnostyki medycznej. W systemach takich kluczową rolę odgrywają zastosowane metody klasyfikacji. Ich zadaniem jest przydzielanie obiektów do odpowiednich klas. Parametry klasyfikatora, czyli funkcji decyzyjnej na podstawie której nowe obiekty będą przydzielane do odpowiedniej klasy, znajdują się na podstawie zbioru danych zawierającego wartości atrybutów obiektów oraz etykiety klas, do których obiekty te należą.

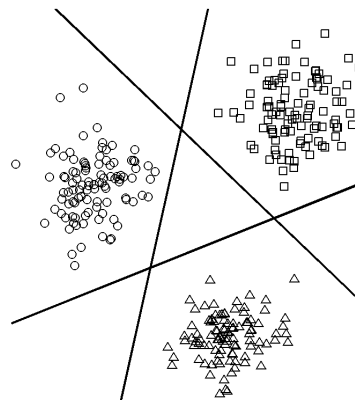
Przykładem podstawowym jest rozdzielanie obiektów należących do dwóch klas, a najprostszą postacią funkcji rozdzielającej

objekty jest prosta w przypadku dwuwymiarowym lub hiperpłaszczyzna w przypadku wielowymiarowym. W rzeczywistości mamy jednak najczęściej do czynienia z danymi, które nie są liniowo separowalne, czyli nie istnieje jedna hiperpłaszczyzna całkowicie rozdzielająca dane należące do dwóch klas. Liczba klas, do których przydzielane są obiekty także często przekracza liczbę klas dla przypadku podstawowego. Dlatego niezwykle istotnym zadaniem jest budowa klasyfikatorów w przypadku wieloklasowym.

## 2. Sformułowanie problemu

Niech będzie danych  $k$  klas  $C_l$ ,  $l=1, \dots, k$ , zawierających opis danych w postaci wektorów  $x \in R^n$  z wartościami wybranych atrybutów. Wektory te nazywane są wektorami cech lub wektorami atrybutów i stanowią jedyne źródło informacji o obiekcie.

Podstawowym problemem klasyfikacji jest zbadanie w jaki sposób na podstawie wartości atrybutów można określić przynależność obiektu do wybranej klasy. W przypadku klasyfikatora liniowego zakłada się, że funkcjonal  $\Phi: R^n \rightarrow R$ ,  $\Phi(x) = w^T x - b$  może być skonstruowany w taki sposób, by dla obiektów należących do klasy  $C_1$  otrzymać wartość funkcjonału  $\Phi(x) \geq 0$ , natomiast dla pozostałych obiektów  $\Phi(x) < 0$ . W przypadku dwuklasowym ( $k=2$ ) wystarczy jeden tego typu funkcjonal by oddzielić dwie klasy, jednak w przypadku, gdy  $k$  jest liczbą większą niż 2 występuje potrzeba skonstruowania listy takich funkcjonałów, by znając sekwencje znaków jednoznacznie można było wydzielić każdą klasę.



Rys. 1. Trzy klasy, każda odseparowana od pozostałych za pomocą prostej  
Fig. 1. Three classes, each separated from the others by a straight line

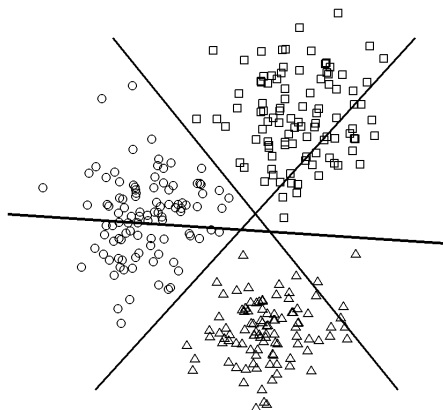
W przypadku gdy  $k=3$ , można skonstruować funkcjonal separujący wszystkie obiekty z pierwszej klasy od obiektów z pozostałych dwóch klas. Następnie można powtórzyć proces wyznaczający

analogicznie parametry kolejnych dwóch funkcjonałów – oddzielającego drugą klasę od pozostałych i ostatecznie trzecią klasę od pozostałych, co zostało zaprezentowane na rysunku 1. Innym bardziej efektywnym rozwiązaniem jest w kroku drugim oddzielenie klasy drugiej od trzeciej, podczas gdy pierwsza jest już oddzielona od pozostałych w pierwszym kroku. W pierwszym podejściu klasa pierwsza jest scharakteryzowana przez sekwencję znaków (1,0,0), druga przez sekwencję (0,1,0), natomiast trzecia przez (0,0,1). Podejście drugie identyfikuje klasę pierwszą przez (1,\*), drugą poprzez (0,1) zaś trzecią przez (0,0). Za pomocą znaku \* określana jest dowolna wartość funkcjonału.

a)



b)

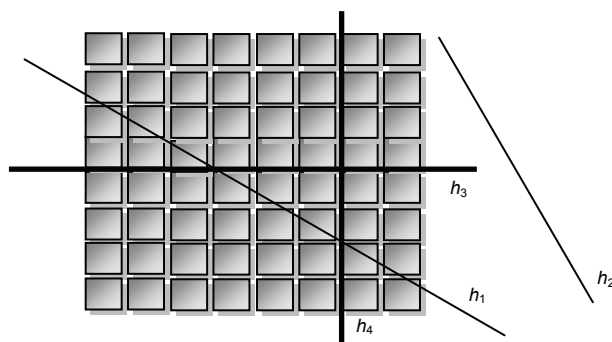


Rys. 2. Trzy klasy, każda odseparowana w kilku krokach  
Fig. 2. Three classes, each separated in several steps

Podejście drugie jest podejściem znacznie bardziej efektywnym – potrzeba jedynie dwóch funkcjonałów do rozdzielania klas, rys. 2a. Z drugiej strony, podejście to nie jest symetryczne, gdyż wyróżniona jest pierwsza klasa. Może się to wydawać wadą, otwiera to jednak nowe możliwości. Jeśli nie uda się oddzielić wybranej klasy od pozostałych, można rozpocząć ponownie podział przestrzeni wybierając inną klasę i próbować następnie oddzielać poprzednią klasę od mniejszego podzbioru danych. Na rysunku 2a klasa ulokowana w środku, której obiekty oznaczone zostały symbolem trójkąta, nie może zostać oddzielona od pozostałych klas w jednym kroku. Można oczywiście poszukiwać funkcjonału nieliniowego – w ogólnym przypadku może to jednak prowadzić do zbyt mocnego dopasowania klasyfikatora do danych [3].

Należy rozpatrzyć również sytuację, gdy żadna z klas nie może być odseparowana od pozostałych za pomocą jednego cięcia, rys. 2b. Linie podziału na rysunku 2b pokazują, że lepszym rozwiązaniem byłoby oddzielenie dwóch klas przy pominięciu pozostałych. W takim przypadku każda klasa byłaby charakteryzowana poprzez zapis (1,1,\*), (1,\*,1) lub (\*,1,1).

Wraz ze zwiększaniem liczby klas liczba możliwości ich podziału wzrasta. Możliwości te stają się również coraz bardziej złożone. Załóżmy, że danych jest  $k=64$  klas i że są one rozmieszczone w postaci pól szachownicy, rys. 3.



Rys. 3. 64 klasy rozmieszczone jak pola szachownicy  
Fig. 3. 64 classes located on a "chess board"

Oczywiste jest, że żadna z klas nie może być całkowicie oddzielona od pozostałych za pomocą jednego cięcia przy wykorzystaniu pojedynczego klasyfikatora liniowego ani funkcji kwadratowej, czy też innej funkcji gładkiej. Jedynie klasy znajdujące się w narożnikach mogą zostać odseparowane w pewnym stopniu od pozostałych. W takim przypadku zamiast próbować oddzielać jedną klasę od pozostałych, rozsądne wydaje się podjęcie próby rozdzielania grupy klas od siebie za pomocą prostych. Problemem jest fakt, że istnieje  $2^{64}$  możliwych podzbiorów, tak więc brak jest możliwości sprawdzania wszystkich podziałów i połączeń klas. Jako alternatywy można próbować poszukiwać linii podziału, dodając przy poszukiwaniach warunków na odpowiednią efektywność i jak najmniejszą liczbę błędów przy podziale obiektów należących do różnych klas.

Rozważając dane przedstawione na rysunku 3 można zauważyć, że linia  $h_1$  dzieląca klasy na szachownicy przecina 11 klas. Prosta  $h_2$  (znajdująca się najbardziej z prawej strony rysunku) nie przecina żadnej z klas, nie posiada jednak żadnej właściwości dyskryminacyjnej.

Cel, do którego należałoby dążyć to połączenie dużej zdolności dyskryminacyjnej przy jak najmniejszym stopniu rozdzielania obiektów wewnątrz klasy. Oznacza to dążenie do uzyskania jak najmniejszej liczby obiektów z tej samej klasy leżących po dwóch stronach hiperpłaszczyzny przy jednoczesnym zapewnieniu jak największej liczby par obiektów z różnych klas leżących po różnych stronach hiperpłaszczyzny. Obydwie pogrubione proste ( $h_3$  i  $h_4$ ) spełniają kryterium pierwsze, jednak prosta poziomo dzieląca szachownicę ( $h_3$ ) posiada dodatkową zaletę, polegającą na tym, że dzieli 32 klasy względem 32, podczas gdy pionowa ( $h_4$ ) oddziela tylko 16 klas od 48.

Jeśli podział nadal byłby kontynuowany aż do osiągnięcia obszarów zawierających obiekty należące tylko do jednej klasy, wówczas podział na równe części (w połowie szachownicy) byłby najlepszy. W kolejnych pięciu krokach można uzyskać podział na 16 klas, 8, 4, 2 i ostatecznie jedną klasę. Tak więc potrzeba  $\log_2(k)$  cięć, aby wydzielić dowolnie wybraną klasę. Można oczywiście zastosować inny kształt klasyfikatora, na przykład prostokątny [4], ale nie jest to przedmiotem tej pracy.

Podsumowując, istnieją dwie możliwości. Po pierwsze, można dokonać wstępnego podziału przestrzeni a następnie kontynuować podział dzieląc osobno otrzymane części. W tym przypadku jako pierwszy klasyfikator powinien zostać wybrany najlepszy, tzn. spełniający kryterium minimum globalnego dla badanego funkcjonału. Po drugie, można poszukiwać minimów lokalnych, co oznacza poszukiwanie klasyfikatorów separujących mniej różnych klas, ale nienaruszających warunku, by obiekty z tej samej klasy były po jednej stronie hiperpłaszczyzny. Oznacza to, że chcemy jednocześnie separować obie części, co jest znacznie trudniejsze w realizacji. Szachownica przedstawiona na rysunku 3 w przypadku drugiego wariantu może być podzielona na 64 obszary przynależności do klas za pomocą tylko 14 prostych, podczas gdy podejście pierwsze będzie wymagało znalezienia tych samych prostych dla różnych zestawów klas otrzymanych w poprzednich krokach algorytmu. Głównym problemem w przy-

padku drugiego podejścia jest to, że nie zawsze istnieje rozwiązanie, a jeśli nawet istnieje, nie jest oczywiste w jaki sposób uniezależnić poszukiwanie kolejnych klasyfikatorów od uzyskanych uprzednio podziałów przestrzeni.

Praca dotyczy pierwszego podejścia – podejmowana jest więc próba znalezienia najlepszego podziału przestrzeni w obrębie całego zbioru danych, a następnie kontynuuje się podział dzieląc osobno otrzymane części. Algorytm może być uruchamiany rekurencyjnie dla kolejnych części podzielonej przestrzeni. Najważniejsze jest jednak odpowiednie zdefiniowanie funkcjonu, który będzie przyjmował niewielkie wartości w przypadku, gdy znaleziony klasyfikator będzie charakteryzował się dużą zdolnością dyskryminacyjną, jednocześnie będzie zapewniał jak najmniejszą liczbę źle sklasyfikowanych obiektów, nie będzie zbyt złożony obliczeniowo oraz będzie dogodny do minimalizacji.

### 3. Konstrukcja funkcjonu

Niech cały zbiór danych będzie reprezentowany przez macierz  $D$  o wymiarach  $m \times (n+1)$ , zawierającą wartości atrybutów i indeksy klas. W przypadku analizy danych rzeczywistych mamy do czynienia ze skończonym zbiorem danych, to znaczy że zamiast pól reprezentujących klasy mamy do czynienia z grupami pojedynczych obiektów. Niech  $m$  oznacza liczbę obiektów w zbiorze danych, natomiast  $m_i$  moc zbioru  $C_i$ . Ogólnie należy również mieć na względzie rozproszenie obiektów, tak więc w zastosowaniach rzeczywistych rzadko możliwe jest uzyskanie 100% jakości klasyfikacji. Definiując funkcjon trzeba zapewnić, by w przypadku występowania w danych tego typu błędów, zniwelować efekt ich wpływu na zmianę położenia linii separującej obiekty.

W [1] zaprezentowane zostało pojęcie *dipoli* jako uporządkowanych par wektorów. Mogą być one rozpatrywane jako czyste lub mieszane, w zależności od tego czy obydwa obiekty tworzące dipol należą do tej samej klasy czy też nie. Istnieje więc  $k$  rodzajów dipoli czystych oraz  $k(k-1)$  rodzajów dipoli mieszanych. Dipole mieszane mogą mieć wpływ na wartość funkcjonu przyjmując różne znaki dla obydwu elementów składowych. Z drugiej strony, przyjmowanie różnych znaków w przypadku dipoli czystych będzie rozpatrywane jako błąd. Następnie można zliczać prawidłowe i błędne przypadki ustalając pewną równowagę pomiędzy wpływem dipoli czystych i mieszanych na wartość funkcjonu. Tak sformułowane zadanie jest jednak trudne do optymalizacji, gdyż funkcja celu przyjmuje jedynie całkowite wartości, jest więc nieciągła.

Ponadto, należałoby oszacować jak źle jest nieprawidłowe cięcie oraz jak dobre jest cięcie prawidłowe. Na przykład w przypadku dipola czystego  $(x,y)$  z  $\Phi(x)=0,001$  oraz  $\Phi(y)=1000$  istnieje potencjalne ryzyko, że niewielkie zakłócenie w  $\Phi$  może spowodować, że dipol stanie się mieszany, gdyż obiekt  $x$  znajdzie się po drugiej stronie hiperpłaszczyzny. Analogicznie, dipol mieszany przy niewielkim zakłóceniu może stać się czysty, jeśli dla jednego z jego obiektów składowych wartość  $|\Phi(x)|$  będzie mała (zbyt bliska zera). Ponownie przy niewielkich zakłóceniach obiekt może zostać usytuowany po niewłaściwej stronie hiperpłaszczyzny.

Mając na względzie przedstawiony powyżej opis, można zaproponować minimalizację kryterium następującej postaci

$$F(\Phi, D) = \sum_i \sum_j \sum_{x \in C_i} \sum_{y \in C_j} F_{ij}(\Phi, x, y; D), \quad (1)$$

gdzie  $F$  jest funkcją afinicznie liniowego funkcjonu  $\Phi$ , czyli poszukiwanego klasyfikatora oraz danych  $D$ . Przestrzeń wszystkich funkcjonów tego typu może być sparametryzowana przez wektor  $p=(w,b) \in R^{n+1}$ . Wartość  $F$  zależy oczywiście od danych, to znaczy od zbioru wszystkich wektorów cech oraz ich przynależności do  $k$  różnych klas.

Funkcja  $F$  jest obliczana jako podwójna suma (względem indeksów klas) wyrazów będących podwójną sumą składowych

postaci  $F_{ij}(x,y)$ . Każda taka składowa jest funkcją dwóch wartości rzeczywistych, które przyjmuje funkcjon  $\Phi$  na elementach dipola – obiekcie  $x$  i  $y$ . W przypadku gdy  $i=j$ , mamy do czynienia z dipolem czystym. Wówczas wartość  $F_{ii}$  powinna być dodatnia, jeśli  $\Phi(x)$  i  $\Phi(y)$  różnią się znakami. Błąd liczony jest jako odległość od określonego znaku a jedną z możliwości jest zastosowanie funkcji wypukłych i odcinkowo liniowych [1, 4, 5].

Przedstawiony powyżej sposób prezentacji składowych  $F_{ij}$  umożliwia jeszcze jedno podejście. Opierając się na całym zbiorze danych można założyć, że klasa  $C_i$  będzie pozytywna (negatywna) i obliczyć wówczas błędy w sposób opisany powyżej. Oszczędzi to podejmowania decyzji w przypadku pojedynczych dipoli, dla których oba obiekty należą do klasy  $C_i$ . Istnieje również możliwość obliczenia całkowitego błędu dla dipoli czystych z tej klasy dla obu orientacji oraz wybranie mniejszej wartości.

W przypadku gdy  $i \neq j$ , pożądane są różne znaki dla  $\Phi(x)$  i  $\Phi(y)$ . Jeśli jednak w połowie przypadków (dla połowy dipoli) gdzie  $x$  należy do  $C_i$  a  $y$  do  $C_j$ , wartość  $\Phi(x)$  jest dodatnia a  $\Phi(y)$  jest ujemna, a dla pozostałej połowy sytuacja jest odwrotna, świadczy to o istniejącym problemie z orientacją dla dipoli mieszanych. Można wówczas postąpić w następujący sposób. Na początku, wybiera się spośród  $\{i,j\}$  indeks klasy z największą wartością  $\Phi$  i określa się tę klasę jako pozytywną. Następnie dodaje się wyrazy określające wartości funkcji błędu oparte na funkcjach wypukłych i odcinkowo liniowych dla obiektów z klasy dodatniej, dla których wartości  $\Phi$  są zbyt małe oraz dla obiektów z ujemnej klasy, dla których wartości  $\Phi$  są zbyt duże. Można założyć pewną dodatnią wartość  $\varepsilon$  i rozpatrywać wartości mniejsze od niej jako zbyt małe w przypadku klasy dodatniej oraz negatywną wartość  $\varepsilon$  i brać pod uwagę wartości większe od niej jako zbyt duże w przypadku klasy ujemnej. Oczywiście, dla każdej pary  $(i,j)$  gdy  $i \neq j$  nie ma potrzeby rozpatrywania  $(j,i)$ , dlatego druga suma we wzorze (1) może być ograniczona do warunku  $j < i$ .

**Uwaga 1:** Ustalanie orientacji dodatniej/ujemnej jest wrażliwe na obserwacje odstające. W przypadku danych rzeczywistych zawierających losowe błędy, może istnieć potrzeba zastosowania definicji mniej czułej na tego typu obserwacje, na przykład opartej na środkach ciężkości lub wybieraniu na przykład największej (najmniejszej) wartości w każdej z klas. Można również ustalić orientacje w ten sposób by uzyskać jak najmniejszą wartość kar.

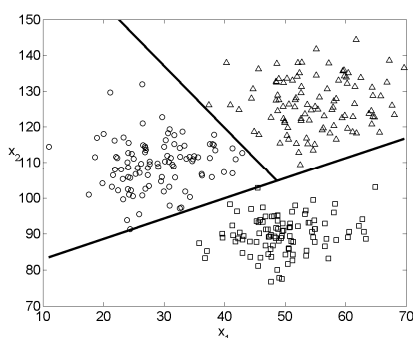
**Uwaga 2:** Podstawową wadą opisanej metody jest ewidentnie duża liczba wyrazów w czterokrotnej sumie (1). W przypadku danych przedstawionych na rysunku 3 mamy  $64 \cdot 63/2$  par mieszanych klas. Zakładając, że w każdej z klas jest 100 obiektów do sumy będzie dodanych 10000 wyrazów dla każdej pary indeksów. Stąd do oszacowania wartości funkcji celu w jednym przebiegu algorytmu potrzeba 200000 wyrazów. Z drugiej strony, podejście opisane w [4, 5] wymagało sumowania 6400 wyrazów z daną a priori wiedzą, która klasa ma być usytuowana po dodatniej stronie hiperpłaszczyzny.

**Uwaga 3:** W ujęciu klasycznym obliczanie sumy błędów jako sumy wypukłych i odcinkowo liniowych funkcji kar jest dogodne do minimalizacji. Prezentowane zadanie z natury nie jest wypukłe. By to potwierdzić wystarczy przesunąć dowolną prostą z rysunku 3. Wartość kryterium będzie zmniejszać się lub zwiększać więcej niż jednokrotnie.

### 4. Wyniki

#### Przykład 1

W celu zaprezentowania działania opisanej powyżej metody przedstawiony jest przykład pierwszy. Wygenerowany zbiór danych zawiera obiekty należące do trzech klas po 100 obiektów w każdej klasie.



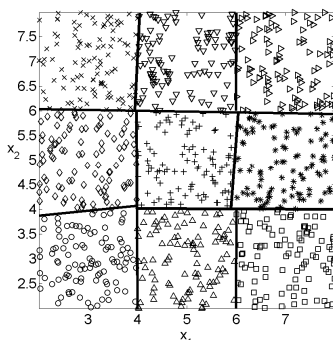
Rys. 4. Podział 3 klas  
Fig. 4. Three classes division

W pierwszym kroku oddzielona od pozostałych została klasa oznaczona symbolem kwadratu, natomiast w kolejnym kroku rozdzielone zostały pozostałe dwie klasy dając ostatecznie 100% jakość klasyfikacji.

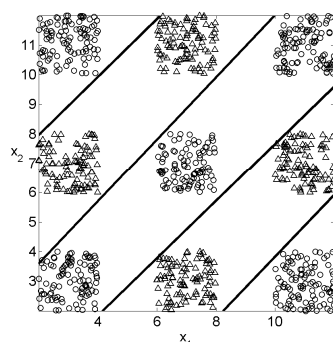
#### Przykład 2

Jako drugi prezentowany jest przykład z większą liczbą obserwacji niż w poprzednim przykładzie – zbiór zawiera 900 obserwacji. Dane ułożone zostały w postaci szachownicy zawierającej 9 klas, rys. 5a oraz dwie klasy, rys. 5b.

a)



b)



Rys. 5. Podział danych w postaci szachownicy, a) 9 klas; b) 2 klasy  
Fig. 5. "Chess board" data division, a) 9 classes; b) 2 classes

W przypadku 9 klas w pierwszym kroku zostały oddzielone 3 klasy od pozostałych 6. Następnie, rekurencyjnie były rozdzielane 3 klasy znajdujące się w górnej części szachownicy – odcinane były kolejno pojedyncze klasy. Sześć klas znajdujących się w dolnej części szachownicy rozdzielone zostały na 2 i 4 klasy. W kolejnych krokach dwie klasy zostały rozdzielone na osobne, natomiast grupa 4 klas była rekurencyjnie rozdzielona na grupy po 2 klasy, natomiast później na pojedyncze klasy.

W przypadku przedstawionym na rysunku 5b, na którym dane rozmieszczone są w postaci szachownicy i należą do dwóch klas, początkowy podział wydzielił dwie grupy skupisk – 6 skupisk po dodatniej stronie hiperpłaszczyzny oraz 3 skupiska po stronie ujemnej. Kolejne trzy podziały przedstawione na rysunku 5b doprowadziły do wydzielenia obszarów przynależności do pojedynczych klas ze 100% jakością klasyfikacji.

Obliczenia prowadzono z wykorzystaniem pakietu Matlab. Ulepszenie implementacji komputerowej przez zastosowanie siatek (meshgrid) dało dwudziestokrotne skrócenie czasu obliczeń w stosunku do pierwotnej wersji oprogramowania.

## 5. Wnioski

Znaną i obecnie często stosowaną praktyką jest szukanie parametrów hiperpłaszczyzny separującej dwie klasy poprzez minimalizację funkcjonalu błędu. Na funkcjonal ten składają się wyrazy opisujące niepoprawnie sklasyfikowane obiekty. Dostępne są efektywnie działające implementacje w postaci metody SVM [2, 6] czy funkcji kar typu CPL, które można stosować do dużych zbiorów danych [4, 5]. W podstawowym przypadku podejście oparte na dipolach nie jest niezbędne. Jednak gdy liczba klas obiektów wzrasta, kryterium oparte na dipolach może być zastosowane przede wszystkim do podziału ogromnych zbiorów danych na mniejsze podzbiory, w których przede wszystkim będzie mniejsza liczba klas. Rozważania zostały rozpoczęte od teoretycznego sformułowania kryterium opartego na dipolach prezentowanego w pracy [1]. Następnie zostały przeprowadzone testy numeryczne, a początkowe kryterium zostało zmodyfikowane w taki sposób, by znaleziona hiperpłaszczyzna separowała jak największą liczbę obiektów należących do różnych klas, nie rozdzielając obiektów z tej samej klasy. Do wyznaczania funkcjonalu nie jest stosowane zliczanie błędów klasyfikacji w postaci liczb całkowitych, ale odbywa się to poprzez ważenie – brane są pod uwagę odległości obiektów od hiperpłaszczyzny w przypadku, gdy usytuowane są one po jej nieprawidłowej stronie. Podejście to zostało przetestowane na wygenerowanych zbiorach danych a otrzymane wyniki skłoniły autorów do implementacji w postaci wywołań rekurencyjnych.

Znaczenie efektywnej i automatycznej klasyfikacji danych w wielu dziedzinach wykorzystujących systemy wspomagania decyzji, takich jak projektowanie, produkcja, kontrola jakości czy medycyna, diagnostyka, trudno jest przecenić. Prezentowany w pracy problem jest trudny do rozwiązania, gdyż z natury nie jest wypukły. Przedstawione rozwiązanie może być zastosowane w wielu dziedzinach, gdyż nie wymaga wkładu ani interwencji specjalisty. W szybki i łatwy sposób rozdziela problem wieloklasowy na sekwencję coraz mniejszych problemów, do uzyskania podstawowych problemów dwuklasowych, do których mogą być następnie zastosowane klasyczne metody klasyfikacji.

Praca finansowana w ramach pracy W/WI/1/2009.

## 6. Literatura

- [1] Bobrowski L.: Eksploracja danych oparta na wypukłych i odcinkowo liniowych funkcjach kryterialnych. Wyd. Pol. Białostockiej, Białystok, 2005.
- [2] Burges C.J.: A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.
- [3] Mitchell T. M.: *Machine learning*. McGraw-Hill Science, 1997.
- [4] Topczewska M.: *Efektywne metody klasyfikacji i wizualizacji danych wielowymiarowych*, rozprawa dokt., Pol. Białostocka, 2009.
- [5] Topczewska M., Frischmuth K.: Numerical aspects of weight calculation in classification methods. In *PTSK Conference*, Krynica Görska, Poland, Sept. 26-29, 2007.
- [6] Vapnik V. N.: *Statistical learning theory*. Wiley J., 1998.