

Dawid Rasała¹⁾, Witold Malina²⁾

UOGÓLNIONY LINIOWY KLASYFIKATOR FISHERA GENERALISED FISHER LINEAR CLASSIFIER

STRESZCZENIE W literaturze wielokrotnie omawiano klasyfikatory obrazów o rozkładach normalnych. Na ogół, kiedy dwie klasy są znacznie oddalone od siebie, to ich separację można przeprowadzić za pomocą jednej hiperpłaszczyzny. W artykule rozpatrywane są przypadki trudne, kiedy rozkłady znacznie nachodzą na siebie. Aby błąd klasyfikacji był wówczas mniejszy, do rozdzielania klas lepiej użyć dwóch niż jednej płaszczyzny. Na początku został opisany algorytm, który bada i wyznacza liczbę przecięć dwóch funkcji Gaussa jednej zmiennej dla różnych przypadków. Potem algorytm ten został włączony do algorytmu uczenia i klasyfikacji dla zadania dwuklasowego. Następnie został on uogólniony do zadań wieloklasowych. Przeprowadzone eksperymenty na płaszczyźnie dla zadań trudnych, gdy liczba klas $L = 2, 3, 4$ wykazały, że zaproponowany algorytm dawał lepsze wyniki niż algorytm klasyczny z jedną płaszczyzną rozdzielającą.

Słowa kluczowe:

liniowa analiza dyskryminacyjna, wektor dyskryminacyjny Fishera, obliczanie liczby progów, klasyfikator z jednym progiem, klasyfikator z dwoma progami, rozszerzone kryterium Fishera.

ABSTRACT Bayesian classifiers for normal distribution patterns have often been discussed in literature. In general, when two classes are considerably apart from each other, they can be separated with a single plane. In this paper we will examine some difficult cases, i.e. when their distributions significantly overlap. In such cases, to minimize the classification error, it is better to use two planes instead of one to separate the classes. At the beginning, the paper describes an algorithm used to investigate and determine the number of intersections of two Gaussian functions for different cases. Further in the article, this algorithm is included in the learning and classification algorithm for a two-class task. Then the algorithm is generalized for multi-class tasks. The experiments carried out on a plane for difficult tasks, when the number of classes $L = 2, 3, 4$, show that the proposed algorithm produces better results than the conventional algorithm with one separating plane.

Keywords:

linear discriminant analysis, Fisher discriminant vector, calculating the number of thresholds, single-threshold classifier, double-threshold classifier, extended Fisher criterion.

DOI: 10.5604/0860889X.1119446

¹⁾ Akademia Pomorska, 76-200 Słupsk, ul. Arciszewskiego 22b, e-mail: dawid@rasala.pl

²⁾ Społeczna Akademia Nauk, 76-200 Słupsk, ul. Szczecińska 60, e-mail: j.malina6@upcpoczta.pl

WSTĘP

Studia liniowych klasyfikatorów są bardzo ważnym zagadnieniem w statystycznych metodach rozpoznawania obrazów [1, 10, 11]. W wielu zastosowaniach preferuje się liniowe klasyfikatory z uwagi na ich prostotę i szybkość działania. Jednym z podstawowych problemów w statystycznym rozpoznawaniu obrazów jest zbudowanie klasyfikatora liniowego dla dwóch klas obrazów c_1 i c_2 o rozkładach normalnych. Wiele takich problemów można rozwiązać, konstruując na podstawie kryterium Fishera równanie wektora własnego. Następnie rozwiązanie tego równania daje wektor własny \mathbf{d} taki, że rzutowane na niego rozkłady klas będą maksymalnie odseparowane [1, 2, 9, 10]. Analizując rozkłady, można wyznaczyć płaszczyznę rozdzielającą dwie klasy z minimalnym średnim błędem. Opisane podejście jest satysfakcjonujące, jeżeli klasy są dobrze odseparowane od siebie.

W pracy tej pójdziemy dalej i rozpatrzemy przypadek trudny, tzn. taki, że rozkłady klas mocno nachodzą na siebie (rys. 1a). W takim przypadku po operacji rzutowania klas na wektor dyskryminacyjny \mathbf{d} widzimy (rys. 1b), że aby otrzymać mniejszy błąd klasyfikacji, należy użyć dwóch progów (w przestrzeni źródłowej te progi reprezentują dwie równoległe płaszczyzny decyzyjne). W praktyce wyznaczenie dwóch progów nie zawsze jest proste i potrzebne.

W artykule został opracowany **algorytm 1.**, który wyznacza liczbę progów dla dwóch klas obrazów po operacji

INTRODUCTION

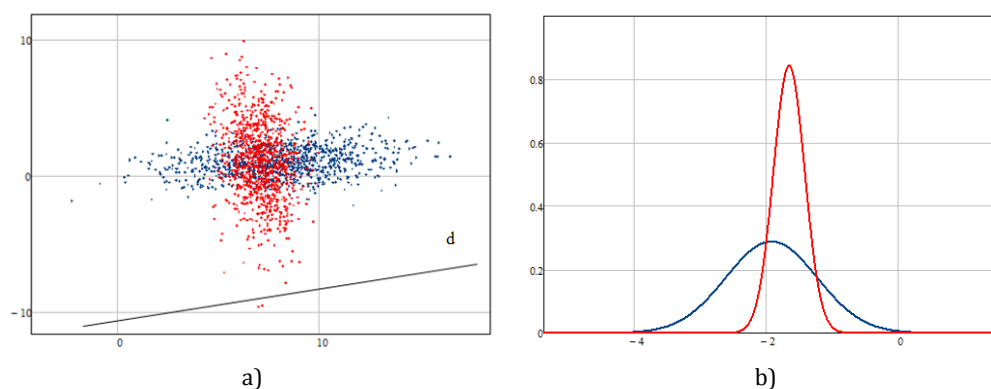
Studies on linear classifiers are a very important issue in statistical methods for pattern recognition [1, 10, 11]. In many applications linear classifiers are preferred for their simplicity and fast action. One of the main problems in statistics-based pattern recognition is building a linear classifier for classes of patterns c_1 and c_2 having normal distribution. Many such problems can be solved by constructing an eigenvector equation based on the Fisher criterion. Then the solution to this equation produces such own vector \mathbf{d} that the class distributions mapped on it will be maximally separated [1, 2, 9, 10]. Through analyzing distributions a plane separating two classes with a minimum mean error can be determined. The described approach is satisfactory if classes are separated well from each other.

In this work we will move further and consider a difficult case, i.e. one where class distributions overlap to a large extent (fig. 1a). In such a case, after the class mapping operation on discriminant vector \mathbf{d} , we can see that to obtain a smaller classification error it is necessary to use two thresholds (in the source space these thresholds represent two parallel decision planes). In practice, to determine two planes is not always simple or needed.

In this article **algorithm 1** was developed. It determines the number of thresholds for two classes of patterns after a mapping operation on vector \mathbf{d} . Then using **algorithm 1**, a two-class

rzutowania ich na wektor \mathbf{d} . Następnie został opracowany algorytm dwuklasowy uczenia i klasyfikacji z zastosowaniem **algorytmu 1**. Na podstawie zaproponowanego algorytmu dwuklasowego opracowano algorytm wieloklasowy uczenia i klasyfikacji oraz jego uproszczoną wersję. Przeprowadzone eksperymenty w przestrzeni 2D pokazały, że dla przypadków trudnych zaproponowany klasyfikator dawał mniejszy błąd niż klasyfikator z jednym progiem.

algorithm of learning and classification was developed. Based on the proposed two-class algorithm a multiclass algorithm of learning and classification and its simplified version were developed. Experiments performed in 2D space showed that for difficult cases the proposed classifier produced a smaller error than a single-threshold classifier.



Rys. 1. Klasy, których rozkłady przecinają się: a) rozkłady obserwacji w przestrzeni; b) rozkłady po operacji rzutowania na wektor dyskryminacyjny \mathbf{d}

Fig. 1. Classes, whose distributions intersect: a) observation distributions in space; b) distributions after a mapping operation on discriminant vector \mathbf{d}

ANALIZA PRZECIĘĆ DWÓCH FUNKCJI GAUSSA

Szukanie progów

Rozważmy przypadek jednowymiarowy, w którym znalezienie punktów przecięć dwóch funkcji Gaussa jest równoznaczne ze znalezieniem progów dla klasyfikatora. Aby znaleźć punkty przecięć, należy przyrównać do siebie obie funkcje gęstości, a następnie rozwiązać otrzymane równanie:

ANALYSIS OF INTERSECTIONS OF TWO GAUSSIAN FUNCTIONS

Looking for thresholds

Let us consider a one-dimensional case in which finding intersection points for two Gaussian functions is tantamount to finding thresholds for a classifier. In order to find the intersection points, both density functions have to be compared to each other and then the equation obtained must be solved:

$$P_1 \frac{1}{2\pi\sigma_1} e^{-\frac{x-m_1}{2\sigma_1^2}} = P_2 \frac{1}{2\pi\sigma_2} e^{-\frac{x-m_2}{2\sigma_2^2}}, \quad (1)$$

gdzie:

where:

- P_1, P_2 — prawdopodobieństwa *a priori* klas c_1, c_2 /
probabilities *a priori* of classes c_1, c_2 ;
 m_1, m_2 — średnie klas / mean magnitudes of classes;
 σ_1^2, σ_2^2 — wariancje obrazów w klasach / variances of patterns in classes.

Po zlogarytmowaniu (1), a następnie uporządkowaniu jego elementów, otrzymujemy:

After logarithmizing (1), and ordering its elements, we obtain:

$$\sigma_2^2 - \sigma_1^2 x^2 + \sigma_1^2 m_2 - \sigma_2^2 m_1 x + \sigma_2^2 m_1^2 - \sigma_1^2 m_2^2 - 2\sigma_1^2 \sigma_2^2 \ln \frac{P_1 \sigma_2}{P_2 \sigma_1} = 0. \quad (2)$$

W zależności od parametrów równanie kwadratowe (2) może mieć różne rozwiązania.

Depending on parameters of square equation (2) can have different solutions.

1. Jeżeli $m_1 = m_2$ oraz $\sigma_1 = \sigma_2 = \sigma$, to na podstawie (2) otrzymujemy:

1. If $m_1 = m_2$ oraz $\sigma_1 = \sigma_2 = \sigma$, then on the basis of (2) we obtain:

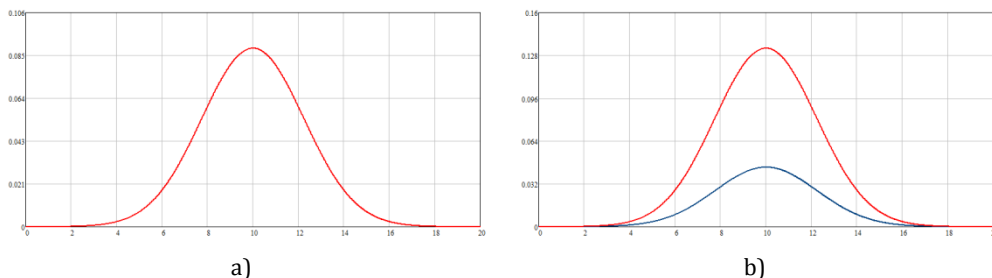
$$2\sigma^4 \ln \frac{P_1}{P_2} = 0. \quad (3)$$

Rozważając równanie (3), widać, że gdy:

Considering equation (3), it follows that when:

- a) $P_1 = P_2$, to równanie jest spełnione. Oznacza to, że wykresy funkcji Gaussa dla powyższych parametrów pokrywają się i występuje taka sytuacja, jaką przedstawiono na rysunku 2a.
b) $P_1 \neq P_2$, to równanie jest sprzeczne. Przypadek ten oznacza, że punktów przecięć nie ma, ponieważ jedna funkcja zawsze przyjmuje mniejsze wartości od drugiej (rys. 2b).

- a) $P_1 = P_2$, then the equation is fulfilled. This means that the diagrams of the Gaussian function for the above parameters overlap and there occurs a situation as presented in figure 2a.
b) $P_1 \neq P_2$, then the equation is contradictory. This case means that there are no points of intersection because one function always takes smaller values than the other (fig. 2b).



Rys. 2. Relacje między rozkładami, gdy: a) $P_1 = P_2$ (rozkłady się pokrywają), b) $P_1 \neq P_2$
 Fig. 2. Relations between distributions, when a) $P_1 = P_2$ (distributions overlap), b) $P_1 \neq P_2$

2. Jeżeli średnie są różne $m_1 \neq m_2$, a wariancje rozkładów identyczne $\sigma_1 = \sigma_2 = \sigma$, to równanie (2) sprowadza się do równania liniowego. Wtedy równanie (2) ma jedno rozwiązanie, czyli występuje jeden próg, który można obliczyć według wzoru:

$$\alpha = \frac{m_1 + m_2}{2} + \frac{\sigma}{m_2 - m_1} \ln \frac{P_1}{P_2}. \quad (4)$$

W szczególności gdy $P_1 = P_2$, to

$$\alpha = \frac{m_1 + m_2}{2}. \quad (5)$$

3. W ogólnym przypadku, jeżeli $m_1 \neq m_2$ i $\sigma_1 \neq \sigma_2$, to równanie (2) ma dwa rozwiązania:

$$\alpha_{1,2} = \frac{\sigma_2^2 m_1 - \sigma_1^2 m_2 \pm \sigma_1 \sigma_2 \sqrt{(m_2 - m_1)^2 + 2(\sigma_2^2 - \sigma_1^2) \ln \frac{P_1 \sigma_2}{P_2 \sigma_1}}}{\sigma_2^2 - \sigma_1^2}. \quad (6)$$

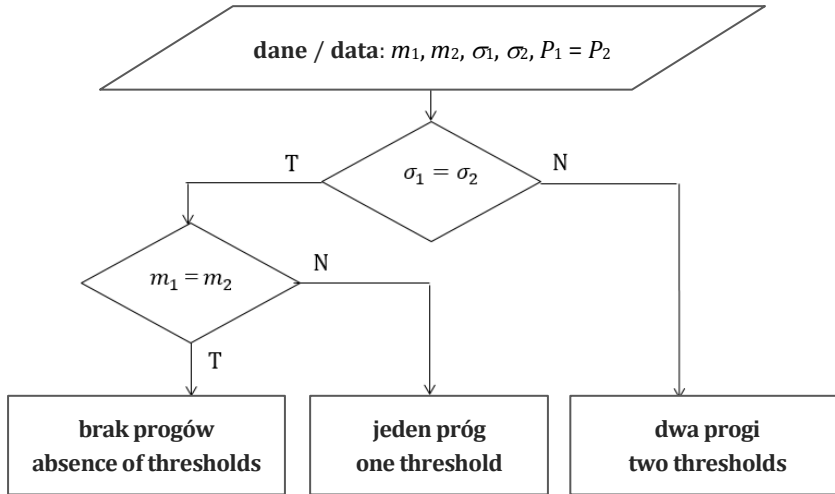
Zgodnie z przedstawionymi rozważaniami teoretyczny algorytm wyznaczania liczby pierwiastków równania pokazano na rysunku 3., a odpowiednie wartości progów można liczyć na podstawie wzorów (4) i (6).

2. If mean magnitudes are different $m_1 \neq m_2$, and distribution variances are identical $\sigma_1 = \sigma_2 = \sigma$, then equation (2) is brought to a linear equation. Then equation (2) has only one solution, which means there occurs one threshold, which can be calculated following the formula:

Especially, when $P_1 = P_2$, then

3. In a general case, if $m_1 \neq m_2$ and $\sigma_1 \neq \sigma_2$, then equation (2) has two solutions:

Following the presented considerations a theoretical algorithm for calculating the number of roots is shown in figure 3, and adequate threshold values can be calculated using the formulas (4) and (6).



Rys. 3. Algorytm wyznaczania liczby pierwiastków równania (1)

Fig. 3. Algorithm for determining the number of roots of the equation (1)

Dyskusja rozwiązań

Przypadek pierwszy, gdy nie ma rozwiązania, pominiemy jako zbyt szczególny i mało interesujący. Przypadek drugi, gdy klasy rozdzielane są jednym progiem, jest powszechnie stosowany, był rozważany wielokrotnie i jest dobrze opracowany. Przyjrzyjmy się przypadkowi trzeciemu, gdy występują dwa pierwiastki równania. Sytuacja taka jest mało dyskutowana w literaturze. Zdaniem autorów ma ona duże znaczenie praktyczne, ponieważ często nie wiemy, czy rozkłady są dobrze czy słabo separowalne, a więc nie wiadomo, czy stosować jeden czy dwa progi. Nasza propozycja taką sytuację uwzględnia i pozwala w sposób ogólny sformułować algorytm uczenia i klasyfikacji.

W ogólnym przypadku, gdy $\sigma_1 \neq \sigma_2$, to równanie (2) ma zawsze dwa rozwiązania, czyli zawsze mogą być obliczane dwa pierwiastki. W zależności od relacji

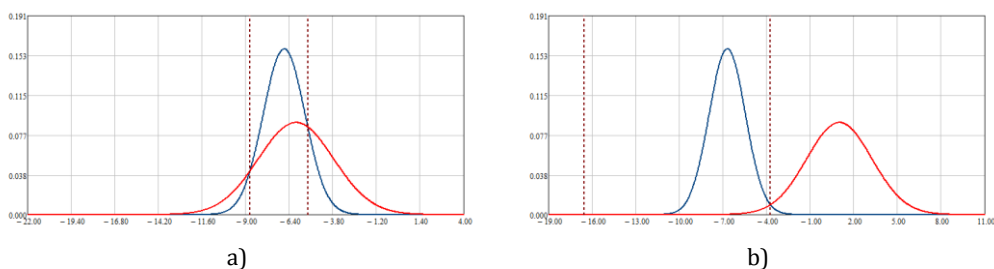
Discussion relating to solutions

We will disregard the first case when there is no solution as too detailed, and not very interesting. The second case, when the classes are separated with one threshold is commonly used. It has been considered many times and is well elaborated. Let us take a look at the third case, when the two equation roots occur. This situation has been rarely discussed in literature. In the authors' view it has large practical significance as we often do not know if distributions are easy or poorly separable, therefore it is not known if one or two thresholds should be used. Our proposal takes such a situation into account and makes it possible to formulate, in a general way, an algorithm of learning and classification.

In a general case, when $\sigma_1 \neq \sigma_2$, then equation (2) has always two solutions, which means two roots can always be calculated. Depending on the relationship

między rozkładami (ich parametrami) można ocenić liczbę praktycznie, tzn. rzeczywiście potrzebnych progów. Na przykład na rysunku 4a widać, że funkcje przecinają się w dwóch miejscach (są dwa progi), natomiast dla przypadku pokazanym na rysunku 4b widać, że wystarczy obliczyć jeden próg (obliczanie drugiego jest zbędne). Mając ten fakt na uwadze, postaramy się sformułować algorytm znajdowania zredukowanej liczby progów.

between distributions (their parameters), we can assess the number of thresholds that is actually needed. For example, in figure 4a it can be seen that the functions intersect at two points (there are two thresholds), whereas it can be seen that for the case shown in figure 4b it is sufficient to calculate one threshold (calculating the other one is not needed). Bearing this fact in mind we will try to formulate an algorithm for finding a reduced number of thresholds.



Rys. 4. Wpływ położenia rozkładów klas na liczbę przecięć: a) $m_1 \approx m_2$; b) $m_1 \neq m_2$

Fig. 4. Influence of class distributions upon the number of intersections:

a) $m_1 \approx m_2$, b) $m_1 \neq m_2$

Algorytm znajdowania zredukowanej liczby progów

Zaprezentowany niżej algorytm ma na celu zbadanie i wyznaczenie liczby progów dla dwóch rozkładów normalnych jednowymiarowych na podstawie ich parametrów (bez konieczności obliczania punktów przecięć krzywych). Dla uproszczenia rozważań zakładamy, że prawdopodobieństwa *a priori* klas są równe ($P_1 = P_2$); przypadek, gdy prawdopodobieństwa te nie są równe, pominiemy.

Algorytm 1. Znajdowanie liczby progów, gdy $P_1 = P_2$.

The algorithm for finding a reduced number of thresholds

The purpose of the algorithm presented below is to examine and determine the number of thresholds for two normal one-dimensional distributions using their parameters (without the necessity to calculate the intersection points of curves). In order to simplify the considerations we assume that the probabilities *a priori* of classes are equal ($P_1 = P_2$); we will disregard the case, when probabilities are not equal.

Algorithm 1. Finding the number of thresholds, when $P_1 = P_2$.

Dane:

m_1, m_2 — wartości średnie klas / mean magnitudes of classes;
 σ_1^2, σ_2^2 — wariancje obrazów w klasach / pattern variances in classes.

Data:

Założenia:

$P_1 = P_2$ — prawdopodobieństwa *a priori* klas są równe /
 probabilities *a priori* of classes are equal;
 $\sigma_2 > \sigma_1$ — wariancja klasy c_2 jest większa od wariancji klasy c_1 /
 variance of class c_2 is larger than that of class c_1 .

Assumptions:

Wprowadzone parametry:

1. Współczynnik zgodności wariancji rozkładów

Introduced parameters:

1. Coefficient of variances agreed in distributions

$$\lambda = \frac{\sigma_1}{\sigma_2}, \tag{7}$$

gdzie:

where:

$$0 < \lambda \leq 1.$$

Jeżeli $\lambda = 1$, to oznacza, że rozkłady pokrywają się.

If $\lambda = 1$, then it means the distributions overlap.

Dalej będziemy przyjmować, że rozkłady pokrywają się, jeżeli $\lambda \geq \lambda_0$, (np. $\lambda_0 = 0,9$), gdzie parametr $\lambda_0 \leq 1$.

Further, we will assume that the distributions overlap, if $\lambda \geq \lambda_0$, (e.g. $\lambda_0 = 0.9$), where parameter $\lambda_0 \leq 1$.

2. Współczynniki oceny rozsunięcia rozkładów

2. Assessment coefficients of spacing distributions

$$\varepsilon_1 = 0.1\sigma_1 = 0.1\lambda\sigma_2; \tag{8}$$

$$\varepsilon_2 = 3\sigma_2 - \sigma_1 = 3\sigma_2(1 - \lambda). \tag{9}$$

Współczynnik ε_1 służy do oceny, czy rozkłady znajdują się na tyle blisko

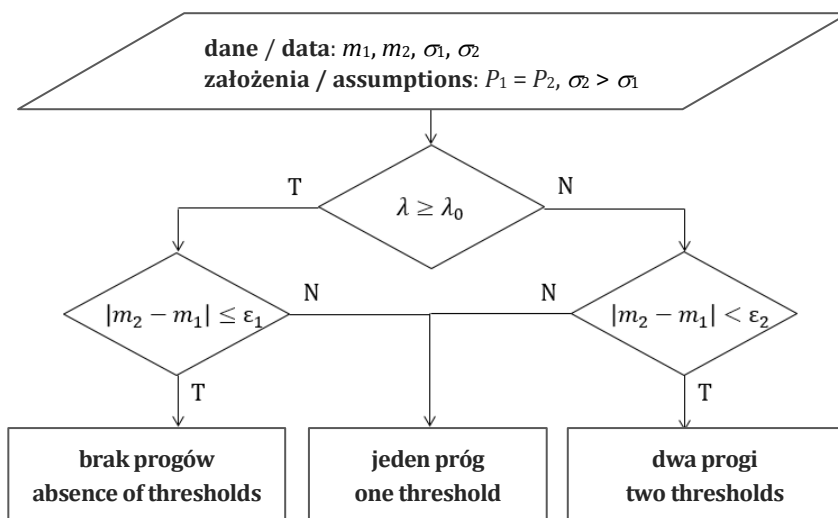
Coefficient ε_1 is used to estimate if the distributions are in such a proximity

siebie, iż można uznać, że pokrywają się (nie przecinają się) i wtedy nie ma punktu przecięcia lub gdy można wyznaczyć tylko jeden próg. Natomiast współczynnik ε_2 służy do badania, czy dla danej pary rozkładów będziemy liczyć jeden czy dwa progi.

Zakładamy, że parametry rozkładów $m_1, m_2, \sigma_1, \sigma_2$ są znane, ponieważ na podstawie zbioru uczącego możemy je estymować. Zaproponowany algorytm wyznaczania liczby przecięć (progów), który uwzględni wprowadzone współczynniki, jest przedstawiony na rysunku 5.

that it can be assumed that they overlap or if only one threshold can be determined. Coefficient ε_2 is used to find out if, for a given pair of distributions we will calculate one, or two thresholds.

We assume that distribution parameters $m_1, m_2, \sigma_1, \sigma_2$ are known, as we can estimate them using the learning set. The proposed algorithm for calculating the number of intersections (thresholds), which takes into account the introduced factors, is presented in figure 5.



Rys. 5. Algorytm znajdowania liczby progów, gdy $P_1 = P_2$

Fig. 5. The algorithm for determining the number of thresholds when $P_1 = P_2$

Uwagi

1. Decyzję o **braku progów** podejmujemy wtedy, gdy rozkłady mają zbliżone wariancje i pokrywają się (lub prawie pokrywają), tzn. gdy spełnione są warunki:

Remarks

1. We make a decision referring to the **absence of thresholds** when the distributions have similar variances and overlap (or almost overlap), i.e. when the following conditions are fulfilled:

$$\lambda \geq \lambda_0 \text{ oraz /and } m_2 - m_1 \leq \varepsilon_1. \quad (10)$$

2. **Jeden próg** wyznaczamy wtedy, gdy:

- a) rozkłady mają różne wariancje i są rozsunięte więcej niż o wartość ε_2 , czyli gdy:

$$\lambda < \lambda_0 \text{ oraz / and } m_2 - m_1 \geq \varepsilon_2; \quad (11)$$

- b) rozkłady mają zbliżone wariancje i są rozsunięte o więcej niż o wartość ε_1 , czyli gdy:

$$\lambda \geq \lambda_0 \text{ oraz / and } m_2 - m_1 > \varepsilon_1. \quad (12)$$

3. Decyzję o wyznaczeniu **dwóch progów** algorytm podejmuje w przypadku, gdy rozkłady mają różne wariancje oraz są rozsunięte o nie więcej niż zadaną wartość ε_2 , czyli gdy:

$$\lambda < \lambda_0 \text{ oraz / and } m_2 - m_1 < \varepsilon_2, \quad (13)$$

gdzie parametry λ , ε_1 , ε_2 zostały zdefiniowane wcześniej.

ALGORYTM UCZENIA I KLASYFIKACJI

Uczenie i klasyfikacja, gdy $L = 2$

Zakładamy, że mamy dwie klasy c_1 i c_2 , których obrazy mają rozkłady normalne w przestrzeni N -wymiarowej. Parametry rozkładów są następujące:

- μ_1, μ_2 — wektory średnie;
 Σ_1, Σ_2 — macierze kowariancji;
 P_1, P_2 — prawdopodobieństwa *a priori*.

Korzystając z **algorytmu 1.** wyznaczania liczby progów, proponujemy następujący algorytm uczenia i klasyfikacji klasyfikatora dwuklasowego.

2. We calculate **one threshold** when:

- a) the distributions have different variances and are extended by more than the value of ε_2 , i.e. when:

$$\lambda < \lambda_0 \text{ oraz / and } m_2 - m_1 \geq \varepsilon_2; \quad (11)$$

- b) the distributions have close variances and are extended by more than the value of ε_1 , i.e. :

$$\lambda \geq \lambda_0 \text{ oraz / and } m_2 - m_1 > \varepsilon_1. \quad (12)$$

3. The algorithm makes a decision to calculate **two thresholds** in a case, when the distributions have different variances and are extended by not more than a predetermined value ε_2 :

$$\lambda < \lambda_0 \text{ oraz / and } m_2 - m_1 < \varepsilon_2, \quad (13)$$

where parameters λ , ε_1 , ε_2 were defined earlier.

THE ALGORITHM OF LEARNING AND CLASSIFICATION

Learning and classification, when $L = 2$

We assume that we have two classes c_1 and c_2 , whose patterns have normal distributions in an N -dimensional space. Distribution parameters are as follows:

- μ_1, μ_2 — mean vectors;
 Σ_1, Σ_2 — covariance matrixes;
 P_1, P_2 — probabilities *a priori*.

Using **algorithm 1** to calculate the number of thresholds, we will propose the following two-class classifier learning and classification algorithm.

Algorytm 2. Uczenie i klasyfikacja, gdy $L = 2$.

Należy:

1. Korzystając z kryterium Fishera, znaleźć wektor dyskryminacyjny

Algorithm 2. Learning and classification, when $L = 2$.

It is necessary:

1. Using the Fisher criterion, find a discriminant vector

$$\mathbf{d} = A^{-1}\Delta, \quad (14)$$

gdzie:

where:

$$\Delta = \mu_1 - \mu_2;$$

$$A = P_1\Sigma_1 + P_2\Sigma_2.$$

2. Zrzutować obserwacje klasy c_1 i c_2 na wektor \mathbf{d} .

2. Map observations of classes c_1 and c_2 on vector \mathbf{d} .

3. Na podstawie **algorytmu 1.** wyznaczyć liczbę progów n_t ($n_t = 0, 1, 2$) i obliczyć ich wartości.

3. Using **algorithm 1**, determine the number of thresholds n_t ($n_t = 0, 1, 2$) and calculate their values.

4. W zależności od liczby progów n_t wybrać sposób konstrukcji funkcji klasyfikujących:

4. Depending on the number thresholds n_t , choose the way of constructing the classification functions:

- a) Jeżeli liczba progów $n_t = 0$, tzn. rozkłady klas są identyczne i pokrywają się, wtedy niech funkcje klasyfikujące g_1 i g_2 mają postać następującą:

- a) If the number of thresholds $n_t = 0$, i.e. class distributions are identical and overlap, then let the classification functions g_1 and g_2 have the following form:

$$g_1(\mathbf{y}) = 1; \quad (15)$$

$$g_2(\mathbf{y}) = 0, \quad (16)$$

gdzie \mathbf{y} oznacza obraz.

where \mathbf{y} stands for a pattern.

- b) Jeżeli liczba progów $n_t = 1$, to przyjmujemy, że funkcje klasyfikujące są następujące:

- b) If the number of thresholds $n_t = 1$, then we assume that the classification functions are as follows:

$$g_1(\mathbf{y}) = \begin{cases} 1, & \text{gdy / when } d^T \mathbf{y} \leq \alpha, \\ 0, & \text{gdy / when } d^T \mathbf{y} > \alpha; \end{cases} \quad (17)$$

$$g_2(y) = \begin{cases} 0, & \text{gd}y / \text{when } d^T y \leq \alpha \\ 1, & \text{gd}y / \text{when } d^T y > \alpha' \end{cases} \quad (18)$$

gdzie próg α obliczamy według wzoru (4) lub (5).

where we calculate threshold α following the formula (4) or (5).

c) Jeżeli liczba progów $n_t = 2$, wtedy niech funkcje klasyfikujące są następujące:

c) If the number of thresholds $n_t = 2$, then may the classification functions be as follows:

$$g_1(y) = \begin{cases} 1, & \text{gd}y / \text{when } \alpha_1 \leq d^T y \leq \alpha_2 \\ 0, & \text{gd}y / \text{when } d^T y < \alpha_1 \vee d^T y > \alpha_2 \end{cases}; \quad (19)$$

$$g_2(y) = \begin{cases} 0, & \text{gd}y / \text{when } \alpha_1 \leq d^T y \leq \alpha_2 \\ 1, & \text{gd}y / \text{when } d^T y < \alpha_1 \vee d^T y > \alpha_2 \end{cases}, \quad (20)$$

gdzie progi dane są wzorem (6).

where the thresholds are expressed using formula (6).

5. Dokonać klasyfikacji obrazów.

5. To classify patterns.

Dla klasyfikatora dwuklasowego możemy dwie funkcje klasyfikujące g_1 i g_2 zastąpić jedną:

In the case of a two-class classifier we can substitute two classification functions g_1 and g_2 for one:

$$g_{12}(y) = g_1(y) - g_2(y) = \begin{cases} +1, & \text{gd}y / \text{when } g_1(y) > g_2(y) \\ -1, & \text{gd}y / \text{when } g_1(y) < g_2(y) \end{cases}. \quad (21)$$

Odpowiednia reguła decyzyjna klasyfikatora jest następująca:

The adequate decision rule in the classifier is as follows:

$$c^*(y) = \begin{cases} c_1, & \text{gd}y / \text{when } g_{12}(y) > 0 \\ c_2, & \text{gd}y / \text{when } g_{12}(y) < 0 \end{cases}. \quad (22)$$

W zależności od liczby progów błąd klasyfikacji dla poszczególnych przypadków możemy opisać wzorami:

Depending on the number of thresholds the classification error for each case can be described with the formulas:

a) Jeżeli $n_t = 0$ (rys. 2a), to:

a) If $n_t = 0$ (fig. 2a), then:

$$P_e = 0.5. \quad (23)$$

b) Jeżeli $n_t = 1$, to:

b) If $n_t = 1$, then:

$$P_e = \int_{-\infty}^{\alpha} P_1 p(z|c_1) dz + \int_{\alpha}^{\infty} P_2 p(z|c_2) dz. \quad (24)$$

c) Jeżeli $n_t = 2$ (rys. 3a), to:

c) If $n_t = 2$ (fig. 3a), then:

$$P_e = \int_{-\infty}^{\alpha_1} P_1 p(z|c_1) dz + \int_{\alpha_1}^{\alpha_2} P_2 p(z|c_2) dz + \int_{\alpha_2}^{\infty} P_1 p(z|c_1) dz. \quad (25)$$

Uczenie i klasyfikacja, gdy $L > 2$

Learning and classification, when $L > 2$

Opisana metoda uczenia dla $L = 2$ klas zostanie wykorzystana do uczenia klasyfikatora wieloklasowego. Uczenie takiego klasyfikatora (**algorytm 2**, [11]) będzie polegało na wyznaczaniu progów i współczynników funkcji klasyfikujących odpowiednio dla poszczególnych par klas $c_i, c_j, i, j = 1, 2, \dots, L, i \neq j$, dalej funkcję klasyfikującą dla pary c_i, c_j będziemy oznaczać $g_{ij}(y)$, przy czym $g_{ij} = -g_{ji}$. Po wyznaczeniu wszystkich funkcji klasyfikujących $g_{ij}(y)$ reguła decyzyjna klasyfikatora będzie następująca:

The described learning method for $L = 2$ classes will be used in a multi-class classifier learning. Learning in the case of such a classifier (**algorithm 2**, [11]) involves determining thresholds and coefficients of classification functions respectively for the particular pairs of classes $c_i, c_j, i, j = 1, 2, \dots, L, i \neq j$, further we will mark the classification function for the pair c_i, c_j using $g_{ij}(y)$, while $g_{ij} = -g_{ji}$. After determining all classification functions $g_{ij}(y)$ the decision rule of the classifier will be as follows:

$$c^*(y) = c_i, \text{ gdy / when } g_{ij}(y) > 0 \text{ dla / for } j = 1, 2, \dots, L; j \neq i. \quad (26)$$

Przeprowadzenie klasyfikacji według reguły (26) sprowadza się do badania zespołów funkcji:

Making the classification in accordance with rule (26) is brought to studying sets of functions:

$$\begin{aligned} \text{zespół 1 / set 1: } & g_{12}, g_{13}, \dots, g_{1L}; \\ \text{zespół 2 / set 2: } & g_{21}, g_{23}, \dots, g_{2L}; \\ & \dots \\ \text{zespół } L \text{ / set } L: & g_{L1}, g_{L2}, \dots, g_{L(L-1)}. \end{aligned} \quad (27)$$

Numer zespołu i , dla którego wszystkie wartości funkcji g_{ij} ($j = 1, 2, \dots, L$ oraz $i \neq j$) będą dodatnie, oznacza klasę c_i , do której zostanie zaliczony nieznan obraz y . Jeżeli nie istnieje taki zespół, to obiekt nie będzie sklasyfikowany, co

The cardinal number in set i , for which all the values of function g_{ij} ($j = 1, 2, \dots, L$ and $i \neq j$) are positive, indicates class c_i , in which an unknown pattern y will be included. If such a set does not exist, then the object will not be classified,

oznacza, że klasyfikator podejmie decyzję neutralną, tzn. w przestrzeni obrazów będą występowały obszary nieprzydzielone do żadnej klasy.

Jeżeli ustalimy kolejność badania zespołów i występujących w nich funkcji, to złożoność opisanego algorytmu klasyfikacji można uprościć. Po uproszczeniu algorytm (27) przyjmie następującą postać (**algorytm 3.**):

$$\begin{aligned}
 \text{zespół 1 / set 1: } & g_{12}, g_{13}, g_{14}, \dots, g_{1L}; \\
 \text{zespół 2 / set 2: } & g_{23}, g_{24}, \dots, g_{2L}; \\
 \text{zespół 3 / set 3: } & g_{34}, \dots, g_{3L}; \\
 & \dots \\
 \text{zespół } L-1 \text{ / set } L-1: & g_{(L-1)L}.
 \end{aligned} \tag{28}$$

W algorytmie (28) mamy mniej o jeden zespół funkcji oraz mniej liczne zespoły funkcji. Tak jak wcześniej, w procesie klasyfikacji przeglądanie zespołów rozpoczynamy od zespołu 1. Jeżeli w zespole tym wszystkie funkcje będą dodatnie, to znaczy, że obraz y zostanie przydzielony do c_1 . Jeżeli warunek ten nie zostanie spełniony, to wtedy badamy zespół 2, ale teraz funkcję g_{12} możemy pominąć, ponieważ wiemy, że obraz nie należy do klasy c_1 . Postępując analogicznie, dla zespołu badamy funkcje g_{ij} tylko dla $j = i + 1, \dots, L$. Natomiast ostatni zespół $L-1$ ma jedną funkcję g_{L-1L} . Jeżeli funkcja g_{L-1L} będzie dodatnia, to obraz zaliczymy do klasy c_{L-1} , jeżeli będzie ujemna, to do klasy c_L . Warto zwrócić uwagę, że ten sposób klasyfikacji obrazów nie ma decyzji neutralnych, ale błąd klasyfikacji może zależeć od kolejności badanych par klas.

which means that the classifier will make a natural decision, i.e. in the pattern space there will be areas not assigned to any class.

If we establish the order in which to study the sets and functions occurring in them, then the complexity of the described classification algorithm can be simplified. After simplification, algorithm (27) will take the following form (**algorithm 3**):

In algorithm (28) we have one set of functions less and quantitatively smaller sets of functions. As earlier, we start reviewing sets in the process of classification with set 1. If in the set all functions are positive, it means that pattern y will be assigned to c_1 . If this condition is not fulfilled, then we study set 2, but this time we can disregard function g_{12} , as we know that the pattern does not belong to class c_1 . Analogically, as regards the set we study function g_{ij} only for $j = i + 1, \dots, L$. The last set $L-1$ has one function g_{L-1L} . If function g_{L-1L} is positive, we will assign the pattern to class c_{L-1} , if it is negative, to class c_L . It is worth mentioning that this way of classifying patterns does not have neutral decisions, but a classification error may depend on the order in which pairs of classes are studied.

EKSPERYMENTY

Badanie opisanych algorytmów uczenia i klasyfikacji (27, 28) oraz zastosowanie ich do zadań wieloklasowych przeprowadzono w przestrzeni dwuwymiarowej dla liczby klas $L = 2, 3, 4$. Zbiór treningowy każdej klasy składał się z 512 obrazów wygenerowanych za pomocą generatora liczb losowych o rozkładzie normalnym. Podczas naszych eksperymentów szczególną uwagę zwróciliśmy na przypadki, kiedy rozkłady obrazów zachodziły na siebie i były trudne do rozdzielenia. W eksperymentach założono, że $\lambda_0 = 0.9$, tzn. że akceptowana tolerancja różnicy wariancji dla rozkładów nie przekracza 10%. Odpowiednie współczynniki oceny rozsunięć rozkładów ε_1 i ε_2 liczymy zgodnie ze wzorami (8), (9).

Badanie algorytmu uczenia i jakości klasyfikacji przeprowadzono dla dwóch przypadków: dla jednego progu i dla dwóch progów, przy czym w drugim przypadku dla algorytmu (27) i uproszczonego (28). Otrzymane obszary decyzyjne zostały na rysunkach pokolorowane zgodnie z kolorami klas w przestrzeni źródłowej. Jeżeli wystąpiły obszary z decyzją neutralną, to otrzymały one kolor biały (bez koloru). Jakość klasyfikacji poszczególnych algorytmów oceniano za pomocą estymatora średniego prawdopodobieństwa błędu na zbiorze treningowym.

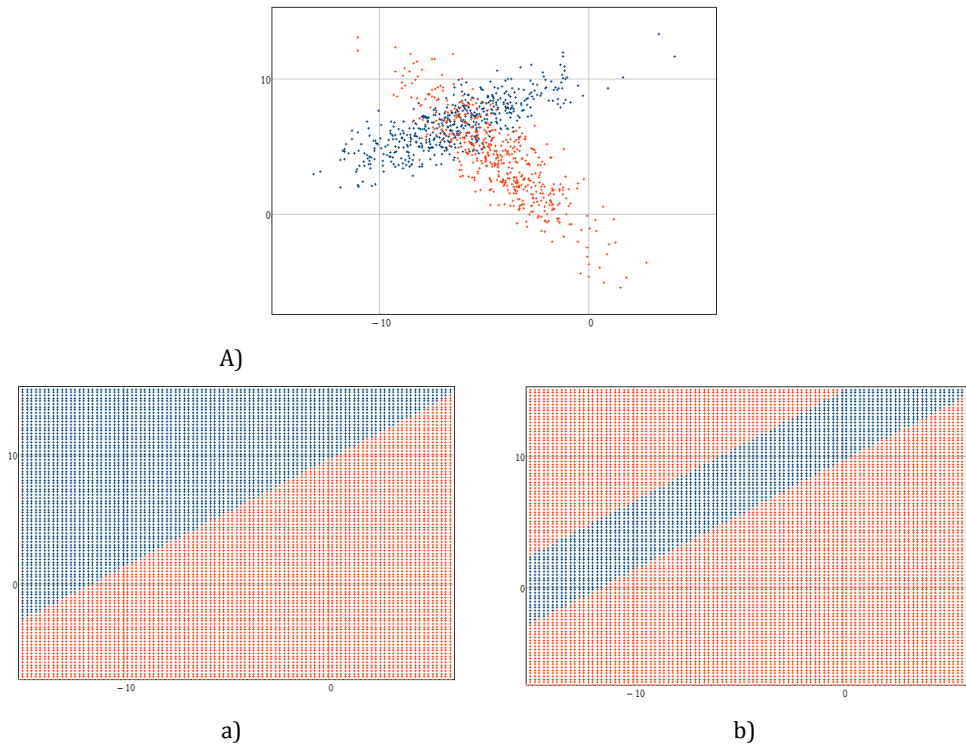
Rozkłady obrazów oraz wyniki badania klasyfikatora dla $L = 2$ pokazano na rysunku 6. Z porównania błędów klasyfikacji widać, że klasyfikator z dwoma progami okazał się nieco lepszy.

EXPERIMENTS

Studying the described learning and classification algorithms (27, 28) and applying them to multi-class tasks was done in two-dimensional space for the number of classes $L = 2, 3, 4$. A training set for each class was composed of 512 patterns generated with a generator of random numbers having normal distribution. During our experiments we paid special attention to cases, where the pattern distributions overlapped and were difficult to separate. As for the experiments, it was assumed that $\lambda_0 = 0.9$, i.e. the acceptable tolerance for difference in variance for a distribution does not exceed 10%. We calculate adequate estimation coefficients of extension of distributions ε_1 and ε_2 following formulas (8), (9).

Studying the algorithm of learning and classification quality was done for two cases: for one threshold and for two thresholds, however, in the latter case for algorithm (27) and simplified (28). The decision areas obtained are colored in the patterns according to class colors in the source space. If there occurred areas with a neutral decision, they received white color (colorless). The quality of classification referring to particular algorithms was estimated using the mean error probability estimator on the training set.

The pattern distributions and the results of studying classifier $L = 2$ are shown in figure 6. It follows from the comparison of the classification errors that the double-threshold classifier proves to be a little better.



Rys. 6. Dwie klasy obrazów $L = 2$: A) rozkłady klas w przestrzeni źródłowej; obszary decyzyjne klasyfikatora: a) z jednym progami (algorytm (27)), $P_e = 20\%$; b) z dwoma progami (algorytm (27)), $P_e = 17\%$

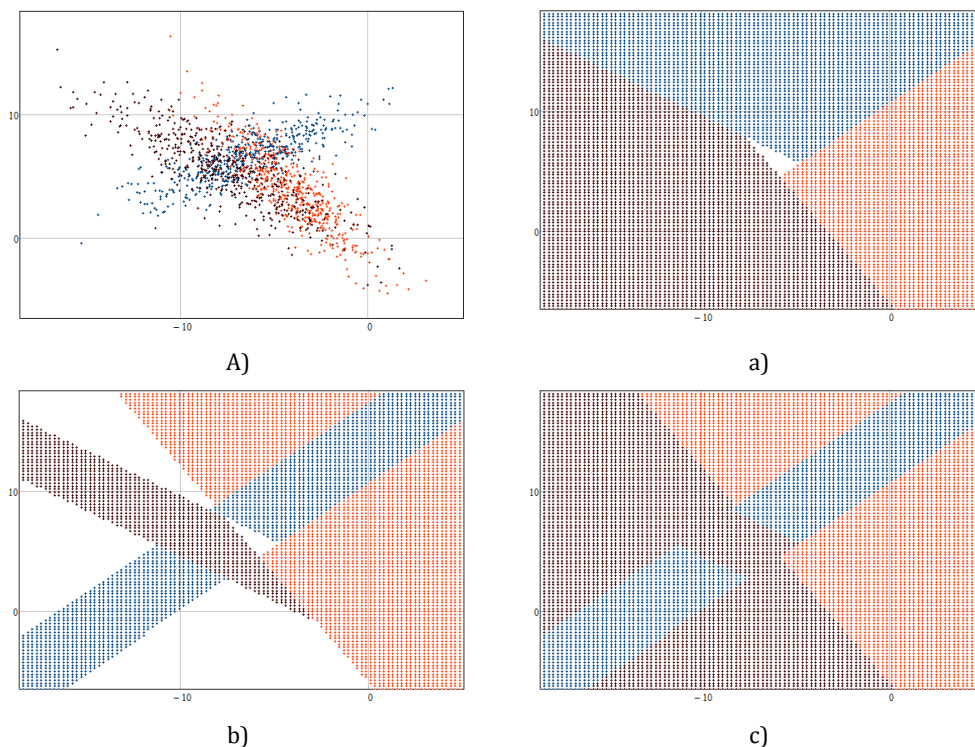
Fig. 6. Two classes of patterns $L = 2$: A) class distributions in the source space; classifier decision areas: a) single-threshold (algorithm (27)), $P_e = 20\%$; b) double-threshold (algorithm (27)), $P_e = 17\%$

W celu sprawdzenia działania algorytmu uczenia i klasyfikacji, gdy $L > 2$, przeprowadzono eksperymenty na trzech i czterech klasach. W przeprowadzonych eksperymentach każdą parę klas rozdzielano jednym progiem (jedną płaszczyzną), a w drugim przypadku dwoma progami (jeżeli wystąpiły odpowiednie warunki). Otrzymane obszary decyzyjne i odpowiednie wyniki klasyfikacji pokazano na rysunkach 7. i 8. Z porównania wyników widać, że klasyfikacja z dwoma progami była lepsza od klasyfikatora z jednym progiem. Jeszcze lepsze wyniki otrzymano, gdy został zastosowany klasyfikator uproszczony. Ale klasyfikator

In order to check the performance of the learning and classification algorithm, when $L > 2$, an experiment was carried out on three and four classes. In the experiments each pair of classes was separated with one threshold (one plane), and in the other case with two thresholds if adequate conditions occurred. The decision areas obtained, and adequate classification results are shown in figures 7 and 8. It follows from the comparison of the results that the classification with two thresholds was better than the single-classifier. Even better results were obtained when the simplified classifier was used. However,

ten wymagał dodatkowego wysiłku w celu dobrania takiej kolejności analizowania par klas, aby błąd był najmniejszy.

this classifier required additional effort in order to select such an order of analyzing pairs of classes that the error should be the least.

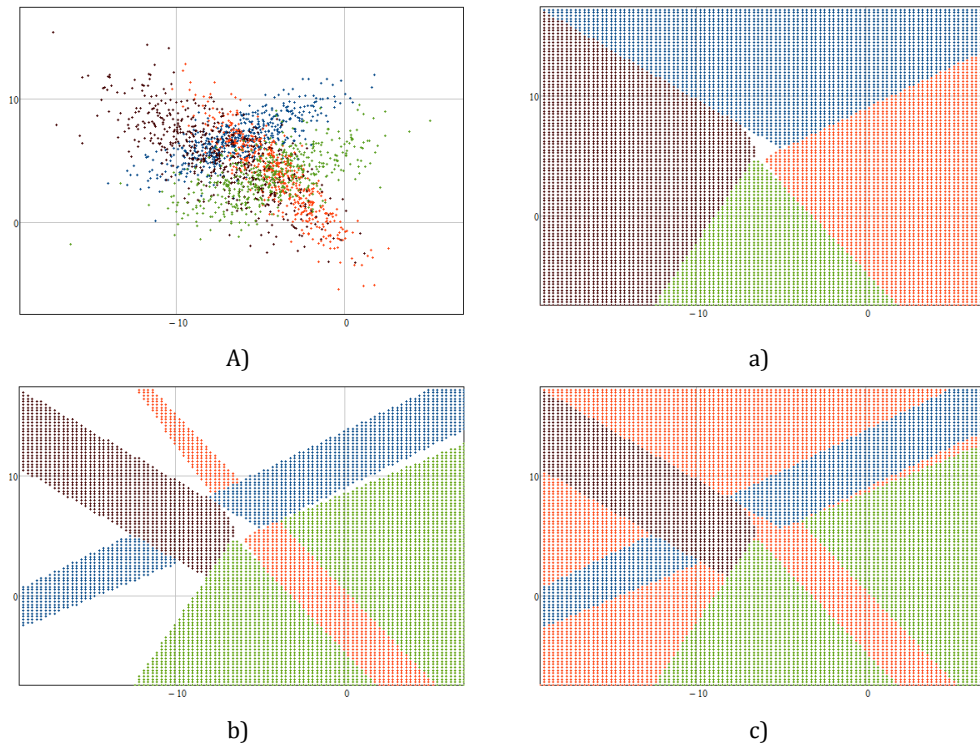


Rys. 7. Trzy klasy obrazów $L = 3$: A) rozkłady klas w przestrzeni źródłowej; obszary decyzyjne klasyfikatora: a) z jednym progami (algorytm (27)), $P_e = 44\%$; b) z dwoma progami (algorytm (27)), $P_e = 41\%$; c) uproszczonego z dwoma progami (algorytm (28)), $P_e = 37\%$

Fig. 7. Three classes of patterns $L = 3$: A) class distributions in the source space; classifier decision areas: a) single-threshold (algorithm (27)), $P_e = 44\%$; b) double-threshold (algorithm (27)), $P_e = 41\%$; c) simplified double-threshold (algorithm (28)), $P_e = 37\%$

Na podstawie przeprowadzonych eksperymentów można powiedzieć, że klasyfikator z dwoma progami dla trudnych przypadków daje lepsze wyniki niż klasyfikator z jednym progiem oraz że zastosowanie klasyfikatora uproszczonego może dodatkowo zmniejszyć błąd.

Based on the experiments carried out, it can be said, that a double-threshold classifier produces better results for difficult cases than a single-threshold classifier and using a simplified classifier can additionally reduce the error.



Rys. 8. Cztery klasy obrazów $L = 4$: A) rozkłady klas w przestrzeni źródłowej; obszary decyzyjne klasyfikatora: a) z jednym progami (algorytm (27)), $P_e = 53\%$; b) z dwoma progami (algorytm (27)), $P_e = 46\%$; c) uproszczonego z dwoma progami (algorytm (28)), $P_e = 44\%$

Fig. 8. Four classes of patterns $L = 4$: A) class distributions in the source area; classifier decision areas: a) single-threshold (algorithm (27)), $P_e = 53\%$; b) double-threshold (algorithm (27)), $P_e = 46\%$; c) simplified double-threshold (algorithm (28)), $P_e = 44\%$

WNIOSKI

Po wyznaczeniu wektora dyskryminacyjnego dla każdej pary klas i zrzutowaniu obrazów można na podstawie samych parametrów rozkładów ocenić liczbę progów potrzebnych do skonstruowania klasyfikatora (**algorytm 1**). Na bazie tego algorytmu skonstruowano klasyfikator, który automatycznie dobiera liczbę progów do rozdzielania każdej pary klas, które występują w zadaniu.

CONCLUSIONS

After determining a discrimination vector for each pair of classes and mapping the patterns, the number of thresholds needed to construct a classifier (**algorithm 1**) can be estimated, using only distribution parameters. This algorithm was used to construct a classifier which automatically selects the number of thresholds needed to separate each class of pairs which occur in the task.

Przeprowadzone eksperymenty pokazują, że dla trudnych przypadków algorytm klasyfikacji z dwoma progami daje mniejszy błąd niż klasyczny klasyfikator z jednym progiem. Warto także zwrócić uwagę, że algorytm klasyfikacji uproszczony (28) z dwoma progami dawał lepsze wyniki od algorytmu (27), przy czym dla algorytmu (28) należy dodatkowo dobrać kolejność analizowania par klas. Jeżeli klasy w przestrzeni źródłowej będą dobrze odseparowane od siebie, to zaproponowany algorytm uczenia i klasyfikacji działa analogicznie jak algorytm z jednym progiem.

Obecnie pewną wadą algorytmu wyznaczania wektora \mathbf{d} na podstawie klasycznego kryterium Fishera [1, 9] jest brak rozwiązania, gdy $\Delta = 0$ (14), a dla $\Delta \approx 0$ obliczony wektor \mathbf{d} nie musi być optymalny. Dlatego zawsze, kiedy występuje przypadek z dwoma progami lub brak progów, wektor \mathbf{d} należy liczyć według rozszerzonego kryterium Fishera [6, 8].

The experiments show that a double-threshold classification algorithm produces a smaller error for difficult cases than a classic single threshold classifier. It is also worth mentioning that simplified double-threshold algorithm (28) produced better results than algorithm (27), however, for algorithm (28) the order in which to analyze pairs of classes must be additionally selected. If classes in the source space are separated well from each other, then the proposed learning and classification algorithm will perform analogically as the single-threshold algorithm.

At present a certain disadvantage relating to determining vector \mathbf{d} using the Fisher classic criterion [1, 9] is the absence of a solution when $\Delta = 0$ (14), and the calculated vector does not have to be optimal for $\Delta \approx 0$. Therefore, always, when there occurs a double-threshold case or absence of thresholds, vector \mathbf{d} must be calculated following the Fisher extended criterion [6, 8].

BIBLIOGRAFIA / REFERENCES

- [1] Duda R., Hart P., Stork D., *Pattern Classification*, second ed., John Wiley and Sons, New York 2000.
- [2] Foley D. H., Sammon J. W., *An optimal set of discriminant vectors*, 'IEEE Trans Computer', 1975, C-24(3), pp. 281–289.
- [3] Kim H., Drake B. L., Park H., *Multiclass classifiers based on dimension reduction with generalized LDA*, 'Pattern Recognition', 2007, Vol. 40, pp. 2939–2945.
- [4] Li T., Zhu S., Ogihara M., *Using discriminant analysis for multi-class classification: an experimental investigation*, 'Knowledge and Information Systems', 2006, 10(4), pp. 453–472.
- [5] Longstaff I. D., *On Extensions to Fisher's Linear Discriminant*, 'IEEE Transactions on Pattern Analysis and Machine Intelligence', 1987, Vol. 9, No 2, pp. 321–325.
- [6] Malina W., *On an Extended Fisher Criterion for Feature Selection*, 'IEEE Transactions on Pattern Analysis and Machine Intelligence', 1981, Vol. 3, No 5, pp. 611–614.
- [7] Martinez A. M., Zhu M., *Where Are Linear Feature Extraction Methods Applicable?*, 'IEEE Transactions on Pattern Analysis and Machine Intelligence', 2005, Vol. 27, No 12, pp. 1934–1944.
- [8] Okada T., Tomita S., *An extended Fisher criterion for feature extraction — Malina's method and its problems*, 'Electronic and Communications in Japan', 1984, No 3, pp. 159–165.

- [9] Sammon J. W., *An optimal discriminant plane*, 'IEEE Transactions on Comp.', Sept. 1970, pp. 826–829.
- [10] Theodoridis S., Koutroumbas K., *Pattern recognition*, AP 1999.
- [11] Tou J. T., Gonzales R. C., *Pattern Recognition principles*, Addison — Weseley 1974.