

C₃
I N S T Y T U T C Y B E R N E T Y K I T E C H N I C Z N E J

POLITECHNIKI WROCŁAWSKIEJ

Komunikat Nr 398

SELEKCJA CECH W ROZPOZNAWANIU

OBRAZÓW

Praca doktorska/

Zofia Wilimowska

Promotor: Prof.dr hab.inż. Zdzisław Bubnicki

Słowa kluczowe: rozpoznawanie obrazów, selekcja,
ryzyko, oszacowanie ryzyka

W r o c ł a w 1 9 7 6

Nr 3304

Mgr inż. Zofia Wilimowska

Instytut Organizacji i Zarządzania

Zakład Organizacji Produkcji w Przemśle Maszynowym

Komunikat wpłynął do Redakcji Prac Naukowych Instytutu
Cybernetyki Technicznej Politechniki Wrocławskiej
dnia 14.08.1976r.

Poczuwam się do miłego obowiązku serdecznego podziękowania mojemu Promotorowi Panu Prof.dr hab.inż. Zdzisławowi Bubnickiemu za liczne wskazówki i rady oraz za pomoc okazaną mi w czasie realizacji niniejszej pracy. Dziękuję również Panu Doc.dr inż. Tadeuszowi Czarnemu Kierownikowi Zakładu Organizacji Produkcji w Przemysle Maszynowym za przychylność i rady, które niejednokrotnie wykorzystywałam w trakcie pisania pracy. Wyrażam również szczerą wdzięczność wszystkim kolegom, zawsze chętnym służyć radą i pomocą.

Praca dotyczy teorii rozpoznawania obrazów, a ściślej metod selekcji cech, na podstawie których dokonuje się klasyfikacji. W pracy dokonano przeglądu znanych metod i sformułowano nowe dla postawionych wcześniej dwu zadań selekcji, a mianowicie: zadania odrzucania m cech spośród n tak, aby wzrost ryzyka był minimalny, oraz odrzucania maksymalnej liczby cech takich, aby wzrost ryzyka nie przekroczył zadanej liczby $\varepsilon > 0$.

Oba te zadania sformułowano dla problemu rozpoznawania z pełną informacją probabilistyczną. Dla zadań tych zbudowano osiem algorytmów selekcji (po cztery dla każdego z zadań) : cztery z nich są algorytmami optymalnymi, cztery suboptymalnymi. Algorytmy suboptymalne zbudowano na podstawie zdefiniowanych wcześniej oszacowań ryzyka, które porównano z innymi znanymi oszacowaniami. Sześć algorytmów zrealizowano w postaci programów napisanych w języku FORTRAN-1900 na maszynie cyfrową serii ODRA - 1300.

S p i s t r e ś c i

	Strona
1. Metody i zadania selekcji cech w rozpoznawaniu	5
1.1. Wstęp	5
1.2. Sformalizowane przedstawienie problemu rozpoznawania	11
1.3. Przegląd metod selekcji	17
1.4. Sformalizowane przedstawienie zadań selekcji	23
2. Zastosowanie metody podziału i ograniczeń do selekcji cech	29
2.1. Krótki przegląd metod programowania dyskretnego	29
2.2. Sprowadzenie zadań selekcji do zadań typu kombinatorycznego	43
2.3. Algorytmy selekcji cech dla zadania 1 oraz zadania 2	46
3. Algorytmy selekcji cech dla szczególnych przypadków rozpoznawania	68
3.1. Optymalny algorytm selekcji w oparciu o współczynnik Bhattacharyya φ [26]	68
3.2. Suboptymalny algorytm selekcji	83
4. Suboptymalny algorytm selekcji w oparciu o zbieżne oszacowanie ryzyka	93
4.1. Definicja i własności miary $M_K(J/X)$	94
4.2. Porównanie niektórych oszacowań ryzyka	101
4.3. Algorytmy selekcji wykorzystujące oszacowanie $1 - M_K(J/X)$	108
5. Omówienie programów i ich wyników	116
6. Wnioski i uwagi końcowe	161
Literatura	166

1. METODY I ZADANIA SELEKCJI CECH W ROZPOZNAWANIU.

1.1. Wstęp.

Rozpoznawanie obrazów rozumiane jest w niniejszej pracy jako tworzenie algorytmów klasyfikujących obiekty na podstawie ich charakterystycznych cech. Algorytmy te mogą być realizowane w postaci urządzeń fizycznych, w postaci programów na maszynę cyfrową bądź w formie zapisu słownego.

Algorytmy rozpoznawania obrazów znalazły zastosowanie w wielu dziedzinach [7,9,31].

- 1/ w diagnostyce lekarskiej,
- 2/ w przewidywaniu pogody,
- 3/ w rozpoznawaniu dźwięków mowy,
- 4/ w czytaniu pisma,
- 5/ w klasyfikowaniu zdjęć fotograficznych,
- 6/ w rozpoznawaniu sytuacji technologicznych i innych.

Stosowanie tych algorytmów do rozpoznawania sytuacji technologicznych jest jedną z ważniejszych dziedzin zastosowania. Umożliwia ono sterowanie systemami produkcyjnymi w czasie rzeczywistym dzięki możliwości zastosowania do tych celów maszyn cyfrowych.

Przed przystąpieniem do rozpoznawania otrzymuje się od osób kompetentnych, ekspertów pewien zbiór cech charakteryzujących obiekty, które mają być na podstawie tych cech klasyfikowane. Jeśli na przykład obiektem jest chory człowiek a zadaniem rozpoznawania jest postawienie diagnozy, czyli określenie choroby tego człowieka, to przed przystąpieniem do realizacji zadania

od lekarzy uzyskuje się zbiór objawów np.: temperatura ciała, morfologia i ciśnienie krwi, bóle głowy, itd., na podstawie których należy stawiać diagnozę, czyli przeprowadzić klasyfikację. Jeśli zaś rozpoznawać się będzie zachodzące sytuacje technologiczne to od konstruktorów, technologów, organizatorów otrzymuje się zbiór cech obejmujących własności konstrukcyjne, technologiczne, organizacyjne, itp., charakteryzujące procesy technologiczne; na podstawie tego zbioru przeprowadzać się będzie klasyfikację. Jednakże liczba elementów takiego zbioru jest zwykle bardzo duża. Niektóre z tych elementów nie wnoszą żadnej albo bardzo mało informacji o klasie obiektu. Są więc z punktu widzenia rozpoznawania mało przydatne, a nawet komplikują algorytm, co:

- 1/ przedłuża czas realizacji algorytmu,
- 2/ zwiększa koszty obliczeń,
- 3/ niejednokrotnie uniemożliwia realizację algorytmu (budowę klasyfikatora).

Zachodzi więc konieczność wyboru z pełnego zbioru pewnego podzbioru cech, na podstawie którego buduje się odpowiednie algorytmy. Często wstępnego wyboru mogą dokonać sami specjaliści, którzy z uwagi na swoje doświadczenie specjalistyczne z góry wiedzą, że pewne cechy można pominąć, że są one mało przydatne. Eliminować należałoby również te cechy, które nie są mierzalne, albo też pomiar ich wartości wymaga skomplikowanych i kosztownych urządzeń pomiarowych. Podzbiór po takim wstępnym wyborze może okazać się w dalszym ciągu zbyt liczny, aby na jego podstawie można było realizować algorytm rozpoznawania w postaci urządzenia czy programu na maszynę cyfrową (komputer dysponuje ograniczoną

pamięcią). W dalszym ciągu istnieje więc potrzeba zmniejszania liczności takiego podzbioru.

Dalsze zmniejszanie należy przeprowadzać na podstawie pewnych kryteriów wyboru. Jedną z metod zmniejszania jest redukcja [5,6,8,34,35] zbioru cech. Redukcja zbioru cech rozumiana jest jako procedura zmniejszania zbioru cech, na podstawie których realizowany będzie algorytm rozpoznawania, najczęściej przez nieliniową transformację przestrzeni n -wymiarowej w m -wymiarową, $n > m$. Szczególnym przypadkiem redukcji jest selekcja. Selekcję cech można zdefiniować jako metodę ^{wyboru} podzbioru cech ze zbioru większego, zapewniającą minimalizację lub maksymalizację odpowiedniego kryterium. Inaczej mówiąc zadaniem selekcji jest wybór podzbioru cech wnoszących jak największą ilość informacji ^(*), na podstawie których realizowany będzie algorytm rozpoznawania. Wybór ten dotyczyć może także podzbioru cech wnoszących jak najmniejszą ilość informacji ^(*).

W drugim przypadku wybrane cechy należy wyeliminować. Jednym z podstawowych kryteriów selekcji jest ryzyko. W zależności od apriorycznej informacji o rozpoznawanym obiekcie istnieją różne algorytmy selekcji. W ostatnich latach ukazało się wiele prac poświęconych algorytmom selekcji w zadaniach z pełną informacją probabilistyczną [4,15,26,28,36,39]. Proponowane w nich algorytmy są bardzo podobne, a różnią się tylko postawią kryterium selekcji. Zazwyczaj kryteria proponowane przez autorów algorytmów

(*) - Ilość informacji jest tutaj rozumiana jako wartość odpowiedniego kryterium.

są różnymi oszacowaniami ryzyka, a więc algorytmy te są suboptymalne względem ryzyka. E. Rasek [36] proponuje algorytm selekcji oparty na statystykach zdefiniowanych dla każdej cechy. Statystyki te muszą spełniać odpowiednie warunki. C.C. Babu [4] rozważa przypadek dwu klas, dla których odpowiednie gęstości warunkowe są rozkładami normalnymi. Jako kryterium selekcji przyjmuje dywergencję. Przypadek dwu klas rozważa również Kailath [26], który jako kryterium selekcji proponuje miarę zwaną współczynnikiem Bhattacharyya. Keinosuke Fukunaga [28] rozważa również przypadek z rozkładami normalnymi i stosując transformatę Karhunen-Loevego sprowadza macierz kowariancji do macierzy diagonalnej, a następnie pokazuje, że w przypadku dwu klas należałoby selekcje przeprowadzać osobno dla każdej klasy. Vilmansen [39] przedstawił kilka algorytmów, w których jako kryteria selekcji przyjął miarę Kołomogorowa, dywergencję, informację, współczynnik Bhattacharyya i porównał te algorytmy. Devijver [15] zdefiniował pewne oszacowanie ryzyka, które nazwał odległością Bayesa i zbudował algorytm selekcji traktując to oszacowanie jako kryterium.

Wszystkie wyżej wymienione algorytmy polegają na odrzuceniu tych cech, dla których kryteria przyjmują wartości ekstremalne. Ponieważ kryteria te zazwyczaj nie są ani multiplikatywne, ani addytywne ze względu na cechy, algorytmy są zazwyczaj suboptymalne. Co więcej, jak już wcześniej powiedziano, kryteria są oszacowaniami ryzyka, a więc proponowane algorytmy są suboptymalne również względem ryzyka. Drugą zasadą tych algorytmów jest odrzucenie tych cech, które powodują najmniejszą zmianę kryterium w porównaniu z pełnym zbiorem cech. Takie bezpośrednie podejście wymaga przeprowadzenia bardzo dużej liczby obliczeń. Zapewnia ono optymalność selekcji względem oszacowania, ale w dalszym

ciągu tylko suboptymalność względem ryzyka.

W tej rozprawie rozważa się przypadek, w którym zarówno klasy, jak i zbiory cech (wektory) traktowane są jako zmienne losowe, dla których znana jest a priori pełna informacja probabilistyczna. Zawęzi to nieco zakres stosowania wyników, jednakże zauważyć należy, że niektóre algorytmy rozpoznawania w przypadkach niepełnej informacji lub zupełnego braku informacji a priori, w dużej części opierają się na wyestymowanych parametrach rozkładu czy rozkładach wyestymowanych na podstawie ciągów uczących, tzn. po wyestymowaniu parametrów rozkładów prawdopodobieństwa traktuje się te zadania jak zadania z pełną informacją probabilistyczną. Można więc powiedzieć, że przypadki z pełną informacją probabilistyczną obejmują dość szeroki zakres zagadnień praktycznych.

Dla tych przypadków z pełną informacją probabilistyczną i przypadków, w których algorytmy rozpoznawania opierają się na pełnej informacji wcześniej wypracowanej na podstawie ciągów uczących, sformułowano dwa podstawowe zadania:

- 1/ odrzucenie m cech spośród n , $m < n$, takich aby wzrost ryzyka był minimalny,
- 2/ odrzucenie możliwie największej liczby cech, takich aby wzrost ryzyka nie przekroczył z góry zadanej liczby $\varepsilon > 0$.

Zadanie pierwsze interpretować można jako sytuację, w której specjalistom zależy na jak najdokładniejszym przeprowadzeniu klasyfikacji (mało prawdopodobnych pomyłek), ale liczba cech, która stanowi podstawę klasyfikacji jest za duża i uniemożliwia realizację algorytmu rozpoznawania w postaci urządzenia czy programu na maszynie cyfrową bądź też powoduje zbyt długi czas

klasyfikacji. Wyznacza się więc liczbę cech, jaką należy odrzucić, aby można było algorytm zrealizować. Problem polega tylko na odpowiednim wyborze tych cech, aby jakość rozpoznawania jak najmniej się pogorszyła.

Zadanie drugie natomiast dotyczy sytuacji, w której specjalistom nie zależy na zbyt dużym prawdopodobieństwie poprawnej klasyfikacji. Zakłada się, że zadowalać ich będzie klasyfikacja przeprowadzana z pewną z góry założoną dokładnością, mniejszą od osiąganą na dotychczasowym zbiorze cech.

Można więc kosztem dokładności zmniejszyć liczbę cech, przez co osiągnie się dodatkowe korzyści związane z dużym uproszczeniem algorytmów rozpoznawania a przede wszystkim ze zmniejszeniem kosztów klasyfikacji. W takiej sytuacji należy oczywiście dążyć do odrzucenia możliwie największej liczby cech. Jeśli różnicę między dokładnością rozpoznawania na pełnym zbiorze cech a dokładnością z góryadaną oznaczymy przez ε , wówczas zadanie sprowadzi się do odrzucenia możliwie największej liczby cech takich, aby ich pominięcie nie pogorszyło dokładności klasyfikacji więcej niż o ε .

Dla tak postawionych zadań selekcji cech w niniejszej pracy zbudowano algorytmy optymalnej selekcji a także zdefiniowano pewne oszacowania ryzyka i przyjęto je jako kryteria selekcji. Jedno z tych oszacowań posiada ważną własność, a mianowicie własność zbieżności do ryzyka przy pewnym parametrze dążącym do nieskończoności. W pracy zbudowano osiem algorytmów selekcji. Cztery algorytmy są algorytmami, w których jako kryterium selekcji przyjęto ryzyko. Cztery pozostałe są algorytmami suboptymalnymi. Sześć z tych algorytmów - dwa optymalne i cztery suboptymalne -

zrealizowano w postaci programów na maszynę cyfrową.

W rozdziale pierwszym przedstawiono ogólnie problemy związane z rozpoznawaniem i selekcją, następnie sformalizowano problem rozpoznawania obrazów, omówione algorytmy selekcji proponowane przez różnych autorów, przedstawiono w sposób sformalizowany zadania niniejszej pracy i omówiono szczególne przypadki rozpoznawania rozpatrywane w dalszych rozdziałach.

W rozdziale drugim sprowadzono zadania selekcji do zagadnień programowania dyskretnego, dokonano wyboru metody (metoda podziału i ograniczeń), a następnie przedstawiono optymalne algorytmy selekcji dla zadań sformułowanych w rozdziale pierwszym.

W rozdziale trzecim omówiono szczególne przypadki rozpoznawania i dla tych przypadków zbudowano algorytmy optymalne oraz algorytmy suboptymalne dla każdego z zadań, na podstawie zdefiniowanego wcześniej oszacowania ryzyka.

W rozdziale czwartym przedstawiono definicję i własności pewnego oszacowania zbieźnego do ryzyka, przeprowadzono porównanie niektórych oszacowań ryzyka stosowanych jako kryteria selekcji oraz zdefiniowanego w tym rozdziale oszacowania. Następnie w oparciu o to oszacowanie zaproponowano algorytmy selekcji.

1.2. Sformalizowane przedstawienie problemu rozpoznawania

Rozpoznawanie jest to proces tworzenia algorytmów, których zadanie polega na klasyfikowaniu obiektów na podstawie zbioru charakterystycznych cech.

Zakłada się, że zbiór cech, który jest podstawą klasyfikacji, stanowi zbiór n -liczb rzeczywistych $(x_1, x_2, \dots, x_n)$, zbiór ten

nazywa się obrazem. Każdy obraz może być reprezentowany przez punkt w n -wymiarowej przestrzeni rzeczywistej $\mathcal{X}^n \subset \mathcal{R}^n$.

Współrzędne tego punktu (cechy, składowe obrazu) w prostokątnym układzie współrzędnych są liczbami rzeczywistymi $x_1, x_2, \dots, x_n$.

Wyprowadzony ze środka układu współrzędnych do punktu

$(x_1, x_2, \dots, x_n)$ wektor x , nazywa się wektorem obrazu.

Zbiór klas jest skończonym zbiorem dyskretnym $\Omega_L = (\omega_1, \omega_2, \dots, \omega_L)$

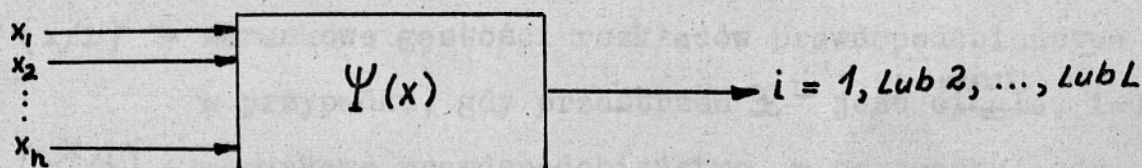
Najczęściej w praktyce operuje się zbiorem indeksów klas,

$\bar{\Omega}_L = \{1, 2, \dots, L\}$, czyli zbiór klas stanowi skończony zbiór liczb naturalnych.

Zadaniem rozpoznawania jest utworzenie takiego algorytmu

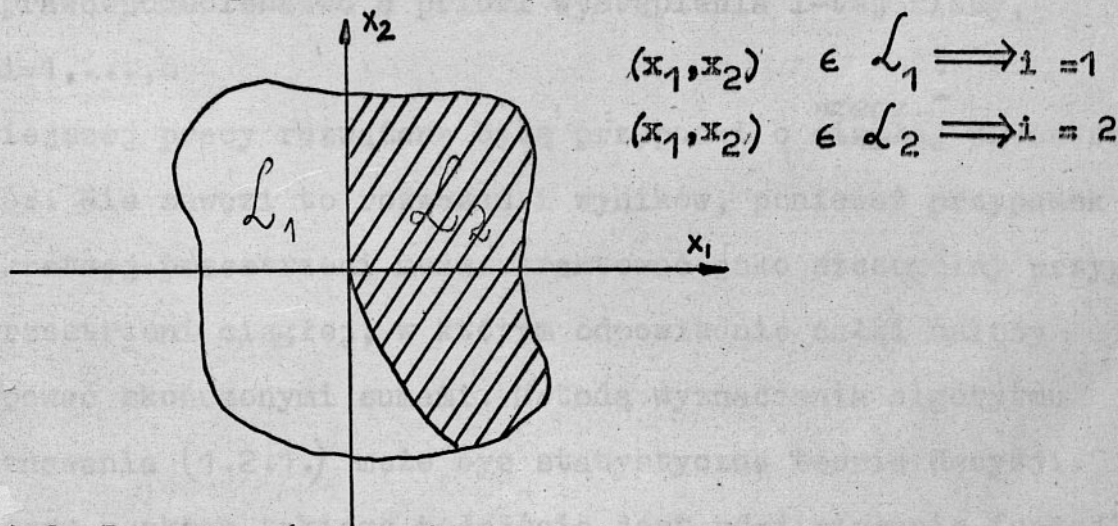
$$i = \Psi(x), \quad i = 1, \dots, L, \quad /1.2.1/$$

który zbiorowi n -liczb rzeczywistych $x = \{x_1, x_2, \dots, x_n\}$ przyporządkuje jeden z elementów zbioru $\bar{\Omega}_L$.



Rys.1.1 Algorytm klasyfikacji

Niech $\mathcal{L}_i$ oznacza zbiór punktów w $\mathcal{X}^n$, którym przyporządkowana jest liczba i , $i = 1, \dots, L$; wówczas każdemu numerowi klasy odpowiada zbiór punktów w $\mathcal{X}^n$, rys.2.



Rys.1.2. Przestrzeń obrazów.

W zależności od ilości ^{jakości} informacji apriorycznej o obiektach można rozważać przypadki:

- 1/ deterministyczne,
- 2/ z pełną informacją probabilistyczną,
- 3/ z niepełną informacją probabilistyczną,
- 4/ zupełny brak informacji a priori, dysponuje się tylko odpowiednim ciągiem uczącym.

W przypadkach 2,3,4 zakłada się, że obrazy oraz klasy są zmiennymi losowymi, czyli obraz ^{i jego klasę} reprezentuje para zmiennych losowych (X, J) .

Pełna informacja probabilistyczna zakłada znajomość odpowiednich rozkładów [21], a mianowicie

$f(x/i)$ - warunkowe gęstości rozkładów prawdopodobieństwa w przypadku, gdy przestrzeń $\mathcal{X}^n$ jest ^{niecz.} ciągła, $i=1, \dots, L$

$p(x^k/i)$ - warunkowe prawdopodobieństwo, w przypadku, gdy przestrzeń $\mathcal{X}^n$ jest dyskretna; $i=1, \dots, L$,
 $k=1, \dots, s$, s - liczba obrazów /punktów/
w przestrzeni $\mathcal{X}^n$, /1.2.2./

p_i - prawdopodobieństwo a priori wystąpienia i -tej klasy,
 $i=1, \dots, L$

W niniejszej pracy rozważane będą przypadki o ^{niecz.} ciągłej przestrzeni obrazów. Nie zawęzi to rozważań i wyników, ponieważ przypadek o dyskretnej przestrzeni można traktować jako szczególny przypadek przestrzeni ciągłej, w którym odpowiednie całki należy zastępować skończonymi sumami. Metodą wyznaczania algorytmu rozpoznawania (1.2.1.) może być statystyczna teoria decyzji. Pierwszym punktem takiego podejścia jest zdefiniowanie funkcji

strat $\alpha(i/j)$,

$$\alpha(i/j) = \begin{cases} \alpha, & \text{gdy } i=j \\ \beta, & \text{gdy } i \neq j, \end{cases}$$

/1.2.3./

$$i, j = 1, 2, \dots, L,$$

tzn. funkcja reprezentuje "karę" wynikłą z zaklasyfikowania do klasy i -tej obiektu, który w rzeczywistości należy do klasy j -tej oraz "nagrodę" za zgodną klasyfikację.

Algorytm minimalizujący średnią wartość $\alpha(i/j)$ nazywa się algorytmem optymalnym.

Wyrażenie

$$L_x(i) = \sum_{j=1}^L \alpha(i/j) p(j/x) \quad /1.2.4./$$

przedstawia średnią stratę warunkową dla każdego obrazu x .

$p(j/x)$ - oznacza prawdopodobieństwa a posteriori wystąpienia j -tej klasy pod warunkiem, że pojawił się obraz x ,

$$p(j/x) = \frac{p_j f(x/j)}{\sum_{i=1}^L p_i f(x/i)} \quad /1.2.5./$$

Algorytm rozpoznawania (1.2.1) w tym przypadku ma postać

$$i_0 = \Psi(x) \iff L_x(i_0) \leq L_x(i), \quad /1.2.6./$$

$$i = 1, 2, \dots, L.$$

Jeśli algorytm (1.2.6.) minimalizuje $L_x(i)$, to również minimalizuje wartość oczekiwaną średniej straty warunkowej (1.2.7) po wszystkich obrazach, którą oznacza się zazwyczaj R .

$$\begin{aligned} R_{\psi(x)} &= \int_{\mathcal{X}^n} \sum_{j=1}^L \alpha(i/j) p(j/x) f(x) dx = \\ &= \sum_{j=1}^L \int_{\mathcal{X}^n} \alpha(i/j) f(x/j) p_j dx, i = \psi(x) \end{aligned} \quad /1.2.7./$$

i nazywa ryzykiem średnim albo krótko ryzykiem.

Wartość

$$R^* = \inf_{\psi(x)} R_{\psi(x)} \quad /1.2.8./$$

nazywa się ryzykiem bayesowskim.

Dla szczególnej postaci funkcji strat, a mianowicie

$$\alpha(i/j) = \begin{cases} 0, & \text{gdy } i=j \\ 1, & \text{gdy } i \neq j, i, j=1, \dots, L, \end{cases} \quad /1.2.9./$$

średnia strata warunkowa (1.2.4.) wyraża się wzorem

$$\begin{aligned} L_x(i) &= \sum_{j=1}^L \alpha(i/j) p(j/x) \quad /1.2.5./, /1.2.9./ \\ &= \frac{1}{f(x)} \sum_{j=1}^L p_j f(x/j) = \frac{1}{f(x)} [f(x) - p_i f(x/i)] \end{aligned} \quad /1.2.10./$$

Algorytm rozpoznawania minimalizujący tę stratę ma postać

$$\begin{aligned} \psi(x) = i_0 &\iff p_{i_0} f(x/i_0) = \max_i p_i f(x/i), \\ i_0, i &= 1, 2, \dots, L, i_0 \neq i. \end{aligned} \quad /1.2.11./$$

Okazuje się, że algorytm ten nie tylko minimalizuje ryzyko średnie R , ale, co więcej,

$$R = R^* \quad /1.2.12./$$

przedstawia prawdopodobieństwo błędnej klasyfikacji P_e .

$$\begin{aligned} R^* &= \int_{\mathcal{X}^n} [f(x) - f(x/i_0) p_{i_0}] \frac{1}{f(x)} f(x) dx = \quad /1.2.13./ \\ &= \int_{\mathcal{X}^n} f(x) dx - \int_{\mathcal{X}^n} f(x/i_0) p_{i_0} dx = 1 - \int_{\mathcal{X}^n} \max_i f(x/i) p_i dx = P_e. \end{aligned}$$

Np. dla $L=2$

$$\begin{aligned} R^* &= 1 - \int_{\mathcal{X}^n} \max_i f(x/i) p_i dx = 1 - \int_{Dx_1} p_1 f(x/1) dx + \int_{Dx_2} p_2 f(x/2) dx = \\ &= \int_{Dx_1} p_2 f(x/2) dx + \int_{Dx_2} p_1 f(x/1) dx, \end{aligned}$$

gdzie

$$\begin{aligned} Dx_1 &= \left\{ x: p_1 f(x/1) \geq p_2 f(x/2) \right\} \\ Dx_2 &= \left\{ x: p_2 f(x/2) > p_1 f(x/1) \right\}, \end{aligned}$$

jasne, że

$$Dx_1 \cup Dx_2 = \mathcal{X}^n.$$

W przypadku niepełnej informacji probabilistycznej, tzn. np. w przypadku, gdy znana jest postać funkcyjna gęstości $f(x/i)$ $i=1, \dots, L$, ale nieznane są parametry tych rozkładów, np.: wartości oczekiwane, wariancje, należy je wyznaczyć na podstawie ciągu uczącego, znajdując ich odpowiednie estymatory.

W przypadku, gdy znane są parametry rozkładów, ale nieznane są postaci funkcyjne, należy dokonać wyboru odpowiednich postaci w oparciu o testy zgodności.

W przypadku zupełnego braku informacji apriorycznej, gdy dysponuje się tylko ciągiem uczącym, istnieją dwa różne podejścia.

Jedną z metod jest estymacja parametrów rozkładu, a następnie wybór odpowiedniej postaci rozkładu w oparciu o testy zgodności, czyli wypracowanie pełnej informacji apriorycznej tak jak w poprzednich przypadkach.

Druga metoda różni⁹ się zasadniczo od poprzednich, polega na stosowaniu odpowiednich algorytmów nie wymagających znajomości rozkładów. Takimi algorytmami są [10,3,14,23].

- 1/ algorytm Nearest Neighbour [10],
- 2/ " Nearest Mean [10],
- 3/ " Least Interval [10],
- 4/ " funkcji potencjalnych [3],
- i inne

czyli tzw. algorytmy z trenerem oraz algorytmy uczące się, adaptacyjne [14,23] .

1.3.

Przegląd metod selekcji.

Jak już wspomniano we wstępie, w niniejszej rozprawie rozważane będą te przypadki rozpoznawania, dla których znam jest pełna informacja probabilistyczna, przypadki przedstawione w rozdziale 1.2. Istnieje wiele algorytmów selekcji dla tych przypadków. Jednakże ze względu na skomplikowaną postać ryzyka, skomplikowaną w sensie analitycznego obliczania wartości ryzyka, a także na nieaddytywność i niemultiplikatywność ryzyka, względem cech, istniejące algorytmy opierają się głównie na oszacowaniach ryzyka, nie wprost na ryzyku. Poniżej przedstawione zostaną niektóre najczęściej stosowane oszacowania brane jako kryteria selekcji.

1. Oszacowanie wyznaczone na podstawie miary Kołmogorowa, oznaczaną zwykle przez $K_D(X, J)$, [28, 39].

$$K_D(X, J) = \frac{1}{2} \sum_{i=1}^L p_i \int_{\mathbb{R}^n} |f(x/i) - f(x)| dx. \quad /1.3.1./$$

Dla $p_1 = p_2 = \dots = p_L = 1/L$ oszacowanie ryzyka jest następujące

$$R^* \leq 1 - \frac{1}{L} - \frac{1}{L-1} K_D(X, J). \quad /1.3.2./$$

Łatwo pokazać, że jeśli $L=2$, $p_1 = p_2 = \frac{1}{2}$, wówczas

$$R^* = \frac{1}{2} - K_D(X, J). \quad /1.3.3./$$

2. Oszacowanie wyznaczone na podstawie miary zwanej informacją $J(X, J)$, [18, 25, 34, 36, 39].

$$J(X, J) = \sum_{i=1}^L p_i \int_{\mathbb{R}^n} f(x/i) \log \frac{f(x/i)}{f(x)} dx. \quad /1.3.4./$$

Oszacowanie otrzymane w wyniku zastosowania informacji przyjmuje następującą postać

$$R^* \leq \frac{1}{2 \log 2} [H(P_L, 1-P_L) + (1-P_L) \log(L-1) - J(X/J)], \quad /1.3.5./$$

gdzie

$$P_L = \max(p_1, p_2, \dots, p_L),$$

$$H(p_1, \dots, p_L) = - \sum_{i=1}^L p_i \log p_i.$$

3. Oszacowanie wyznaczone na podstawie współczynnika Bhattacharyya η , [12, 15, 26, 27, 34, 39].

$$S = \sum_{i=1}^L p_i \int_{\mathbb{R}^n} \sqrt{f(x/i) f(x)} dx . \quad /1.3.6./$$

Oszacowanie ryzyka w tym przypadku jest następujące

$$R^* \leq \frac{1}{L-1} S + \frac{L^2 + 3L + 1}{L^2 - L} . \quad /1.3.7./$$

W przypadku $L=2$, jak pokazano w artykule [26], postać współczynnika wyrazić można następująco

$$S_2 = \int_{\mathbb{R}} \sqrt{f(x/1) f(x/2)} dx , \quad /1.3.8./$$

a wynikające stąd oszacowanie ryzyka

$$R^* \leq S_2 \sqrt{p_1 p_2} . \quad /1.3.9./$$

Zauważyć należy, że współczynnik S_2 jest multiplikatywny względem cech, w przypadku gdy cechy są niezależne.

4. Oszacowanie wyznaczone na podstawie miary zdefiniowanej przez Devijver'a [15], $B(J/X)$, którą autor nazwał odległością Bayes'a.

$$B(J/X) = \int_{\mathbb{R}^n} \frac{1}{f(x)} \sum_{i=1}^L [f(x/i) p_i]^2 dx . \quad /1.3.10/$$

Oszacowanie wynikające z tej miary jest następujące

$$R^* \leq 1 - B(J/X) . \quad /1.3.11./$$

Co więcej, udowodniono [15], że największe odchylenie, czyli największa różnica między ryzykiem a tym oszacowaniem, $\Delta_{R_B}(x)$ zawiera się między $1/8$ a $1/4$.

$$\frac{1}{8} \leq \Delta R_B \leq \frac{1}{4} \quad . \quad /1.3.12./$$

Udowodniono również, że $B(J/X)$ można zinterpretować jako prawdopodobieństwo poprawnej klasyfikacji w algorytmie Nearest Neighbour.

5./ Oszacowanie otrzymane w wyniku zastosowania miary zdefiniowanej przez Ito, oznaczanej zwykle przez Q_k , [15], dla $L=2$

$$Q_k = \frac{1}{2} - \frac{1}{2} \int_{\mathbb{R}^n} \left\{ [p_1 f(x/1) - p_2 f(x/2)] [p_1 f(x/1) - p_2 f(x/2)] \right\}^{\frac{1}{2k+1}} dx$$

/1.3.13./

Oszacowanie ryzyka wynikające z zastosowania tej miary ma postać

$$R^* \leq Q_k, \quad k=1, \dots, \infty, \quad , \quad /1.3.14./$$

$$\lim_{k \rightarrow \infty} Q_k = R^*.$$

/1.3.15/

Biorąc zatem wystarczająco duże k , można otrzymać oszacowanie dowolnie bliskie R^* .

6. Oszacowanie otrzymane w wyniku zastosowania średniej warunkowej entropii [40,34] oznaczanej zwykle przez $h(J/X)$.

$$h(J/X) = \sum_{i=1}^L \left\{ p(i/x) [1 - p(i/x)] \right\} \quad /1.3.16/$$

Oszacowanie wynikające z zastosowania entropii ma postać

$$R^* \leq h(J/X). \quad /1.3.17/$$

7. Oszacowanie wynikające z zastosowania miary niejednorodności zdefiniowanej przez Minkowskiego i, oznaczanej przez

$$MN_k(J/X)$$

$$MN_k(J/X) = E_x \left\{ \sum_{i=1}^L \left| p(i/x) - \frac{1}{L} \right|^{2(k+1)/2k+1} \right\} \quad /1.3.18/$$

Oszacowanie na R^* ma postać

$$R^* \leq \frac{L-1}{L} - MN_0(J/X) \quad /1.3.19/$$

8. Często również jako kryterium selekcji wykorzystuje się entropię Shannona

$$H(J/X) = E_x \left\{ \sum_{i=1}^L p(i/x) \log p(i/x) \right\} \quad /1.3.20/$$

Wszystkie algorytmy selekcji, w których jako kryterium przyjmowane są wyżej wymienione oszacowania, opierają się na trzech zasadniczych metodach postępowania.

1. Obliczanie dla każdej składowej obrazu, dla każdej cechy x_i , $i=1, \dots, n$ odpowiednich oszacowań i odrzucania tych cech, dla których oszacowania te osiągają wartości maksymalne.
2. Wyznaczenie dla każdego możliwego rozwiązania odpowiedniego oszacowania i wybór tego, dla którego przyjmuje ono wartość najmniejszą, innymi słowy bezpośredni przegląd wszystkich rozwiązań.
3. Kolejne odrzucanie cech, to znaczy odrzucenie 1 cechy, wybór najlepszego rozwiązania, następnie odrzucenie dwóch cech, poprzednio wybranej i następnej i znów wybór najlepszego rozwiązania, itd.

Białasiewicz [8] pokazuje, że każde wklęsłe względem $p(i/x)$ oszacowanie ryzyka można przyjąć jako kryterium selekcji.

E. Rasek [36] proponuje następujący algorytm selekcji:

- Dla każdej cechy x_i , $i=1, \dots, n$ definiuje się statystykę Q_i spełniające warunki
 1. Q_i - jest pojedynczą liczbową statystyką,
 2. Q_i - jest monotoniczną funkcją P_i , gdzie P_i oznacza prawdopodobieństwo poprawnej klasyfikacji i -tej cechy,
 3. Q_i - jest funkcją $f(x_i/j)$
 4. $Q_i(J, X) = Q_i[g(J), h(x)]$, gdzie g i h są pewnymi arbitralnymi transformacjami
- Odrzuca się te cechy, dla których statystyka Q_i przyjmuje najmniejsze wartości.

Łatwo zauważyć, że wszystkie zdefiniowane wcześniej oszacowania opierają się na tej zasadzie.

C.C. Babu [4] rozważa przypadek $L=2$ oraz zakłada, że gęstości $f(x/1)$, $f(x/2)$ są gęstościami rozkładu normalnego. Algorytm, który autor proponuje, jest następujący:

1. Przekształcić przestrzeń obrazów $\mathbb{R}^n$ w przestrzeń $\mathcal{Y}$, aby w tej przestrzeni obraz stanowiły zmienne losowe niezależne

$$\mathcal{Y} = P\mathbb{R}^n.$$

/1.3.21./

2. Należy znaleźć takie przekształcenia Q , taką macierz $[(n-m) \times n]$ -wymiarową (jeśli liczba odrzuconych cech ma równać się m), której wiersze są liniowo niezależne

$$Z = QY,$$

/1.3.22/.

aby dywergencja

$$J_D = \sum_{i=1}^L \int_Z [\bar{f}(z/1) p_1 - \bar{f}(z/2)] p_2 \log \frac{p_1 \bar{f}(z/1)}{p_2 \bar{f}(z/2)} dz$$

/1.3.23./

przyjęła wartość najmniejszą.

Jeśli przez $s_{i=1,2}$ oznaczy się macierz kowariancji rozkładów $g(y/1)$, $g(y/2)$, przez μ_i wektor wartości oczekiwanych tych rozkładów, to macierz Q będzie optymalna, tzn. będzie minimalizowała J_D wówczas, gdy jako wiersze macierzy Q wybierze się te wektory własne macierzy s_i , które odpowiadają największym wartościom własnym macierzy $s_i^{-1} \mu_i$.

Istnieje wiele innych algorytmów selekcji [2, 11, 12, 13, 22, 25, 28, 30, 32, 33, 38, 41, 42, 44] dla przypadków, w których nie jest znana pełna informacja probabilistyczna a priori, np.: [44], stosowano metodę programowania dynamicznego dla przypadku rozpoznawania przy zupełnym braku informacji; dysponuje się tylko ciągiem uczącym, gdzie jako kryterium przyjęto pewną addytywną miarę.

Wiele prac poświęconych jest również selekcji cech dla szczególnych przypadków rozpoznawania, np.: rozpoznawania liter.

1.4. Sformalizowane przedstawienie zadań selekcji.

Przed przystąpieniem do dokładnego sformalizowania zadań selekcji, które w sposób szczególny przedstawiono we wstępie wprowadzonej zostanie kilka oznaczeń i definicji.

Niech, jak poprzednio, $\mathcal{X} = \mathcal{X}^n$ oznacza przestrzeń obrazów, przy czym obraz stanowi punkt w tej przestrzeni $x = (x_1, x_2, \dots, x_n)$.

Wówczas przestrzeń $\mathcal{X}^i$ oznaczać będzie przestrzeń $(n-1)$ -wymiarową, po wyeliminowaniu cechy $x_i, i=1, \dots, n$

$R^*(i)$ - oznacza ryzyko, które otrzyma się przeprowadzając rozpoznanie (klasyfikację) w oparciu o $(n-1)$ cech.

Niech dalej

$\mathcal{X}_i$ - oznacza przestrzeń jednowymiarową dla cechy $x_i, i=1, \dots, n$, natomiast

x^i - oznacza wartość wektora $(n-1)$ -wymiarowego po wyeliminowaniu z wektora x cechy o numerze $i, i=1, \dots, n$.

Wówczas z definicji

$$R^*(i) \triangleq 1 - \int_{\mathcal{X}^i} \max_j \left[\int_{\mathcal{X}_i} f(x/j) p_j dx_i \right] dx^i, \quad j = 1, 2, \dots, L. \quad /1.4.1./$$

Analogicznie, niech

$$\mathcal{X}^{k_1, k_2, \dots, k_s} \quad /1.4.2./$$

- oznacza przestrzeń obrazów $(n-s)$ -wymiarową otrzymaną z przestrzeni $\mathcal{X}$ przez odrzucenie s cech o indeksach $k_1, k_2, \dots, k_s = 1, \dots, n$ /1.4.3./

$\mathcal{X}_i$ - jak wyżej.

Niech dalej

$$x^{k_1, k_2, \dots, k_s} \quad /1.4.4./$$

- oznacza wartość wektora obrazu $(n-s)$ -wymiarowego po wyeliminowaniu (odrzuceniu) cech o indeksach $k_1, k_2, \dots, k_s$, wówczas

$$R^*(k_1, k_2, \dots, k_s) \quad /1.4.5./$$

oznacza ryzyko. jakie otrzyma się w wyniku przeprowadzenia klasyfikacji w oparciu o $(n-s)$ cech, po wyeliminowaniu $x_{k_1}, x_{k_2}, \dots, x_{k_s}$.

Zdefinicji

$$R^*(k_1, \dots, k_s) \triangleq 1 - \int_{x_{k_1}, \dots, x_{k_s}} \max_{i=1, \dots, L} [p_i \int \dots \int f(x/i) dx_{k_1} \dots dx_{k_s}] dx_{k_1}, \dots, dx_{k_s} \quad /1.4.6./$$

Niech

V^n - oznacza zbiór indeksów poszczególnych składowych

$$V^n = \{1, 2, \dots, n\}, \quad /1.4.7./$$

natomiast

$$U_t^q = \{r_1, r_2, \dots, r_t\}^q \quad /1.4.8./$$

- oznacza q -tą kombinację t -elementową zbioru indeksów V^n , n -elementowego

Biorąc pod uwagę (1.4.8.) używać się będzie również zapisu postaci

$$R^{*U_t^q} \equiv R^*(r_1, r_2, \dots, r_t)^q \equiv R(r_1^q, r_2^q, \dots, r_t^q), \quad /1.4.9./$$

$$r_i^q \in U_t^q \subset V^n, \quad i=1, \dots, t$$

Przy powyższych oznaczeniach podstawowe zadania tej pracy można sformułować w następujący sposób.

Z a d a n i e 1 .

Niech dany będzie wektor n -cech mierzalnych $x=(x_1, x_2, \dots, x_n)$, oraz liczba cech, które należy odrzucić - m .

Zadanie polega na znalezieniu takiej kombinacji m -elementowej zbioru V^n , aby odrzucenie cech o indeksach zawartych w tej kombinacji spowodowało najmniejszy wzrost ryzyka R^* , to znaczy należy znaleźć takie q_0 , że

$$R^* U_{m0}^{q_0} = \min_q R^* U_m^q, \quad q = 1, 2, \dots, \binom{n}{m} \quad /1.4.10./$$

Z a d a n i e 2 .

Niech dany będzie n -elementowy wektor cech oraz liczba ξ , o którą może wzrosnąć ryzyko

$$\xi = R_1 - R^*, \quad /1.4.11./$$

gdzie

R_1 - oznacza ryzyko po selekcji

Zadanie polega na znalezieniu maksymalnej liczby cech, które można odrzucić przy założeniu, że ryzyko nie wzrośnie więcej niż o ξ , tzn. należy znaleźć takie q_0 , t_0 , że

$$\begin{cases} t_0 = \max t, \quad t=1, \dots, n-1 \\ R^* U_{t_0}^{q_0} - R^* \leq \xi \end{cases} \quad /1.4.12./$$

Rozważając selekcję cech, do tej pory milcząco zakładało się, że ryzyko po przeprowadzeniu selekcji cech wzrasta. Niżej udowodnione zostanie twierdzenie, że wzrost taki w rzeczywistości zachodzi.

Tw.1.4.1.

Ryzyko po odrzuceniu cech $i_1, i_2, \dots, i_s$, $s=1, 2, \dots, n-1$,

$i_s \in V^n$, jest większe równe ryzyku przed selekcją.

$$R^*(i_1, i_2, \dots, i_s) \gg R^* \quad /1.4.13/$$

Dowód:

Dowód tego twierdzenia przeprowadzony zostanie na podstawie twierdzenia Jensena.

Całkowe twierdzenie Jensena mówi, że jeśli

$p(x)$ - dodatnia funkcja określona w $[a, b]$,

$\varphi(x)$ - dowolna funkcja określona w $[a, b]$, a jej wartości leżą w pewnym przedziale D ,

$g(\cdot)$ - funkcja wypukła względem $(\cdot)$,

wówczas

$$g \left\{ \frac{\int_a^b p(x) \varphi(x) dx}{\int_a^b p(x) dx} \right\} \leq \frac{\int_a^b p(x) g(\varphi(x)) dx}{\int_a^b p(x) dx}.$$

/1.4.1.4/

Kładąc teraz w miejsce:

$p(x)$ - funkcję gęstości $f(x)$, która określona jest w przedziale $(-\infty, \infty)$, $x = (x_1, x_2, \dots, x_n)$,

$\varphi(x)$ - funkcję $p(i/x)$, określoną w przedziale $(-\infty, \infty)$

i przyjmującą wartości z przedziału $[0, 1]$, $x = (x_1, \dots, x_n)$,

oraz w miejsce funkcji

$g(\cdot)$ - funkcję $\max_j y_j$, $j = 1, \dots, L$, $y_j = p(j/x)$, wypukłą względem y_j (funkcja ta jest wypukła, ponieważ jest normą), otrzymuje się zgodnie z (1.4.14), że

$$\max_j \left\{ \frac{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(x) p(j/x) dx_{i_1} \dots dx_{i_s}}{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(x) dx_{i_1} \dots dx_{i_s}} \right\} \leq$$

/1.4.15./

$$\leq \frac{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(x) \max_j p(j/x) dx_{i_1} \dots dx_{i_s}}{\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} f(x) dx_{i_1} \dots dx_{i_s}}, \quad j = 1, \dots, L,$$

a więc

$$R^*(i_1, i_2, \dots, i_s) \gg R^* \quad \text{q.e.d.}$$

W dalszych rozdziałach tej pracy zbudowano optymalne algorytmy selekcji dla zadań sformułowanych w tym rozdziale, ze szczególnym uwzględnieniem pewnych przypadków, a mianowicie,

- 1/ $L=2$, czyli występują tylko dwie klasy, przy czym dodatkowo zakłada się, że elementy wektora cech są zmiennymi losowymi niezależnymi,
- 2/ przypadek $L \gg 2$, składowe obrazu $x = (x_1, \dots, x_n)$ są zmiennymi losowymi niezależnymi lub zależnymi,

Dla tych przypadków zbudowano algorytmy suboptymalne i optymalne. Algorytmy suboptymalne zbudowano w oparciu o wcześniej zdefiniowane oszacowania ryzyka. Algorytmy przedstawiono dla zadania 1 i dla zadania 2.

Zdefiniowane tutaj zadania selekcji w ogólnym przypadku polegają na znalezieniu podzbioru cech, na którym funkcja celu, w zadaniu selekcji ryzyko, przyjmuje wartość najmniejszą.

Problemami szukania ekstremów funkcji na skończonym zbiorze dyskretnym zajmuje się programowanie matematyczne.

2. ZASTOSOWANIE METODY PODZIAŁU I OGRANICZEŃ DO SELEKCJI CECH

2.1. Krótki przegląd metod programowania dyskretnego.

Przeglądu dokonano na podstawie [19]. Zagadnienie programowania matematycznego rozumiane jest jako poszukiwanie ekstremum funkcji celu.

$$g(z_1, z_2, \dots, z_n), \quad /2.1.1./$$

przy warunkach

$$h_1(z_1, \dots, z_n) \leq 0$$

$$h_2(z_1, \dots, z_n) \leq 0 \quad /2.2.2./$$

$$\dots \dots \dots$$

$$h_w(z_1, \dots, z_n) \leq 0$$

i dodatkowym warunkiem

$$z = (z_1, z_2, \dots, z_n) \in D \subset \mathcal{R}^n. \quad /2.1.3./$$

Obszar rozwiązań dopuszczalnych oznacza się literą G .

Programowanie dyskretne należy do klasy zadań nieregularnych programowania matematycznego. W zadaniach tych warunki dyskretności precyzuje się ze względu na każdą zmienną, a mianowicie

$$z_j \in D_j, \quad j=1, 2, \dots, n_1, \quad n_1 \leq n,$$

gdzie $\bigwedge_j D_j$ jest skończonym zbiorem zawierającym więcej niż jeden element, albo zbiorem przeliczalnym.

Jeśli $n_1 = n$, to zadanie nazywa się w pełni dyskretnym, jeśli zaś $n_1 < n$ - częściowo dyskretnym.

W programowaniu dyskretnym wyróżnić można następujące klasy zadań:

1. Zadania z niepodzielnosćmi.
2. Ekstremalne zadania kombinatoryczne.
3. Zadania z niejednorodną nieciągłą funkcją celu.
4. Zadania określone na obszarach nieklasycznych.
5. Pewne wieloekstremalne zadania.
6. Zadania dyskretnie w ścisłym tego słowa znaczeniu; poszukiwanie ekstremów na zbiorach skończonych.

Istnieje wiele (algorytmów) metod rozwiązywania tych zadań. Jedną z najczęściej wykorzystywanych jest metoda płaszczyzn odcinających, która wykorzystuje metody programowania liniowego. Pewnymi modyfikacjami tej metody są I, II, i III algorytm Gomory'ego, algorytm Dantzig'a i inne.

Algorytmy te poszukują ekstremum liniowej funkcji celu i z tego względu w problemach optymalnej selekcji, optymalnej ze względu na ryzyko lub inne nieliniowe oszacowania ryzyka, nie mogą być brane pod uwagę.

Szeroką klasę metod programowania dyskretnego stanowią metody kombinatoryczne. Zadania typu kombinatorycznego polegają na poszukiwaniu wartości ekstremalnej pewnej funkcji celu na zbiorze skończonym lub poszukiwaniu tych punktów skończonego zbioru, dla których funkcja celu osiąga ekstremum. Najprostszą metodą rozwiązywania zadań tego typu jest przegląd zupełny wszystkich rozwiązań i wybór zadania optymalnego.

Jednakże metoda taka wymaga wykonania bardzo dużej liczby obliczeń funkcji celu, a następnie ich porównania. Niekiedy

przy dużych licznościach odpowiednich zbiorów jest to wprost niewykonalne. Celowe więc jest szukanie i stosowanie innych metod, które umożliwiają zmniejszenie liczby obliczeń, a w rezultacie szybsze osiągnięcie szukanego rozwiązania.

Centralne miejsce wśród metod kombinatorycznych zajmuje metoda podziału i ograniczeń, nazywana też często metodą gałęzi i granic. Metoda ta jest w zasadzie ideą postępowania. Dla każdego indywidualnego zadania należy ją adoptować.

Inne metody rozwiązujące zadania kombinatoryczne, to:

1. Addytywny algorytm Balasa,
2. Algorytm Faure'a i Malgrange'a,
3. Algorytm kolejnych obliczeń Czernina,
4. Metoda programowania dynamicznego.

Algorytm addytywny Balasa opracowany został dla poszukiwania ekstremum addytywnej funkcji celu.

$$c_1 t_1 + c_2 t_2 + \dots + c_n t_n, \quad t_j \in \{0, 1\}, \quad j = 1, \dots, n \quad /2.1.5./$$

przy pewnych warunkach ograniczających.

Metoda Faure'a i Malgrange'a służy również do rozwiązywania zadań o liniowej funkcji celu.

Metodą kolejnych obliczeń Czernina sformułować można w następujący sposób.

Dany jest skończony zbiór U , np.: zbiór indeksów pewnych parametrów, które przyjmują wartości 0 lub 1.

Każdy podzbiór $U_1 \subset U$ interpretuje się jako zbiór tych parametrów, które przyjmują jedną z wartości, np. 1.

Na dowolnym podzbiorze $U_1 \subset U$ określa się funkcję $g(U)$.

Należy znaleźć taki zbiór $U_1^* \subset U$, na którym funkcja g osiąga ekstremum globalne, np. maximum.

Zakłada się, że funkcja g spełnia następujący warunek. Dla dowolnych $U_1, U_2 \subset U$ zachodzi

$$g(U_1) + g(U_2) \geq g(U_1 \cup U_2) + g(U_1 \cap U_2). \quad /2.1.6./$$

Warunek ten pozwala na eliminację wielu podzbiorów rozwiązań, Metoda programowania dynamicznego wymaga przeprowadzenia obliczeń do końca. Warto ją stosować w przypadkach o prostszej postaci funkcji celu, ponieważ w tej metodzie wykonuje się obliczenia dwukrotnie "od końca do początku i od początku do końca".

Metoda podziału i ograniczeń ma na celu znalezienie w dużym zbiorze rozwiązań charakteryzujących się określoną cechą, elementu o ekstremalnej wartości tej cechy.

Dokonuje się tego przez odpowiednie kolejne dzielenie zbioru wszystkich rozwiązań na coraz mniejsze podzbiory aż do otrzymania zbioru jednoelementowego będącego poszukiwanym rozwiązaniem zadania. Znalezienie najkrótszej drogi do uzyskania rozwiązania wymaga przyjęcia odpowiedniej strategii wyboru. W tym celu, zależnie od sformułowania funkcji celu, dokonuje się kolejnego podziału podzbiorów według minimalnych bądź też maksymalnych wartości ograniczających wartości cech występujących w poszczególnych podziorach.

Ideę tej metody można przedstawić w następujący sposób.

Zminimalizować funkcję

$$g(z),$$

/2.1.7./

przy warunkach

$$z = (z_1, z_2, \dots, z_n) \in G, \quad /2.1.8./$$

G - jest pewnym zbiorem skończonym.

U podstaw tej metody leżą następujące czynności, które w wielu przypadkach pozwalają zmniejszyć obszar przeglądu.

I. Obliczenie dolnego kresu (oszacowania) $\xi(G)$

$$g(z) \geq \xi(G). \quad /2.1.9./$$

II. Podział na podzbiory rozwiązań G .

III. Przeliczenie kresów (oszacowań).

W tej metodzie występują dwa podstawowe podejścia, dwie strategie, strategia A i strategia B [45].

Strategia A wymaga zapamiętania dużej liczby podzbiorów, ale czas trwania rozwiązania jest krótki.

Strategia B potrzebuje zapamiętania mniejszej ilości podzbiorów rozwiązań, natomiast czas obliczeń jest dłuższy niż dla strategii A. Wyboru odpowiednich strategii należy dokonywać dla każdego zadania indywidualnie, szczególnie wtedy, gdy algorytm ma być realizowany w formie programu na m.c. Wówczas bowiem z jednej strony komputer dysponuje ograniczoną pamięcią, a z drugiej strony z czasem obliczeń związane są duże koszty.

Formalny schemat strategii A metody podziału i ograniczeń jest następujący [19], [45].

Zerowy krok. Obliczenie kresu dolnego funkcji celu oszacowania

$$\xi(G) = \xi(G^0).$$

Jeśli uda się znaleźć takie rozwiązanie $\bar{z} \in G$, że

$$g(\bar{z}) = \xi(G) \quad /2.1.10/$$

to $\bar{z}$ jest rozwiązaniem optymalnym.

Jeśli nie, to w określony sposób rozbija się zbiór $G=G^0$ na skończoną liczbę podzbiorów

$$G^0 = G_1^1 \cup G_2^1 \cup \dots \cup G_{r_1}^1 \quad /2.1.11./$$

1-wszy krok. Obliczenie $\xi(G_i^1)$, $i = 1, 2, \dots, r_1$.

Jeśli uda się znaleźć takie rozwiązanie $\bar{z} \in G_r^1$, dla $r \in \{1, r_1\}$,

$$g(\bar{z}) = \xi(G_r^1) \leq \xi(G_i^1), \quad i = 1, \dots, r_1, \quad /2.1.12./$$

to $\bar{z}$ jest rozwiązaniem optymalnym. Jeśli nie, to należy wybrać "najbardziej perspektywiczny" dla dalszego rozbitcia zbiór $G_{\nu(1)}^1$, dla którego

$$\xi(G_{\nu(1)}^1) = \min_i \xi(G_i^1), \quad i = 1, \dots, r_1. \quad /2.1.13./$$

Zbiór $G_{\nu(1)}^1$ dzieli się na kilka podzbiorów

$$G_{\nu(1)}^1 = G_{\nu(1),1}^1 \cup G_{\nu(1),2}^1 \cup \dots \cup G_{\nu(1),s(1)}^1, \quad s(1) /2.1.14./$$

Wszystkie dotąd otrzymane podzbiory

$$G_1^1, G_2^1, \dots, G_{\nu(1)-1}^1, G_{\nu(1)+1}^1, \dots, G_{r_1}^1, G_{\nu(1),1}^1, \dots, G_{\nu(1),s(1)}^1 /2.1.15/$$

oznaczają się przez

$$G_1^2, G_2^2, \dots, G_{r_2}^2 \quad /2.1.16./$$

i przechodzi do następnego kroku.

K-ty krok. Obliczenie kresów (oszacowań) $\xi(G_i^k)$, $i = 1, \dots, r_k$

Jeśli przy tym uda się znaleźć takie rozwiązanie $\bar{z}$, że

$$\bar{z} \in G_r^k, \quad r \in \{1, r_k\} \quad i$$

$$g(\bar{z}) = \xi(G_r^k) \leq \xi(G_i^k), \quad i=1, \dots, r_k, \quad /2.1.17./$$

to $\bar{z}$ jest rozwiązaniem optymalnym.

Jeśli nie, to ponownie wybiera się rokujący największą nadzieję na uzyskanie rozwiązania optymalnego zbiór według zasady

$$\xi(G_{v(k)}^k) = \min_{i=1, \dots, r_k} \xi(G_i^k). \quad /2.1.18./$$

Zbiór $G_{v(k)}^k$ dzieli się na kilka podzbiorów.

$$G_{v(k)}^k = G_{v(k),1}^k \cup G_{v(k),2}^k \cup \dots \cup G_{v(k),s(k)}^k \quad /2.1.19./$$

Wszystkie otrzymane do tej pory zbiory

$$G_1^k, G_2^k, \dots, G_{v(k)-1}^k, G_{v(k)+1}^k, \dots, G_{r_k}^k, G_{v(k),1}^k, \dots,$$

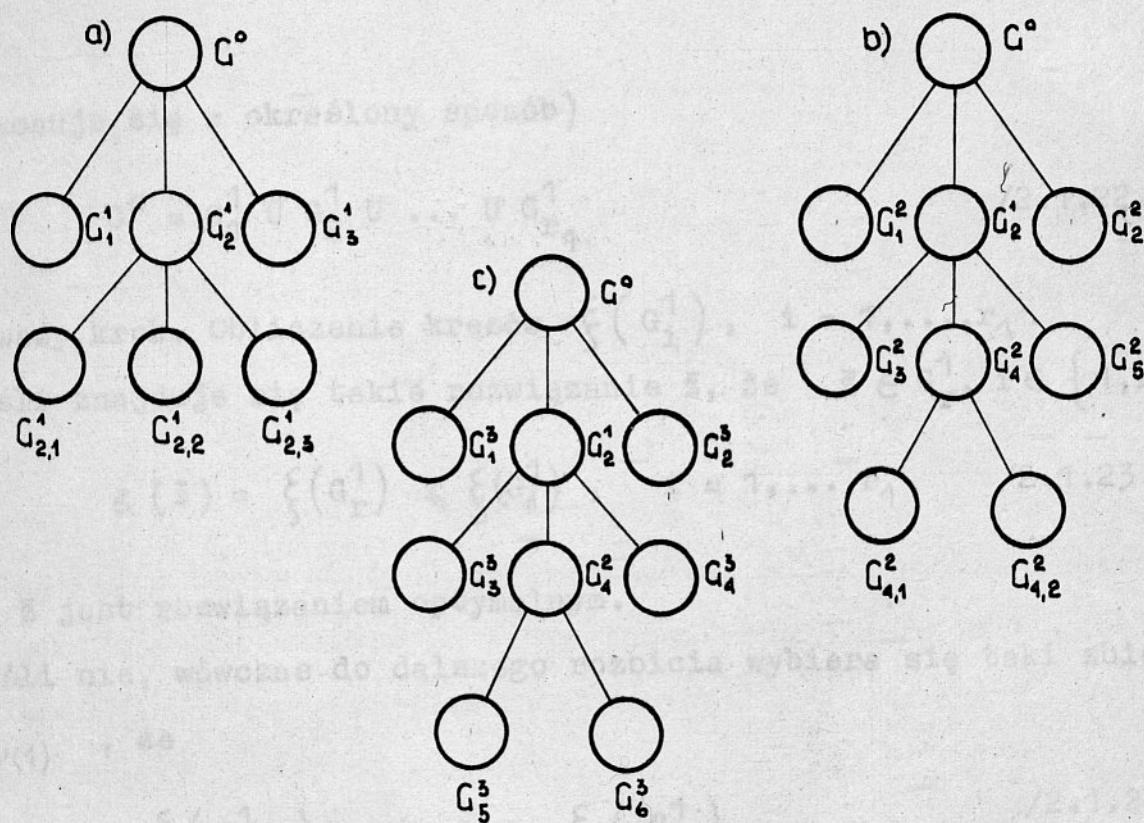
$$\dots, G_{v(k),s(k)}^k \quad /2.1.20./$$

oznacza się przez

$$G_1^{k+1}, G_2^{k+1}, \dots, G_{r_{k+1}}^{k+1} \quad /2.1.21./$$

i przechodzi do następnego kroku $(k+1)$ -wszego.

Postępowanie takie kontynuuje się aż do otrzymania zbioru jednoelementowego. Z przedstawionego wyżej schematu widać, że strategia ta wymaga zapamiętania dużej liczby rozwiązań w każdym kroku algorytmu. Kilka kroków tego algorytmu przedstawia rysunek 2.1.1.



Rys.2.1.1. Graficzne przedstawienie 3-kroków algorytmu

Ogólny schemat strategii B jest bardzo podobny. Algorytm według strategii B można podzielić na dwa etapy. W etapie pierwszym dokonuje się przeglądu wzdłuż jednej gałęzi i znajduje w niej rozwiązanie optymalne. W drugim etapie na podstawie rozwiązania uzyskanego w etapie pierwszym dokonuje się przeglądu pozostałych gałęzi.

Formalny schemat algorytmu podziału i ograniczeń według strategii B można przedstawić w następujący sposób

I etap.

o-ty krok. Obliczenie kresu $\xi(G) = \xi(G^0)$.

Jeśli nie znajduje się takiego rozwiązania $\bar{z}$, że $g(\bar{z}) = \xi(G)$, to rozbiła się zbiór G na skończoną liczbę podzbiorów (rozbiła

dokonyuje się w określony sposób)

$$G^0 = G_1^1 \cup G_2^1 \cup \dots \cup G_{r_1}^1 \quad /2.1.22./$$

1-wszy krok. Obliczenie kresów $\xi(G_i^1)$, $i = 1, \dots, r_1$.

Jeśli znajduje się takie rozwiązanie $\bar{z}$, że $\bar{z} \in G_r^1$, $r \in \{1, r_1\}$

$$g(\bar{z}) = \xi(G_r^1) \leq \xi(G_i^1), \quad i = 1, \dots, r_1, \quad /2.1.23./$$

to $\bar{z}$ jest rozwiązaniem optymalnym.

Jeśli nie, wówczas do dalszego rozbicia wybiera się taki zbiór

$G_{v(1)}^1$, że

$$\xi(G_{v(1)}^1) = \min_{i=1, \dots, r_1} \xi(G_i^1). \quad /2.1.24./$$

Dzieli się ten zbiór na skończoną liczbę podzbiorów

$$G_{v(1)}^1 = G_{v(1),1}^1 \cup G_{v(1),2}^1 \cup \dots \cup G_{v(1),r_2}^1 \quad /2.1.25./$$

i oznacza je przez

$$G_1^2, G_2^2, \dots, G_{r_2}^2, \quad /2.1.26./$$

a następnie przechodzi się do wykonania następnego kroku.

k-ty krok. Obliczenie kresów $\xi(G_i^k)$, $i=1, \dots, r_k$.

Jeśli znajdzie się takie rozwiązanie $\bar{z}$, że $\bar{z} \in G_r^k$, $r \in \{1, r_k\}$.

$g(\bar{z}) = \xi(G_r^k)$, to $\bar{z}$ jest pierwszym zadaniem optymalnym.

Jeśli nie, to do dalszego rozbicia wybiera się taki zbiór,

dla którego

$$\xi(G_{v(k)}^k) = \min_{i=1, \dots, r_k} \xi(G_i^k). \quad /2.1.27./$$

Rozbija się zbiór $G_{V(k)}^k$ na podzbiory

$$G_{V(k)}^k = G_{V(k),1}^k \cup G_{V(k),2}^k \cup \dots \cup G_{V(k),r_k}^k \quad /2.1.28./$$

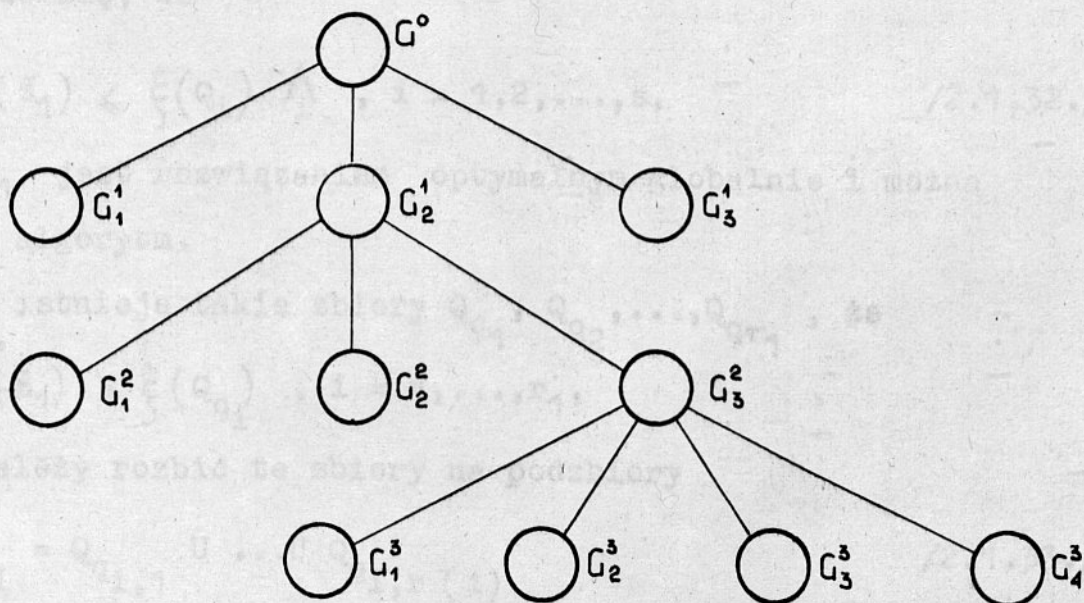
i oznacza je przez

$$G_1^{k+1}, G_2^{k+1}, \dots, G_{r_k}^{k+1}, \quad /2.1.29./$$

a następnie przechodzi do następnego kroku.

Postępowanie takie kontynuuje się dopóty, dopóki w którymś kroku nie osiągnie się pierwszego optymalnego rozwiązania bądź nie dojdzie do rozwiązania jednoelementowego.

Etap pierwszy ilustruje rysunek 2.1.2.



Rys. 2.1.2. Graficzne przedstawienie I etapu algorytmu wg strategii B.

Etap II.

0-wy krok. Zbiory nie rozbijane w etapie pierwszym, a więc zbiory

$$G_1^1, G_2^1, \dots, G_{v(1)-1}^1, \dots, G_{r_1}^1, G_1^2, \dots, G_{v(2)-1}^2, G_{v(2)+1}^2, \dots, G_{r_2}^2, \dots$$

$$\dots, G_1^k, \dots, G_{v(k)-1}^k, G_{v(k)+1}^k, \dots, G_{r_k}^k, \dots, G_{r_w-1}^{w-1}, \quad /2.1.30./$$

gdzie v oznacza liczbę kroków wykonaną w etapie pierwszym,
 r_k oznacza się przez

$$Q_1, Q_2, \dots, Q_s. \quad /2.1.31/$$

Rozwiązanie optymalne w etapie pierwszym oznacza się przez $\bar{z}_1$
i oblicza dla tego rozwiązania funkcję celu $g(\bar{z}_1)$.
1-wszy krok. Porównuje się $g(\bar{z}_1)$ z wartościami kresów $\xi(Q_i)$.
Jeśli okaże się, że

$$g(\bar{z}_1) \leq \xi(Q_i) \bigwedge_i, \quad i = 1, 2, \dots, s, \quad /2.1.32./$$

wówczas $\bar{z}_1$ jest rozwiązaniem optymalnym globalnie i można
zakończyć algorytm.

Jeśli zaś istnieją takie zbiory $Q_{q_1}, Q_{q_2}, \dots, Q_{q_{r_1}}$, że

$$g(\bar{z}_1) > \xi(Q_{q_i}), \quad i = 1, \dots, r_1,$$

wówczas należy rozbić te zbiory na podzbiory

$$Q_{q_i} = Q_{q_i,1} \cup \dots \cup Q_{q_i,r(i)} \quad /2.1.33./$$

i oznaczyć przez

$$Q_1^1, Q_2^1, \dots, Q_{s_1}^1 \quad /2.1.34./$$

2-gi krok. Obliczenie kresów $\xi(Q_i^1)$, $i = 1, \dots, s_1$ i porównanie
ich z wartością funkcji celu na zbiorze $\bar{z}_1$, $g(\bar{z}_1)$.

Jeśli $g(\bar{z}_1) \leq \xi(Q_i^1) \bigwedge_i$, wówczas $\bar{z}_1$ w dalszym ciągu jest
rozwiązaniem optymalnym globalnie i można zakończyć algorytm.

Jeśli zaś istnieją takie zbiory $Q_{q_1}^1, Q_{q_2}^1, \dots, Q_{q_{r_2}}^1$, że

$$g(\bar{z}_1) > \xi(Q_{q_i}^1), \quad i = 1, \dots, r_2, \quad /2.1.35./$$

wówczas w dalszym ciągu należy rozbić te zbiory na podzbiory pamiętając, iż niektóre z nich mogą być jednoelementowe.

Jeśli rzeczywiście któryś ze zbiorów $Q_{q_i}^1$, $i = 1, \dots, r_2$ jest zbiorem jednoelementowym, wówczas jasne, że zbioru tego nie można w dalszym ciągu rozбивać. Dla takich zbiorów należy obliczyć funkcję celu i porównać z $g(\bar{z}_1)$. Jeśli $g(\bar{z}_1)$ jest mniejsze do odpowiednich wartości funkcji celu na tych zbiorach wówczas $\bar{z}_1$ jest w dalszym ciągu rozwiązaniem optymalnym, które służyć będzie do porównań z wartościami kresów funkcji celu dla dalszych podzbiorów. Jeśli zaś funkcja celu na zbiorze jednoelementowym $Q_{q_r}^1$ jest mniejsza niż $g(\bar{z}_1)$, wówczas zbiór $Q_{q_r}^1$ uznaje się za optymalny i oznacza go przez $\bar{z}_1$.

Rozbicia na dalsze podzbiory dokonuje się w następujący sposób:

$$Q_{q_i}^1 = Q_{q_i,1}^1 \cup \dots \cup Q_{q_i,r(i)}^1, \quad i = 1, \dots, r_2 \quad /2.1.36./$$

i wykonuje następny krok, oznaczając przedtem te podzbiory przez

$$Q_1^2, Q_2^2, \dots, Q_{s_2}^2 \quad /2.1.37./$$

K-ty krok. Obliczenie kresów $\xi(Q_i^{k-1})$, $i = 1, \dots, s_{k-1}$.

Porównanie tych kresów z funkcją celu zbioru jednoelementowego $\bar{z}_1$

Jeśli

$$g(\bar{z}_1) \leq \xi(Q_i^{k-1}), \quad \bigwedge_i, \quad i = 1, \dots, r_{k-1} \quad /2.1.38./$$

wówczas $\bar{z}_1$ jest w dalszym ciągu rozwiązaniem optymalnym i algorytm można zakończyć.

Jeśli istnieją takie zbiory $Q_{q_1}^{k-1}, Q_{q_2}^{k-1}, \dots, Q_{q_{r_k}}^{k-1}$,
że

$$g(\bar{z}_1) \gg \xi(Q_{q_i}^{k-1}), i = 1, \dots, r_k, \quad /2.1.39./$$

wówczas należy rozbić te zbiory na podzbiory.

Jeśli okaże się, że istnieją zbiory jednoelementowe

$Q_{q_{w_1}}^{k-1}, \dots, Q_{q_{w_t}}^{k-1}$, wówczas dla tych zbiorów należy obliczyć

funkcje celu i porównać je z $g(\bar{z}_1)$. Jeśli okaże się, że

$$g(\bar{z}_1) \leq \min_{w_i} g(Q_{q_{w_i}}^{k-1}), i = 1, \dots, t, w_i \in \{1, \dots, r_k\}, \quad /2.1.40./$$

wówczas $\bar{z}_1$ jest w dalszym ciągu rozwiązaniem optymalnym.

W przeciwnym wypadku za $\bar{z}_1$, czyli rozwiązanie optymalne służące do dalszych porównań, przyjmuje się rozwiązanie, na którym funkcja celu osiąga wartość najmniejszą.

Pozostałe zbiory $Q_{q_i}^{k-1}, i = 1, \dots, r_k, i \neq w_1, \dots, w_t$

rozbija się na podzbiory.

$$Q_{q_i}^{k-1} = Q_{q_{i,1}}^{k-1} \cup \dots \cup Q_{q_{i,r(k)}}^{k-1}, i = 1, \dots, r_k, i \neq w_1, \dots, w_t$$

/2.1.41/

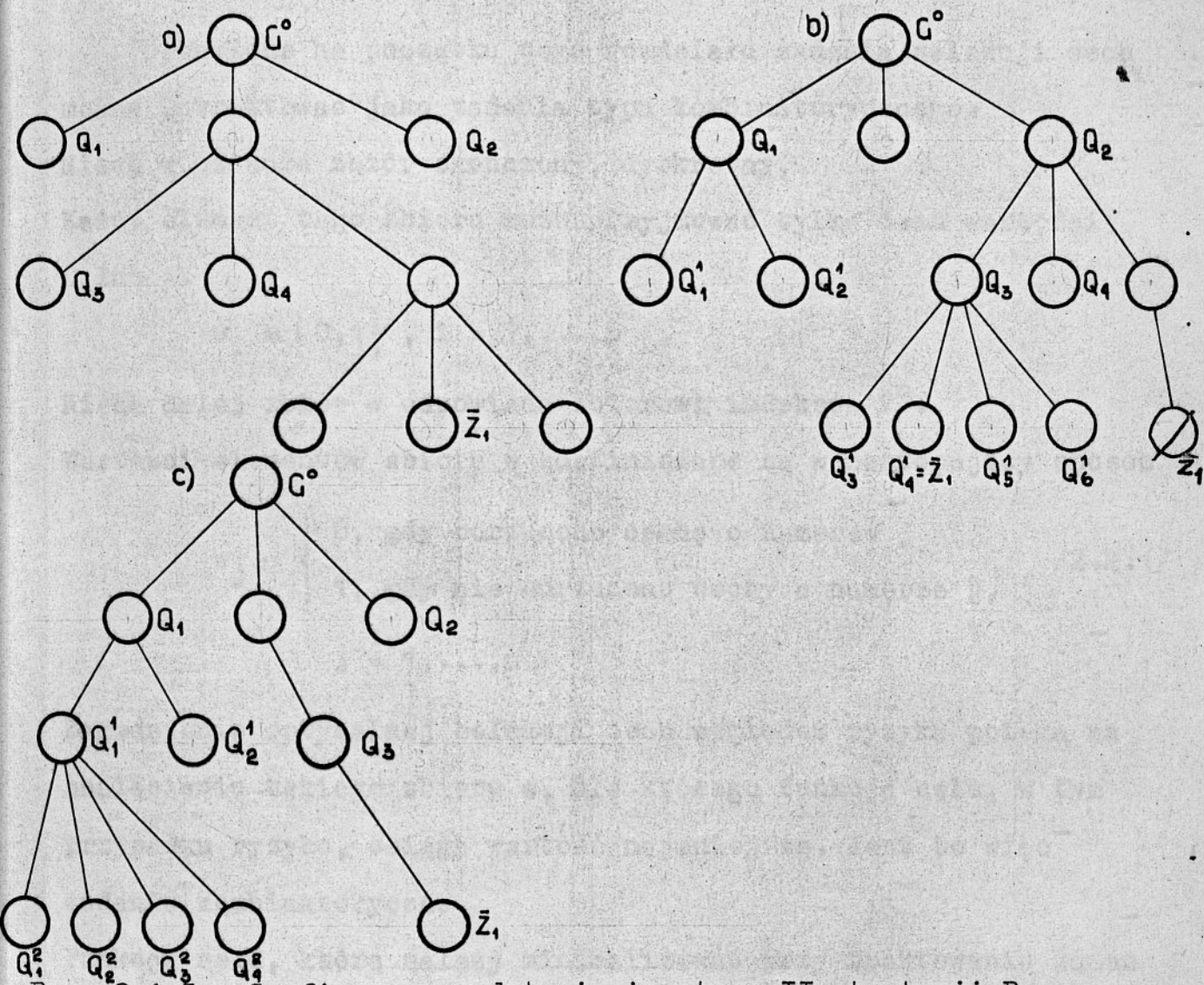
i oznacza je przez

$$Q_1^k, Q_2^k, \dots, Q_{s_k}^k$$

/2.1.42./

Postępowanie takie kontynuuje się aż do otrzymania jednoelementowego rozwiązania globalnego.

Rys.2.1.3. ilustruje etap II, biorąc za podstawę rysunek 2.1.2.



Rys. 2.1.3.: Graficzne przedstawienie etapu II strategii B.

Podstawową zaletą metody podziału i ograniczeń jest to,

że w każdym kroku algorytmu otrzymuje się pewne rozwiązanie.

W wypadku, jeśli czas obliczeń na maszynie cyfrowej przekracza założony wcześniej czas realizacji zadania, można przerwać te obliczenia uzyskując rozwiązanie suboptymalne.

2.2. Sprowadzenie zadań selekcji do zadań typu kombinatorycznego.

Postawione na początku tego rozdziału zadania selekcji cech można potraktować jako zadania typu kombinatorycznego.

Niech w oznacza zbiór skończony, dyskretny,

Każdy element tego zbioru może przyjmować tylko dwie wartości:

0 lub 1.

$$w_i = \{0, 1\}, \quad i = 1, \dots, n.$$

Niech dalej zbiór w odpowiada zbiorowi indeksów V^n .

Wartości elementów zbioru w zdefiniowane są w następujący sposób

$$w_j = \begin{cases} 0, & \text{gdy odrzucono cechę o numerze } j \\ 1, & \text{gdy nie odrzucono cechy o numerze } j, \end{cases} \quad /2.2.1/$$
$$j = 1, \dots, n.$$

Zagadnienie optymalnej selekcji cech względem ryzyka polega na znalezieniu takiego zbioru w , dla którego funkcja celu, w tym przypadku ryzyko, osiąga wartość najmniejszą. Jest to więc zadanie kombinatoryczne.

Funkcja celu, którą należy minimalizować przy traktowaniu zadań selekcji jako zadań kombinatorycznych, nie da się przedstawić w sposób ogólny. Należy ją zdefiniować dla każdego zbioru w oddzielnie.

Rozważając zadanie 1., a więc zadanie odrzucania spośród n -cech m cech takich, aby wzrost ryzyka był minimalny, można funkcję celu formułować w następujący sposób:

Zakłada się, że tylko m elementów zbioru $w = \{w_1, w_2, \dots, w_n\}$ może jednocześnie przyjąć wartości zerowe.

Jeśli $w_{k_1}, w_{k_2}, \dots, w_{k_m} = 0$, $k_1, \dots, k_m = 1, \dots, n$, $k_1 \neq k_2 \neq \dots \neq k_m$, pozostałe elementy tego zbioru są równe 1, wówczas funkcja celu dla takiego zbioru ma postać

$$g(w) = R^*(w) \triangleq 1 - \int_{x^{k_1, \dots, k_m}} \max_i \left[\int \dots \int f(x/i) p_i dx_{k_1} \dots dx_{k_m} \right] dx^{k_1 \dots k_m} = R^*(k_1, \dots, k_m) \quad i=1, \dots, L \quad /2.2.2./$$

Ograniczenie w tym wypadku ma postać

$$\sum_{j=1}^n w_j = n-m, \quad /2.2.3./$$

Zadanie polega na znalezieniu takiego zbioru w spełniającego ograniczenie (2.2.3), aby funkcja celu (2.2.2.) przyjęła wartość najmniejszą.

Zadanie 2 sformułować można w następujący sposób. Zakłada się, że t elementów zbioru w może przyjmować wartości zerowe, $t = 1, \dots, n-1$

Funkcja celu, którą należy minimalizować, ma postać

$$g(w) = \sum_{i=1}^n w_i, \quad /2.2.4./$$

przy ograniczeniu

$$R^*(w) - R^* \leq \varepsilon, \quad /2.2.5./$$

gdzie $R^*(w)$, jeśli $w_{k_1}, w_{k_2}, \dots, w_{k_t} = 0$, $k_i = 1, \dots, n$,

$t < n$, $i = 1, \dots, t$, pozostałe elementy zbioru w równe jedności, wyraża się następująco

$$R^*(w) = R^*(k_1, k_2, \dots, k_t) \quad /2.2.6./$$

Ze sformułowania w powyższy sposób zadań optymalnej selekcji wynika, że tylko niektóre z wymienionych wcześniej metod programowania dyskretnego można w tych zadaniach wykorzystać. W obu zadaniach występują nieliniowości. W zadaniu 1 nieliniowa jest funkcja celu, natomiast w zadaniu 2 nieliniowe są ograniczenia. Można odrzucić wszystkie metody płaszczyzn odcinających, ponieważ pokazano, że zadania te należy traktować jako zadania typu kombinatorycznego i szukać odpowiedniej metody wśród metod rozwiązujących tego typu zadania.

Algorytm kolejnych obliczeń Czernina wymaga pewnych dodatkowych założeń o funkcji celu, a założeń tych ryzyko nie spełnia. Niech np.: $U = \{1, 2, 3, 4, 5\}$, U_1 - zbiór tych indeksów, dla których odpowiednie elementy zbioru są równe 0, np.: $U_1 = \{1, 2, 3\}$, $U_2 = \{2, 3, 4\}$

Wówczas

$$R^*_{U_1} + R^*_{U_2} \quad \mathcal{R} \quad R^*(U_1 \cup U_2) + R^*(U_1 \cap U_2)$$

czyli

$$R^*_{\{1, 2, 3\}} + R^*_{\{2, 3, 4\}} \quad \mathcal{R} \quad R^*_{\{1, 2, 3, 4\}} + R^*_{\{2, 3\}}$$

ale niczego nie można powiedzieć na temat relacji $\mathcal{R}$.

Ze względu na skomplikowaną postać funkcji celu bądź ograniczeń można również pominąć metodę programowania dynamicznego, która wymaga dwukrotnego "od początku do końca" i "od końca do początku", przeprowadzenia obliczeń, aby otrzymać jedno rozwiązanie. Jest w związku z tym bardzo praco- i czasochłonna.

Jedną z najbardziej odpowiednich metod rozwiązania zadań selekcji wydaje się być metoda podziału i ograniczeń i tę metodę w dalszym ciągu zaadoptowano do tych zadań. Ponieważ algorytmy selekcji przedstawione w niniejszej pracy są budowane pod kątem realizacji ich w postaci programu na maszynę cyfrową, istotnymi ograniczeniami są pamięć m.c. i czas obliczeń.

Ponieważ, jak już wspomniano wcześniej, metoda podziału i ograniczeń w każdym kroku daje pewne rozwiązanie suboptymalne, a więc przy zbyt długim czasie obliczeń można je przerwać, proponuje się zastosować metodę podziału i ograniczeń według strategii B.

Strategia A, jak już wspomniano wcześniej, wymaga zapamiętania bardzo dużej liczby odpowiednich zbiorów, w związku z czym w przypadku realizacji algorytmów selekcji w postaci programów na maszynę cyfrową, algorytmów zbudowanych według strategii A, może się zdarzyć, że pewne problemy praktyczne nie mieszczą się w pamięci maszyny cyfrowej. Strategię A można stosować tylko w przypadku małych n . Z tego powodu wybór strategii B wydaje się być słuszniejszy.

2.3. Algorytmy selekcji cech dla zadania 1 oraz dla zadania 2.

Przed przystąpieniem do przedstawienia odpowiednich algorytmów udowodnione zostaną następujące lematy.

Lemat 2.3.1.

Ryzyko po odrzuceniu L_1 cech, $x_{k_1}, \dots, x_{k_{L_1}}$ jest mniejsze lub równe ryzyku po odrzuceniu $L_1 + L_2$ cech, $x_{k_1}, \dots, x_{k_{L_1}}$,

$$x_{k_{L_1+1}}, \dots, x_{k_{L_1+L_2}},$$

$$R^*(k_1, \dots, k_{L_1+L_2}) \gg R^*(k_1, \dots, k_{L_1}) \quad /2.3.1./$$

Dowód:

Dowód tego lematu wynika wprost z twierdzenia 1.4.1, jeśli tylko we wzorze (1.4.13.) w miejsce R^* wstawi się $R^{*(k_1, \dots, k_{L_1})}$ a w miejsce $R^*(i_1, \dots, i_s)$ wstawi się $R^*(k_1, \dots, k_{L_1+L_2})$.

Lemat 2.3.2.

Jeśli $\{k_1, \dots, k_r\}$ jest kombinacją r -elementową indeksów, a $\{\tilde{k}_1, \tilde{k}_2, \dots, \tilde{k}_r\}$ jest dowolną permutacją tych indeksów, wówczas

$$R^*(k_1, \dots, k_r) = R^*(\tilde{k}_1, \tilde{k}_2, \dots, \tilde{k}_r) \quad /2.3.2./$$

Dowód.

Dowód tego lematu wynika wprost z definicji ryzyka, jeśli tylko funkcje gęstości są odcinkami ciągłe, wówczas bowiem

$$\max_i \int_{x_{k_1}} \dots \int_{x_{k_r}} f(x/i) p_i dx_{k_1} \dots dx_{k_r} = \max_i \int_{x_{\tilde{k}_1}} \dots \int_{x_{\tilde{k}_r}} f(x/i) p_i dx_{\tilde{k}_1} \dots dx_{\tilde{k}_r} \quad i = 1, \dots, L$$

/2.3.3./

Sformułowane wyżej lematy pozwolą na pewne uproszczenie algorytmów selekcji, które budowane będą na podstawie metody podziału i ograniczeń.

W pierwszej kolejności rozważać się będzie zadanie 1, a więc zadanie optymalnego odrzucenia m -cech spośród n . Wszystkich

możliwych rozwiązań w tym przypadku jest tyle, ile jest kombinacji m -elementowych zbioru n -elementowego, czyli

$$\binom{n}{m} = \frac{n!}{m!(n-m)!} \quad /2.3.4./$$

Zgodnie z lematem 2.3.1. pomija się odpowiednie permutacje. Niech zbiór G stanowi zbiór wszystkich rozwiązań dopuszczalnych. Bezpośrednia metoda rozwiązania tego problemu polegałaby na przeglądzie całego zbioru G i wyborze tego elementu, dla którego ryzyko osiągałoby wartość najmniejszą. Dla dużych n , takie podejście jest bardzo czasochłonne a nawet niekiedy niewykonalne. Przez zastosowanie metody podziału i ograniczeń rozmiar tego przeglądu można znacznie zmniejszyć.

Zakłada się, że celem odpowiednich algorytmów jest wyznaczenie cech, które należy odrzucić. Jako postawę adaptacji metody podziału i ograniczeń przyjęte zostaną następujące punkty:
I Kres dolny (oszacowanie) funkcji celu na zbiorze G stanowi R^* .

$$\xi(G) = R^* \quad /2.3.5./$$

II. Podział zbioru G na podzbiory dokonywać się będzie według następującej zasady

$$G \equiv G^0 = G_1 \cup G_2 \cup \dots \cup G_n, \quad /2.3.6./$$

gdzie

G_i - zbiór tych rozwiązań, w których na pewno odrzucono cechę i -tą, $i = 1, \dots, n$

III. Kresem dolnym funkcji celu (oszacowaniem) dla każdego

zbioru G_i , będzie $R^*(i)$ - ryzyko po odrzuceniu i -tej cechy.

Algorytm selekcji dla zadania 1.

Algorytm ten opiera się na metodzie podziału i ograniczeń przeprowadzonej według strategii B.

Etap I.

0-wy krok. Obliczenie kresu $\xi(G^0)$, $G^0 \equiv G$, a następnie rozbięcie zbioru G^0 na podzbiory, zgodnie ze wzorem (2.3.6)

$$G^0 = G_1^1 \cup G_2^1 \cup \dots \cup G_n^1$$

1-krok. Obliczenie kresów $\xi(G_i^1)$

$$\xi(G_i^1) = R^*(i), \quad i = 1, \dots, n, \quad /2.3.7./$$

a następnie wybór zbioru do dalszego rozbięcia, zbiór ten jest taki, że

$$\xi(G_{k_1}^1) = \min_i \xi(G_i^1), \quad i = 1, \dots, n,$$

inaczej

$$R^*(k_1) = \min_i R^*(i).$$

Rozbicie zbioru $G_{k_1}^1$ przeprowadza się w następujący sposób:

$$G_{k_1}^1 = G_{k_1,1}^1 \cup \dots \cup G_{k_1, k_1-1}^1 \cup G_{k_1, k_1+1}^1 \cup \dots \cup G_{k_1, n}^1 \quad /2.3.9./$$

gdzie

$G_{k_1, i}^1$ oznacza zbiór tych rozwiązań, w których na pewno

odrzucono cechy k_1, i , $i = 1, \dots, n$, $i \neq k_1$.

W dalszym ciągu zbiory te oznaczają się przez

$$G_{s_1}^2, G_{s_2}^2, \dots, G_{s_{n-1}}^2 \quad /2.3.10/$$

k-ty krok. Obliczenie kresów $\xi(G_{s_i}^k)$, $i = 1, \dots, n-k$,

gdzie jak poprzednio

$G_{s_i}^k$ - oznacza zbiór rozwiązań, w których na pewno odrzucono cechy o indeksach $k_1, k_2, \dots, k_{k-1}, s_i, s_i \in V^n$,

$$s_i \neq k_1, \dots, k_{k-1}, \quad i = 1, \dots, n-k$$

$$\xi(G_{s_i}^k) = R^*(k_1, \dots, k_{k-1}, s_i). \quad /2.3.11/$$

Do dalszego rozbicia wybiera się ten zbiór $G_{k_k}^k$, dla którego

$$\xi(G_{k_k}^k) = \min_i \xi(G_{s_i}^k), \quad i = 1, \dots, n-k. \quad /2.3.12./$$

Rozbicie tego zbioru przeprowadza się w następujący sposób:

$$G_{k_k}^k = G_{k_k, s_1}^k \cup G_{k_k, s_2}^k \cup \dots \cup G_{k_k, s_{n-k-1}}^k, \quad /2.3.13./$$

gdzie

G_{k_k, s_i}^k - oznacza zbiór tych rozwiązań, w których na pewno odrzucono $(k+1)$ cech o indeksach $k_1, \dots, k_k, s_i$,

a następnie zbiory te oznacza się przez

$$G_{s_1}^{k+1}, G_{s_2}^{k+1}, \dots, G_{s_{n-k-1}}^{k+1}. \quad /2.3.14./$$

m-ty krok. Dla każdego zbioru $G_{s_i}^m$, $i = 1, \dots, n-m$ oblicza się

oszacowanie $\xi(G_{s_i}^m)$, zgodnie ze wzorem

$$\xi(G_{s_i}^m) = R^*(k_1, k_2, \dots, k_{m-1}, s_i). \quad /2.3.15./$$

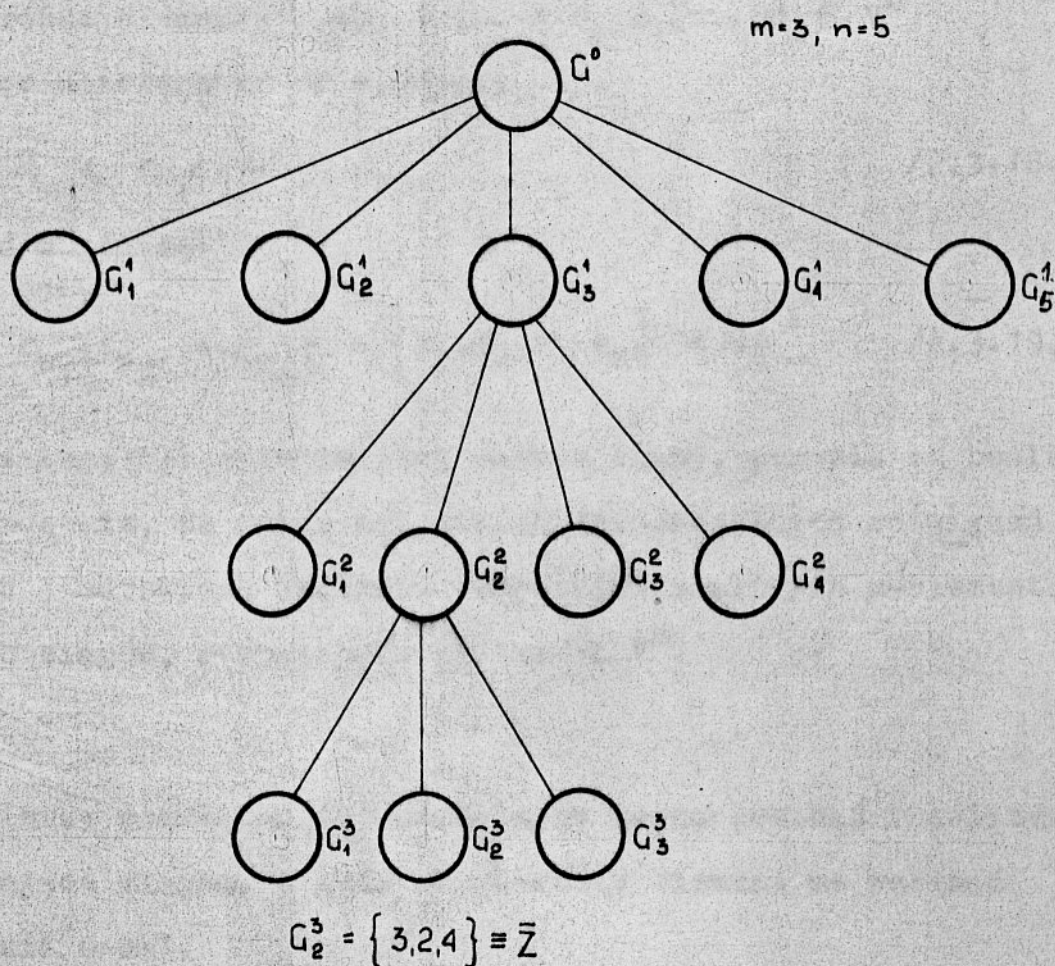
Następnie wyznacza się taki zbiór $G_{k_m}^m$, dla którego

$$\xi(G_{k_m}^m) = \min_i \xi(G_{s_i}^m), \quad i = 1, \dots, n-m, \quad s_i \in V^n, \quad s_i \neq k_1, \dots, k_{m-1}. \quad /2.3.16/$$

Zbiór ten jest zbiorem jednoelementowym. Rozwiązanie otrzymane w ten sposób, a więc rozwiązanie, które otrzymuje się poprzez odrzucenie cech o indeksach $k_1, k_2, \dots, k_m$ otrzymanych w wyniku I etapu algorytmu, jest pierwszym rozwiązaniem optymalnym, oznaczy się go przez $\bar{z}$.

Na tym kończy się I etap tego algorytmu.

Graficzny obraz I etapu przedstawia rysunek 2.3.1.



Rysunek 2.3.1.: Graficzne przedstawienie etapu I.

Jak już poprzednio wspomniano, zbiór wszystkich rozwiązań stanowi zbiór wszystkich kombinacji m -elementowych zbioru n -elementowego, V^n . Kombinację m -elementową U_m^q zbioru nazywa się każdy podzbiór m -elementowy o różnych elementach. Ponieważ zbiór zgodnie z (1.4.7) zawiera różne elementy, więc również $\bigwedge_q U_m^q$ zdefiniowane przez (1.4.8) zawiera różne elementy.

Każdą kombinację U_m^q da się przedstawić w postaci rosnącego ciągu.

Dowód:

Z poprzednich rozważań wynika, że każda kombinacja $U_m^q = \{r_1, r_2, \dots, r_m\}$ zawiera różne elementy $r_1 \neq r_2 \neq \dots \neq r_m, r_1, \dots, r_m \in V^n$.

Można więc uszeregować te elementy

$$r_{w_1} < r_{w_2} < \dots < r_{w_m} \quad /2.3.18./$$

i zapisać w postaci

$$\{r_{w_1}, r_{w_2}, \dots, r_{w_m}\} = \{r_1, r_2, \dots, r_m\} = U_m^q \quad /2.3.19./$$

Wniosek z tego stwierdzenia jest bardzo ważny, pozwala on bowiem na stwierdzenie, że zbiór wszystkich dopuszczalnych rozwiązań w zadaniu 1 odpowiada zbiorowi wszystkich możliwych m -elementowych rosnących ciągów, utworzonych ze zbioru V^n .

Tw. 2.3.2.

W zbiorze rozwiązań dopuszczalnych można pominąć rozwiązania odpowiadające ciągom, w których pierwszy element ma wartość większą niż $n-m+1$.

Dowód:

Zgodnie z poprzednim twierdzeniem zbiór wszystkich rozwiązań dla zadania 1 jest równoważny zbiorowi odpowiednich ciągów rosnących (2.3.19.) .

Dalszy ciąg dowodu przeprowadzony w sposób RaA.

Zakłada się, że istnieje takie rozwiązanie $\{r_1, r_2, \dots, r_m\}$, któremu odpowiada ciąg rosnący $\{r_{w_1}, r_{w_2}, \dots, r_{w_m}\}$ taki, że

$$r_{w_1} = (n-m+1) + k, \quad 1 \leq k \leq n-m, \quad /2.3.20/$$

wówczas nawet w najlepszym wypadku, tzn. gdy kolejne elementy ciągu różniąc się tylko o 1, zachodzi

$$(n-m+1+k) + (m-1) \cdot 1 < n,$$

stąd otrzymuje się, że

$$n+k < m,$$

co jest sprzeczne, ponieważ zgodnie z (2.3.20) $k \geq 1$.

q.e.d.

Drugi etap algorytmu polega na porównaniu kresu rozwiązania $\bar{z}$, otrzymanego w pierwszym etapie z kresami wszystkich zbiorów oznaczonych w tym etapie, przez $G_1^1, G_2^1, \dots, G_n^1$ (2.1.31) /

Biorąc pod uwagę sposób tworzenia zbiorów oraz lemat 2.3.1.

należy sądzić, że w drugim etapie algorytmu na pewno rozbijając należałoby zbiory $G_{k_1}^1, G_{k_2}^1, \dots, G_{k_m}^1$. Jednakże z twierdzenia

2.3.2. wynika, że można pominąć pewne zbiory rozwiązań, a

mianowicie te, w których odpowiednie ciągi indeksów (2.3.19.) pierwsze elementy mają większe od $n-m+1$. Stwierdzenie takie pozwala z góry wyeliminować $n-m+1$ zbiorów już przy pierwszym

rozbić.

Można więc wyeliminować właśnie te zbiory, o których wiemy, że oszacowania funkcji celu na tych zbiorach na pewno będą mniejsze lub równe oszacowaniom osiąganym na zbiorze $\bar{z} = \{k_1, k_2, \dots, k_m\}$.

Przed przystąpieniem do etapu II należy więc "przenumerować" składowe wektora cech $x = (x_1, x_2, \dots, x_n)$.

Niech teraz obraz stanowi wektor

$$\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n), \quad /2.3.21./$$

gdzie

$$\tilde{x}_{n-i} = x_{k_{i+1}}, \quad i = 0, 1, \dots, n-m.$$

Pozostałe składowe $\tilde{x}_1, \dots, \tilde{x}_{n-m}$ odpowiadają składowym x_j , $j = 1, \dots, n, j \neq k_1, \dots, k_m$ w sposób dowolny.

Etap II.

0-krok. Zbiór wszystkich rozwiązań dopuszczalnych oznacza się teraz przez $\tilde{G} \equiv \tilde{G}^0 = G^0$, a następnie rozбивa się ten zbiór na podzbiory w następujący sposób

$$\tilde{G}^0 = \tilde{G}_1^1 \cup \tilde{G}_2^1 \cup \dots \cup \tilde{G}_n^1, \quad /2.3.22./$$

gdzie

- $\tilde{G}_1^1$ - zbiór rozwiązań, w których na pewno odrzucono cechę $\tilde{x}_1$,
- $\tilde{G}_i^1$ - zbiór tych rozwiązań, w których na pewno odrzucono składową $\tilde{x}_i$, $i=2, \dots, n$, ale na pewno nie odrzucono cechy $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_{i-1}$.

Zauważyć należy, że przy takim rozbiściu zbioru $\tilde{G}^0$ zbiór $\tilde{G}_{n-m+1}^1$ jest zbiorem jednoelementowym zawierającym rozwiązanie $\bar{z}$. Natomiast zbiory $\tilde{G}_s$, $s = n-m+2, \dots, n$ są zbiorami pustymi (tw.2.3.2.).

1 krok. Obliczenie kresów funkcji celu dla zbiorów G_i^1 , $i = 1, \dots, n-m$.

Jasne, że w przypadku, gdy zapamiętano wartości kresów obliczone w poprzednim etapie, wystarczy je tylko odpowiednio podstawić

$$\xi(G_i^1) = R^*(s_i) = \tilde{R}^*(i), \text{ jeśli } \tilde{x}_i = x_{s_i}. \quad /2.3.23./$$

Następnie porównuje się $\xi(\bar{z})$ oraz $\xi(\tilde{G}_i)$

$$\xi(\bar{z}) = R^*(k_1, \dots, k_m) \quad R \quad \xi(\tilde{G}_i^1) = \tilde{R}^*(i), \quad i = 1, \dots, n-m \quad /2.3.24./$$

1. Jeśli R jest " $\leq$ " $\bigwedge_i$, wówczas $\bar{z}$ jest rozwiązaniem optymalnym globalnie, można zakończyć algorytm.
2. Jeśli $V_{J_1} = \{q_1^1, \dots, q_{s_1}^1\}$, że R jest " $>$ " wówczas zbiór indeksów

pozostałych zbiorów oznacza się przez $\bar{J}_1 = \{w_1, \dots, w_{n-m-s}\}$, a następnie rozbijają się zbiory $\tilde{G}_{r_i}^1$, $r_i \in J_1$, zgodnie z zasadą

$$\tilde{G}_{r_i}^1 = \tilde{G}_{q_i^1, q_{i+1}^1}^1 \cup \tilde{G}_{q_i^1, q_{i+2}^1}^1 \cup \dots \cup \tilde{G}_{q_i^1, q_n^1}^1, \quad /2.3.25./$$

inaczej

$$G_{r_1}^1 = \bigcup_j G_{q_{i,j}}^1, \quad j = q_{i+1}^1, \dots, n, \quad i = 1, \dots, s_1.$$

Biorąc pod uwagę Lemat 2.3.1. można w dalszych rozważaniach wyeliminować zbiory, dla których

$$j \in \bar{J}_1.$$

Zbiory $\tilde{G}_{q_{i,j}}^1, j \in \bar{J}_1, i = 1, \dots, s_1, j = q_{i+1}^1, \dots, n$

oznaczają się przez $\tilde{G}_{\mu(2)}^2, \mu(2) = \{1, 2, \dots, s_1\}$.

K-ty krok. Obliczenie kresów funkcji celu dla zbiorów $\tilde{G}_{\mu(k)}^k$

$$\xi \left(G_{\mu(k)}^k \right) = \tilde{R}^* \left(q_i^1, q_i^2, \dots, q_i^k \right), \quad i \in \mu(k), \quad /2.3.26./$$

gdzie, podobnie jak poprzednio,

$$\tilde{R}^* \left(q_i^1, \dots, q_i^k \right) = R^* \left(s_i^k, \dots, s_i^k \right) \quad \text{jeśli} \quad \tilde{x}_{q_i^w} = x_{s_i^w}.$$

/2.3.27/

Następnie porównuje się $\xi(\bar{z}) = R^* \left(k_1, \dots, k_m \right)$ oraz $\xi \left(\tilde{G}_{\mu(k)}^k \right)$,

które zdefiniowane jest wzorem (2.3.26.)

$$\xi(z) R \xi \left(\tilde{G}_{\mu(k)}^k \right).$$

1. Jeśli R jest " $\leq$ " $\bigwedge_{j \in \mu(k)}$ wówczas $\bar{z}$ jest w dalszym ciągu rozwiązaniem optymalnym globalnie i algorytm można zakończyć.

2. Jeśli V $J_k = \left\{ q_{i_1}^k, q_{i_2}^k, \dots, q_{i_{s_k}}^k \right\}$ że R jest " $>$ " wówczas rozbija się

zbiory $\tilde{G}_{r_i^k}^k, r_i^k \in J_k$ zgodnie z zasadą

$$\tilde{G}_{r_i}^k = \bigcup_j \tilde{G}_{q_i^k, j}^k, \quad j = q_{i+1}^k, \dots, n, \quad i = 1, \dots, s_k \quad /2.3.28/$$

Biorąc pod uwagę Lemat 2.3.1., można pominąć w dalszych rozważaniach zbiory, dla których $j \in \bar{J}_1$.

Zbiory $\tilde{G}_{q_i^k, j}^k$, $i = 1, \dots, s_k, j = q_{i+1}^k, \dots, n, j \in \bar{J}_1$ oznaczają

się przez $\tilde{G}_{\mu(k+1)}^{k+1}$ i przechodzi do następnego kroku.

m-ty krok. Obliczenie kresów $\xi(G_{\mu(m)}^m)$.

$$\xi(G_{\mu(m)}^m) = R^* (q_1^1, \dots, q_i^m), \quad i \in \mu(m). \quad /2.3.29./$$

Porównanie kresów (2.3.29) z $\xi(\bar{z})$

$$\xi(\bar{z}) \leq \xi(G_{\mu(m)}^m).$$

1. Jeśli R jest " $<$ " wówczas $\bar{z}$ jest rozwiązaniem optymalnym globalnie; koniec algorytmu.

2. Jeśli V , że R jest " $=$ " wówczas istnieje $\tilde{J}_m = \{ \tilde{q}_1^m, \dots, \tilde{q}_{s_m}^m \}$

(s_m+1) rozwiązań optymalnych i wszystkie te rozwiązania można wyznaczyć; koniec algorytmu.

3. Jeśli V , że R jest " $>$ " wówczas rozwiązania $J_m = \{ q_1^m, \dots, q_{s_m}^m \}$

optymalnego należy szukać wśród rozwiązań $\tilde{G}_{q_1^m}^m, \dots, \tilde{G}_{q_{s_m}^m}^m$

porównując kresy funkcji celu osiągnięte na tych zbiorach.

Rozwiązaniem optymalnym w tym przypadku będzie rozwiązanie

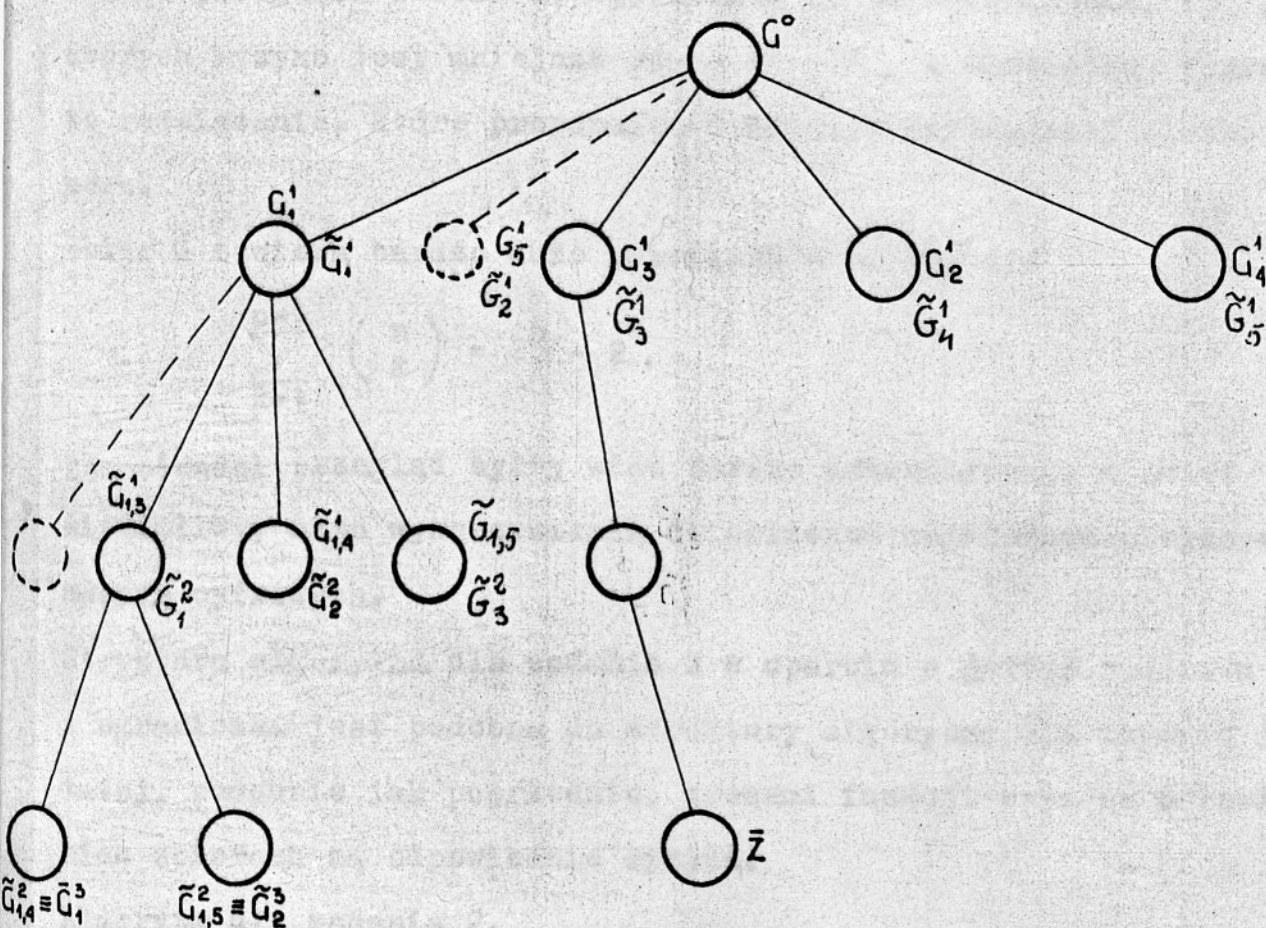
$\tilde{G}_{q_r}^m$, dla którego zachodzi

$$\{(\tilde{G}_{q_r}^m) = \min_i \{ (G_{q_i}^m), i \in \mu(m) \}. \quad /2.3.30./$$

Uwaga:

Zarówno w etapie I, jak i w etapie II w m-tym kroku odpowiedni kresy są wprost równe funkcjom celu.

Graficzny obraz etapu II przedstawia rysunek 2.3.2.



Rysunek 2.3.2.: II etap algorytmu dla zadania 1.

Algorytm ten zrealizowano w formie programu na maszynę cyfrową serii ODRA-1300. Program napisano w języku FORTRAN-1900.

Schemat blokowy tego programu 2009 oraz omówienie go i przedstawienie przykładowych wydruków znajduje się w rozdziale V.

Tabulogram programu umieszczony jest w Dodatku.

W podobny sposób tworzy się algorytm selekcji dla zadania 2., tzn. zadania odrzucania maksymalnej liczby cech takich, aby wzrost ryzyka nie przekroczył z góry zadanej liczby ε .

Niech, podobnie jak poprzednio, G oznacza zbiór wszystkich możliwych rozwiązań. W przypadku bezpośrednim należałoby dokonać całego przeglądu zbioru G , wybrać z nich te rozwiązania, na których ryzyko jest mniejsze równe $R^* + \varepsilon$, a następnie wybrać to rozwiązanie, które proponuje odrzucenie największej liczby cech.

Zbiór G zawiera bardzo dużo rozwiązań a mianowicie

$$\sum_{k=1}^{n-1} \binom{n}{k} = 2^n - 2.$$

Bezpośredni przegląd byłby więc bardzo czasochłonny, a nawet niemożliwy, mimo wykorzystania do obliczeń najnowocześniejszych maszyn cyfrowych.

Struktura algorytmu dla zadania 2 w oparciu o metodę podziału i ograniczeń jest podobna do struktury algorytmu dla zadania 1, tutaj, podobnie jak poprzednio, kresami funkcji celu na odpowiednich zbiorach są odpowiednie ryzyka.

Algorytm dla zadania 2.

0-krok. Rozbicie zbioru $G^0 \equiv G$ na podzbiory w następujący sposób.

$$G^0 = G_1^1 \cup G_2^1 \cup \dots \cup G_n^1, \quad /2.3.31./$$

gdzie

G_1^1 - zbiór tych rozwiązań, w których na pewno odrzucono cechę

o indeksie $i = 1, \dots, n$.

1. krok. Obliczenie $\xi(G_i^1)$

$$\xi(G_i^1) = R^*(i), i = 1, \dots, n : \quad /2.2.32./$$

Sprawdzenie, czy istnieje taki zbiór indeksów $J_1 = \{q_1^1, q_2^1, \dots, q_{s_1}^1\}$

że

$$\xi(G_i^1) \gg R^* + \varepsilon, i \in J_1 . \quad /2.3.33./$$

Jeśli zbiór J_1 pokrywa cały zbiór V^n , wówczas stwierdza się,

że nie można odrzucić żadnej cechy, koniec algorytmu, albo

w przypadku równości w (2.3.33.), odrzucenie jednej cechy.

Zbiór pozostałych indeksów oznacza się przez $\bar{J}_1$.

$$\bar{J}_1 = V^n \setminus J_1 . \quad /2.3.34./$$

Do dalszego rozbitcia wybiera się taki zbiór $G_{k_1}^1$, dla którego

$$\xi(G_{k_1}^1) = \min_{i \in \bar{J}_1} \xi(G_i^1) .$$

Rozbitcia dokonuje się w następujący sposób.

$$G_{k_1}^1 = \bigcup_{\substack{i \in J_1 \\ i \neq k_1}} G_{k_1, i}^1 , \quad /2.3.35./$$

gdzie

$G_{k_1, i}^1$ - zbiór tych rozwiązań, w których na pewno odrzucono cechy o indeksach $k_1, i \in \bar{J}_1$.

Zbiory te oznacza się przez

$$G_{i_1}^2, G_{i_2}^2, \dots, G_{i_r}^2, i_j \in \bar{J}_1, j = 1, \dots, r ,$$

k -krok. $k \leq n-1$. Obliczenie kresów $\xi(G_{i_j}^k)$, $i_j \in \bar{J}_{k-1}$

$$\xi(G_{i_j}^k) = R^* (k_1, k_2, \dots, k_{k-1}, i_j), \quad i_j \in \bar{J}_{k-1}. \quad /2.3.36/$$

Sprawdzenie, czy istnieje taki zbiór indeksów $\tilde{J}_k = \{q_1^k, \dots, q_{s_k}^k\}$,

że

$$\xi(G_i^1) \gg R^* + \varepsilon$$

$$J_k = \bigcup_{p=1}^{k-1} J_p \cup \tilde{J}_{p-1}, \quad \bar{J}_k = V^n \setminus J_k. \quad /2.3.37,/$$

Jeśli zbiór J_k , pokrywa cały zbiór V^n wówczas I etap algorytmu jest zakończony, należy wrócić do poprzedniego kroku i wyznaczyć zbiór $(k-1)$ cech, dla których ryzyko najmniej wzrosło, albo w przypadku równości w (2.3.37.) wyznaczyć k -cech, dla których ta równość zachodzi i przyjąć je za pierwsze rozwiązanie optymalne $\bar{z}$.
Jeśli $V^n \setminus J_k \neq \emptyset$, wówczas sprawdza się, czy $k = n-1$.

1. Jeśli $k = n-1$, wówczas koniec I etapu, jako pierwsze rozwiązanie optymalne przyjmuje się zbiory otrzymane w tym kroku.

Rozwiązanie takie jest również rozwiązaniem optymalnym globalnie, a więc również koniec całego algorytmu. Nie można bowiem odrzucić więcej niż $n-1$ cech.

2. Jeśli $k < n-1$, wówczas należy rozbić zbiór $G_{k_k}^k$, $k_k \in \bar{J}_{k-1}$, dla którego

$$\xi(G_{k_k}^k) = \min_{i_j \in \bar{J}_{k-1}} \xi(G_{i_j}^k), \quad /2.3.38./$$

w następujący sposób

$$G_{k_k}^k = \bigcup_{i \in \bar{J}_k} G_{k_k, i}^k$$

$$i \neq k_1, \dots, k_k,$$

gdzie

$G_{k_k, i}^k$ - oznacza zbiór rozwiązań, w których na pewno odrzucono cechy $k_1, k_2, \dots, k_k, i$, $i \in \bar{J}_k$,

a następnie wykonać kolejny krok.

Etap I algorytmu trwa do momentu otrzymania pierwszego rozwiązania optymalnego $\bar{z}$, czyli ciągu indeksów cech, które algorytm proponuje odrzucić, $\bar{z} = (k_1, k_2, \dots, k_w)$, w -liczba cech, które należy odrzucić.

Biorąc za podstawę rozwiązanie $\bar{z}$ oraz licznosc tego zbioru równą w , doprowadza się zadanie 2 do postaci zadania 1, czyli do zadania, w którym znana jest liczba cech, które należy odrzucić. W tym przypadku liczbę tę wyznaczono w I etapie algorytmu, jest ona równa w .

Zadaniem drugiego etapu jest po pierwsze szukanie korzystniejszego z punktu widzenia kryterium selekcji zbioru cech do odrzucenia o licznosci równej w (jak w algorytmie dla zadania 1.) , po drugie, sprawdzenie, czy istnieje możliwość zwiększenia liczby w . Jeśli okaże się, że taka możliwość istnieje, wówczas w tym etapie należy również wyznaczyć zbiór cech do odrzucenia o licznosci większej niż w .

Podobnie jak w algorytmie dla zadania 1, przed przystąpieniem do etapu drugiego należy "przenumerować" składowe wektora cech $\bar{x} = (x_1, x_2, \dots, x_n)$.

Niech teraz obraz reprezentuje wektor

$\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)$, gdzie

$$\tilde{x}_{n-i} = x_{k_1+i}, \quad i = 0, 1, \dots, n-w$$

Pozostałe składowe wektora $\tilde{x}$, odpowiadają pozostałym cechom wektora w sposób dowolny.

$$\tilde{x}_i = x_j, \quad i = 1, \dots, n-m, \quad j = 1, \dots, n, \quad j \neq k_1, k_2, \dots, k_w$$

Ponieważ znana jest teraz liczba cech w , które należy odrzucić, wynikająca z pierwszego etapu algorytmu, optymalna liczba w może być co najwyżej większa od w , a więc w drugim etapie algorytmu zawęży się liczba rozwiązań, które należy przegłądać.

Odpadają wszystkie rozwiązania o liczbie cech mniejszej od w .

Pozwala to na pominięcie rozwiązań, w których uszeregowany rosnąco ciąg indeksów zaczyna się od liczby większej niż $(n-w)$, przy czym podobnie jak poprzednio, rozwiązanie o pierwszej liczbie równej $(n-w+1)$ zawiera tylko jedno rozwiązanie równe pierwszemu optymalnemu $\bar{z}$, zgodnie z Tw. 2.3.2.

Co więcej, zgodnie z twierdzeniem 1.4.1. można pominąć te rozwiązania, których odpowiednie ciągi indeksów zaczynają się od liczby $p \in J_1$, zbiór J_1 zdefiniowany jest przez (2.3.33.) a nawet zawierają $p \in J_1$.

Etap II.

Zbiór rozwiązań dopuszczalnych $\bar{G}$ jest równy różnicy zbiorów.

$$\bar{G} = G \setminus G_{J_1}, \quad /2.3.40./$$

gdzie

G - zbiór rozwiązań, w których odrzucono więcej niż w cech,

G_{J_1} - zbiór rozwiązań, w których proponuje się odrzucenie cechy $\tilde{x}_p$, $p \in J_1$.

Zbiór indeksów V^n odpowiada zbiorowi indeksów $\bar{V}^n$,

gdzie

$\bar{V}^n$ - uporządkowany zbiór indeksów cech $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n$.

Niech teraz

$$\tilde{V}_n = \bar{V}^n \setminus \tilde{J}_1, \quad /2.3.41./$$

gdzie

$\tilde{J}_1$ - zbiór indeksów cech $\tilde{x}_1$, odpowiadających zbiorowi (2.3.33).

0-krok. Rozbicie zbioru $\bar{G}$ na skończoną liczbę podzbiorów

$$\bar{G} = \bigcup_{i \in \tilde{V}^n} \bar{G}_i^1, \text{ gdzie}$$

$\bar{G}_i^1$ - zbiór rozwiązań, w których na pewno odrzuca się cechę $\tilde{x}_i$, $i \in \tilde{V}^n$.

1-krok. Obliczenie kresów $\xi(\bar{G}_i^1)$, $i \in \tilde{V}^n$,

$$\xi(\bar{G}_i^1) = R^*(i), \quad i \in \tilde{V}^n. \quad /2.3.42./$$

Porównanie

$$\xi(\bar{G}_i^1) \text{ R } \xi(\bar{z}), \quad i \in \tilde{V}_n.$$

1. Jeśli R jest " $\geq$ ", wówczas z jest rozwiązaniem optymalnym globalnie, koniec algorytmu.

2. Jeśli R jest "<", to znaczy istnieje taki zbiór

$J_1' = \{q_1', q_2', \dots, q_{s_1}'\}$, że $\xi(\bar{z}) > \xi(\bar{G}_i^1)$ i J_1' , wówczas rozbija się zbiory $\bar{G}_i^1$, $i \in J_1'$, na podzbiory

$$\bar{G}_1^1 = \bigcup_j \bar{G}_{1,j}^1, \quad i \in J_1', \quad j \in J_1', \quad j \neq i,$$

$\bar{G}_{1,j}^1$ - zbiór rozwiązań, w których na pewno odrzucono cechy $i, j \in J_1'$, a następnie oznaczenie tych zbiorów przez

$$\bar{G}_j^2, \quad j \in J_1'. \quad /2.3.43./$$

k-ty krok. Obliczenie kresów $\xi(\bar{G}_i^k)$, $i \in J_{k-1}$.

$$\xi(\bar{G}_i^k) = \tilde{R}^*(i), \quad \text{gdzie}$$

$$\tilde{R}^*(i) \equiv R^*(l_1^i, l_2^i, \dots, l_k^i), \quad l_j \in J_{j-1}', \quad j = 2, \dots, k$$

$$l_1 \in \tilde{V}^n, \quad i = 1, \dots, k. \quad /2.3.44./$$

Porównanie

$$\xi(\bar{G}_i^k) \quad R \quad \xi(\bar{z}).$$

1. Jeśli $\bigwedge_{i \in \bar{J}_{k-1}} R$ jest " $\gg$ " wówczas $\bar{z}$ jest rozwiązaniem

optymalnym, nie da się odrzucić więcej niż w cech i koniec algorytmu.

2. Jeśli $\bigvee_{J_k' = \{q_1', \dots, q_s'\}}$, że R jest " $<$ ", wówczas rozbija się

zbiory $\bar{G}_i^k$, $i \in J_k'$ na podzbiory

$$\bar{G}_i^k = \bigcup_j \bar{G}_{i,j}^k, \quad i, j \in J_k', \quad i \neq j, \quad /2.3.45./$$

$\bar{G}_{i,j}^k$ - zbiór rozwiązań, w których na pewno odrzucono cechy o indeksach $(l_1^i, \dots, l_k^i, j)$, $i = 1, \dots, k$, $l_s^i \in J_{s-1}'$, $s = 2, \dots, k$, $j \in J_k'$,

a następnie oznacza się te podzbiory przez $\bar{G}_j^{k+1}$, $j \in J_k'$.

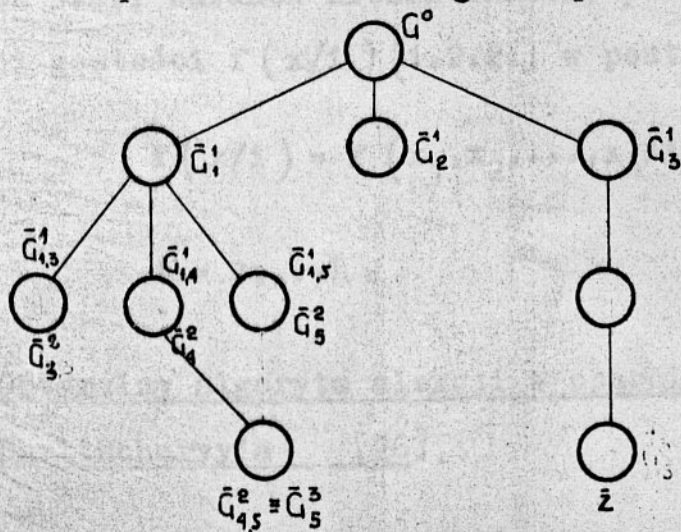
Sprawdzenie czy,

$$\xi(G_i^s) \leq R^* + \varepsilon.$$

1. Jeśli R jest " $>$ " $\bigwedge_{i \in J_{s-1}}$, wówczas rozwiązaniem optymalnym $\bar{z}$ jest rozwiązanie uzyskane w poprzednim kroku; koniec algorytmu.
2. Jeśli R jest " $=$ ", wówczas zbiór tych rozwiązań, dla których zachodzi równość, jest rozwiązaniem optymalnym $\bar{z}$, można odrzucić s -cech; koniec algorytmu.
3. Jeśli R jest " $<$ " dla pewnych $i \in J'_s = \{q_1, \dots, q_s\}$, wówczas znów należy rozbić zbiory $\bar{G}_i^s$, $i \in J'_{s-1}$, oznaczyć otrzymane w ten sposób podzbiory przez $\bar{G}_i^{s+1}$, $i \in J'_s$ i wykonać następny krok.

Postępowanie takie kontynuuje się dopóty, dopóki nie nastąpi w którymś kroku koniec algorytmu, albo gdy liczba kroków drugiego etapu jest równa $n-1$, nie można bowiem odrzucić więcej niż $n-1$ cech. Każde rozwiązanie otrzymane w tym kroku jest rozwiązaniem optymalnym.

Graficzne przedstawienie tego etapu obrazuje rys. 2.3.3.



Rys. 2.3.3. II etap algorytmu dla zadania 2.

Optymalność obu tych algorytmów wynika z optymalności metody podziału i ograniczeń [19]. Realizacja algorytmów w formie programów na m.c jest bardzo skomplikowana, dlatego ze względu na trudności związane z zaprogramowaniem ich oraz ze względu na ograniczony czas obliczeń na m.c uruchomiono tylko jeden program realizujący algorytm dla zadania 1.

3. ALGORYTMY SELEKCJI CECH DLA SZCZEGÓLNYCH PRZYPADKÓW ROZPOZNAWANIA

W poprzednim rozdziale przedstawiono algorytmy selekcji dla zadania 1. i dla zadania 2., dla przypadków, w których nie nakładano żadnych warunków na obrazy i odpowiednie funkcje gęstości. Niniejszy rozdział poświęcony jest pewnym szczególnym przypadkom rozpoznawania, a mianowicie przypadkom spełniającym następujące warunki.

3.1. Liczba klas $L=2$

3.2. Obraz stanowi uporządkowany zbiór zmiennych losowych

niezależnych - cechy są zmiennymi losowymi niezależnymi.

Warunek 3.2., warunek niezależności, pozwala na przedstawienie funkcji gęstości $f(x/i)$ (1.2.2.) w postaci iloczynu.

$$f(x/i) = f(x_1, x_2, \dots, x_n/i) = \prod_{j=1}^n f(x_j/i),$$

$$i = 1, \dots, L.$$

/3.1.0/

3.1. Optymalny algorytm selekcji w oparciu o współczynnik

Bhattacharyy'a [26].

Dla przypadków sformułowanych wyżej, to znaczy spełniających warunki 3.1. oraz 3.2. zbudować można prostsze algorytmy selekcji cech. Jak pokazuje wzór (3.1.0), odpowiednie funkcje gęstości są multiplikatywne względem cech. Natomiast warunek 3.2 pozwala na zastosowanie jako odpowiednich kresów w metodzie podziału i ograniczeń współczynnika Bhattachary'ego zdefiniowanego w rozdziale 1 wzorem (1.3.8.) a omówionego w artykule [26]

$$\mathcal{S} \triangleq \int_{\mathcal{X}} \sqrt{f(x/1) \cdot f(x/2)} dx \quad /3.1.1./$$

W artykule [26] pokazano również, że

$$\sqrt{1 - 4p_1 p_2 \mathcal{S}^2} \gg 2K_D \gg 1 - 2\mathcal{S} \sqrt{p_1 p_2}, \quad /3.2.2./$$

K_D - miara Kołmogorowa (1.3.1.), [26].

Można teraz udowodnić następujące twierdzenie

Tw. 3.1.1.

Dla $L=2$

$$R^* \gg \frac{1}{2} \left[1 - \sqrt{1 - 4p_1 p_2 \mathcal{S}^2} \right]. \quad /3.1.3./$$

Dowód

Dla $L=2$, K_D (1.3.1.) jest równe

$$\begin{aligned} K_D &= \frac{1}{2} \int_{\mathcal{X}} [p_1 |f(x/1) - f(x)| + p_2 |f(x/2) - f(x)|] dx \\ &= \frac{1}{2} \int_{\mathcal{X}} [p_1 |f(x/1) - p_1 f(x/1) - p_2 f(x/2)| + p_2 |f(x/2) - p_1 f(x/1) - \\ &\quad - p_2 f(x/2)|] dx = \frac{1}{2} \int_{\mathcal{X}} [p_1 |f(x/1)(1-p_1) - f(x/2)p_2| + p_2 \cdot \\ &\quad \cdot |f(x/2)(1-p_2) - p_1 f(x/1)|] dx = \frac{1}{2} \int_{\mathcal{X}} [p_1 p_2 |f(x/1) - f(x/2)| + \\ &\quad + p_1 p_2 |f(x/2) - f(x/1)|] dx = \end{aligned}$$

$$= \frac{1}{2} \int_{\mathbb{R}} 2 p_1 p_2 |f(x/1) - f(x/2)| dx =$$

$$= p_1 p_2 \int_{\mathbb{R}} |f(x/1) - f(x/2)| dx.$$

Stąd wniosek, że

K_D dla $p_1 = p_2 = \frac{1}{2}$ oznaczone przez K_D' jest większe niż dla $p_1 \neq p_2$

$$K_D' > K_D. \quad /3.1.5./$$

A więc również dla $p_1 \neq p_2$ (1.3.3.)

$$R^* \leq \frac{1}{2} - K_D. \quad /3.1.6./$$

Inaczej mówiąc

$$K_D \geq \frac{1}{2} - R^* \quad /3.1.7./$$

Wstawiając teraz we wzorze (3.2.1) w miejsce K_D wyrażenie $\frac{1}{2} - R^*$ (można to zrobić zgodnie z 3.1.7.) otrzymuje się, że

$$\sqrt{1 - 4p_1 p_2 g^2} \geq 2 \left(\frac{1}{2} - R^* \right) = 1 - 2R^*.$$

Odejmując od obu stron nierówności jedność, a następnie dzieląc obie strony przez -2 , otrzymuje się

$$\frac{1}{2} - \frac{1}{2} \sqrt{1 - 4p_1 p_2 g^2} \leq R^* \quad /3.1.8./$$

q.e.d.

A więc jako dolne ograniczenie R^* można przyjąć

$$D_g = \frac{1}{2} \left[1 - \sqrt{1 - 4p_1 p_2 g^2} \right]. \quad /3.1.9./$$

Korzystając z warunku 3.2. udowodnione zostanie następujące twierdzenie

Tw. 3.1.2.

Jeśli obraz x spełnia warunek 3.2, wówczas

$$D_{\varrho} = \frac{1}{2} \left[1 - \sqrt{1 - 4p_1 p_2 \prod_{i=1}^n \varrho_i^2} \right], \quad /3.1.10/$$

gdzie

$$\varrho_i \triangleq \int_{\mathcal{X}_i} \sqrt{f(x_i/1) \cdot f(x_i/2)} dx_i. \quad /3.1.11/$$

Dowód:

Zgodnie z definicją

$$\varrho \triangleq \int_{\mathcal{X}} \sqrt{f(x/1) \cdot f(x/2)} dx.$$

Korzystając ze wzoru (3.1.0) otrzymuje się, że

$$\varrho = \int_{\mathcal{X}} \sqrt{f(x_1/1) \cdot f(x_2/1) \dots f(x_n/1) \cdot f(x_1/2) \cdot f(x_2/2) \dots f(x_n/2)} dx_1 \dots dx_n$$

Grupując gęstości pod pierwiastkiem względem cech, otrzymuje się

$$\varrho = \int_{\mathcal{X}} \sqrt{f(x_1/1) \cdot f(x_1/2)} \cdot \sqrt{f(x_2/1) \cdot f(x_2/2)} \dots \sqrt{f(x_n/1) \cdot f(x_n/2)} dx_1 \dots dx_n$$

Ponieważ poszczególne czynniki funkcji podcałkowej zależą tylko od jednej zmiennej można więc napisać

$$\begin{aligned} \varrho &= \int_{\mathcal{X}_1} \sqrt{f(x_1/1) \cdot f(x_1/2)} dx_1 \int_{\mathcal{X}_2} \sqrt{f(x_2/1) \cdot f(x_2/2)} dx_2 \dots \\ &\dots \int_{\mathcal{X}_n} \sqrt{f(x_n/1) \cdot f(x_n/2)} dx_n. \end{aligned}$$

Zgodnie z oznaczeniem (3.1.11)

$$\vartheta = \prod_{i=1}^n \vartheta_i.$$

Zatem

$$\vartheta^2 = \left(\prod_{i=1}^n \vartheta_i \right)^2 = \prod_{i=1}^n \vartheta_i^2. \quad /3.1.12/$$

Wstawiając (3.1.12) do (3.1.8) otrzymuje się (3.1.10).

Multiplikatywność ϑ^2 pozwala na duże uproszczenie algorytmu selekcji.

Łatwo zauważyć, że im mniejsze ϑ tym mniejsze D_ϑ ,

D_ϑ jest malejącą funkcją ϑ . W zadaniach selekcji można więc posługiwać się bezpośrednio współczynnikiem ϑ , a nie całym D_ϑ co jeszcze bardziej upraszcza algorytm selekcji.

Niech, jak w poprzednim rozdziale, G oznacza zbiór wszystkich rozwiązań możliwych. Dla zadania 1 zbiór ten zawiera $\binom{n}{m}$ rozwiązań. Rozwiązania te odpowiadają m -elementowym kombinacjom zbioru indeksów cech $\dots$. Jeśli element zbioru G odpowiadający kombinacji $\{r_1, r_2, \dots, r_m\}$, $r_1, \dots, r_m \in V^n$ oznaczy się zgodnie z (1.4.8) przez U_m^q (q - oznacza numer kombinacji), wówczas ograniczenie od dołu ryzyka po odrzuceniu cech $r_1, \dots, r_m$, czyli

$$R^* (r_1, \dots, r_m) = R^* U_m^q,$$

oznaczone przez $D_{\vartheta U_m^q}$ definiuje się w następujący sposób.

$$D_{\vartheta U_m^q} = \frac{1}{2} \left[1 - \sqrt{1 - 4 p_1 p_2 \left[\int_{x_1, \dots, x_m} \sqrt{\int_{x_1} \dots \int_{x_m} f(x/1) dx_1 \dots dx_m} \int_{x_1} \dots \int_{x_m} f(x/2) dx_1 \dots dx_m} dx_1 \dots dx_m \right]} \right]$$

/3.1.13/

gdzie

$x^{r_1, \dots, r_m}$ - zdefiniowane jest przez (1.4.2),

$x_{r_1}, \dots, x_{r_2}$ - zdefiniowane jest przez (1.4.3),

$x^{r_1, \dots, r_m}$ - zdefiniowane jest przez (1.4.4).

Dla obrazów, których cechy są zmiennymi losowymi niezależnymi, można udowodnić następujące twierdzenie.

Tw. 3.1.3

$$D_{S U_m^q} = \frac{1}{2} \left[1 - \sqrt{1 - 4p_1 p_2 \prod_{\substack{i=1 \\ i \neq r_1, \dots, r_m}}^n S_i^2} \right], \quad /3.1.14/$$

gdzie

$$U_m^q = \{ r_1, r_2, \dots, r_m \}.$$

Dowód

Dowód tego twierdzenia wynika wprost z definicji (3.1.13) oraz (3.1.10), a mianowicie.

$$\begin{aligned} D_{S U_m^q} &\stackrel{(3.1.10)}{=} \frac{1}{2} \left[1 - \sqrt{1 - 4p_1 p_2 \prod_{\substack{i=1 \\ i \neq r_1, \dots, r_m}}^n S_i^2 \cdot \prod_{s=1}^m \left(\sqrt{\int_{x_{r_s}} f(x_{r_s}) dx_{r_s}} \int_{x_{r_s}} f(x_{r_s}/2) dx_{r_s} \right)^2} \right] = \\ &= \frac{1}{2} \left[1 - \sqrt{1 - 4p_1 p_2 \prod_{\substack{i=1 \\ i \neq r_1, \dots, r_m}}^n S_i^2 \cdot \prod_{s=1}^m (1 \cdot 1)^2} \right] = \\ &= \frac{1}{2} \left[1 - \sqrt{1 - 4p_1 p_2 \prod_{\substack{i=1 \\ i \neq r_1, \dots, r_m}}^n S_i^2} \right]. \end{aligned}$$

q.e.d.

Z twierdzenia tego wypływa bardzo ważny wniosek, a mianowicie, w przypadku gdy dąży się, podczas przeprowadzania selekcji, do otrzymania jak najmniejszego $D_{\mathcal{G}} U_m^q$, należy odrzucać te cechy, których odpowiednie $\mathcal{G}_i$ przyjmują wartości największe, innymi słowy należy pozostawić cechy o najmniejszych $\mathcal{G}_i$. Algorytmy selekcji niżej przedstawione wykorzystują tę własność. Algorytm selekcji dla zadania 1.

Krok 1. Dla każdej cechy x_i obliczenie współczynnika Bhattacharyy'a

$$\mathcal{G}_i \text{ (3.1.11) , } i = 1, \dots, n,$$

$$\mathcal{G}_i = \int_{x_i} \sqrt{f(x_i/1) \cdot f(x_i/2)} dx_i,$$

a następnie uporządkowanie ciągu tych współczynników w ciąg rosnący

$$\mathcal{G}_{i_1} \leq \mathcal{G}_{i_2} \leq \dots \leq \mathcal{G}_{i_n}, \quad i_j \in V^n, \quad j = 1, \dots, n$$

Krok 2. Rozbija się zbiór rozwiązań G na $(n-m+1)$ podzbiorów^(*)

$$G = \bigcup_{s=1}^{n-m+1} G_s^1, \text{ gdzie}$$

G_s^1 - zbiór tych rozwiązań, w których na pewno odrzucono cechę i_s -tą, $s > 1$ a pozostawiono składowe $i_{s-1}, \dots, i_{s-r}$,

$$r = 1, \dots, s-1$$

(*) Zgodnie z twierdzeniem 2.3.2., można pominąć niektóre zbiory, $n-m+1$ odpowiednio utworzonych podzbiorów zawiera wszystkie dopuszczalne rozwiązania.

Uwaga:

Dla $s=1$, zbiór G_1^1 , definiuje się jako zbiór rozwiązań, w których na pewno odrzucono cechę i_1 .

Każdy zbiór G_s^1 zawiera $\binom{n-s}{m-1}$ rozwiązań, będących elementami zbioru G .

$$G_s^1 = \{g_{s_1}, g_{s_2}, \dots, g_{s_{\binom{n-s}{m-1}}}\}.$$

Krok 3. Obliczenie odpowiednich kresów dla każdego zbioru G_s^1 , $\xi(G_s^1)$, $s = 1, \dots, n-m+1$. Jako kresy przyjmuje się odpowiednie D_g ograniczające ryzyko od dołu.

Tw. 3.1.1.

Dla zaznaczenia, że ograniczenie dotyczy zbioru G_s^1 , przyjmuje się oznaczenie $D_{G_s^1}$

$$D_{G_s^1} = \min_i D_{g_{s_i}}, \quad i = 1, \dots, n-m+1,$$

gdzie

$$D_{g_{s_j}} = \frac{1}{2} \left[1 - \sqrt{1 - 4p_1 p_2 \prod_{s_j}^2} \right],$$

$$s_j \in V^n$$

$$s_j \neq r_1, \dots, r_m$$

natomiast

$$g_{s_j} = \{r_{j_1}, r_{j_2}, \dots, r_{j_m}\}, \quad j = 1, 2, \dots, \binom{n-s}{m-1}.$$

Uwaga.

Ograniczenia $D_{\mathcal{G}_s}^1$ zachowują pewien porządek, a mianowicie

$$D_{\mathcal{G}_1}^1 \gg D_{\mathcal{G}_2}^1 \gg \dots \gg D_{\mathcal{G}_{n-m+1}}^1$$

krok 4. Wyznaczenie pierwszego rozwiązania optymalnego $\bar{z}$.

Pierwsze rozwiązanie optymalne $\bar{z}$, zawarte jest w zbiorze G_{n-m+1}^1 ponieważ ograniczenie tego zbioru ze względu na uporządkowanie $\mathcal{G}_{i_j}$ jest najmniejsze. Oczywisty jest fakt, że zbiór ten jest jednoelementowy, a więc pierwsze rozwiązanie optymalne $\bar{z}$ równe jest zbiorowi indeksów,

$$\bar{z} = G_{n-m+1}^1 = \{ \mathcal{G}_{(n-m+1)_1} \}$$

$$\bar{z} = \{ i_{n-m+1}, i_{n-m+2}, \dots, i_n \}$$

krok 5. Obliczenie ryzyka $R^*(i_{n-m+1}, i_{n-m+2}, \dots, i_n) = R_1^*$

krok 6. Porównanie

$$D_{\mathcal{G}_s}^1 R R_1^*, s=1, \dots, n-m$$

1. Jeśli R jest " $\gg$ ", wówczas $\bar{z}$ jest rozwiązaniem optymalnym globalnie i koniec algorytmu
2. Jeśli istnieje taki zbiór indeksów $J = \{ q_1, \dots, q_k \}$, że R jest " $<$ ", należy wykonać następny krok.

7 krok. Ze względu na to, że ograniczenia zbiorów G_s^1 , $s=1, \dots, n-m$ uporządkowane są malejąco, wiadomo, że do dalszych rozważań wybrać należy k kolejnych zbiorów.

$$G_{n-m}^1, G_{n-m-1}^1, \dots, G_{n-m-k+1}^1$$

Należy teraz obliczyć ryzyko odpowiadające ograniczeniu zbioru G_{n-m}^1 , $R_1 = R_{D \mathcal{G}}^* G_{n-m}^1$, gdzie

$$D \mathcal{G} G_{n-m}^1 = \min_j D \mathcal{G} \mathcal{G}_{(n-m)_j}, \mathcal{G}_{(n-m)_j} = r_{j_1}, \dots, r_{j_m}, j = 1, \dots, \binom{m}{m-1},$$

$$R_1 = R^*(r_{j_1}, r_{j_2}, \dots, r_{j_m}),$$

a następnie porównać

$$R_1 \quad R \quad R_1^*,$$

1. Jeśli R jest " $<$ " wówczas przyjąć zbiór odpowiadający R_1 za optymalne rozwiązanie $\bar{z}$, poprzednie zapomnieć, a R_1 oznaczyć przez R_1^* i wykonać następny krok.

2. Jeśli R jest " $\geq$ " wówczas należy sprawdzić, czy $R_1 < D \mathcal{G} G_{n-m-1}^1$

a/ jeśli tak, to rozwiązanie $\bar{z}$ jest rozwiązaniem optymalnym globalnie i koniec algorytmu

b/ jeśli nie, należy obliczyć ryzyka dla każdego rozwiązania zawartego w zbiorze G_{n-m-1}^1 , wyznaczyć rozwiązanie dla którego $R^* \mathcal{G}_{(n-m-1)_j}$ jest najmniejsze i dla tego rozwiązania oznaczając to ryzyko przez R_1^* , wykonać jeszcze raz krok 7.

8 krok. Przeprowadzić porównanie

$$R_1^* = R_{D \mathcal{G}}^* G_{n-m}^1 \quad R \quad D \mathcal{G} G_s^1, \quad s = n-m-1, \dots, n-m-k+1,$$

1. Jeśli V , że R jest " $<$ ", wówczas wykonać

$$J = \{s_1, \dots, s_r\}$$

znów krok 7 dla odpowiedniego zbioru $G_{s_i}^1$, $s_i \in J$.

2. Jeśli $\bigwedge_{s=n-m, \dots, n-m-k+1}$ R jest " $<$ ", wówczas zbiór rozwiązań odpowiadający ograniczeniu ostatnio rozpatrywanemu jest rozwiązaniem optymalnym i koniec algorytmu.

Algorytm ten jest o wiele szybszy od algorytmu sformułowanego w poprzednim rozdziale. Wynika to z tego, że od razu na początku otrzymuje się dokładniejsze oszacowanie funkcji celu dla zbiorów, a poza tym wymyga on mniejszej ilości obliczeń ryzyka, co, jak już pokazano w rozdziale 1., nastrocza wiele trudności.

Algorytm ten zrealizowano w formie programu na m.c. serii Odra - 1300, Program PØ Ø2.

Schemat blokowy, omówienie wyników, przykłady wydruków przedstawiono w rozdziale 5. Tabulogram programu zamieszczony jest w dodatku.

W podobny sposób można zbudować algorytm optymalnej selekcji dla zadania 2., a mianowicie zadania odrzucania maksymalnej liczby cech przy zadanym maksymalnym wzroście ryzyka.

Algorytm.

1 krok. Obliczenie g_i (1.3.11) dla każdej cechy x_i , $i=1, \dots, n$, a następnie uporządkowanie ich rosnąco.

$$g_{i_1} \leq g_{i_2} \leq \dots \leq g_{i_n}.$$

2 krok. Wyznaczenie pierwszej liczby optymalnej cech do odrzucenia.

2.1. Obliczenie $R^{*(i_n)}$ a następnie sprawdzenie relacji między

$$R^{*(i_n)} R R^* + \varepsilon,$$

A/ jeśli R jest " $>$ ", wówczas należy sprawdzić czy

$$R^{*(i_n)} \leq D_{\bigotimes_{i_s}}, \bigwedge_s, s = 1, \dots, n-1,$$

a/ jeśli tak, koniec algorytmu, nie można odrzucić żadnej cechy

b/ jeśli $J = \bigvee \{q_1, \dots, q_p\}$ dla których tak nie jest, wówczas

$\bigwedge_{q_i \in J}$ należy obliczyć $R^{*(q_i)}$, a następnie wykonać dla niego

podkrok 2.1.

B/ jeśli R jest " $=$ ", przyjąć $w=1$ i wykonać następny krok.

C/ jeśli R jest " $<$ ", wykonać następny podkrok.

2.2. Obliczenie $R^{*(i_n, i_{n-1})}$, a następnie porównanie

$$R^{*(i_n, i_{n-1})} \underset{R}{R} R^* + \varepsilon,$$

A/ Jeśli R jest " $>$ ", wówczas przyjąć $w=1$ i wykonać następny krok.

B/ Jeśli R jest " $<$ ", wykonać następny podkrok.

C/ Jeśli R jest " $=$ ", $w=2$ i wykonać następny krok.

2.k. Obliczenie $R^{*(i_n, \dots, i_{n-k+1})}$, a następnie wykonać porównanie

$$R^{*(i_n, i_{n-1}, \dots, i_{n-k+1})} \underset{R}{R} R^* + \varepsilon.$$

Jeśli

A. R jest " $>$ ", wówczas przyjąć $w=k-1$ i wykonać następny krok

B. R jest " $=$ ", przyjąć $w=k$ i wykonać następny krok

C. R jest " $<$ ", wykonać następny podkrok

Podkroków w kroku drugim może być najwyżej $n-1$, ponieważ taka

jest możliwa maksymalna liczba cech, jaką można odrzucić.

W wyniku tego kroku otrzymuje się pierwszą liczbę cech, które można odrzucić-w, oraz ryzyko po odrzuceniu tego zbioru.

3.krok. Podobnie jak w algorytmie dla zadania 1., należy sprawdzić, czy istnieją inne cechy i czy jest ich więcej, które można odrzucić, a ryzyko nie przekroczy $R^* + \varepsilon$. W tym celu, podobnie jak poprzednio, należy zbiór rozwiązań G , rozbić na podzbiory

$$G = \bigcup_{s=1}^{n-m+1} G_s^1,$$

G_s^1 - zbiór tych rozwiązań, w których na pewno odrzucono cechę o indeksie i_s , a zostawiono cechy $i_{s-1}, i_{s-2}, \dots, i_{s-r}$,
 $s > 1$.

4 krok. Obliczenie ograniczeń dla każdego zbioru $G_s^1, D_{G_s^1}$ tak jak w kroku 3 w algorytmie dla zadania 1.

5 krok. Porównanie.

$$D_{G_s^1} R R^* (i_n, \dots, i_{n-w+1}) = R_1^*$$

5.1. Jeśli $\bigwedge_{s=1, \dots, n-w} R$ jest " $\gg$ " zbiór cech o indeksach

$i_n, i_{n-1}, \dots, i_{n-w+1}$ jest zbiorem optymalnym globalnie.

5.2. Jeśli $\bigvee_{J=\{q_1, \dots, q_k\} \in V^n}$, że R jest " $<$ ", wówczas należy

wykonać następny krok.

6. krok. Obliczenie ryzyk odpowiadających elementom zbioru G_{n-w}^1 , a następnie wybranie spośród nich takiego rozwiązania, dla którego ryzyko przyjmuje wartość najmniejszą, G_m^1 .

Porównanie

$$R^* \quad G_m^1 \quad R \quad R_1^* .$$

6.1. Jeśli R jest " $<$ ", wykonać następny krok.

6.2. Jeśli R jest " $>$ ", porównać

$$D \quad G_{n-m-1}^1 \quad R \quad R^* \quad G_m^1$$

.A/. Jeśli R jest " $>$ ", wówczas R_1 i odpowiadający mu zbiór jest rozwiązaniem optymalnym $\bar{z}$, koniec algorytmu.

.B/. Jeśli R jest " $<$ ", należy powtórzyć krok szósty dla zbioru G_{n-m-1}^1 , itd.

7. krok. Okazuje się, że można próbować odrzucić jeszcze jedną cechę. Należy więc wyznaczyć zbiór cech odpowiadający rozpatrywanemu zbiorowi G_r^1 , r może się równać $n-m$, $n-m-1, \dots, 1$ w zależności od liczby powtórzeń kroku 6.

Z wcześniejszych rozważań wynika, że zbiór G_r^1 zawiera $\binom{n-r}{w-1}$ rozwiązań elementarnych.

$$G_r^1 = \{g_{r_1}, g_{r_2}, \dots, g_{r \binom{n-r}{w-1}}\} .$$

Jeśli założy się, że liczba cech, które można odrzucić, może być większa, wówczas zbiór G_r^1 należy rozbić na podzbiory

$$G_r^1 = \{g_{r_1}, \dots, g_{r \binom{n-r}{w-1}}\} = \left\{ \{g_{r_1}\}, \{g_{r_2}\}, \dots, \{g_{r \binom{n-r}{w-1}}\} \right\} ,$$

gdzie

$$g_{r_i} = \{i_{r_1}, \dots, i_{r_w}\},$$

natomiast

$$\{g_{r_i}\} = \left\{ \{g_{r_i}, i_{r_{w+1}}\}, \dots, \{g_{r_i}, i_m\} \right\}.$$

Znaczy to, że zbiór G_r^1 zawiera $\binom{n-r}{w}$ rozwiązań.

Należy więc obliczyć ryzyka odpowiadające poszczególnym elementom zbioru G_r^1 , a następnie wybrać element g_{r_0} odpowiadający najmniejszej wartości ryzyka

$$R^* g_{r_0} = \min_{i,j} R^* (g_{r_i}, i_j), \quad i = 1, \dots, \binom{n-r}{w}, \quad j = r_w+1, \dots, m.$$

Przeprowadzić równanie

$$R^* g_{r_0} \leq R^* + \epsilon.$$

7.1. Jeśli R jest " $<$ ", wykonać następny krok

7.2. Jeśli R jest "=", uznać to rozwiązanie za optymalne i koniec algorytmu.

7.3. Jeśli R jest ">", poprzednie rozwiązanie uzyskane w poprzednim kroku uznać za optymalne; koniec algorytmu.

9 krok. Znow podzielić rozwiązanie uważane za najlepsze uzyskane w wyniku poprzednim kroku g_{r_0} na podzbiory, zwiększając liczbę cech do odrzucenia o 1, w sposób przedstawiony w poprzednim kroku i powtórzyć dla niego wszystkie rozważania jak w kroku 8.

Postępować tak aż do momentu otrzymania rozwiązania optymalnego.

Optymalność obu przedstawionych tu algorytmów wynika z optymalności metody podziału i ograniczeń.

Jakkolwiek zaproponowane w tym rozdziale algorytmy selekcji dla zadania 1 i zadania 2, są o wiele prostsze niż algorytmy przedstawione w poprzednim rozdziale, to i one są dosyć skomplikowane, ale są to algorytmy optymalne względem ryzyka.

Nie zawsze wymagana jest optymalność, często wystarcza suboptymalność selekcji, która jest szybsza i łatwiejsza. Jak pokazano w rozdziale 1, algorytmów selekcji suboptymalnej jest bardzo dużo. Poniżej przedstawiony zostanie jeden z takich algorytmów.

3.2. Suboptymalny algorytm selekcji.

Przedstawiony w tym rozdziale algorytm selekcji odnosi się do przypadków rozpoznawania o liczbie klas $L=2$ oraz cechach niezależnych.

Suboptymalność algorytmu polega na tym, że jako kryterium selekcji, przyjęte zostanie zdefiniowane wcześniej oszacowanie ryzyka, oznaczone tutaj przez $\bar{\sigma}$, a nie samo ryzyko.

Definicja

$$\bar{\sigma} \triangleq \int_{\mathcal{X}} |p_1 f(x/1) - p_2 f(x/2)| dx. \quad /3.2.1./$$

Dla tak zdefiniowanego $\bar{\sigma}$ udowodnione zostanie następujące twierdzenie.

Tw. 3.2.1.

$$\mathcal{J} = 1 - 2R^*.$$

/3.2.2./

Dowód.

Zgodnie z definicją (3.2.1.)

$$\mathcal{J} = \int_{\mathcal{X}} |p_1 f(x/1) - p_2 f(x/2)| dx,$$

co w dalszym ciągu można zapisać w postaci

$$\mathcal{J} = \int_{D_{x_1}} (p_1 f(x/1) - p_2 f(x/2)) dx + \int_{D_{x_2}} (p_2 f(x/2) - p_1 f(x/1)) dx, \quad /3.2.3./$$

gdzie

$$D_{x_1} = \left\{ x : p_1 f(x/1) \geq p_2 f(x/2) \right\} = \mathcal{X} \setminus D_{x_2},$$

$$D_{x_2} = \left\{ x : p_2 f(x/2) > p_1 f(x/1) \right\} = \mathcal{X} \setminus D_{x_1}.$$

Korzystając z (3.2.4.) oraz z postaci $\mathcal{J}$ przedstawionej wzorem (3.2.3.) przeprowadzi się następujące przekształcenia

$$\begin{aligned} \mathcal{J} &= \int_{D_{x_1}} p_1 f(x/1) dx - \int_{D_{x_1}} p_2 f(x/2) dx + \int_{D_{x_2}} f(x/2) p_2 dx - \int_{D_{x_2}} p_1 f(x/1) dx = \\ &= p_1 \left[\int_{D_{x_1}} f(x/1) dx - \int_{D_{x_2}} f(x/1) dx \right] + p_2 \left[\int_{D_{x_2}} f(x/2) dx - \int_{D_{x_1}} f(x/2) dx \right] = \\ &= p_1 \left[\int_{\mathcal{X} \setminus D_{x_2}} f(x/1) dx - \int_{D_{x_2}} f(x/1) dx \right] + p_2 \left[\int_{\mathcal{X} \setminus D_{x_1}} f(x/2) dx - \int_{D_{x_1}} f(x/2) dx \right] = \\ &= p_1 \left[\int_{\mathcal{X}} f(x/1) dx - \int_{D_{x_2}} f(x/1) dx - \int_{D_{x_1}} f(x/1) dx \right] + \\ &+ p_2 \left[\int_{\mathcal{X}} f(x/2) dx - \int_{D_{x_1}} f(x/2) dx - \int_{D_{x_2}} f(x/2) dx \right] = \end{aligned}$$

$$\begin{aligned}
 &= p_1 \left[1 - 2 \int_{D_{x_2}} f(x/1) dx \right] + p_2 \left[1 - 2 \int_{D_{x_1}} f(x/2) dx \right] = \\
 &= p_1 + p_2 - 2 \left[\int_{D_{x_2}} f(x/1) dx + \int_{D_{x_1}} f(x/2) dx \right] = \\
 (1.2.14) \quad &= 1 - 2R^*.
 \end{aligned}$$

q.e.d.

Można też napisać, że

$$R^* = \frac{1}{2} [1 - \mathcal{J}]. \quad /3.2.5./$$

Ryzyko R^* jest malejącą funkcją $\mathcal{J}$, to znaczy im większe $\mathcal{J}$ tym mniejsze R^* , a więc zamiast minimalizować R^* można maksymalizować $\mathcal{J}$. I tak na przykład: dla zadania 1, które mówi, że należy odrzucić m-cech takich, aby wzrost ryzyka był minimalny, najmniejszy, można szukać tych m-cech do odrzucenia, które spowodują najmniejsze zmniejszenie wartości $\mathcal{J}$. Wcześniej pokazano, że odrzucenie cechy powoduje wzrost ryzyka. Można również pokazać, że odrzucenie cechy powoduje zmniejszenie wartości $\mathcal{J}$.

Niech

$\mathcal{J}^k$ - oznacza $\mathcal{J}$ po odrzuceniu cechy k-tej

$$\mathcal{J}^k \triangleq \int_{\mathcal{E}^k} \left| p_1 \int_{x_k} f(x/1) dx_k - p_2 \int_{x_k} f(x/2) dx_k \right| dx^k. \quad /3.2.6./$$

Tw. 3.2.2.

$$\mathcal{J} \geq \mathcal{J}^k, \quad /3.2.7./$$

gdzie

$\mathcal{J}$ - zdefiniowana jest wzorem (3.2.1.), a $\mathcal{J}^k$ wzorem (3.2.6.)

Dowód.

Moduł jest funkcją wypukłą, a więc na podstawie twierdzenia Jensena, można pokazać, że

$$\int_{x_k} |p_1 f(x/1) - p_2 f(x/2)| dx_k \gg \left| p_1 \int_{x_k} f(x/1) dx_k - p_2 \int_{x_k} f(x/2) dx_k \right|. \quad /3.2.8./$$

Ponieważ obie strony nierówności (3.2.8.) są dodatnie, to całkując stronami wyrażenie (3.2.8.) otrzymuje się ten sam znak nierówności

$$\int_{x^k} \int_{x_k} |p_1 f(x/1) - p_2 f(x/2)| dx_k dx^k \gg \int_{x^k} \left| p_1 \int_{x_k} f(x/1) dx_k - p_2 \int_{x_k} f(x/2) dx_k \right| dx^k$$

Lewa strona tej nierówności zgodnie z (3.2.1.) jest równa 0 a prawa zgodnie z (3.2.6.) = 0^k, q.e.d.

Jeśli założy się, że wektor x jest wektorem zmiennych losowych niezależnych, wówczas zachodzi następujące twierdzenie.

Tw. 3.2.3.

$$\delta \leq p_1 \sum_{i=1}^n \int_{x_i} |f(x_i/1) - f(x_i/2)| dx_i + |p_1 - p_2| \triangleq \bar{\delta}. \quad /3.2.9./$$

Dowód.

Dowód tego twierdzenia otrzyma się stosując cały ciąg przekształceń w stosunku do δ .

$$\begin{aligned} \delta &= \int_{x} |p_1 f(x/1) - p_2 f(x/2)| dx \quad (3.2) \\ &= \int_{x} |p_1 f(x_1/1) \cdot f(x_2/1) \dots f(x_n/1) - p_2 f(x_1/2) f(x_2/2) \dots f(x_n/2)| dx = \\ &= \int_{x} |p_1 \underline{f(x_1/1)} \cdot f(x_2/1) \dots f(x_n/1) - p_1 \underline{f(x_1/2)} \cdot f(x_2/1) \dots f(x_n/1) + \end{aligned}$$

$$\begin{aligned}
 & + p_1 f(x_1/2) \cdot \underline{f(x_2/1)} \cdot f(x_3/1) \dots f(x_n/1) - p_1 \underline{f(x_2/2)} \cdot f(x_1/2) \cdot \\
 & \cdot f(x_3/1) \dots f(x_n/1) + p_1 f(x_2/2) \cdot f(x_1/2) \cdot \underline{f(x_3/1)} \dots f(x_n/1) - \\
 & - p_1 \underline{f(x_3/2)} \cdot f(x_2/2) \cdot f(x_1/2) \dots f(x_n/1) + \dots + \\
 & + p_1 f(x_1/2) \cdot f(x_2/2) \dots f(x_{n-1}/2) \cdot \underline{f(x_n/1)} - p_1 f(x_1/2) \dots f(x_{n-1}/2) \cdot \\
 & \cdot \underline{f(x_n/2)} + \underline{p_1} f(x_1/2) \cdot f(x_2/2) \dots f(x_n/2) - \underline{p_2} f(x_1/2) \dots f(x_n/2) \Big| dx,
 \end{aligned}$$

Zauważyć należy, że składniki sumy pod całką mają parami takie same elementy z wyjątkiem podkreślonych. Można więc te wspólne elementy wyciągnąć przed nawias, odpowiednio grupując składniki. Wówczas

$$\begin{aligned}
 \oint_{\mathcal{F}} &= \int \left| p_1 f(x_2/1) \cdot f(x_3/1) \dots f(x_n/1) [f(x_1/1) - f(x_1/2)] + \right. \\
 & + p_1 f(x_1/2) \cdot f(x_3/1) \dots f(x_n/1) [f(x_2/1) - f(x_2/2)] + \\
 & + p_1 f(x_1/2) \cdot f(x_2/2) f(x_n/1) \dots f(x_n/1) [f(x_3/1) - f(x_3/2)] + \dots \\
 & \dots + p_1 f(x_1/2) \cdot f(x_2/2) \dots f(x_{n-1}/2) [f(x_n/1) - f(x_n/2)] + \\
 & + f(x_1/2) \cdot f(x_2/2) \dots f(x_n/2) [p_1 - p_2] \Big| dx \leq \\
 & \leq \int \left\{ \left| p_1 f(x_2/1) f(x_3/1) \dots f(x_n/1) [f(x_1/1) - f(x_1/2)] + \right. \right. \\
 & + p_1 f(x_1/2) \cdot f(x_3/1) \dots f(x_n/1) [f(x_2/1) - f(x_2/2)] + \dots
 \end{aligned}$$

$$\begin{aligned}
 & \dots + p_1 f(x_1/2) \cdot f(x_2/2) \dots f(x_{n-1}/2) |f(x_n/1) - f(x_n/2)| + \\
 & + f(x_1/2) f(x_2/2) \dots f(x_n/2) |p_1 - p_2| \Big\} dx = \\
 & = p_1 \int_{x_2} f(x_2/1) dx_2 \int_{x_3} f(x_3/1) dx_3 \dots \int_{x_n} f(x_n/1) dx_n \int_{x_1} |f(x_1/1) - f(x_1/2)| dx_1 + \\
 & + p_1 \int_{x_1} f(x_1/2) dx_1 \cdot \int_{x_3} f(x_3/1) dx_3 \dots \int_{x_n} f(x_n/1) dx_n \int_{x_2} |f(x_2/1) - f(x_2/2)| dx_2 + \\
 & \dots + p_1 \int_{x_1} f(x_1/2) dx_1 \int_{x_2} f(x_2/2) dx_2 \dots \int_{x_{n-1}} f(x_{n-1}/2) dx_{n-1} \int_{x_n} |f(x_n/1) - f(x_n/2)| dx_n + \\
 & + \int_{\bar{x}} f(x/2) dx |p_1 - p_2| = \\
 & = p_1 \int_{x_1} |f(x_1/1) - f(x_1/2)| dx_1 + p_1 \int_{x_2} |f(x_2/1) - f(x_2/2)| dx_2 + \dots + p_1 \int_{x_n} |f(x_n/1) - \\
 & - f(x_n/2)| dx_n + |p_1 - p_2| = |p_1 - p_2| + p_1 \sum_{i=1}^n \int_{x_i} |f(x_i/1) - f(x_i/2)| dx_i ,
 \end{aligned}$$

q.e.d.

Twierdzenie to ma bardzo ważne znaczenie, ponieważ pokazuje że istnieje addytywne oszacowanie $\bar{\delta}$, addytywne względem cech.

Można teraz sformułować bardzo proste algorytmy selekcji zarówno dla zadania 1, jak i dla zadania 2.

Jednakże algorytmy te będą suboptymalne.

Algorytm dla zadania 1.

1 krok. Obliczenie dla każdej cechy $x_i, i=1, \dots, n$, δ_{oi}

$$\delta_{oi} \triangleq \int_{x_i} |f(x_i/1) - f(x_i/2)| dx_i .$$

/3.2.10./

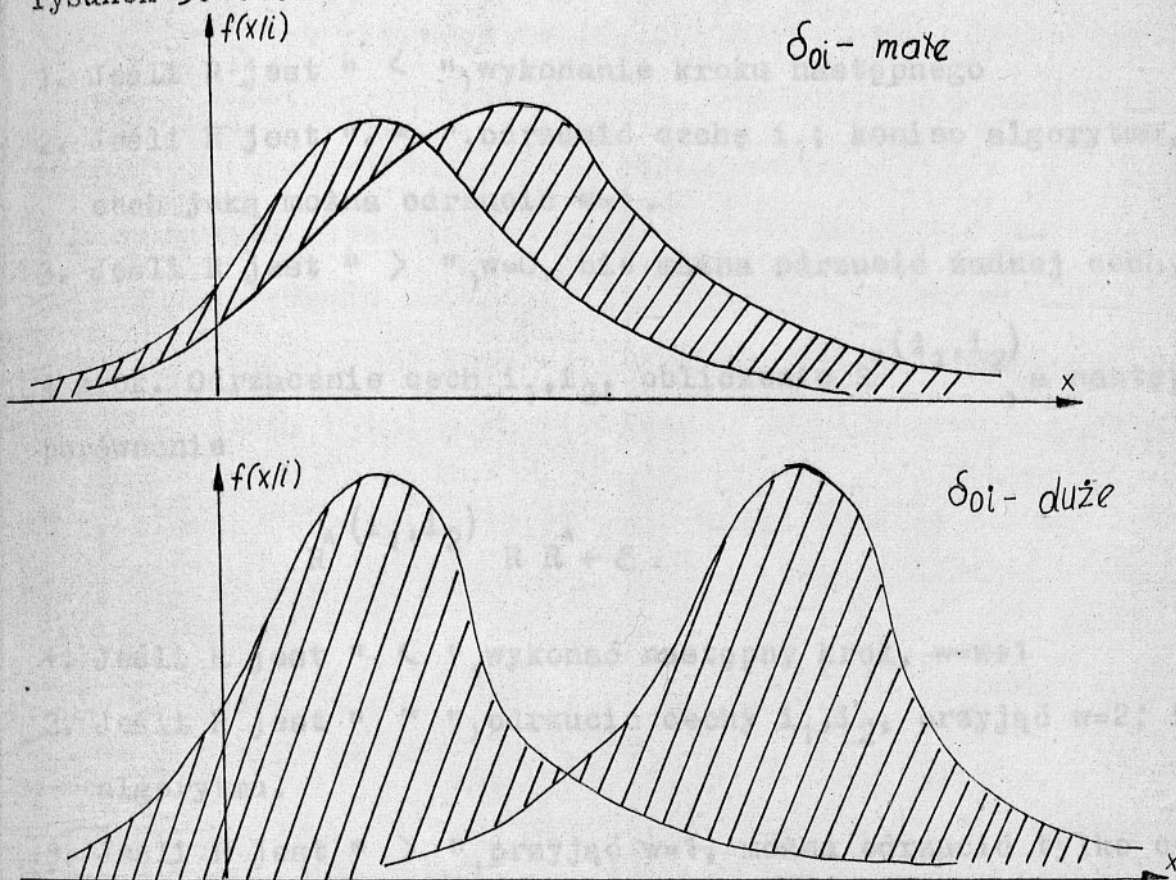
2 krok. Spośród cech $x_1, \dots, x_n$ wybranie takich m -cech, $x_{k_1}, \dots, x_{k_m}$ dla których odpowiednie $\sigma_{o_{k_j}}$, $j = 1, \dots, m$ są najmniejsze.

3 krok. Odrzucić cechy $x_{k_1}, x_{k_2}, \dots, x_{k_m}$. Koniec algorytmu.

Algorytm ten jest optymalny ze względu na oszacowanie (3.2.1.), ale tylko suboptymalny względem σ , a tym samym suboptymalny względem ryzyka.

Jak już wcześniej zostało powiedziane, w zadaniu 1 należy odrzucić te m -cech, które spowodują najmniejszą różnicę między σ przed i σ po selekcji. A zatem również oszacowanie dąży w tym samym kierunku, tzn. należy odrzucić te m -cech, które spowodują najmniejsze zmniejszenie wartości oszacowania $\bar{\sigma}$. Ponieważ oszacowanie to jest addytywne względem cech, należy odrzucić te cechy, dla których odpowiednie składniki, czyli σ_{o_i} osiągają najmniejsze wartości. Wydaje się, że wykorzystywanie tego algorytmu jest uzasadnione, ponieważ widać, że σ są wtedy małe, gdy rozkłady prawdopodobieństwa dla tej samej cechy w różnych klasach są "bliskie", odpowiednie funkcje gęstości mocno "zachodzą na siebie", rys. 3.2.1.

rysunek 3.2.1.



Rys. 3.2.1. Funkcje gęstości o różnych δ_{oi} .

Algorytm dla zadania 2.

1. Krok. Obliczenie δ_{oi} dla każdej cechy $x_i, i=1, \dots, n$

$$\delta_{oi} = \int_{x_i} |f(x_i/1) - f(x_i/2)| dx_i.$$

2 krok. Utworzenie rosnącego ciągu $\delta_{oi_k}, k=1, \dots, n$

$$\delta_{oi_1} \leq \delta_{oi_2} \leq \dots \leq \delta_{oi_n}$$

3 krok. Odrzucenie cechy x_{i_1} , obliczenie $R^{*(i_1)}$ a następnie porównanie.

$$R^{*(i_1)} R R^* + \varepsilon.$$

1. Jeśli R jest " $<$ ", wykonanie kroku następnego
 2. Jeśli R jest " $=$ ", odrzucić cechę i_1 ; koniec algorytmu, liczba cech jaką można odrzucić $w=1$.
 3. Jeśli R jest " $>$ ", $w=0$, nie można odrzucić żadnej cechy.
- 4 krok. Odrzucenie cech i_1, i_2 , obliczenie $R^{*(i_1, i_2)}$, a następnie porównanie

$$R^{*(i_1, i_2)} R R^* + \varepsilon.$$

1. Jeśli R jest " $<$ ", wykonać następny krok, $w=w+1$
2. Jeśli R jest " $=$ ", odrzucić cechy i_1, i_2 , przyjąć $w=2$; koniec algorytmu.
3. Jeśli R jest " $>$ ", przyjąć $w=1$, można odrzucić tylko cechę i_1 ; koniec algorytmu.

2 + k-krok. Odrzucenie $i_1, i_2, \dots, i_k$, obliczenie $R^{*(i_1, \dots, i_k)}$, a następnie porównanie

$$R^{*(i_1, \dots, i_k)} R R^* + \varepsilon.$$

1. Jeśli R jest " $<$ ", wykonać następny krok, zwiększając w o jeden.
2. Jeśli R jest " $=$ ", odrzucić cechy $i_1, i_2, \dots, i_k$, przyjąć $w=k$; koniec algorytmu.
3. Jeśli R jest " $>$ ", przyjąć $w=k-1$, odrzucić cechy o indeksach $i_1, \dots, i_{k-1}$; koniec algorytmu.

Krok $k+2$ wykonywać dopóty, dopóki nie nastąpi koniec algorytmu, liczba w wskaże ile cech należy odrzucić, albo gdy liczba $w = n-1$, ponieważ dalej odrzucać nie można, rozpoznawanie należy przeprowadzić co najmniej za pomocą jednej cechy.

Algorytmy dla zadań 1,2 w oparciu o oszacowanie $\bar{\sigma}$ zrealizowano w formie programów na m.c. serii ODRA-1300, program Z007, Z008. Schematy blokowe omówienie i przykłady wydruków zamieszczono w rozdziale 5. Tabulogramy programów zamieszczono w Dodatku.

Zdefiniowane wzorem (3.2.1.) σ odpowiada zdefiniowanemu wcześniej (1.3.1.) mierze K_D .

Tw. 3.2.4.

Łatwo pokazać, że dla $L=2$, $p_1=p_2=\frac{1}{2}$

$$K_D = \frac{1}{2} \sigma \quad /3.2.11./$$

Dowód. Zgodnie z (3.1.4.) w rozważanym przypadku

$$K_D = p_1 p_2 \int_{\mathbb{X}} |f(x/1) - f(x/2)| dx$$

dla $p_1, p_2 = \frac{1}{2}$

$$K_D = \frac{1}{2} \cdot \frac{1}{2} \int_{\mathbb{X}} |f(x/1) - f(x/2)| dx \quad /3.2.12./$$

Natomiast σ (3.2.1.)

$$\sigma = \int_{\mathbb{X}} |p_1 f(x/1) - p_2 f(x/2)| dx$$

dla $p_1, p_2 = \frac{1}{2}$

$$\sigma = \frac{1}{2} \int_{\mathbb{X}} |f(x/1) - f(x/2)| dx$$

A zatem

$$K_D = \frac{1}{2} \sigma$$

q.e.d.

Przedstawione w tym rozdziale algorytmy zarówno optymalne, jak i suboptymalne dotyczą szczególnych przypadków rozpoznawania, co mocno zawęża pole ich zastosowania, chociaż w praktyce takich przypadków spotyka się bardzo wiele.

Następny rozdział poświęcony zostanie budowie suboptymalnego algorytmu selekcji na podstawie zdefiniowanego w nim oszacowania ryzyka zbieżnego do ryzyka.

4. SUBOPTYMALNY ALGORYTM SELEKCJI W OPARCIU O ZBIEŻNE OSZACOWANIE RYZYKA.

Ze względu na brak addytywnych bądź multiplikatywnych oszacowań ryzyka, ze względu na cechy, brak jest prostych algorytmów selekcji w problemach rozpoznawania o liczbie klas $L \gg 2$, albo cechach zależnych.

Algorytmy optymalne, w których kryterium selekcji jest ryzyko, są długie i skomplikowane ze względu na szczególną postać ryzyka R^* . Duże trudności przy realizacji algorytmów optymalnych na m.c sprawia fakt, że algorytmy te wymagają wielu obliczeń całek wielowymiarowych, szczególnie przy kolejnych nawrotach algorytmów na "wyższe poziomy".

W niniejszym rozdziale zdefiniowano pewną miarę $M_k(J/X)$ i na podstawie tej miary zdefiniowano oszacowanie ryzyka R_k^* , zbieżne do niego przy parametrze $k \rightarrow \infty$. Do własności tej przywiązuje się dużą wagę, ponieważ pozwala ona na dobór odpowiednich wielkości takich, aby oszacowanie dowolnie mało różniło się od ryzyka.

Następnie zaprogramowano algorytmy selekcji cech dla zadania 1

i dla zadania 2 sformułowanych w rozdziale 1.

Algorytmy te są suboptymalne względem ryzyka, a także suboptymalne względem oszacowania. Suboptymalność wynika z założenia, tzn. z faktu, że jako kryterium selekcji przyjęto oszacowanie a nie ryzyko, a także z samej postaci algorytmu.

4.1. Definicja i własności miary $M_k(J/X)$

Definicja miary $M_k(J/X)$.

$$M_k(J/X) \triangleq \int_{\mathfrak{X}} \left[\frac{1}{L} \sum_{i=1}^L p_i^k f^k(x/i) \right]^{1/k} dx, \quad /4.1.1./$$

gdzie

k -dowolna liczba naturalna, $k \gg 2$

Zgodnie z (1.2.5.), (4.1.1.) można zapisać w postaci

$$M_k(J/X) = \int_{\mathfrak{X}} f(x) \left[\frac{1}{L} \sum_{i=1}^L p^k(i/x) \right]^{1/k} dx. \quad /4.1.2./$$

Oznaczając

$$E_x \left\{ \cdot \right\} = \int_{\mathfrak{X}} f(x) (\cdot) dx, \quad /4.1.3./$$

wyrażenie (4.1.1.) można zapisać jeszcze w innej postaci, a mianowicie

$$M_k(J/X) = E_x \left\{ \left[\frac{1}{L} \sum_{i=1}^L p^k(i/x) \right]^{1/k} \right\}. \quad /4.1.4./$$

W dalszych rozważaniach wykorzystywana będzie postać (4.1.4.) ze względu na ułatwienie interpretacji tej miary.

Tak zdefiniowana miara ma pewne własności, które ujęte zostaną w postaci twierdzeń.

Tw.4.1.4.

$$R^* \leq 1 - M_k(J/X). \quad /4.1.5./$$

Dowód:

Ryzyko R^* zdefiniowane przez (1.2.13.) wykorzystując oznaczenie (4.1.13.) można zapisać w postaci

$$R^* = 1 - E \left\{ \max_x \sum_{i=1, \dots, L} p(i/x) \right\}. \quad /4.1.6./$$

Oznaczając przez

$$M_k(J/\bar{x}) = \sqrt[k]{\frac{1}{L} \sum_{i=1}^L p^k(i/\bar{x})}. \quad /4.1.7./$$

otrzymuje się, że

$$M_k(J/x) = \sqrt[k]{\frac{1}{L} \sum_{i=1}^L p^k(i/x)} \leq \sqrt[k]{\frac{1}{L} \cdot L \max_i p^k(i/x)} = \max_i p(i/x), \quad /4.1.8./$$

Uśredniając obie strony nierówności (4.1.8.) po x otrzymuje się

$$E_x \left\{ M_k(J/x) \right\} = M_k(J/X) \leq E_x \left\{ \max_i p(i/x) \right\}, \quad /4.1.9./$$

a mnożąc obie strony tej nierówności przez -1 i dodając do obu stron jedność otrzymuje się (4.1.5.) q.e.d.

Własność (4.1.5.) jest bardzo ważna, mówi ona, że ryzyko można szacować za pomocą miary $M_k(J/X)$.

Następne twierdzenie pokazuje również bardzo ważną własność, a mianowicie własność zbieżności.

Tw. 4.1.2.

$$\lim_{k \rightarrow \infty} [1 - M_k(J/X)] = R^*. \quad /4.1.10./$$

Dowód:

Wyrażenie (4.1.8.) pokazuje, że

$$M_k(J/x) \leq \max_i p(i/x), \quad i=1, \dots, L,$$

otrzymano więc oszacowanie od góry.

Natomiast oszacowanie od dołu jest postaci

$$M_k(J/x) = \sqrt[k]{\frac{1}{L} \sum_{i=1}^L p^k(i/x)} \gg \sqrt[k]{\frac{1}{L} \left[\max_i (p(i/x)^k) \right]} = \sqrt[k]{\frac{1}{L}} \max_i p(i/x) \quad /4.1.11./$$

Obliczając granicę przy $k \rightarrow \infty$ oszacowania od dołu a następnie od góry, otrzymuje się

$$\lim_{k \rightarrow \infty} \left[\max_i p(i/x) \right] = \max_i p(i/x),$$

$$\lim_{k \rightarrow \infty} \left[\sqrt[k]{\frac{1}{L}} \max_i p(i/x) \right] = \max_i p(i/x) \cdot \lim_{k \rightarrow \infty} \sqrt[k]{\frac{1}{L}} = \max_i p(i/x).$$

A więc granica dolnego oszacowania jest równa granicy górnego oszacowania, stąd jasne, że również

$$\lim_{k \rightarrow \infty} M_k(J/x) = \max_i p(i/x). \quad /4.1.12./$$

Uśredniając obustronnie powyższych x obie strony wyrażenia

(4.1.12.) otrzymuje się, że

$$\begin{aligned} E \left\{ \lim_{k \rightarrow \infty} M_k(J/x) \right\} &= \lim_{k \rightarrow \infty} E \left\{ M_k(J/x) \right\} = \lim_{k \rightarrow \infty} M_k(J/X) = \\ &= E \left\{ \max_i p(i/x) \right\}, \end{aligned} \quad (4.1.12.)$$

stąd już łatwo wynika, że

$$\lim_{k \rightarrow \infty} [1 - M_k(J/X)] = R^* .$$

q.e.d.

Zbieżność oszacowania $1 - M_k(J/X)$ jest bardzo ważną własnością. Żadne z oszacowań pokazanych w rozdziale 1 własności tej nie posiada. Jedyny wyjątek stanowi miara zdefiniowana przez J_{to} , Q_n (1.2.13.), ale miara ta jest zdefiniowana tylko dla $L=2$, a więc zakres jej stosowania jest raczej wąski. Łatwo również wykazać następujące twierdzenie.

Tw.4.2.3.

$$\sqrt{\frac{1}{L} \sum_{i=1}^L p_i^k} \ll M_k(J/X) \ll 1 . \quad /4.1.13./$$

Dowód

Miara $M_k(J/X)$ przyjmuje wartość najmniejszą w przypadku, gdy $f(x/i) = f(x/j) = g(x)$, $i, j = 1, \dots, L$, $i \neq j$

Zgodnie z (4.1.1.)

$$M_k(J/X) = \int \left[\frac{1}{L} \sum_{i=1}^L p_i^k f^k(x/i) \right]^{1/k} dx \gg \int \left[\frac{1}{L} \sum_{i=1}^L p_i^k g^k(x) \right]^{1/k} dx =$$

$$= \int_X g(x) \sqrt[k]{\frac{1}{L} \sum_{i=1}^L p_i^k} dx = \sqrt[k]{\frac{1}{L} \sum_{i=1}^L p_i^k}.$$

Lewa strona nierówności (4.1.13.) jest więc udowodniona.

Dowód prawej strony otrzyma się korzystając z (4.1.9.), a mianowicie

$$M_k(J/X) \leq E_x \left\{ \max p(i/x) \right\} \leq 1.$$

Oczywisty jest fakt, że jeśli

$$\begin{cases} p_i = p_j = \frac{1}{L}, f(x/i) = f(x/j), i, j = 1, \dots, L \\ \text{wówczas} \\ M_k(J/X) = \frac{1}{L} \end{cases}$$

/4.1.14./

Wynika to wprost z twierdzenia (4.1.1.), a konkretniej, z lewej strony nierówności (4.1.13.). Wystarczy tylko w (4.1.13.) w miejsce p_i wstawić $1/L$.

Co więcej okazuje się, że w przypadku (4.1.14.) zachodzi

$$R^* = 1 - \frac{1}{L} = 1 - M_k(J/X),$$

/4.1.15./

to znaczy ryzyko R^* przyjmuje wartość największą.

W przypadku, gdy $M_k(J/X) = 1$, wówczas $R^* = 0$.

Zachodzi tak wówczas, gdy $p_k = 1$, $p_i = 0$, $i = 1, \dots, L$, $i \neq k$.

Można więc powiedzieć, że miara $M_k(J/X)$ przyjmuje wartość największą wtedy, gdy ryzyko jest najmniejsze, a wartość najmniejsza, gdy ryzyko jest największe.

Można również oszacować maksymalną odległość oszacowania $1 - M_k(J/X)$ od R^* .

Tw. 4.1.4.

Niech ΔRM_k oznacza maksymalną odległość między oszacowaniem a ryzykiem.

$$\Delta RM_k = |1 - M_k(J/X) - R^*| \quad /4.1.16./$$

wówczas

$$\Delta RM_k \leq 1 - \sqrt[k]{\frac{1}{L}}. \quad /4.1.17./$$

Dowód.

Zgodnie z (4.1.8.) oraz (4.1.11.)

$$\sqrt[k]{\frac{1}{L}} \max_i p(i/x) \leq M_k(J/x) \leq \max_i p(i/x), \quad i=1, \dots, L. \quad /4.1.18./$$

Uśredniając po wszystkich x tę nierówność (4.1.18) otrzymuje się

$$\sqrt[k]{\frac{1}{L}} (1 - R^*) \leq M_k(J/X) \leq 1 - R,$$

a zatem

$$\Delta RM_k = (1 - R^*) - \sqrt[k]{\frac{1}{L}} (1 - R^*) = (1 - R^*) \left(1 - \sqrt[k]{\frac{1}{L}}\right) \leq 1 - \sqrt[k]{\frac{1}{L}}.$$

Okazuje się, że odległość między ryzykiem a oszacowaniem zależy od parametru k , co więcej

$$\lim_{k \rightarrow \infty} \Delta RM_k = \lim_{k \rightarrow \infty} \left(1 - \sqrt[k]{\frac{1}{L}}\right) = 1 - \lim_{k \rightarrow \infty} \sqrt[k]{\frac{1}{L}} = 0. \quad /4.1.19./$$

Istnieje więc możliwość regulowania odległości między R^* a oszacowaniem. Biorąc duże k otrzyma się oszacowanie dowolnie

bliskie R^* .

Tw. 4.1.5.

Jeśli żąda się, aby

$$\Delta RM_k < \varepsilon_1, \quad \varepsilon_1 > 0, \quad /4.1.20./$$

Wówczas należy wziąć

$$k \gg - \frac{\log L}{\log(1-\varepsilon_1)}. \quad /4.1.21./$$

Dowód:

Zgodnie z (4.1.17.)

$$\Delta RM_k \leq 1 - \sqrt[k]{1/L},$$

a więc jeśli

$$1 - \sqrt[k]{1/L} \leq \varepsilon_1 \implies \Delta RM_k \leq \varepsilon_1, \quad /4.1.22./$$

Przekształcając nierówność (4.1.22.) otrzymuje się

$$1 - \sqrt[k]{1/L} \leq \varepsilon_1,$$

$$\sqrt[k]{1/L} \geq 1 - \varepsilon_1.$$

Logarytmując obie strony powyższej nierówności

$$-\frac{1}{k} \log L \geq \log(1-\varepsilon_1),$$

a następnie mnożąc obie strony przez k , dzieląc przez $\log(1-\varepsilon_1)$, otrzymuje się, że (należy zauważyć, że $\log(1-\varepsilon_1)$ jest liczbą ujemną, a więc znak nierówności zmieni się na przeciwny)

$$k \gg - \frac{\log L}{\log(1-\xi_1)}$$

q.e.d.

Np.: Jeśli $L=2$ $\xi_1 = 0,125$, to

$$k \gg - \frac{\log L}{\log(1-\xi_1)} = - \frac{\log 2}{\log 0,875} = 5.$$

Istnieje więc możliwość znajdowania ξ_1 - dostatecznego oszacowania ryzyka.

4.2. Porównanie niektórych oszacowań ryzyka.

Porównanie oszacowań ryzyka przedstawionych w rozdziale pierwszym (1.3.1.), (1.3.4.), (1.3.6.), (1.3.10.), (1.3.16), (1.3.17) oraz oszacowania przedstawionego w rozdziale 4.1. przeprowadzone zostanie na podstawie analizy prostego przykładu, $L=2$.

Korzystając z przekształcenia (1.2.5) oraz oznaczenia (4.1.3.), oszacowania te można zapisać w postaci

$$1. R^* = 1 - E \left\{ \max_{i=1,2} p(i/x) \right\}$$

$$2. 1-B(J/X) = E \left\{ 1 - \sum_{i=1,2} p^2(i/x) \right\}$$

$$3. 1-M_2(J/X) = E \left\{ 1 - \sqrt{1/2 \sum_{i=1,2} p^2(i/x)} \right\}$$

$$4. 1-M_4(J/X) = E \left\{ 1 - \sqrt[4]{1/2 \sum_{i=1,2} p^4(i/x)} \right\}$$

$$5. 1-M_8(J/X) = E \left\{ 1 - \sqrt[8]{1/2 \sum_{i=1,2} p^8(i/x)} \right\}$$

$$6. 1-K_D(J/X) = E \left\{ \frac{1}{2} \left[1 - \sum_{i=1,2} |p(i/x) - p_i| \right] \right\}$$

$$7. \varphi - \frac{1}{2} = E_x \left\{ \sum_{i=1,2} \sqrt{p_i p(i/x)} - \frac{1}{2} \right\}, \dots$$

$$8. \frac{1}{2} - 1,65 J(X) = E_x \left\{ \frac{1}{2} - 1,65 \sum_{i=1,2} p(i/x) \log \frac{p(i/x)}{p_i} \right\}$$

$$9. h(J/X) = 2 E_x \{ p(1/x) p(2/x) \}$$

$$10. \frac{1}{2} - MN_0(J/X) = \frac{1}{2} - E_x \left\{ \left| p(1/x) - \frac{1}{2} \right|^2 + \left| p(2/x) - \frac{1}{2} \right|^2 \right\}$$

$$11. 1 - M_{16}(J/X) = 1 - E_x \left\{ \sqrt[16]{\frac{1}{2} [p(1/x)^{16} + p(2/x)^{16}]} \right\}$$

Obliczenia przeprowadzone zostaną dla $p_1=p_2=\frac{1}{2}$, dla $p(1/x)$ przyjmującego wartości 0,05, 0,1, 0,2, 0,3, 0,4, 0,5,

Należy zauważyć, że w tym przypadku $p(1/x) = 1-p(2/x)$, a zatem na odcinku $p(1/x) \in [0,5, 1]$ wartości odpowiednich oszacowań będą symetryczne w stosunku do $p(1/x) \in [0,5]$.

Wartości odpowiednich oszacowań przedstawia tabela 4.1., (przed uśrednieniem po x).

Tabela 4.1. Wartości różnych oszacowań ryzyka.

Lp	$p(1/X)$	0,05	0,1	0,2	0,3	0,4	0,5
1. R_X^*		0,05	0,1	0,2	0,3	0,4	0,5
2. $1-B$		0,075	0,18	0,32	0,42	0,48	0,5
3. $1-M_2$		0,32	0,36	0,42	0,46	0,49	0,5
4. $1-M_4$		0,205	0,29	0,375	0,425	0,47	0,5
5. $1-M_8$		0,125	0,17	0,26	0,35	0,44	0,5
6. $1-K_D$		0,05	0,1	0,2	0,3	0,4	0,5
7. $\varphi - \frac{1}{2}$			0,394	0,447	0,478	0,494	0,5
8. $\frac{1}{2} - J \cdot 1,65$			0,236	0,365	0,44	0,487	0,5

9.2 $h(J/X)$	0,095	0,18	0,32	0,42	0,48	0,5
10. $\frac{1}{2}MN_0$	0,095	0,18	0,32	0,42	0,48	0,5
11. $1-M_{16}(J/X)$	0,0511	0,1001	0,2001	0,3	0,4	0,5

Wiersze tej tabeli przedstawiają wartości odpowiednich oszacowań dla różnych $p(1/X)$, kolumny zaś obrazują dokładność oszacowania ryzyka R_X^* dla jednej wartości $p(1/X)$. Łatwo zauważyć, że oszacowanie $1-K_D$ w omawianym przypadku jest wprost równe R_X^* .

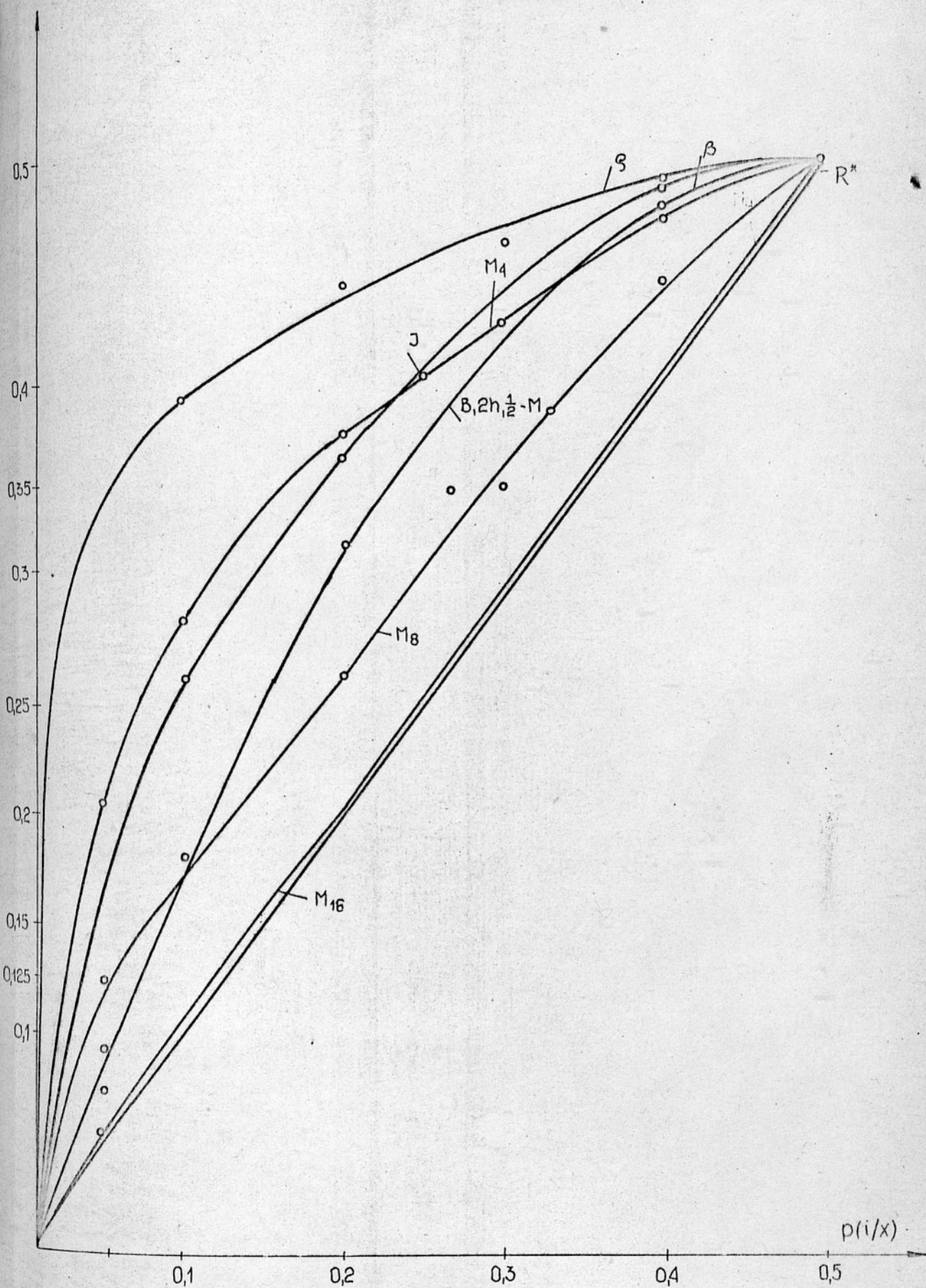
Wynika to ze wzoru otrzymanego dla $p_1 = p_2 = \frac{1}{2}$, ze wzoru /3.2.11/ oraz /3.2.5/.

Aby uzyskać bardziej obrazowe porównanie otrzymanych wyników wartości z tabeli 4.1 przedstawiono na rysunkach 4.2 i 4.3.

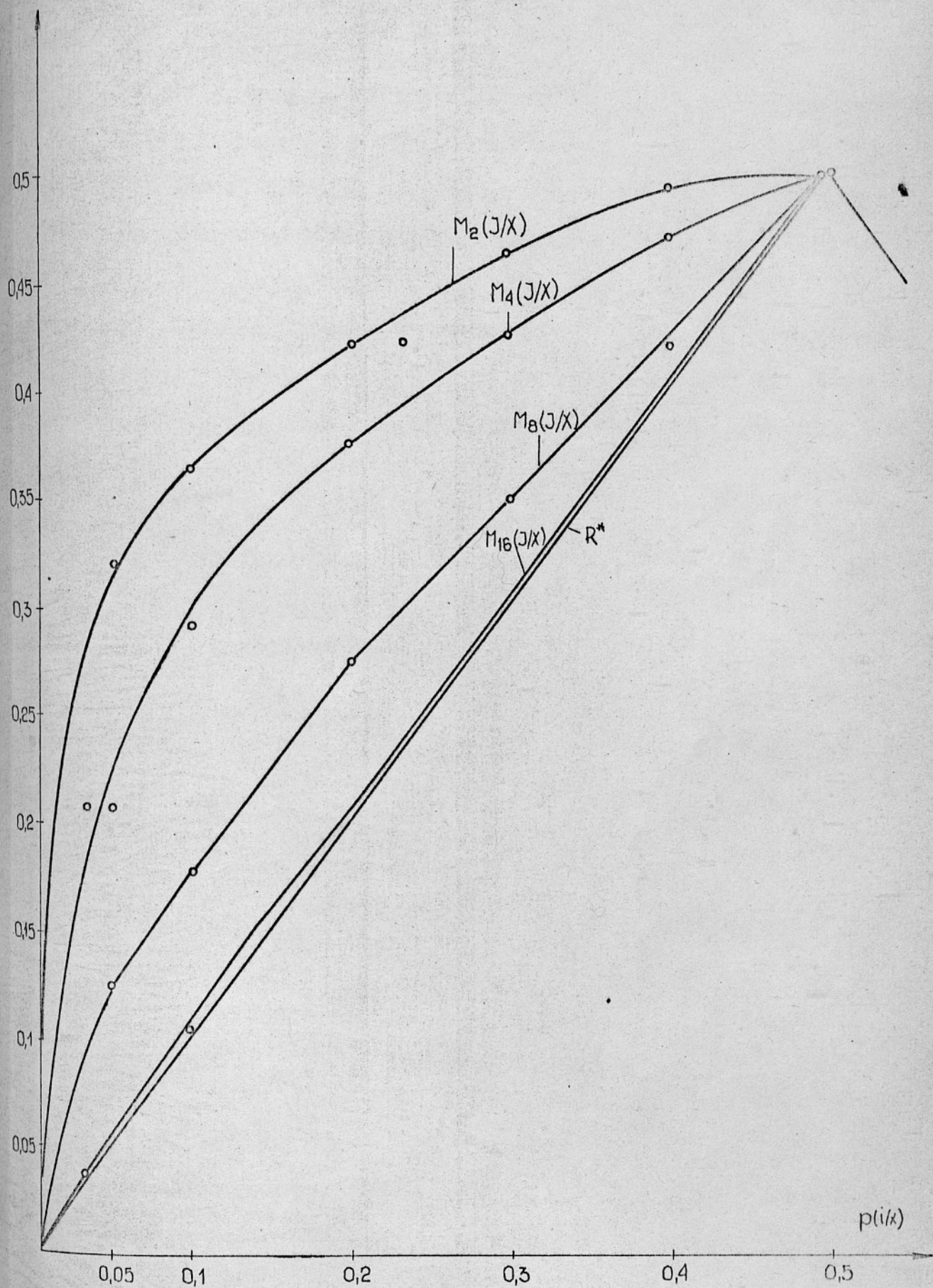
Rys. 4.2 przedstawia dokładność oszacowań ryzyka za pomocą różnych oszacowań, rys. 4.3 pokazuje jak szybko rośnie dokładność oszacowania R_X^* za pomocą miary $M_k(J/X)$, przy rosnącym k , $k=2, 4, 8, 16$.

Z rysunku 4.2 albo z tabeli 4.1 łatwo wywnioskować, że spośród prezentowanych w tym rozdziale oszacowań ryzyko najdokładniej szacowane jest przez $1-M_{16}(J/X)$. Oszacowanie to / $k=16$ / pokrywa się niemal z ryzykiem R_X^* .

Narzuca się więc wniosek, że szacowanie ryzyka R_X^* za pomocą miary $M_k(J/X)$ jest o wiele dokładniejsze od innych, jeśli tylko weźmie się odpowiednio duże k , np. już dla $R_X^* \gg 0,38$ nawet M_4 dokładniej szacuje ryzyko niż najlepszą spośród pozostałych miar, miarą $B(J/X)$.



RYG. 4.1. ZALEŻNOŚĆ RÓŻNYCH TYPÓW OŚZACOWAŃ OD $p(i/x)$.



RYŚ. 4.2. DOKŁADNOŚĆ OSZACOWANIA R^* W ZALEŻNOŚCI OD WIELKOŚCI k

Z tegoż rysunku widać również, że ze wzrostem k oszacowania $1 - M_K(J/x)$ staje się coraz bliższe R_x^* . Widać również, że oszacowanie za pomocą $B(J/x)$ jest takie samo jak za pomocą $h(J/x)$ oraz $M_N(J/x)$. Najgorsze oszacowania powstają dla małych ryzyk. Otrzymane tutaj wnioski potwierdza również następny przykład, w którym porównane zostaną oszacowania uśrednione po wszystkich x .

Przykład: $L = 2$, $p_1 = p_2 = \frac{1}{2}$.

Niech przestrzeń obrazów $\mathcal{X}$ będzie czterowymiarową przestrzenią dyskretną, binarną, a zatem obraz stanowi czterowymiarowy wektor cech $x = (x_1, x_2, x_3, x_4)$, którego elementy (cechy) mogą przyjmować tylko dwie wartości, 0 lub 1.

$x_i = 0$ lub 1 , $i = 1, 2, 3, 4$.

Inaczej mówiąc przestrzeń $\mathcal{X}$ stanowi zbiór wierzchołków 4-wymiarowej kostki.

Dane są prawdopodobieństwa warunkowe.

Dla cechy x_1

$$p_1(0/1) = 0,2$$

$$p_1(0/2) = 0,4$$

dla cechy x_2

$$p_2(0/1) = 0,9$$

$$p_2(1/1) = 0,3$$

Dla cechy x_3

$$p_3(0/1) = 0,5$$

$$p_3(1/1) = 0,3$$

dla cechy x_4

$$p_4(0/1) = 0,1$$

$$p_4(1/1) = 0,2$$

Dla tak postawionego zadania obliczone zostały:

$$R^*, 1 - B(J/x), 1 - M_4(J/x), 1 - M_g(J/x)$$

Odpowiednie prawdopodobieństwa warunkowe $p(x^k/1)$, $p(x^k/2)$ oraz $p(x^k)$ przedstawiono w tabeli 2, $k = 0, \dots, 2^4 - 1$ oznacza numer obrazów w przestrzeni $\mathcal{X}$.

Tabela 2. Wartości $p(x^k/1)$, $p(x^k/2)$, $p(x^k)$

Lp.	Obraz	$p(x^k/1)$	$p(x^k/2)$	$p(x^k)$
0	0000	0,009	0,0072	0,0081
1	0001	0,081	0,0288	0,0549
2	0010	0,009	0,0168	0,0129
3	0011	0,081	0,0672	0,0741
4	0,100	0,001	0,0168	0,0089
5	0101	0,009	0,0672	0,0381
6	0,110	0,001	0,0392	0,0201
7	0111	0,009	0,1568	0,0829
8	1000	0,036	0,0108	0,0234
9	1001	0,324	0,432	0,1836
10	1010	0,036	0,0252	0,0306
11	1011	0,324	0,1008	0,2124
12	1100	0,004	0,0252	0,0146
13	1101	0,036	0,1008	0,0684
14	1110	0,004	0,0588	0,0314
15	1111	0,036	0,2352	0,1356

Wartości policzonego R^* oraz pewnych oszacowań R^* są następujące

$$R^* = 0,1961$$

$$1 - B(J/X) = 0,2911$$

$$\Delta R_B = 0,0950$$

$$1 - M_4(J/X) = 0,3195$$

$$\Delta R_{M_4} = 0,1234$$

$$1 - M_8(J/X) = 0,2083$$

$$\Delta R_{M_8} = 0,0122$$

Porównując otrzymane wyniki z rysunkiem 4.1 można zauważyć, że im większe k tym dokładność oszacowania jest większa. Wzrost k o 4 spowodował zmniejszenie odległości między oszacowaniem a R^* aż 10-krotnie. Co więcej, dla $k = 8$ otrzymano odległość prawie 8 razy mniejszą niż analogiczną odległość otrzymaną w wyniku zastosowania oszacowania $1 - B(J/X)$. Devijver [15] udowodnił, że oszacowanie to dokładniej szacuje R^* niż wiele innych oszacowań nie przedstawionych w niniejszej pracy. Stąd należy wnioskować, że dla odpowiednio dużych k zdefiniowane tutaj oszacowanie jest również lepsze.

4.3. Algorytmy selekcji wykorzystujące oszacowanie $1 - M_K(J/X)$

Łatwo zauważyć, że miara $M_K(J/X)$ /4.1.1/ jest funkcją wypukłą względem $p(i/x)$, $i = 1, \dots, L$.

Stosując twierdzenie Jensena /podobnie jak dla δ / można pokazać, że wartość tej miary maleje wraz z odrzucaniem cech. Można więc zadanie 1 sformułowane w rozdziale 1 przedstawić w nieco zmodyfikowanej postaci.

Należy odrzucić spośród n cech m cech takich, aby zmiana wartości miary $M_K(J/X)$ była najmniejsza.

Zadanie 2 natomiast można przedstawić w następującej formie.

Należy odrzucić możliwie największą liczbę cech takich, aby zmniejszenie wartości $M_K(J/X)$ nie przekroczyło wartości $\varepsilon > 0$. Tak sformułowane zadania selekcji są zadaniami suboptymalnymi względem ryzyka.

Można teraz przystąpić do budowy algorytmów selekcji wykorzystujących miarę $M_K(J/X)$ dla zadań sformułowanych w rozdziale 1.

Algorytm dla zadania 1.

0-krok. Podobnie jak w poprzednich algorytmach niech G oznacza zbiór wszystkich rozwiązań dopuszczalnych.

Rozbija się ten zbiór na skończoną liczbę podzbiorów, w następujący sposób

$$G = G_1^1 \cup G_2^1 \cup \dots \cup G_n^1,$$

gdzie

G_i^1 - oznacza zbiór rozwiązań, w których na pewno odrzucono cechę o indeksie i , $i=1, \dots, n$.

1-krok. Dla każdego zbioru G_i^1 oblicza się oszacowanie funkcji celu określonej na tym zbiorze $\xi(G_i^1)$, $i=1, \dots, n$

$$\xi(G_i^1) = M_k^{i-1}(J/X) \triangleq \int_{x^i} [1/L \sum_{j=1}^L (p_j \int_{x_i} f(x/j) dx_i)^k]^{1/k} dx_i^i,$$

$i = 1, 2, \dots, n$.

Dla dalszego rozbitcia wybiera się ten zbiór $G_{k_1}^1$, dla którego

$$\xi(G_{k_1}^1) = \max_i \xi(G_i^1), \quad i = 1, \dots, n,$$

a następnie rozbija się ten zbiór na podzbiory

$$G_{k_1}^1 = G_{k_1,1}^1 \cup \dots \cup G_{k_1,n}^1 = \bigcup_{i=1}^n G_{k_1,i}^1, \quad i \neq k_1,$$

gdzie

$G_{k_1,i}^1$ - zbiór tych rozwiązań, w których na pewno odrzucono cechy k_1, i , $i = 1, \dots, n$, $i \neq k_1$

Zbiory te oznaczają się przez

$$G_i^2, \quad i = 1, \dots, n, \quad i \neq k_1.$$

s-krok. Obliczenie oszacowań funkcji celu dla zbiorów G_i^s ,

$$\xi(G_i^s), \quad i=1, \dots, n, \quad i \neq k_1, \dots, k_{s-1}$$

wg wzoru

$$\xi(G_i^s) = M_k^{(k_1, \dots, k_{s-1}, i)}(J/X) = \int [1/L \sum_{j=1}^L (p_j \cdot x^{k_1, \dots, k_{s-1}, i})]$$

$$\int \dots \int \int f(x/j) dx_{k_1} \dots dx_{k_{s-1}} dx_i^{1/k} dx_{k_1, \dots, k_{s-1}, i}.$$

Do dalszego rozbitcia wybiera się ten zbiór $G_{k_s}^s$, dla którego

$$\xi(G_{k_s}^s) = \max_i \xi(G_i^s), \quad i=1, \dots, n, \quad i \neq k_1, \dots, k_{s-1},$$

a następnie rozbitcia się ten zbiór na podzbiory

$$G_{k_s}^s = \bigcup_i G_{k_s, i}^s, \quad i = 1, \dots, n, \quad i \neq k_1, \dots, k_s$$

i oznaczają je przez

$$G_i^{s+1}, \quad i=1, \dots, n, \quad i \neq k_1, \dots, k_s$$

m-krok. Obliczenie oszacowań funkcji celu dla zbiorów

$$G_i^m, \quad i = 1, \dots, n, \quad i \neq k_1, \dots, k_{m-1}, \quad \xi(G_i^m)$$

$$\xi(G_i^m) = M_k^{(k_1, k_2, \dots, k_{m-1}, i)}(J/X).$$

Wybór zbioru $G_{k_m}^m$, dla którego

$$\xi(G_{k_m}^m) = \max_i \xi(G_i^m), \quad i=1, \dots, n, \quad i \neq k_1, \dots, k_{m-1},$$

a następnie przyjęcie rozwiązania

$$\bar{z} = (k_1, k_2, \dots, k_m)$$

Uwaga.

Jeśli w którymś z kroków okaże się, że istnieje więcej niż jeden zbiór wytypowany do dalszego rozbicia, wybiera się dowolny.

Algorytm dla zadania 2, jest bardzo podobny. Istnieje tylko ta różnica, że w każdym kroku algorytmu należy sprawdzać, czy nie przekroczono zadanego ε .

Algorytm dla zadania 2

0-krok. Niech G oznacza zbiór wszystkich możliwych rozwiązań

1-krok. Rozbicie zbioru G na podzbiory

$$G = \bigcup_{i=1}^n G_i$$

G_i - oznacza zbiór tych rozwiązań, w których na pewno odrzucona została cecha i -ta, a następnie dla każdego zbioru G_i obliczenie kresu górnego $\xi(G_i) = M_k^i (J/X)$, $i = 1, \dots, n$

$$M_k^i = \int_{x^i} \left[\frac{1}{L} \sum_{j=1}^L \left(p_j \int_{x_i} f(x/j) dx_1 \right)^k \right]^{1/k} dx^i,$$

2-krok. Wybór takiego zbioru G_{k_1} , dla którego

$$M_k^{k_1} = \max_{i=1, \dots, n} M_k^i.$$

Sprawdzenie,

$$M_k^{k_1} \cdot R M_k - \varepsilon.$$

1. Jeśli R jest " $>$ ", wykonanie następnego kroku i podstawienie $p=1$
2. Jeśli R jest "=", przyjęcie $p=1$, $\bar{z}=(k_1)$; koniec algorytmu
3. Jeśli R jest "<", nie można odrzucić żadnej cechy, koniec algorytmu.

3-krok. Rozbicie zbioru G_{k_1} na podzbiory.

$$G_{k_1} = \bigcup_{\substack{i=1 \\ i \neq k_1}}^n G_{k_1,i},$$

gdzie

$G_{k_1,i}$ - oznacza zbiór rozwiązań, w których napewno odrzucono cechy k_1,i .

Obliczenie oszacowań funkcji dla tych zbiorów

$$\xi(G_{k_1,i}) = M_k^{k_1,i} \quad (J/X), \quad i = 1, \dots, n, \quad i \neq k_1$$

4-krok. Wybór takiego zbioru G_{k_1,k_2} dla którego

$$\xi(G_{k_1,k_2}) = \max_{\substack{i=1, \dots, n \\ i \neq k_1}} \xi(G_{k_1,i}).$$

Porównanie

$$M_k^{k_1,k_2} \quad R \quad M_k - \varepsilon.$$

1. Jeśli R jest " $>$ ", przyjęcie $p=2$ i wykonanie następnego kroku.
2. Jeśli R jest "=", przyjęcie $p=2$, $\bar{z}=(k_1,k_2)$; koniec algorytmu.
3. Jeśli R jest "<", przyjęcie $p=1$, $\bar{z}=(k_1)$; koniec algorytmu.

krok $2s-1$. Rozbicie zbioru $G_{k_1, \dots, k_{s-1}} = \bigcup_{\substack{i=1 \\ i \neq k_1, \dots, k_{s-1}}}^n G_{k_1, \dots, k_{s-1}, i}$
i obliczenie oszacowania górnego

$$\xi(G_{k_1, \dots, k_{s-1}, i}), i = 1, \dots, n, i \neq k_1, \dots, k_{s-1},$$

$$\xi(G_{k_1, \dots, k_{s-1}, i}) = M_k(k_1, \dots, k_{s-1}, i) \quad (I/X) .$$

$2s$ -krok. Wybór takiego zbioru $G_{k_1, \dots, k_s}$, dla którego

$$M_k(k_1, \dots, k_s) = \max_{\substack{i=1, \dots, n \\ i \neq k_1, \dots, k_{s-1}}} M_k(k_1, \dots, k_{s-1}, i) .$$

Porównanie

$$M_k(k_1, \dots, k_s) R M_k - \varepsilon .$$

1. Jeśli R jest " $>$ ", przyjęcie $p=s$ i wykonanie następnego kroku
2. Jeśli R jest "=", przyjęcie $p=s$, $\bar{z}=(k_1, \dots, k_s)$; koniec algorytmu
3. Jeśli R jest "<", przyjęcie $p=s-1$, $z=(k_1, \dots, k_{s-1})$; koniec algorytmu.

Uwaga 1

Kroków w tym algorytmie może być co najwyżej $2(n-1)$, to znaczy, że musi pozostać co najmniej jedna cecha, na podstawie której przeprowadzać się będzie rozpoznanie.

Uwaga 2

$$M_k^{(s_1, s_2, \dots, s_r)} \triangleq \int_{x_{s_1}, \dots, x_{s_r}} \left[\frac{1}{L} \sum_{j=1}^L \left(p_j \int \dots \int f/x_j dx_{s_1} \dots dx_{s_r} \right)^k \right]^{1/L} \cdot dx_{s_1}, \dots, dx_{s_r}$$

oznacza wartość miary $M_k(I/X)$ po odrzuceniu r cech $x_{s_1}, x_{s_2}, \dots, x_{s_r}$, $s_1, \dots, s_r \in V^n$.

Oba algorytmy przedstawione w tym rozdziale, tak jak wspomniano wcześniej, są algorytmami suboptymalnymi względem ryzyka R^* , ale o wiele prostszymi niż algorytmy przedstawione w rozdziale 2 i 3.

W porównaniu z algorytmami przyjmującymi jako kryterium selekcji ryzyko, prostota ich polega na tym, że rozwiązanie znajduje się idąc wzdłuż jednej gałęzi. Nie ma potrzeby wracania do "wiszących" zbiorów rozwiązań na wyższych poziomach, w związku z czym następuje skrócenie obliczeń algorytmu.

Zadając duże k , można spodziewać się, że rozwiązanie wynikające z algorytmu tutaj przedstawionego nie będzie się zbyt różniło od rozwiązań otrzymanych w algorytmach suboptymalnych opartych bezpośrednio na ryzyku.

Tw.4.3.1

Jeśli założyć się, że ryzyko w poszczególnych krokach liczone będzie z dokładnością do s -tego miejsca po przecinku, a maksymalną odległość ΔRM_k między oszacowaniem a ryzykiem założyć się równą $\frac{1}{m} 10^{-(s+1)}$, wówczas w etapie I algorytmów omówionych w rozdziale 2.3 dla zadania pierwszego otrzyma się takie samo rozwiązanie jak w algorytmie wyżej sformułowanym (w oparciu o oszacowanie $M_k(I/X)$).

Dowód :

Biorąc
$$k \gg \frac{-\log L}{\log \left(1 - \frac{1}{m} 10^{-(s+1)}\right)}$$

różnica między odpowiednim ryzykiem a oszacowaniem wystąpi dopiero na miejscu $(s+1)$ -wszym po przecinku, a zatem do s -tego miejsca włącznie ryzyko i oszacowanie są sobie równe. W każdym kroku etapu I, algorytmu przedstawionego w rozdziale 2, oraz w każdym kroku algorytmu przedstawionego w rozdziale 4.3 wykonywane są takie same czynności, z tą tylko różnicą, że w pierwszym przypadku jest liczone ryzyko, a w drugim miara M_k , dająca taki sam efekt. Zatem pierwsze rozwiązanie $\bar{z}$ uzyskane w rozdziale 2, jest równe rozwiązaniu $\bar{z}$ w algorytmie z rozdziału 4.3, *q.e.d.*

Rozważając w podobny sposób algorytmy dla zadania 2, otrzyma się podobne wyniki.

Można więc powiedzieć, że algorytmy zbudowane w oparciu o miarę M_k są ε_1 -suboptymalne względem ryzyka, gdzie

$$\varepsilon_1 = \frac{1}{m} 10^{-(s+1)}$$

Algorytmy selekcji dla zadania 1 i zad.2 wykorzystujące miarę $M_k(J/X)$ zrealizowano w formie programów na m.c serii ODRA-1300, programy Z0005, Z0006.

Schematy blokowe, omówienie wyników, przykładowe wydruki przedstawiono w rozdz.5. Tabulogramy programów znajdują się w Dodatku.

5. OMÓWIENIE PROGRAMÓW I ICH WYNIKÓW

5.1.

W pracy przedstawiono osiem algorytmów selekcji, 4 dla zadania 1, czyli zadania optymalnego odrzucania m cech spośród n , 4 dla zadania 2, czyli zadania odrzucania maksymalnej liczby cech takich aby wzrost ryzyka nie przekroczył zadanej z góry liczby $\xi > 0$.

Cztery algorytmy przedstawione w pracy są algorytmami optymalnymi względem ryzyka, natomiast cztery pozostałe suboptymalne.

Sześć algorytmów zrealizowano w formie programów na m.c. serii ODRA-1300.

Programy napisane są w języku FORTRAN-1900.

Zestawienie programów:

1. Program Z009 - realizuje algorytm optymalnej selekcji w oparciu o ryzyko R^* , dla zadania 1
2. Program P002- realizuje algorytm optymalnej selekcji dla $L = 2$, cech niezależnych, w oparciu o oszacowanie za pomocą wsp. Bhattachary'ego η ,
Algorytm realizuje zad.1.
3. Program Z005- realizuje algorytm optymalnej selekcji w oparciu o $M_k(J/X)$, dla funkcji gęstości spełniających odpowiednie warunki - dla zad.1.
4. Program Z006- tak jak Z005- dla zad.2.
5. Program Z007- realizuje algorytm suboptymalnej selekcji dla $L = 2$, w oparciu o oszacowanie δ i dla niezależnych cech - dla zad.1.

6. Program Z008 - tak jak Z007 - dla zad.2.

Wszystkie programy uruchomione zostały na przykładzie rozpoznawania dla $L = 2$ /czyli dwu klas/, dla 10-wymiarowego wektora obrazów $x = (x_1, x_2, \dots, x_{10})$, o cechach niezależnych z rozkładami normalnymi. Prawdopodobieństwo a priori występowania j-tej klasy przyjęto

$$p_1 = 0,3, \quad p_2 = 0,7$$

Parametry rozkładów $f(x_i / j)$, $i = 1, \dots, 10$, $j = 1, 2$

są następujące:

$m_{11} = 5$	,	$\sigma_{11} = 1$	,	$a_{11} = 2$	,	$b_{11} = 8$
$m_{12} = 2$	,	$\sigma_{12} = 2$	,	$a_{12} = -4$	,	$b_{12} = 8$
$m_{21} = 2$	,	$\sigma_{21} = 2$	,	$a_{21} = -4$	,	$b_{21} = 8$
$m_{22} = 5$	,	$\sigma_{22} = 1$	,	$a_{22} = 2$	,	$b_{22} = 8$
$m_{31} = 0$	,	$\sigma_{31} = 1$	,	$a_{31} = -3$	,	$b_{31} = 3$
$m_{32} = 1$	,	$\sigma_{32} = 3$	,	$a_{32} = -8$	,	$b_{32} = 10$
$m_{41} = -2$	,	$\sigma_{41} = 2$	,	$a_{41} = -8$	,	$b_{41} = 6$
$m_{42} = 0$	,	$\sigma_{42} = 3$	,	$a_{42} = -9$	,	$b_{42} = 9$
$m_{51} = 10$	,	$\sigma_{51} = 5$	,	$a_{51} = -5$	,	$b_{51} = 20$
$m_{52} = 5$	,	$\sigma_{52} = 2$	,	$a_{52} = -1$	,	$b_{52} = 11$
$m_{6,1} = 1$	,	$\sigma_{6,1} = 0,5$	,	$a_{6,1} = -1$	,	$b_{6,1} = 3$
$m_{6,2} = 2$	,	$\sigma_{6,2} = 2$	,	$a_{6,2} = -4$	,	$b_{6,2} = 8$
$m_{7,1} = 0$	,	$\sigma_{7,1} = 3$	,	$a_{7,1} = -9$	,	$b_{7,1} = 9$
$m_{7,2} = 2$	,	$\sigma_{7,2} = 1$	,	$a_{7,2} = -1$	,	$b_{7,2} = 5$
$m_{8,1} = 5$	,	$\sigma_{8,1} = 2$	,	$a_{8,1} = -2$	,	$b_{8,1} = 11$
$m_{8,2} = 4$	,	$\sigma_{8,2} = 0,5$	,	$a_{8,2} = 2$	,	$b_{8,2} = 6$
$m_{9,1} = 0$	,	$\sigma_{9,1} = 0,5$	,	$a_{9,1} = -2$	,	$b_{9,1} = 2$
$m_{9,2} = -1$	,	$\sigma_{9,2} = 2$	,	$a_{9,2} = -5$	,	$b_{9,2} = 7$
$m_{10,1} = 4$	,	$\sigma_{10,1} = 4$	,	$a_{10,1} = -12$	,	$b_{10,1} = 20$
$m_{10,2} = 0$	,	$\sigma_{10,2} = 1$	,	$a_{10,2} = -3$	,	$b_{10,2} = 3$

gdzie

m_{ij} - oznacza wartość oczekiwaną i-tej cechy w j-tej klasie

6.11 - " " warianci " " " " "

a_{1j} - " dolną granicę całkowania " " "

b_{ij} - " górna " " " " " " "

$$i = 1, \dots, 10, \quad j = 1, 2.$$

Wszystkie programy, w których należy obliczać całki wielowymiarowe; obliczanie ryzyka R^* , oszacowania $1-M_K(J/K)$ zawierają procedurę obliczania tych całek zbudowaną na podstawie metody Monte-Carlo obliczania całek wielowymiarowych.

Metoda podstawowa obliczania całek wielowymiarowych postaci [46]

$$\int \dots \int_{\Omega} g(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n.$$

Niech P będzie rozkładem równomiernym na Ω , to znaczy niech

$$dP = \frac{dx_1 dx_2 \dots dx_n}{|\Omega|},$$

gdzie $|\Omega|$ jest miarą Ω w przestrzeni n -wymiarowej.

Losując LC punktów $(x_{1j}, x_{2j}, \dots, x_{nj})$, $j = 1, \dots, LC$ z obszaru Ω .

zgodnie z rozkładem równomiernym, za oszacowanie całki przyjmuje się wyrażenie

$$\frac{|\Omega|}{LC} \sum_{j=1}^{LC} g(x_{1j}, x_{2j}, \dots, x_{nj}).$$

Ponieważ w rozważaniach w pracy przypadkach

$$|\Omega| = (b_1 - a_1) \times (b_2 - a_2) \times \dots \times (b_n - a_n) \quad , \text{ to losowane}$$

$x_{1j}, \dots, x_{10j}$ zgodnie z rozkładem równomiernym na Ω oblicza się w następujący sposób

$$x_{1j} = (b_1 - a_1) r_{1j} + a_1,$$

$$x_{nj} = (b_n - a_n) r_{nj} + a_n,$$

gdzie r_{ij} jest zmienną losową o rozkładzie równomiernym na odcinku $[0,1]$.

Dokładność wyznaczania oszacowań całki w istotny sposób zależy od liczby generowanych punktów LC. Im większa liczba LC tym dokładniejsze oszacowanie. Jeśli dla całki jednowymiarowej osiągane są zadowalające wyniki dla $LC = LC_1$, to dla n -wymiarowej całki z funkcji tego samego typu co dla całki jednowymiarowej liczba generowanych punktów powinna być rzędu $LC = LC_1^n$. Ze wzrostem wymiaru całki liczba generowanych punktów powinna szybko rosnąć.

Tabulogramy programów oraz ich dokładne opisy umieszczono w Dodatku. W niniejszym rozdziale przedstawione zostaną procedury selekcji i omówione krótko kolejne programy, przedstawione przykładowe wydruki oraz omówione niektóre wyniki.

Algorytm optymalnej selekcji w oparciu o ryzyko Bayes'a R^* realizują 2 procedury dla zadanej liczby cech M , którą trzeba odrzucić, aby wzrost ryzyka był minimalny.

1. SUBROUTINE SELECTIONMIS (NC, NCO, K, N, M, FUN, E2, W, ZR)

NC - jednowymiarowa tablica indeksów $NC = (1, 2, \dots, N)$

NCO- jednowymiarowa tablica indeksów odrzucanych cech

NCA- jednowymiarowa tablica indeksów pozostawionych cech

K - jednowymiarowa tablica robocza

N - liczba cech

M - liczba cech do odrzucenia

FUN- segment FUNCTION obliczania ryzyka

E2 - błąd obliczania ryzyka

W - wartość ryzyka po odrzuceniu M cech

ZR - tablica jednowymiarowa ryzyka odpowiadająca kolejnemu odrzucaniu cech.

Segment ten dokonuje wyboru pierwszego optymalnego rozwiązania $\bar{z}$.

2. Segment SUBROUTINE SELECTIONM (N, M, L1, K1, K2, KA, KOPT, RR, ROPT, FUN)

N - liczba cech /kilka tysięcy/

M - liczba cech do odrzucenia ($2 \leq M < N$)

L1 - liczba cech, którą należy pozostawić, $L1=N-M$

K1, K2 - tablice pomocnicze jednowymiarowe M-elementowe

KA - tablica pomocnicza " N-elementowa

KOPT - tablica indeksów cech, które należy pozostawić, jednowymiarowa, L1-elementowa

RR - tablica jednowymiarowa ryzyk uzyskanych w poprzednim segmencie

ROPT - wartość ryzyka po selekcji

FUN - segment FUNCTION obliczania ryzyka

Segment ten wyznacza rozwiązanie optymalne $\bar{z}$.

Działanie tych procedur obrazuje program Z009.

Poniżej umieszczono tabulogramy tych procedur, a następnie schemat blokowy programu Z009.

SUBROUTINE SELECTIONM1S (NC, NCO, NCA, K, N, M, FUN, E2, W, ZR)

DIMENSION NCO(N), NC(N), NCA(N), K(N), ZR(20)

1. DO 1 I=1, N

NCA(I)= NC(I)

DO 2 J=1, M

IN = N + 1 - J

W = 10

DO 3 I = 1, IN

CALL MISS(IN, I, NCA, K)

LAC = IN - 1


```
X = FUN (LAC, K, EPS)
IF(J, EQ, 1) ZR(I)= X
IF(W, LE, X) GO TO 3
W = X
E2 = EPS
INDEX = I
3 WRITE(2, 4) X, EPS,(K (J1) , J1 = 1, LAC )
4 FORMAT (5X, 26HPETLA 3, RYZYKO, EPS, K(I)/5X, 2E24, 11,
5X, 20I3 )
NCO(I)= NCA(INDEX)
WRITE(2, 5) W, NCA(INDEX)
5 FORMAT (5X, 26HPETLA 2, RYZMIN, NCA( INDEX), E24,.11, 15)
2 CALL MISS(IN, INDEX, NCA, NCA)
RETURN
END
```

SEGMENT SUBROUTINE SELECTIONM1S

Dopuszcza on N rzędu kilka tysięcy.

Segment SUBROUTINE SELECTIONRM zbudowany jest na podstawie a metody podziału i ograniczeń.

Dzięki pewnemu uporządkowaniu /operowania dwoma jednowymiarowymi tablicami/, segment ten umożliwia rozwiązywanie zadań o N rzędu nawet kilku tysięcy.

SUBROUTINE SELECTIONRM(N, M, L1, K1, K2, KA, KOPT, RR, ROPT, FUN)

DIMENSION K1(M), K2(M), KOPT (L1), RR(L1), KA(N)

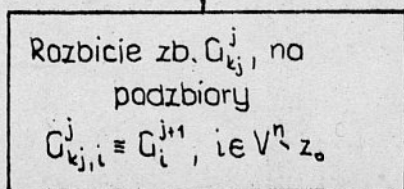
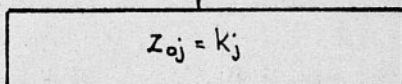
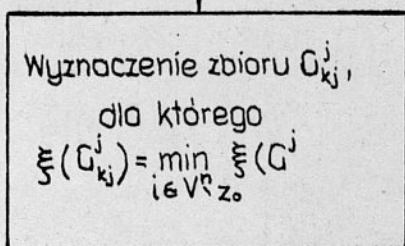
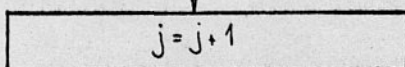
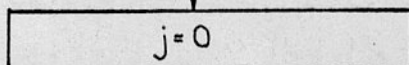
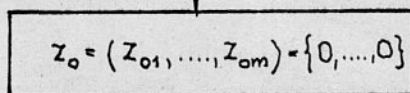
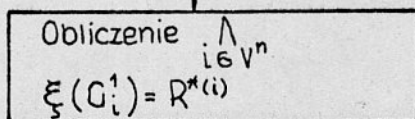
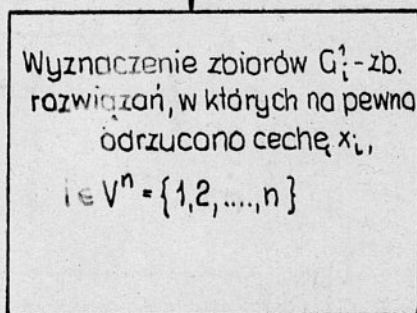
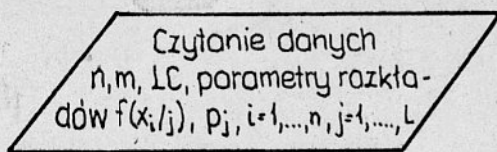
K1(1) = 1

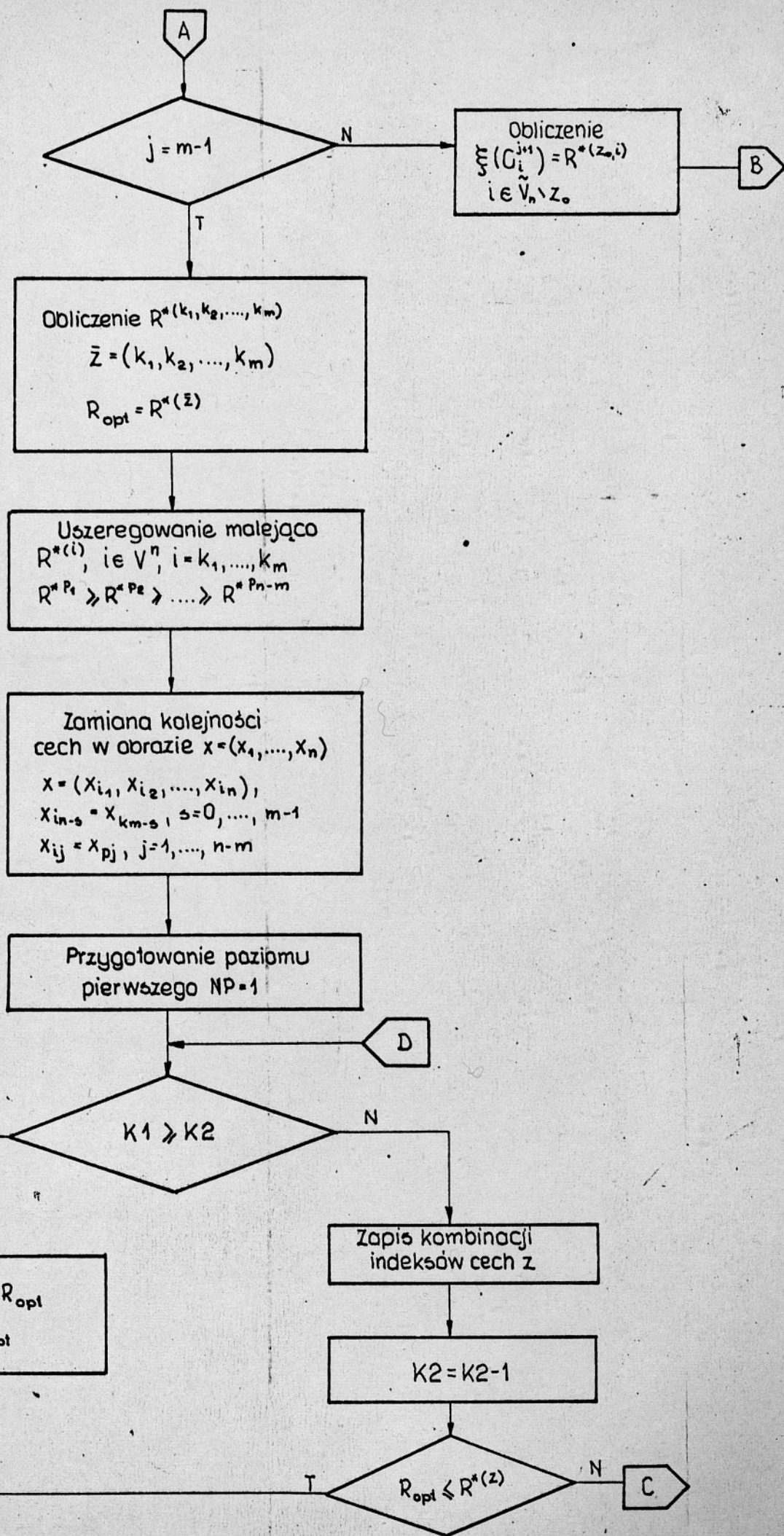
L, K2(1) = L1 + 1

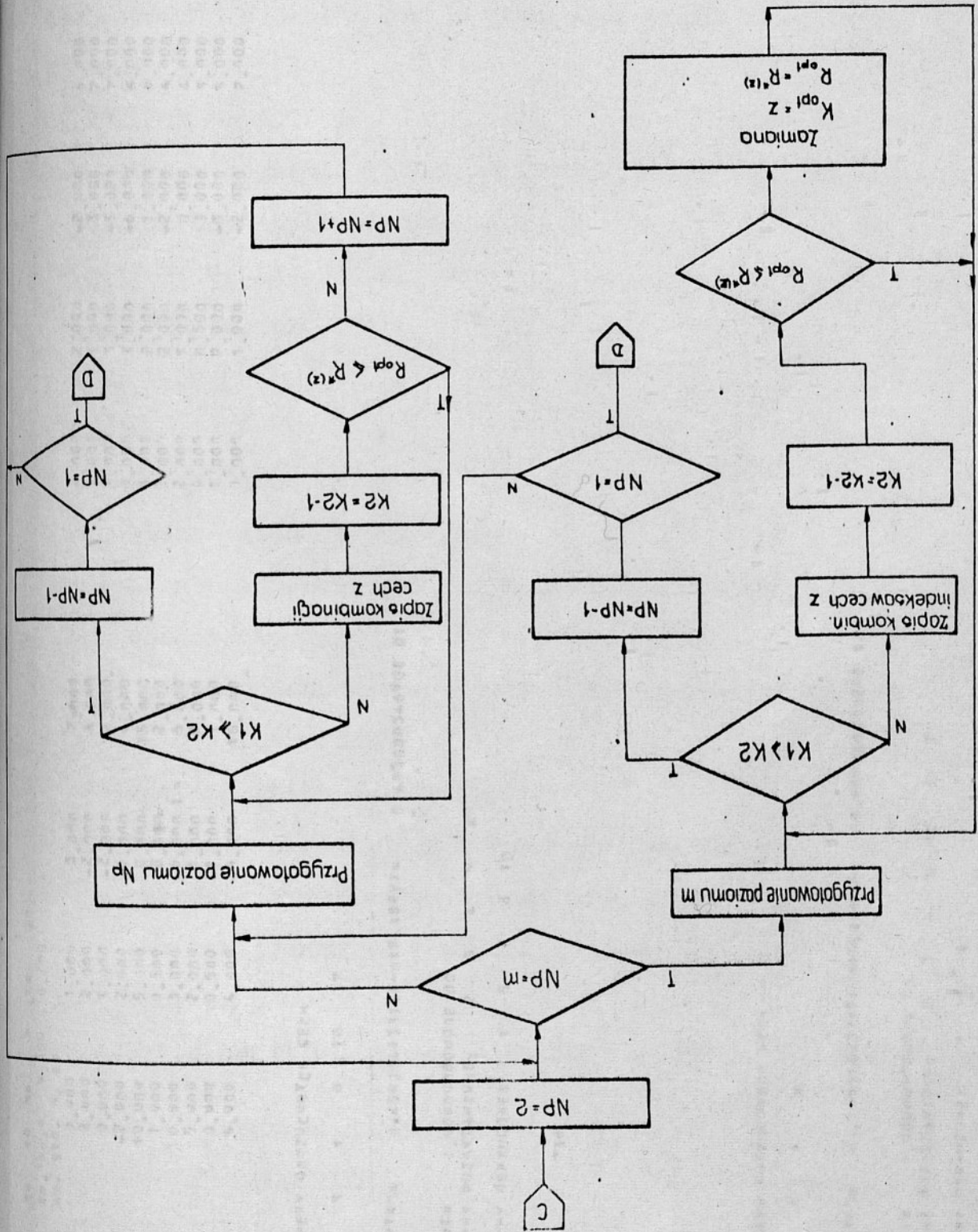
```
1  IF(K1(1), GE, K2(1)) RETURN
   J, K2(1) = K2(1) = 1
   IF(ROPT, IT, RR(J)) RETURN
   NP = 2
2  IF(NP, EQ, M) GO TO 6
3  K2(NP) = 1 + NP
   K1(NP) = K2(NP-1) + 1
4  IF(K1(NP), LT, K2(NP)) GO TO 5
   NP = NP - 1
   IF(NP, EQ, 1) GO TO 1
   GO TO 2
5  K2(NP) = K2(NP - 1)
   J, K, I, I1 = 1
10 I = I + 1
   IF(I, GT, N) GO TO 12
   IF(I, EQ, K2(K)) GO TO 11
   KA(I) = I
   I = J + 1
   GO TO 10
11 IF(K, EQ, NP) I1 = 0
   K = K + I1
   GO TO 10
12 K = N - NP
   WRITE(2, 20) K, (KA(K7), K7 = 1, K)
20 FORMAT(5X, 14HSELECTIONRM, K = I4/(20 I5))
   R = FUN(K, KA, EPS)
   IF(ROPT, LE, R) GO TO 4
   NP = NP + 1
   GO TO 2
```



```
6  K2(M)= L + M
   K1(M)= K2(M - 1) + 1
7  IF(K1(M), GE, K2(M))GO TO 8
   K2(M)= K2(M)- 1
   J, K, I, I1 = 1
13 I = I + 1
   IF(I, GT, N )GO TO 15
   IF(I, EQ, K2(K))GO TO 14
   KA(J)= I
   J = J + 1
   GO TO 13
14 IF(K, EQ, M) I1 = 0
   K = K + I1
   GO TO 13
15 R = FUN(L1, KA, EPS )
   WRITE(2, 20) L1,(KA( K7), K7 = 1, L1)
   IF(ROPT, LE, R )GO TO 7
   DO 16 I = 1, L1
16 KOPT(I)= KA(I)
   ROPT = R
   GO TO 7
8  NP = M - 1
   IF(NP, EQ, 1) GO TO 1
   GO TO 3
   RETURN
END
```







N= 10 M= 5 LC= 200
 DR, APR= 0.300
 RM, STG, A, B= 0.700

3.000	1.000	3.000	7.000	2.000	2.000	-2.000	6.000
2.000	2.000	-2.000	6.000	5.000	1.000	3.000	7.000
0.000	1.000	-2.000	2.000	1.000	3.000	-5.000	7.000
-2.000	2.000	-6.000	4.000	0.000	3.000	-6.000	6.000
10.000	5.000	0.000	15.000	5.000	2.000	1.000	0.000
1.000	0.500	0.000	2.000	2.000	2.000	-2.000	4.000
0.000	3.000	-6.000	6.000	2.000	1.000	0.000	4.000
5.000	2.000	1.000	9.000	4.000	0.500	3.000	5.000
0.000	0.500	-1.000	1.000	1.000	2.000	-3.000	5.000
4.000	4.000	-4.000	16.000	0.000	1.000	-2.000	7.000

NUMERY ODRZUCONYCH CECI

5 1 8 10 7

RYZYKO= 0.45623415315E-01BLADRYZ= 0.46500054690E 01

ROST= 0.00000000000E 00

CECHY POZOSTAWIONE: 2 3 4 9 6

CECHY ODRZUCONE: 1 5 7 8 10

ZAS = 736

= 3

NUMERY ODRZUCONYCH CECI

5 1 8

RYZYKO= -0.11051123719E 01BLADRYZ= 0.16992620024E 02

ROST= 0.00000000000E 00

CECHY POZOSTAWIONE: 2 3 4 9 10 6 7

CECHY ODRZUCONE: 1 5 8

ZAS = 570

Algorytm optymalnej selekcji cech w rozpoznawaniu obrazów, dla $L = 2$ oraz cechach niezależnych, dla zadania odrzucania m cech spośród n takich aby rozrost ryzyka był najmniejszy realizuje segment

SUBROUTINE SELECTION

Parametry z segmentu MASTER przenoszone są do tego segmentu w postaci wspólnych bloków

/CUV11/ N, M, LC, P1, P2, P

/CUVMaster/ O, OX, RO

/CUVR/ KA,

gdzie

N - liczba cech obrazu

M - liczba cech do odrzucenia

LC - liczba generowanych punktów w metodzie Monte-Carlo przy obliczaniu ryzyka

P1, P2 - prawdopodobieństwo a priori występowania pierwszej i drugiej klasy

P - wartość stała = $4 \cdot P1 \cdot P2$

O - parametr określający rodzaj nośnika danych

OX - parametr określający rodzaj urządzenia wyjścia

RO - tablica jednowymiarowa obliczonych ξ

KA - tablica pomocnicza

Segment ten umożliwia przeprowadzenie selekcji dla $N = 100, 2 \leq M < N$.

Tabulogram segmentu SELECTION

- SUBROUTINE SELECTION

INTEGER KOPT(100), KA(100), NRO(100), K(2707), O, OX

REAL R1(101), R(100), RO(101)

COMMON/CUV11/ N, M, LC, P1, P2, P


```
COMMON/CUVR/KA
COMMON/CUVMaster/O, OX, RO
EQUIVALENCE(R1 (1), K(1))
100 FORMAT(10X, 14HRO POCZATKOWE: /(1X, 10E12, 3))
IND = 1
DO 21 I = 1, N
21 NRO(I) = I
DO 22 I = 2, N + 1
22 R1(I) = RO(I) - 1
R1(1) = - 1
DO 27 I = 3, N + 1
J1 = 1
J2 = I
H = R1(I)
IH = NRO(I) - 1
23 J = J1 + J2 / 2
IF(J, EQ, J1) GO TO 25
IF(H, LE, R1(J)) GO TO 24
J1 = J
GO TO 23
24 J2 = J
GO TO 23
25 KK = J + 1
IL = I - KK
DO 26 L = 1, IL
J = I - L
R1(J) + 1 = R1(J)
26 NRO(J) = NRO(J) - 1
R1(KK) = H
```

```
NRO(KK - 1)= IH
27 CONTINUE
DO 28 I = 2, N + 1
28 RO(I - 1)= R1(I)
K(3), J = N - M
NC = J + 1
NC = J + 1
DO 1 I = 1, J
1 KA(I), KOPT (I)= NRO(I)
DO 2 I = NC, N
2 KA(I), KOPT(I)= 0
CALL RYZYKO(ROPT, EPS)
CALL ITIME (ITK)
ITP1 = ITK - ITP
K(2), LR, K2 = NC + 3
K(M*LR + 3) = 0
K(1)= 5
PI = 1
K4 = LR + 1
DO 3 I = 1, NC
PI = PI*RO ( I )
R(1)= PI
DO 4 I = 1, J
F = PI/RO(KIB)
IF((1 - SQRT(1.-P*F)) , LT. ROPT ) GO TO 4
K(1), K1 = IB + 1
IF(K1, GT. LR ) GO TO 19
GO TO 5
4 CONTINUE
```



```
5 NP = 2
6 IP = (NP - 1) * LR
  K3L = K(IP - LR + 3)
  L = J + NP - K3L
  K(IP + 2), K2 = IP + LR
  K(IP + 1), K1 = K2 + 1 - L
  K(IP + 3), K3 = K3L + L
  NC = K3 + 1
  K4 = K2 + 1
  R(NP), PI = R(NP - 1) * RO(N - M + NP) / RO(K3L)
  IF(NP, EQ, M) GO TO 10
  DO 9 I = 1, L
  IB = K4 - I
  K(IB), KIB = NC - I
  F = PI / RO(KIB)
  IF((1. - SORT(1. - P * F)), LT, ROPT) GO TO 9
  K(IP + 1), K1 = IB + 1
  NP = NP + 1
  IF(K1, LE, K2) GO TO 6
  NP = NP - 1
7 NP = NP - 1
  I2 = (NP - 1) * LR + 2
  K(I2), K2 = K(I2) - 1
  IF(K2, GE, K(I2 - 1)) GO TO 8
  IF(NP, EQ, 1) GO TO 19
  GO TO 7
8 K(I2 + 1) = K(K2)
  NP = NP + 1
  GO TO 6
```

```
9      CONTINUE
      NP = NP + 1
      GO TO 6
10     DO 11 I = 1, L
      IB = K4 - I
      K(IB), KIB = NC - I
      F = PI/RO(KIB)
      IF((1.-SQRT(1.-P*F)).GE, ROPT) GO TO 12
11     CONTINUE
      GO TO 13
12     K(IP + 1), K1 = IB + 1
      IF (K1, GT, K2) GO TO 7
13     INDEX = 3
      DO 14 I = 1, N
      INDEX = INDEX + LR
      KA(I) = 0
14     CONTINUE
      CALL RYZYKO(RA, EPS)
      IND = IND + 1
      K(IP + 2), K2, I2 = K2 - 1
      K(IP + 3), K3, I3 = K3 - 1
      IF(RA, LT, ROPT) GO TO 15
      IF(K1, GT, K2) GO TO 7
      GO TO 13
15     ROPT = RA
      DO 16 I = 1, N
16     KOPT(I) = KA I
      IF(I2, LT, K1) GO TO 7
17     F = PI/RO(I3)
      IF((1.-SQRT(1.-P*F)).GE, ROPT) GO TO 18
```



```

I2 = I2 - 1
I3 = I3 - 1
IF(I2, GE, K1) GO TO 17
GO TO 13
18 K(IP + 1), K1 = I2 + 1
IF(K1, GI, K (IP + 2)) GO TO
GO TO 13
19 CALL ITIME ITK
RETURN
END

```

Działanie tej procedury obrazuje program P002

Program ten uruchomiono na przykładzie

$n = 10, m = 5, f(x/1) = N(m_{1j}, \sigma_{1j})$.

Dla funkcji gęstości postaci $N(m, \sigma)$ współczynnik Bhattacharyya dla każdej funkcji gęstości da się wyznaczyć w sposób analityczny.

A mianowicie

$$\xi_1 = \int_{-\infty}^{\infty} \sqrt{f(x_1/1) f(x_1/2)} dx_1 = \int_{-\infty}^{\infty} \sqrt{N(m_{11}, \sigma_{11}) N(m_{12}, \sigma_{12})} dx_1.$$

Oznaczając $x_1 \rightarrow x$, $m_{i1} \rightarrow m_1$, $m_{i2} \rightarrow m_2$, $\sigma_{i1} \rightarrow \sigma_1$,
 $\sigma_{i2} \rightarrow \sigma_2$ oraz funkcję pod pierwiastkiem

przez Λ

$$\Lambda = N(m_1, \sigma_1) \cdot N(m_2, \sigma_2)$$

otrzymuje się

$$S_1 = \int_{-\infty}^{\infty} \sqrt{\Lambda} dx_1$$

W dalszym ciągu rozważać się będzie wyrażenie Λ .

$$\Lambda = \frac{1}{2\pi\sigma_1\sigma_2} \exp \left\{ -\frac{1}{2} \left[\frac{(x - m_1)^2}{\sigma_1^2} + \frac{(x - m_2)^2}{\sigma_2^2} \right] \right\}.$$

Wyrażenie znajdujące się pod eksponentem da się przekształcić do prostszej postaci

$$\begin{aligned} \frac{(x - m_1)^2}{2\sigma_1^2} + \frac{(x - m_2)^2}{2\sigma_2^2} &= \frac{1}{2} \frac{\sigma_2^2 x^2 - 2m_1 \sigma_2^2 x + m_1^2 \sigma_2^2 + \sigma_1^2 x^2 - 2m_2 \sigma_1^2 x + m_2^2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2} \\ &= \frac{1}{2} \frac{x^2(\sigma_2^2 + \sigma_1^2) - 2x(m_1 \sigma_2^2 + m_2 \sigma_1^2) + m_1^2 \sigma_2^2 + m_2^2 \sigma_1^2}{\sigma_1^2 + \sigma_2^2} = \\ &= \frac{x^2 - 2x \frac{m_1 \sigma_2^2 + m_2 \sigma_1^2}{\sigma_2^2 + \sigma_1^2} + \frac{m_1^2 \sigma_2^2 + m_2^2 \sigma_1^2}{\sigma_2^2 + \sigma_1^2}}{\frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2}} \end{aligned}$$

Oznaczając

$$\frac{m_1 \sigma_2^2 + m_2 \sigma_1^2}{\sigma_2^2 + \sigma_1^2} = m_0,$$

$$\begin{aligned}
 & \text{to} \quad (x^2 - 2xm_0 + m_0^2) + \frac{m_1^2 \sigma_2^2 + m_2^2 \sigma_1^2}{\sigma_2^2 + \sigma_1^2} - m_0^2 \\
 & = \frac{(x-m_0)^2 + \frac{m_1^2 \sigma_2^4 + m_2^2 \sigma_1^4}{\sigma_2^2 + \sigma_1^2} + \frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2} - m_0^2}{\frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2}} = \\
 & = \frac{(x-m_0)^2 + \frac{m_1^2 \sigma_2^4 + m_2^2 \sigma_1^4}{\sigma_2^2 + \sigma_1^2} + \frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2} - m_0^2}{\frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2}} =
 \end{aligned}$$

$$= \frac{(x-m_0)^2}{\frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2}} + \frac{\sigma_1^2 \sigma_2^2 (m_2 - m_1)^2}{\sigma_2^2 + \sigma_1^2} \frac{\sigma_2^2 \sigma_1^2}{2\sigma_1^2 \sigma_2^2} =$$

$$= \frac{(x-m_0)^2}{\frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2}} + \frac{(m_2 - m_1)^2}{2(\sigma_2^2 + \sigma_1^2)} .$$

Oznaczając

$$\frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2} = \sigma_0^2 ,$$

$$A = \frac{1}{2\pi \sigma_1 \sigma_2} \exp \left[- \frac{(m_2 - m_1)^2}{2(\sigma_2^2 + \sigma_1^2)} \right] \exp \left[- \frac{(x - m_0)^2}{\sigma_0^2} \right] .$$

Natomiast

$$\sqrt{A} = \frac{1}{\sqrt{2\pi \sigma_1 \sigma_2}} \exp\left[-\frac{(m_2 - m_1)^2}{4(\sigma_2^2 - \sigma_1^2)}\right] \exp\left[-\frac{(x - m)^2}{2\sigma_0^2}\right].$$

Przekształcając w dalszym ciągu, wyrażenie $\sqrt{A}$ można zapisać w postaci

$$\sqrt{A} = B \cdot N(m_0, \sigma_0), \text{ gdzie}$$

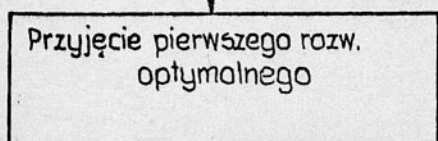
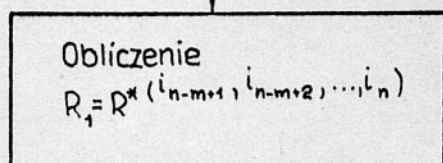
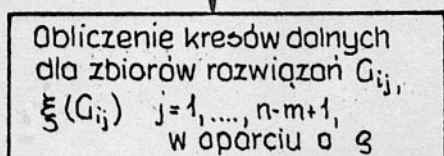
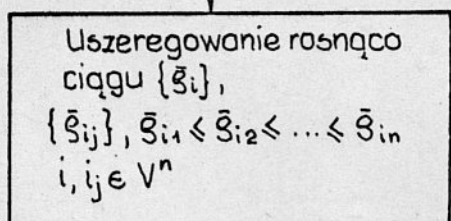
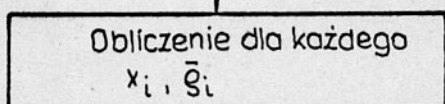
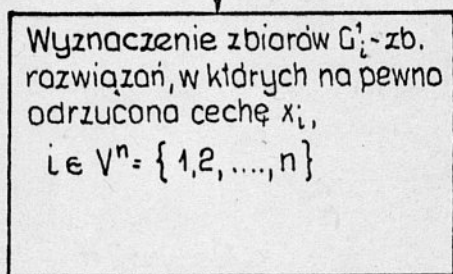
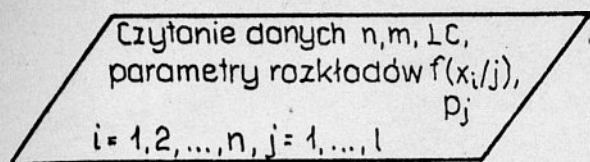
$$\begin{aligned} B &= \frac{1}{\sqrt{\sigma_1 \sigma_2}} \cdot \sqrt{\frac{2\sigma_1^2 \sigma_2^2}{\sigma_2^2 + \sigma_1^2}} \exp\left[-\frac{(m_2 - m_1)^2}{4(\sigma_2^2 - \sigma_1^2)}\right] = \\ &= \frac{\sqrt{2} \sigma_1 \sigma_2}{\sqrt{\sigma_2^2 + \sigma_1^2}} \exp\left[-\frac{(m_2 - m_1)^2}{4(\sigma_2^2 - \sigma_1^2)}\right]. \end{aligned}$$

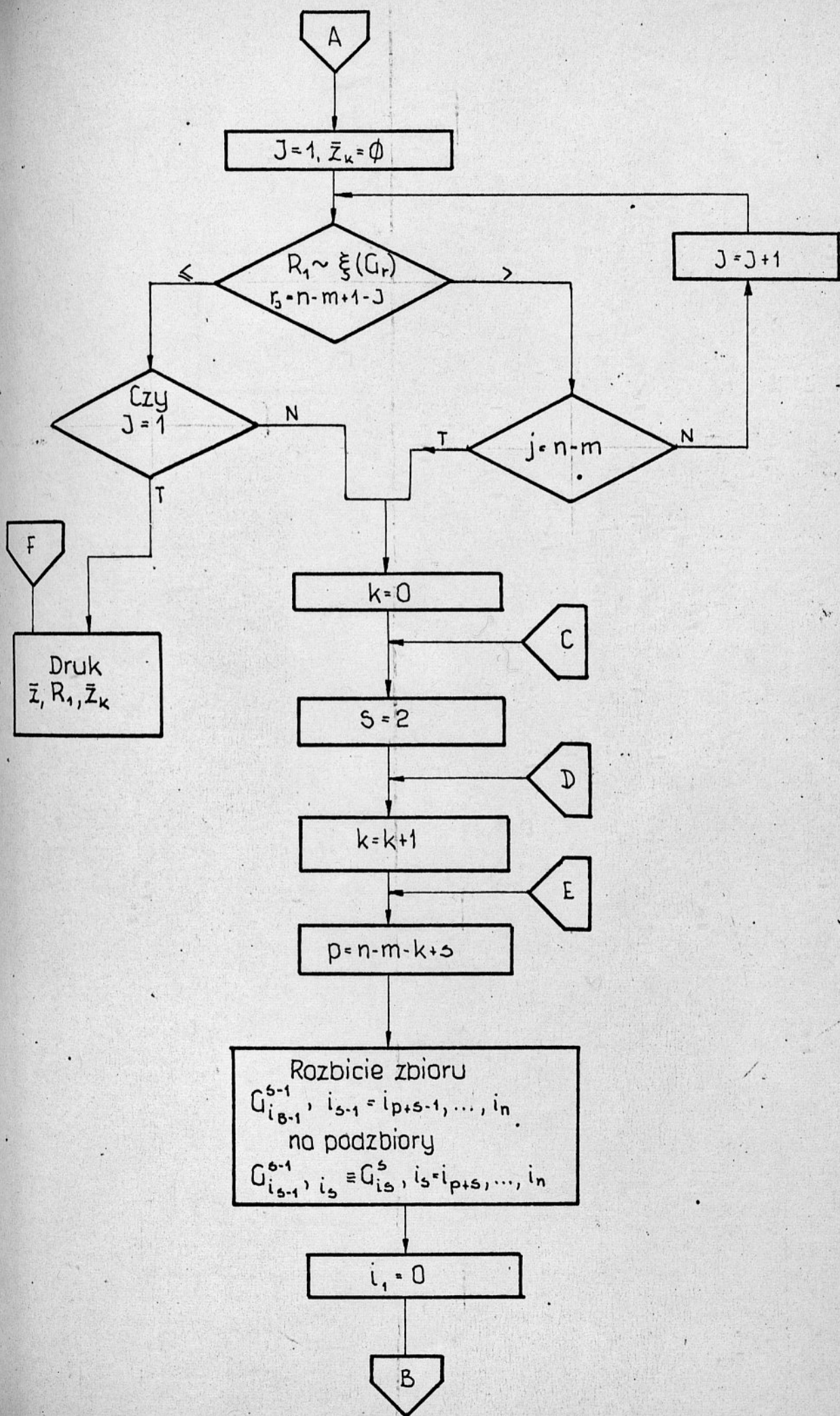
A zatem

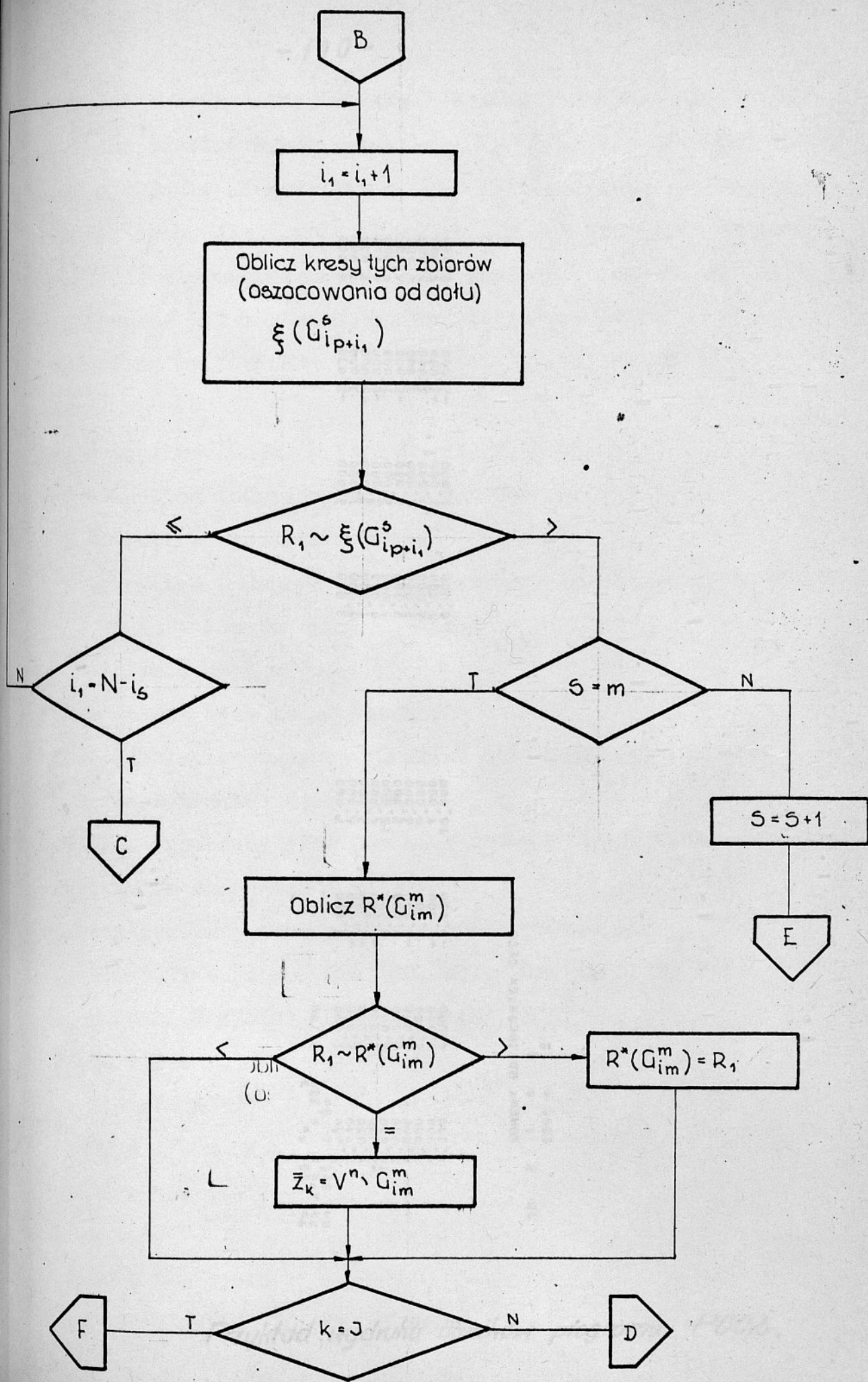
$$\varrho_1 = \int_{-\infty}^{\infty} \sqrt{A} \, dx = \int_{-\infty}^{\infty} B \cdot N(m_0, \sigma_0) \, dx = B,$$

$$\varrho_1 = \sqrt{\frac{2}{\sigma_{12}^2 + \sigma_{11}^2}} \cdot \sigma_{11} \sigma_{12} \cdot \exp\left[-\frac{(m_{12} - m_{11})^2}{4(\sigma_{12}^2 - \sigma_{11}^2)}\right].$$

W programie PO02 wykorzystano tę postać dla wyznaczenia wartości współczynnika ϱ_1 .







V= 10
PR. APR = 0.300
R4, SIG A B=

0.700

5.000
2.000
0.000
-2.000
10.000
1.000
0.000
5.000
0.000
4.000

1.000
2.000
1.000
0.000
0.500
3.000
2.000
0.500
4.000

3.000
-2.000
-2.000
-6.000
0.000
0.000
-6.000
1.000
-1.000
-4.000

7.000
6.000
2.000
4.000
15.000
2.000
6.000
0.000
1.000
16.000

2.000
5.000
1.000
0.000
5.000
2.000
2.000
4.000
1.000
0.000

2.000
1.000
3.000
2.000
1.000
0.500
2.000
1.000

-2.000
3.000
-5.000
-6.000
1.000
-2.000
0.000
3.000
-3.000
-2.000

6.000
7.000
6.000
6.000
6.000
4.000
4.000
5.000
5.000
2.000

NUMERY ODRZUCONYCH CECH
10 2 1 6 3
CZAS = 5

Algorytm suboptymalnej selekcji względem ryzyka, ale optymalnej względem zdefiniowanej miary $M_K(I/X)$ dla dowolnej liczby cech w obrazie (N -może być = 5000), dla zależnych lub niezależnych cech, dla dowolnych rozkładów spełniających warunki (4.3.10), dla zadania odrzucania spośród N cech M , aby wzrost oszacowania był najmniejszy, realizuje procedura

SUBROUTINE SELECTIONM1 (NC, NCO, NCA, K, N, M, FUN)

gdzie

NC - tablica indeksów cech = (1, 2, ..., N)

NCO - tablica indeksów cech do odrzucenia

NCA - tablica indeksów robocza

K - tablica indeksów robocza /zawiera indeksy cech na których należy liczyć miarę $M_K(I/X)$

N - liczba cech w obrazie

M - liczba cech do odrzucenia

FUN - wywołanie segmentu FUNCTION obliczającego wartości odpowiednich oszacowań

Wynikiem procedury jest parametr formalny NCO, tablica ta jest przenoszona do segmentu głównego.

Poniżej przedstawiono tabulogram tej procedury

SUBROUTINE SELECTIONM1(NC, NCO, NCA, K, N, M, FUN)

DIMENSION NCO(N), NC(N), NCA(N), KCN)

DO 1 I = 1, N

1 NCA(I) = NC(I)

DO 2 J = 1, M

IN = N + 1 - J

W = 0

DO 3 I = 1, IN

CALL MISS(IN, I, NCA, K)

```
LAC = IN - 1
X = FUN(LA, K, EPS)
IF (W.GT.X) GO TO 3
W = X
INDEX = 1
3  CONTINUE
   NCO(J) = NCA(INDEX)
2  CALL MISS (IN, INDEX, NCA, NCA)
   RETURN
END
```

Działanie tej procedury obrazuje program Z005

E= 0.2000000000E 00				N,M,LC,NP=				10	5	200	8				
PR AP= 0.300 0.700															
RM, SIG, A, B=															
5.000	1.000	3.000	7.000	2.000	2.000	-2.000	4.000								
2.000	2.000	-2.000	6.000	5.000	1.000	3.000	7.000								
0.000	1.000	-2.000	2.000	1.000	3.000	-5.000	7.000								
-2.000	2.000	-6.000	4.000	0.000	3.000	-6.000	4.000								
10.000	5.000	0.000	15.000	5.000	2.000	1.000	5.000								
1.000	0.500	0.000	2.000	2.000	2.000	-2.000	4.000								
0.000	3.000	-6.000	6.000	2.000	1.000	0.000	4.000								
5.000	2.000	1.000	9.000	4.000	0.500	3.000	4.000								
0.000	0.500	-1.000	1.000	1.000	2.000	-3.000	5.000								
4.000	4.000	-4.000	16.000	0.000	1.000	-2.000	7.000								

A1,B1 =															
1 -2.000	4.000	2 -2.000	9.000	3 -5.000	12.000	4 -6.000	12.000	5 0.000	15.000						
6 -2.000	0.000	7 -6.000	12.000	8 1.000	8.000	9 -3.000	8.000	10 -4.000	20.000						

1K(J/X)= 0.00351522141E 00 EPS= 0.29960976262E 01

NUMERY DOKRZUCONYCH CECI

5 1 3 10 7

CZAS = 732

M = 6

HK(J/X) = 0,84402701510E 00Eps = 0,22270694741E 01

NUMERY ODRZUCONYCH CECI

5 1 8 10 7 0

CZAS = 785

M = 7

HK(J/X) = 0,75878904916E 00Eps = 0,10359039327E 01

NUMERY ODRZUCONYCH CECI

5 1 8 10 7 9 3

CZAS = 820

M = 8

HK(J/X) = 0,82429248004E 00Eps = 0,70209437109E 00

NUMERY ODRZUCONYCH CECI

5 1 8 10 7 9 3 4

CZAS = 838

M = 9

HK(J/X) = 0,9260172020E 00Eps = 0,70748327461E 00

NUMERY ODRZUCONYCH CECI

5 1 8 10 7 9 3 4 6

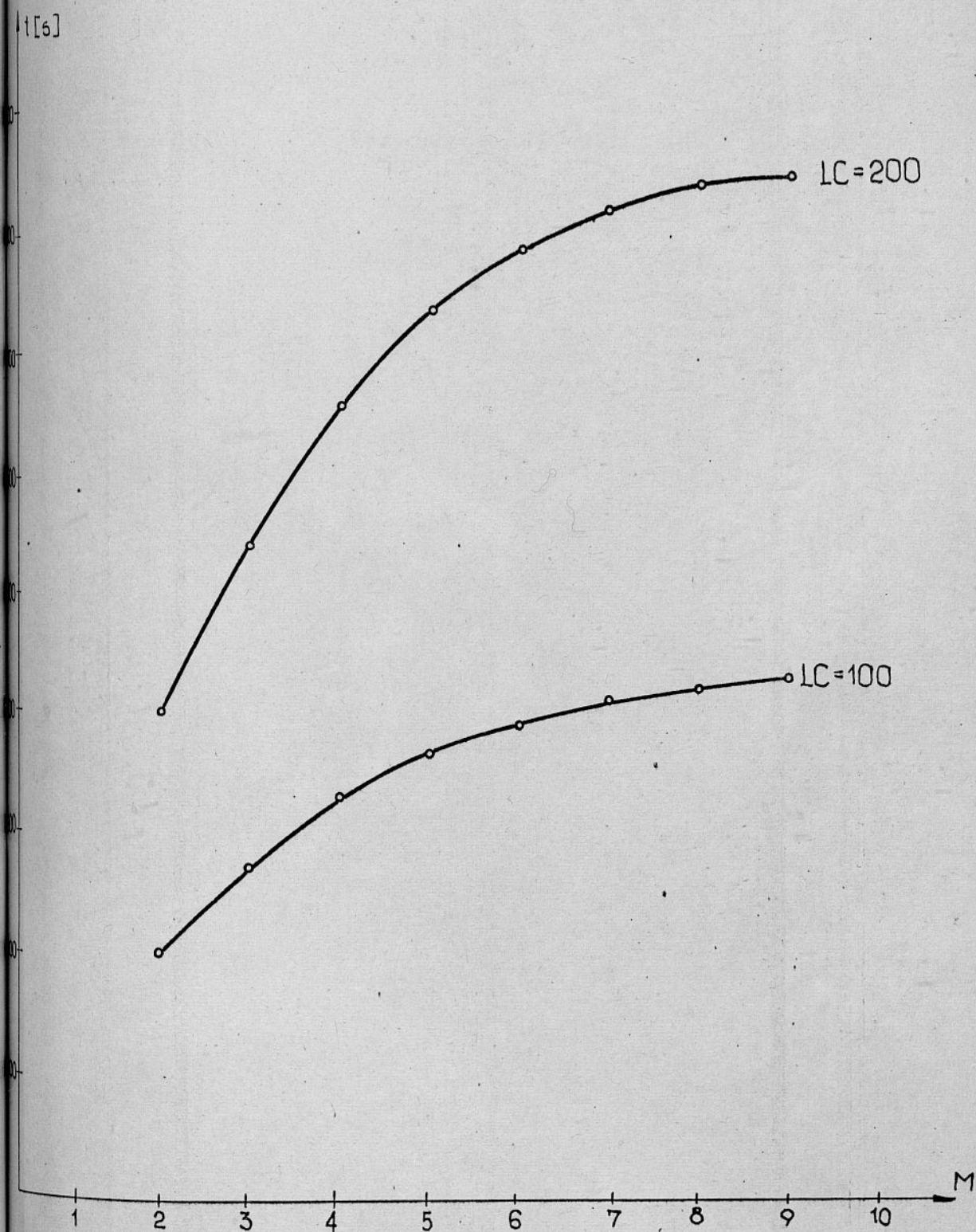
CZAS = 845

Lp.	Liczba M	Czas obliczeń [s]	Numery odrzuconych cech									
1	2	204	5	10								
2	3	276	5	10	7							
3	4	330	5	10	7	3						
4	5	370	5	10	7	3	1					
5	6	396	5	10	7	3	1	9				
6	7	413	5	10	7	3	1	9	6			
7	8	423	5	10	7	3	1	9	6	4		
8	9	427	5	10	7	3	1	9	6	4	8	

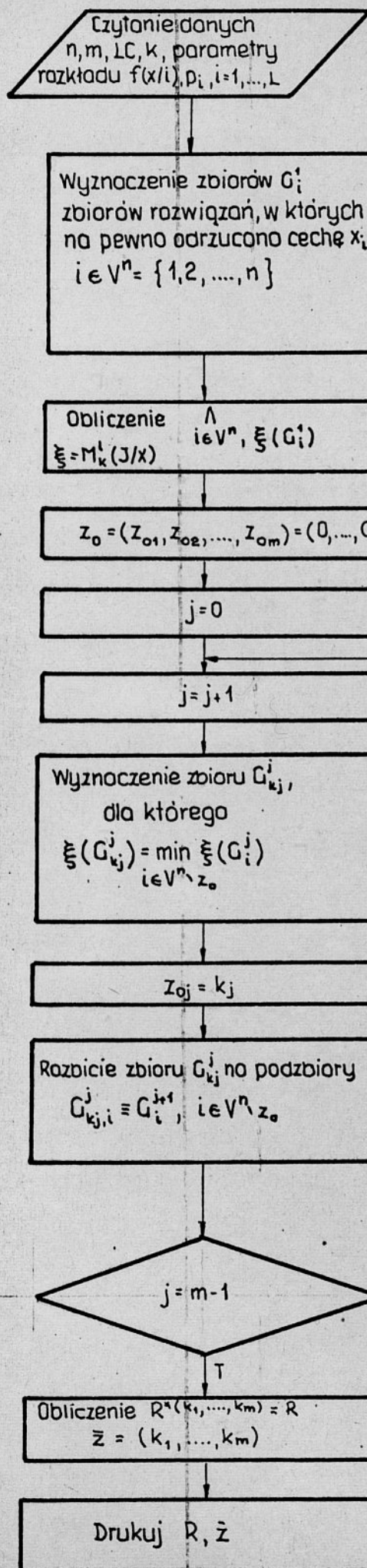
Tab.5.1. Czas obliczeń oraz numery wyselekcjonowanych cech wg programu Z005 /dla LC = 100 /

Lp.	M	Czas [s]	Numery odrzuconych cech									
1	2	404	5	1								
2	3	546	5	1	8							
3	4	653	5	1	8	10						
4	5	732	5	1	8	10	7					
5	6	785	5	1	8	10	7	9				
6	7	820	5	1	8	10	7	9	3			
7	8	838	5	1	8	10	7	9	3	4		
8	9	845	5	1	8	10	7	9	3	4	6	

Tab.5.2. Czas obliczeń oraz numery wyselekcjonowanych cech wg programu Z005 /dla LC = 200/



rys. 5.3. ZALEŻNOŚĆ CZASU OBLICZEŃ OD LICZBY ODRZUCONYCH CECH M .



Algorytm suboptymalnej selekcji względem ryzyka ale optymalnej względem zdefiniowanej miary $M_K(J/X)$ dla dowolnej liczby cech w obrazie dla zależnych bądź niezależnych cech, dla dowolnych rozkładów, dla zadania odrzucania maksymalnej liczby cech takich aby wzrost oszacowania nie przekroczył zadanej liczby ϵ , realizuje segment

SUBROUTINE SELECTIONM2(NC, NCO, NCA, N, E, FUN, K1)

gdzie

NC - tablica jednowymiarowa indeksów ($NC = (1, 2, \dots, N)$)

NCO - tablica cech odrzuconych

NCA - tablica robocza, zawierająca indeksy cech na których należy liczyć odpowiednie $M_K(J/X)$

K - tablica robocza

N - liczba cech w obrazie

E - zadana liczba o którą może wzrosnąć oszacowanie

FUN - procedura FUNCTION obliczająca odpowiednie wartości $M_K(J/X)$

K1 - liczba cech odrzuconych

Wynikiem segmentu są parametry formalne K1 oraz NCO, przenoszone do segmentu głównego.

Poniżej przedstawiono tabulogram tej procedury

Tabulogram segmentu SELECTIONM2

SUBROUTINE SELECTIONM2 (NC, NCO, NCA, K, N, E, FUN, K1)

DIMENSION NCO(N), NC(N), NCA(N), K(N)

K1 = 0

DO 1I = 1, N

1 NCA(I) = NC(I)

W1 = FUN(N, NCA, EPS)

WRITE(2, 4) W1, EPS


```
4  FORMAT (/5X, 9HMK(IJX)= E24.11, 5X, 5HEPS = E24.11/)
    N1 = N - 1
    DO 2 J = 1, N1
    IN = N + 1 - J
    W = 0
    DO 3 I = 1, IN
    CALL(MISS IN , I, NCA, K)
    LAC = IN - 1
    X = FUN(LAC, K, EPS)
    IF(W, GT, X) GO TO 3
    W = X
    INDEX = 1
3  CONTINUE
    IF((W1 - W), GT, E) RETURN
    K(1) = K1 + 1
    NCO(J) = NCA(INDEX)
2  CALL MISS(IN, INDEX, NCA, NCA)
    RETURN
    END
```

Działanie tej procedury obrazuje program Z006.

Przykładowy wydruk wyników programu Z006

E= 0.20000000000E 00 M,M,LC,NP= 10 5 200 8
 PR.APR.= 0.300 0.700
 RM, SIG, A, B=

5.000	1.000	3.000	7.000	2.000	2.000	-2.000	6.000
2.000	2.000		6.000	5.000	1.000	-3.000	7.000
0.000	1.000		2.000	1.000	3.000	-5.000	7.000
-2.000	2.000		4.000	0.000	3.000	-6.000	6.000
10.000	5.000	0.000	5.000	5.000	2.000	-1.000	6.000
1.000	0.500	0.000	2.000	2.000	2.000	-2.000	6.000
0.000	3.000	-6.000	6.000	2.000	1.000	0.000	4.000
5.000	2.000	1.000	9.000	4.000	0.500	-3.000	5.000
0.000	0.500	-1.000	1.000	1.000	2.000	-3.000	5.000
4.000	4.000	-4.000	16.000	0.000	1.000	-2.000	2.000

A1,B1 =

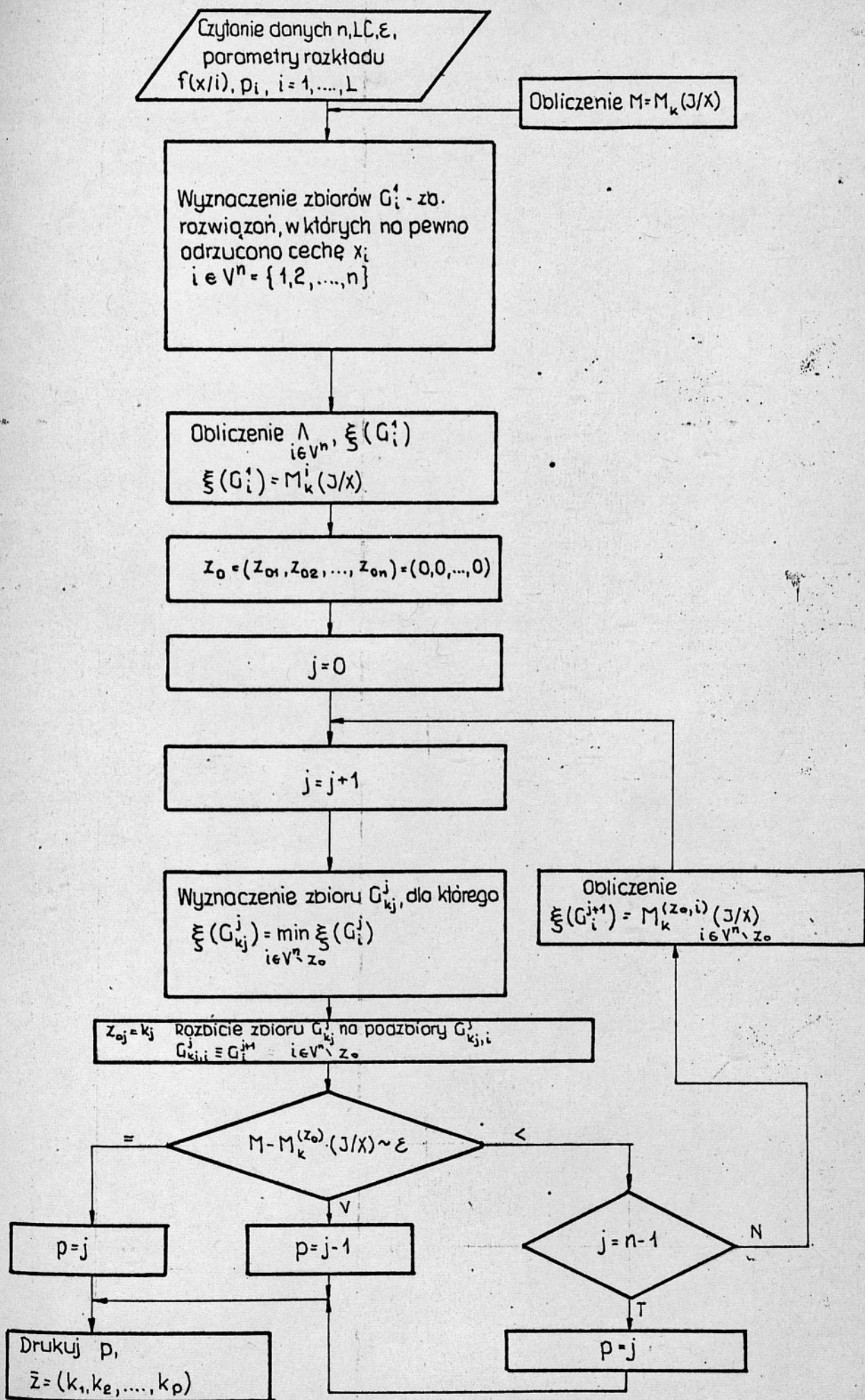
1	-2.000	9.000	2	-2.000	9.000	3	-5.000	12.000	4	-6.000	12.000	5	0.000	15.000
6	-2.000	8.000	7	-6.000	12.000	8	1.000	8.000	9	-3.000	8.000	10	-4.000	20.000

MK(JIX) = 0.29585231028E 00 EPS = 0.22463930979E 01

LICZBA ODRZ. CECH: 9

NUMERY ODRZUCONYCH CECH: 5 1 8 10 7 9 3 4 6

= 893



Algorytm suboptymalnej selekcji na podstawie oszacowania $\bar{\sigma}$ dla dowolnej liczby cech N ($N \sim 5000$), dla niezależnych cech $L=2$, dla dowolnych rozkładów, dla zbadanej liczby cech M do odrzucenia takich, aby wzrost ryzyka był najmniejszy realizuje segment

SUBROUTINE WYBORMNAIM(CR, M, N, K, X),

gdzie

- R - tablica obliczonych wcześniej /w MASTERZE/ oszacowań σ_{0i} dla każdej cechy, metodą F4INTSMP
M - liczba cech do odrzucenia
N - liczba cech obrazu
K - tablica odrzuconych cech
X - wartości oszacowań dla cech o indeksach zawartych w macierzy K

Wynikiem segmentu są tablice K oraz X przenoszone do segmentu głównego.

Poniżej przedstawiono tabulogram tego segmentu.

SUBROUTINE WYBORMNAIM (R, M, N, K, X)

DIMENSION R(N), X(M), K(M), N2(1000)

DO 1 J = 1, M

X(J) = R(J)

1 K(J) = J

DO 2 I = M + 1, N

DO 2 L = 1, M

IF(R(I) .GT. X(L)) GO TO 2

XK = X (L)

X(L) = R(I)

R(I) = XK

J = K(L)

K(L)= N2(I)

N2(I) = J

2 CONTINUE

RETURN

END

Działanie tej procedury obrazuje program Z007.

E=*.1000000E-02 IND= 3 N= 10M= 5

RM,SIG,A, B

5.000	1.000	3.000	7.000	2.000	2.000	-2.000	6.000
2.000	2.000	-2.000	6.000	5.000	1.000	3.000	7.000
0.000	1.000	-2.000	2.000	1.000	3.000	-5.000	7.000
-2.000	2.000	-6.000	4.000	0.000	3.000	-6.000	6.000
10.000	5.000	0.000	15.000	5.000	2.000	1.000	9.000
1.000	0.500	0.000	2.000	2.000	2.000	-2.000	6.000
0.000	3.000	-6.000	6.000	2.000	1.000	0.000	4.000
5.000	2.000	1.000	9.000	4.000	0.500	3.000	5.000
0.000	0.500	-1.000	1.000	1.000	2.000	-3.000	5.000
4.000	4.000	-4.000	16.000	0.000	1.000	-2.000	2.000

EPS= 0.200000000000E 00

NUMERY CECH ODRZUC: WARTOSCI OSZACOWAN

4 0.66233E 00 3 0.96774E 00 5 0.10371E 01 7 0.10902E 01 6 0.11976E 01

ZAS = 13

M= 2

NUMERY CECH ODRZUC: WARTOSCI OSZACOWAN

4 0.66233E 00 3 0.96774E 00

ZAS = 12

M= 3

NUMERY CECH ODRZUC: WARTOSCI OSZACOWAN

4 0.66233E 00 3 0.96774E 00 5 0.10371E 01

ZAS = 13

M= 4

NUMERY CECH ODRZUC: WARTOSCI OSZACOWAN

4 0.66233E 00 3 0.96774E 00 5 0.10371E 01 7 0.10902E 01

ZAS = 12

M= 6

NUMERY CECH ODRZUC: WARTOSCI OSZACOWAN

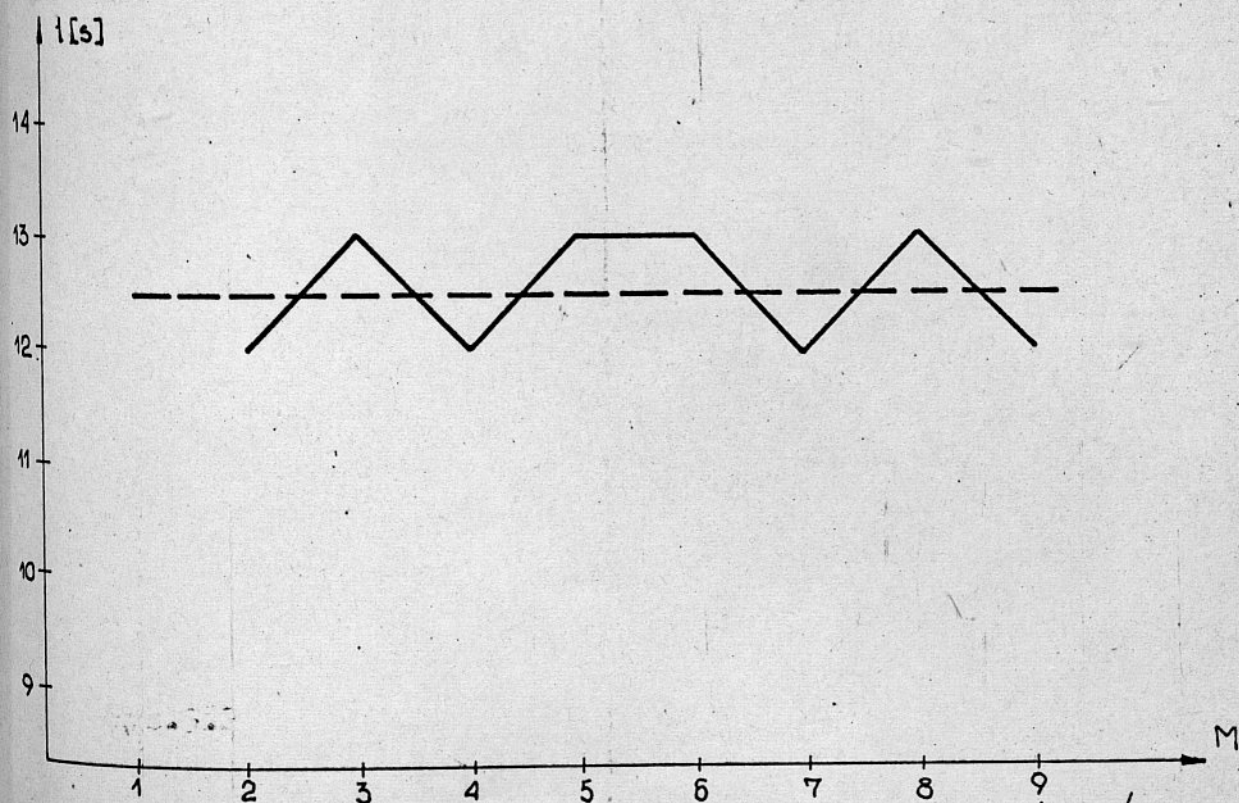
4 0.66233E 00 3 0.96774E 00 5 0.10371E 01 7 0.10902E 01 6 0.11976E 01

3 0.11976E 01

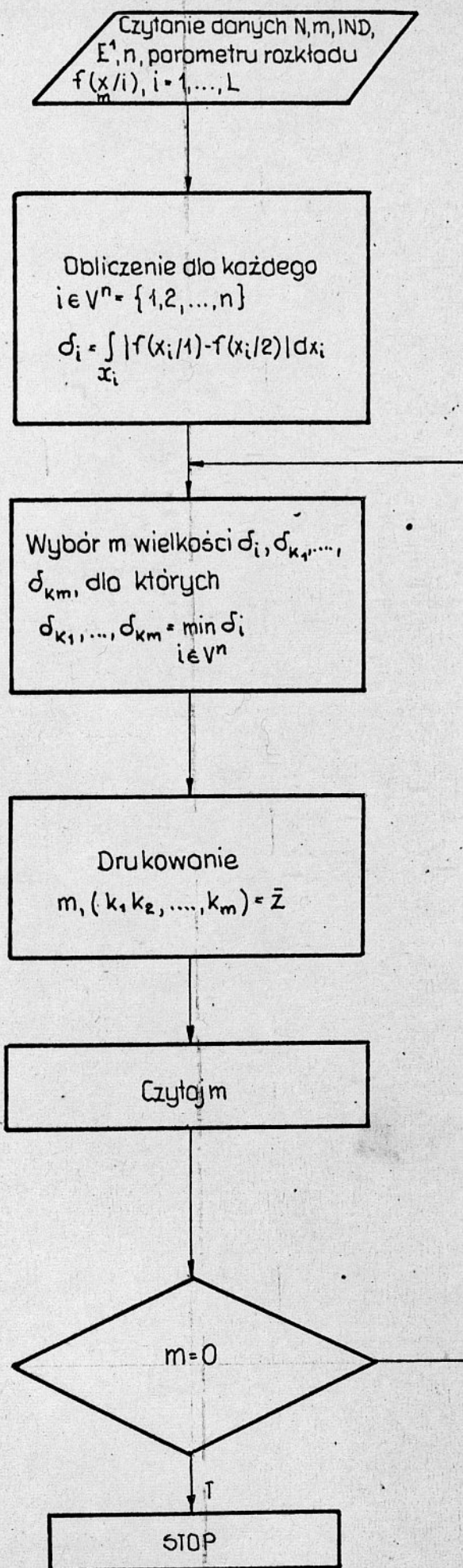
ZAS = 13

Lp.	Liczba odrzuconych cech M	Czas [s]	Numery odrzuconych cech									
1	2	12	4	3								
2	3	13	4	3	5							
3	4	12	4	3	5	7						
4	5	13	4	3	5	7	6					
5	6	13	4	3	5	7	6	8				
6	7	12	4	3	5	7	6	8	9			
7	8	13	4	3	5	7	6	8	9	2		
8	9	12	4	3	5	7	6	8	9	2	1	

Tab.5.3. Czas obliczeń oraz numery odrzuconych cech wg programu Z007



Rys. 5.6. Zależność czasu obliczeń od liczby odrzucanych cech.



Rys. 5.7. Schemat blokowy programu Z007

Algorytm suboptymalnej selekcji względem ryzyka w oparciu o oszacowanie $\bar{\sigma}$, dla dowolnej liczby cech N ($N \sim 5000$) niezależnych dla dowolnych rozkładów, $L = 2$, dla zadania odrzucania maksymalnej liczby cech takich aby wzrost ryzyka nie przekroczył ε realizuje segment

SUBROUTINE SELEKCJAM3(M, K, RO)

gdzie

M - tablica indeksów cech odpowiadająco uporządkowanej tablicy wartości RO

K - liczba odrzuconych cech

RO - tablica uporządkowanych rosnąco wartości $\bar{\sigma}_{0i}$ obliczonych w segmencie MASTER za pomocą procedury bibliotecznej F4INTSMP

Wynikiem procedury jest macierz M oraz liczba K przenoszona do masteru.

Tabulogram procedury

SUBROUTINE SELEKCJAM3(M, K, RO)

DIMENSION M(20), P(2), A1(20), B1(20)

COMMON/ALA/EPS, N, IC, P, A1, B1, E

K = 0

1 K = K + 1

CALL OBLRYZ(K, M, R2, E1)

IF((RO - R).GT.EPS) GO TO 2

IF(K.EQ.(N - 1)) GO TO 3

GO TO 1

2 K = K - 1

3 RETURN

END

Działanie tej procedury obrazuje program Z008.

E= 0.100E-07 EPS= 0.200E 00 N. LC 10 200

PR.ARR.= 0.300 0.700

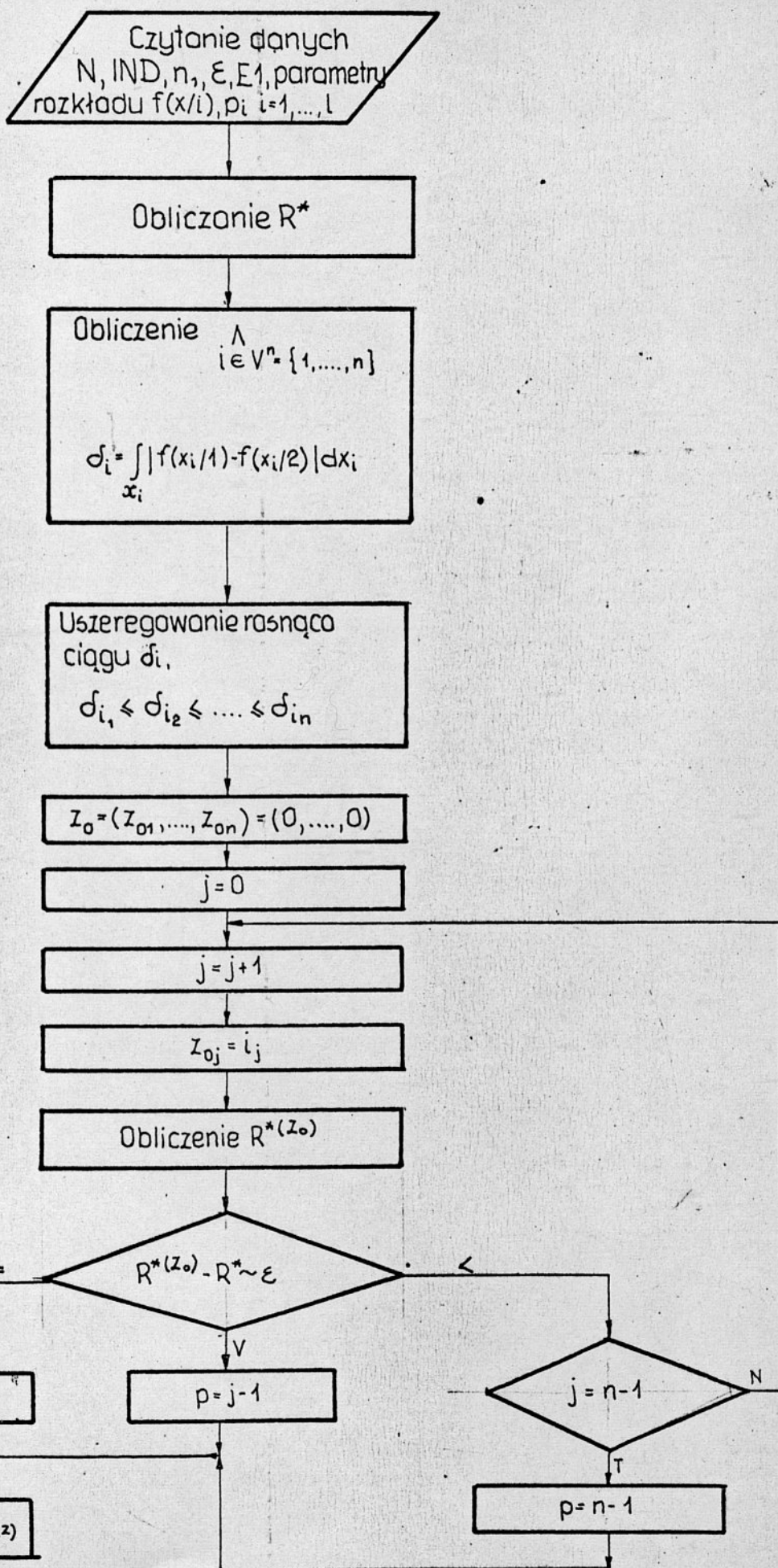
RM, SIG, A, B=

5.000	1.000	3.000	7.000	2.000	2.000	-2.000	6.000
2.000	2.000	-2.000	-6.000	5.000	1.000	3.000	7.000
0.000	1.000	-2.000	2.000	1.000	3.000	-5.000	7.000
-2.000	2.000	-6.000	4.000	0.000	3.000	-6.000	6.000
10.000	5.000	0.000	15.000	5.000	2.000	1.000	2.000
1.000	0.500	0.000	2.000	2.000	2.000	-2.000	6.000
0.000	3.000	-6.000	6.000	2.000	1.000	0.000	4.000
5.000	2.000	1.000	9.000	4.000	0.500	3.000	5.000
0.000	0.500	-1.000	1.000	1.000	2.000	-3.000	5.000
4.000	4.000	-4.000	16.000	0.000	1.000	-2.000	2.000

RYZYKO CAŁK. 0.66018145898F 00RIAD. 0.24509827435F 01

K= 1 RYZYKO: 0.87455549604F 00RLAD= 0.99314809026E 00 NR.CECHY: 4

ODRZUCONE CECHY: 4



Przedstawione w tym rozdziale procedury można zastosować dla dowolnie dużych N , tylko P002, to znaczy SUBROUTINE SELEKCJA ogranicza $N \leq 100$.

Otrzymane wyniki programów pokazują, że algorytmy te działają dosyć wolno, czas realizacji zależy przede wszystkim od parametrów nie związanych z istotą selekcji, a mianowicie czas zależy we wszystkich programach /oprócz Z007/ od zadanego LC , to znaczy od liczby generowanych punktów w metodzie Monte Carlo. Zbyt małe LC powoduje bardzo niedokładne obliczenia odpowiednich wartości, przez co porównywanie ich potem traci sens.

Jednakże należy sądzić, że w bliskiej przyszłości będą działały maszyny cyfrowe jeszcze o większej szybkości obliczeń, albo zbudowane zostaną szybciej zbieżne podprogramy liczenia całek wielowymiarowych bez użycia metod Monte Carlo.

6. WNIOSKI I UWAGI KOŃCOWE

W pracy sformułowano dwa problemy selekcji cech w rozpoznawaniu obrazów przy znajomości pełnej informacji probabilistycznej, nazwane zadaniem 1 i zadaniem 2. Dla zadań tych zbudowano osiem algorytmów. Cztery dla zadania 1 i cztery dla zadania 2. Algorytmy sformułowane w rozdziale 2.3 są algorytmami optymalnymi względem ryzyka. Optymalny algorytm selekcji dla zadania 1 zrealizowano w postaci programu na maszynę cyfrową, program Z009. Z przedstawionych w rozdziale 5 rezultatów tego programu wynika, że wąskim przekrojem obliczeń jest czas realizacji algorytmu. Czas ten wynika przede wszystkim z przyjętej metody obliczania wartości ryzyka. Wydaje się, że zastosowanie metody Monte-Carlo obliczania całek wielowymiarowych do wyznaczania odpowiednich wartości ryzyka daje możliwości dokładniejszego wyznaczania ryzyka, niż zastąpienia go częstością błędnej klasyfikacji. Jednakże, aby osiągnąć w miarę dokładne wyniki /w tym celu budowano algorytmy optymalnej selekcji cech/ liczba generowanych punktów w metodzie Monte-Carlo powinna być bardzo duża. Pociąga to za sobą duży czas obliczeń.

Ze względu na ograniczone możliwości korzystania z maszyny cyfrowej, zrezygnowano z dokładności obliczeń na rzecz przeprowadzenia innych eksperymentów. Z tego też powodu /mała dokładność obliczeń ryzyka/ zrezygnowano z realizacji w formie programu algorytmu optymalnej selekcji dla zadania 2. Program taki, przy założeniu niewielkiej liczby generowanych punktów w metodzie Monte-Carlo nie dawałby poprawnych odpowiedzi, jako że dokładność wyznaczania odpowiednich funkcji celu i ich oszacowań w algorytmie dla zadania 2 jest sprawą istotną.

Algorytmy sformułowane w rozdziale 3.1 dla dwóch klas, $L = 2$, oraz przy założeniu niezależności cech obrazu, są również algorytmami optymalnymi. Algorytm dla zadania 1 zrealizowano w formie programu na maszynę cyfrową - program P002. Nie uruchomiono programu realizującego algorytm dla zadania 2 z przyczyn analogicznych jak uprzednio. Mniejszy czas obliczeń uzyskany przy realizacji programu P002, pomimo takich samych danych wejściowych, wynika po pierwsze z przyjęcia innej metody samego algorytmu /wykorzystanie multiplikatywnych względem cech oszacowań ryzyka/, a po drugie ze specyfikacji przyjętych rozkładów postaci $N(m, \sigma)$, dla których, jak pokazano w rozdziale 5, odpowiednie oszacowania można wyznaczyć w sposób analityczny omijając czasochłonne całkowanie.

Pozostałe algorytmy selekcji są algorytmami suboptymalnymi względem ryzyka zarówno dla zadania 1, jak i dla zadania 2. Ze względu na prostszą postać tych algorytmów zrealizowano je w formie programów na maszynę cyfrową, programy Z005, Z006, Z007, Z008. Programy Z005, Z007 realizują algorytmy dla zadania 1, programy Z006, Z008 algorytmy dla zadania 2 na podstawie zdefiniowanych w pracy oszacowań ryzyka $\bar{\delta}$ oraz $M_K(J/X)$ odpowiednio.

Z przeprowadzonych badań wynika, że optymalne algorytmy selekcji, ze względu na długi czas obliczeń, warto stosować w przypadkach niezbyt dużej liczby cech obrazu i niewielkiej liczby odrzucanych cech /czy też niedużego ε /. W pozostałych przypadkach /duże n / - wygodniej jest stosować algorytmy suboptymalne. Należałoby jednak przeprowadzić pewną gradację zastosowań sformułowanych algorytmów suboptymalnych.

Przy bardzo dużych n (n - liczba cech obrazu/, rzędu kilkuset, najwygodniej i najszybciej uzyskuje się rezultaty stosując algorytmy sformułowane w rozdziale 3.2, wykorzystujące oszacowanie δ .

Dla n rzędu kilkudziesięciu wskazane jest stosowanie algorytmów sformułowane w rozdziale 3.1 wykorzystujące zdefiniowane wcześniej oszacowanie ryzyka $1 - M_K(J/X)$, ponieważ otrzymane wyniki będą lepsze niż przy zastosowaniu δ , a czas obliczeń krótszy niż w algorytmach optymalnych.

Jak już wspomniano wcześniej, wąskim przekrojem przedstawionych algorytmów jest czas obliczeń na maszynie cyfrowej. Czas obliczeń związany jest z liczbą punktów generowanych w metodzie Monte-Carlo obliczania całek wielowymiarowych. Liczba ta jest istotna, ponieważ w tych algorytmach metoda podziału i ograniczeń wymaga wielokrotnego obliczania wartości ryzyka, bądź oszacowań ryzyka /całek wielowymiarowych/, tym bardziej, że za podstawę algorytmów przyjęto metodę podziału i ograniczeń według strategii B, która wymaga długiego czasu obliczeń, natomiast niezbyt dużej pamięci maszyny cyfrowej. Dla przyjętych przykładów ($n = 10$, zadeklarowane $n=20$ /, algorytmy wymagały pamięci rzędu 7 K bitów.

Programy realizujące algorytmy selekcji są tak skonstruowane, że mogą rozwiązywać zadania dla n rzędu kilkuset /należy tylko zadeklarować większe rozmiary tablic/, bez przekroczenia pamięci 64 K bitów. Niedokładność obliczeń wartości ryzyka i wartości odpowiednich oszacowań w przykładowych obliczeniach spowodowana jest zbyt małą zadaną liczbą generowania punktów w metodzie Monte-Carlo / $LC = 100, 200$ dla $n = 10$ /
/autor nie dysponował dłuższym czasem obliczeń na m.c./.

Przykłady obliczeń mają więc charakter raczej tylko weryfikacyjny dla programów i ilustracyjny dla algorytmów. Najbardziej czasochłonny jest algorytm optymalnej selekcji, w którym kryterium selekcji jest ryzyko.

Możnaby czas ten skrócić, gdyby udało się znaleźć oszacowanie od dołu ryzyka, oszacowanie multiplikatywne bądź addytywne względem cech, takie jakie znaleziono dla dwu klas, $L = 2$ oraz niezależnych cech. Stosowanie innych oszacowań, które nie są multiplikatywne bądź addytywne względem cech, miałyoby się z celem, ponieważ w dalszym ciągu, dla przyjętej strategii podziału zbioru rozwiązań, należałoby wyznaczać całki wielowymiarowe, których obliczanie jedynie w minimalnym stopniu zmniejszałoby czas obliczeń na maszynie cyfrowej.

Proponowane w niniejszej pracy algorytmy selekcji w porównaniu z istniejącymi mają swoje dobre i złe strony.

Algorytmy optymalnej selekcji dla zadania 1 jak i dla zadania 2 są algorytmami, w których rozwiązanie nągół uzyskuje się szybciej niż w przypadku bezpośredniego przeglądu wszystkich możliwych rozwiązań

Niewątpliwą zaletą algorytmów suboptymalnych zdefiniowanych na podstawie oszacowania $\bar{\delta}$ jest szybkość i prostota uzyskania rozwiązania. Natomiast wadą, mała dokładność uzyskanych wyników. Strategia tych algorytmów jest identyczna ze strategią algorytmów już istniejących /obliczanie oszacowań dla każdej cechy i odrzucanie ekstremalnej w sensie przyjętego kryterium/. Jednakże istniejące algorytmy są suboptymalne względem przyjętych kryteriów /oszacowań/ i suboptymalne względem ryzyka, /superpozycja suboptymalności/. Natomiast zaproponowane algorytmy są algorytmami optymalnymi względem $\bar{\delta}$, ale suboptymalnymi

względem ryzyka.

Algorytmy selekcji wykorzystujące zdefiniowane w niniejszej pracy oszacowanie ryzyka $1 - M_K (J/X)$ są algorytmami suboptymalnymi względem oszacowania, a więc i ryzyka. Zaletą ich jest natomiast sformułowane kryterium selekcji zbieżne do ryzyka. Można powiedzieć, że dla odpowiednio dużych wartości parametru k algorytmy te są suboptymalne względem ryzyka, przy czym suboptymalność wynika tylko z przyjętej strategii algorytmu. Nie nakłada się dodatkowo błąd wynikający z przyjęcia jako kryterium selekcji oszacowania nie zaś samego ryzyka.

Problem budowy algorytmów optymalnej selekcji wymaga dalszych badań zmierzających do skrócenia czasu obliczeń tych algorytmów zrealizowanych, w szczególności, w postaci programów na maszynę cyfrową.

L i t e r a t u r a

1. Abdali S. Kamal : "Feature Extraction Algorithms",
Pattern Recognition, Pergamon Press, 1971, vol.3. pp.3-21.
2. Andrews H.C. "Multidimensional Rotations in Feature Selection"
IEEE. Trans.on Comput. September 1971.
3. Aizerman, Brawerman, Rozonoer "Metod potencjalnych funkcji
w teorii obuczenia maszyn" Izd.Nauka, Moskwa 1970.
4. Babu C.C. "An Application the Divergence to Feature Selection
in Pattern Recognition" IEEE, Syst.M. and Cyb., November
1972, No 5.
5. Białasiewicz J. "O metodach redukcji danych opartych na
procesach uczenia w zastosowaniu do syntezy klasyfikatorów
optymalnych" Archiwum Automatyki i Telemechaniki, t.12, z.4,
r. 1968.
6. Białasiewicz J. "Problemy obróbki informacji w rozpoznawaniu
obrazów", Rozpoznawanie Obrazów. Adaptivnyje Sistemy.
A.N.CCOP - Międzynarodowa Federacja po Awtomatizieskomu
Uprawleniu. Izd. Nauka, Moskwa 1971.
7. Białasiewicz J. "Liniowe funkcje decyzyjne w zastosowaniu
do rozpoznawania sytuacji technologicznych" - Archiwum
Automatyki i Telemechaniki, t.11, z.4, 1966.

8. Białasiewicz J. "Pewne koncepcje redukcji danych w zastosowaniu do rozpoznawania sytuacji technologicznych", Podstawy sterowania Tom 2., z.4, 1972.
9. Bubnicki Z. "Rozpoznawanie sytuacji technologicznych w systemach kompleksowej automatyzacji", Systemy Automatyki Kompleksowej, PAN, Ossolineum 1969.
10. Bubnicki Z. "Least Interval Pattern Recognition and its Application to Control Systems", Materiały IV Kongresu IFAC, z. 21, Warszawa 1969.
11. Calvert Thomas W. "Nonorthogonal Projections for Feature Extraction in Pattern Recognition" IEEE Trans. on Comp., May 1970, Short Notes.
12. Chen C.H. "Theoretical Comparison of Class Feature Selection in Pattern Recognition" IEEE. Trans. on Comput., September 1971, Short Notes.
13. Chu J.T., Chueh J.C. "Error Probability in Decision Functions for Character Recognition". Journal of Association for Computing Machinery, vol. 14, No 2, April 1967, pp.273-280.
14. Cyplik J.Z. "Osnovy teorii obuczajuszczijejsja sistiem" Izd.Nauka, Moskwa 1970.
15. Devijver Pierre A. "On a New Class of Bayes Risk in Multi-hypothesis Pattern Recognition". IEEE Trans. on Comput., vol.C-23, No 1, January 1971.
16. Degroot M. "Optimalnyje statističeskie rieszenija" Izd.MIR, Moskwa 1974.

17. Daly J.A. "Some Dual Problems in Pattern Recognition"
Pattern Recognition, Pergamon Press 1971, vol.3, pp.73-84.
18. Feinstein "Osnovy teorii informacii" Izd.Nauka, Moskwa 1957.
19. Finkelsztejn J.J., Korbut A.A."Programowanie dyskretne"
PWN, Warszawa 1974.
20. Fisz Marek "Rachunek prawdopodobieństwa i statystyka
matematyczna" PWN, Warszawa 1967, t.18.
21. Greblicki W. "Asymptotyczne optymalne algorytmy rozpoznawania
i identyfikacji w warunkach probabilistycznych". Prace Naukowe
ICT Polit.Wrocławskiej, nr 18 - seria: Monografie, nr 3,
Wrocław 1974.
22. Henderson T.L. Lainiotis D.G."Comments on Linear Feature
Extraction" IEEE Trans.on. Inf.Theory, Nov.1969.
23. Hasiewicz Z.M., Kurzyński "Wyniki eksperymentalnego badania
pewnych algorytmów rozpoznawania z uczeniem dla prostych
łańcuchów Markowa", Prace Naukowe ICT, Polit.Wrocławskiej,
Nr 9 Wrocław 1973.
24. Joshi A.K. "A Note on a Certain Theorem Stated by Kullback"
Correspondence IEEE Trans.on Inf. Theory. January.
25. Kulikowski J.L."Cybernetyczne układy rozpoznające"
PWN, Warszawa 1972.
26. Kailath Thomas "The divergence and Bhattacharyya Distance in
Signal Selection" IEEE Trans. on Com. Technology vol.com.15,
No 1, February 1967.
27. Keinosuke Fukunaga "Estimation of Classification Error"
IEEE Trans. ob Comput. vol C-20, No 12, December 1971.

28. Keinosuke Fukunaga "Application of the Karhunen-Loeve Expansion to Feature Selection and Ordering"
IEEE Trans.on Comput., vol. C-19, No 4, April 1970.
29. Lainiotis D. "A Class of Upper Bounds on Probability of Error for Multihypothesis Pattern Recognition" IEEE Trans. on Inf.Theory Nov. 1969, vol.II-15, No 6.
30. Lewis P.M. "The Characteristic Selection Problem in Recognition Systems", IRE Trans.on Inf.Theory, Febr.1969.
31. Nilson N.J. "Maszyny uczące się", PWN Warszawa 1968.
32. Patrick E.A. "Nonparametric Feature Selection" IEEE Trans. on Inf. Theory, vol.IT-15, No 5, September 1969.
33. Patrick E.A. "Interactive Pattern Analysis and Classification Utilizing Prior Knowledge". Pattern Recognition, Pergamon Press 1971, vol.3, pp 53-71.
34. Perez A. "Information - Theoretic Risk Estimates in Statistical Decision" Kybernetyka, No 1, 3/1967.
35. Perez A. "Information Theory Methods in Reducing Complex Decision Problems", Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes, 1965, Prague 1967, 55-87.
36. Rasek E. "A Contribution to the Problem of Feature Selection with Similarity Functionals in Pattern Recognition", Pattern Recognition, Pergamon Press 1971, vol.3, pp. 31-36.
37. Sammon J.W. , Proctor A.H., Roberts D.F. "An Interactive - Graphic Subsystem for Pattern Analysis". Pattern Recognition Pergamon Press 1971, vol.3, pp 37-52.

38. Toussaint T.G., "Comments, "On Theoretical Comparison of a Class of Feature Selection Criteria in Pattern Recognition" IEEE, Trans. on Comput., June 1972, vol. 1 C-21, No 6.
39. Villmansen T.R. "Feature Evaluation with Measures of Probability Dependence" IEEE Trans. on Comput, vol. 1, pp 381 - 388.
40. Vajda I., Correspondence "Note on Discrimination Information and Variation" IEEE Trans. of Inf. Theory, November 1970
41. Wee W.G. "On Feature Selection in a Class of Distribution Free Pattern Classifiers" IEEE Trans. on Inf. Theory, vol. IT-16, No 1, January 1970.
42. Zobrist A.L. "The Organization of Extracted Features for Pattern Recognition" Pattern Recognition, Pergamon Press 1971, vol.3, pp 23-30.
43. Kanal L. "Pattern in Pattern Recognition: 1968-1974" IEEE Trans. on Inf. Theory, vol.IT-20, No 6, November 1974
44. Chang C.Y. "Dynamic Programming as Applied to Feature Subject Selection in a Pattern Recognition System" IEEE Trans.Syst.Man Cybern, vol. SMC-3, pp 166-171, March 1973.
45. Grabowski J." Optymalne sterowanie sekwencjami operacji w dyskretnych procesach produkcyjnych" , Rozprawa doktorska ICT Pol. Wrocławskiej ,Wrocław 1973r.
46. Zieliński R., "Metody Monte Carlo," WNT, Warszawa 1970.

Spis rysunków i tabel.

1. Rys.1.1 Algorytm klasyfikacji	12
2. Rys.1.2 Przestrzeń obrazów	13
3. Rys.2.1.1 Graficzne przedstawienie trzech kroków algorytmu	36
4. Rys.2.1.2 Graficzne przedstawienie I etapu algorytmu wg strategii B	38
5. Rys.2.1.3 Graficzne przedstawienie II etapu algorytmu wg strategii B	42
6. Rys.2.3.1 Graficzne przedstawienie etapu I	51
7. Rys.2.3.2 II etap algorytmu dla zadania 1	58
8. Rys.2.3.3 II etap algorytmu dla zadania 2	67
9. Rys.3.2.1 Funkcje gęstości o różnych σ_{oi}	90
10. Rys.4.1 Zależność różnych typów oszacowań od $p(i/x)$	104
11. Rys.4.2 Dokładność oszacowania R^* w zależności od wielkości k	105
12. Rys.5.1 Schemat blokowy programu Z009	124
13. Rys.5.2 Schemat blokowy programu P002	137
14. Rys.5.3 Zależność czasu obliczeń od liczby odrzucanych cech	146
15. Rys.5.4 Schemat blokowy programu Z005	147
16. Rys.5.5 Schemat blokowy programu Z006	151
17. Rys.5.6 Zależność czasu obliczeń od liczby odrzucanych cech	155
18. Rys.5.7 Schemat blokowy programu Z007	156
19. Rys.5.8 Schemat blokowy programu Z008	159

Tabele

1. Tab.4.1 Wartości różnych oszacowań ryzyka	102
2. Tab.4.2 Wartości $p(x^k/1)$, $p(x^k/2)$, $p(x^k)$	107
3. Tab.5.1 Czas obliczeń oraz numery wyselekcjonowanych cech w programie Z005 (LC=100)	145
4. Tab.5.2 Czas obliczeń oraz numery wyselekcjonowanych cech w programie Z005 (LC=200)	145
5. Tab.5.3 Czas obliczeń oraz numery wyselekcjonowanych cech w programie Z007	155

D O D A T E K

W części tej omówione zostały kolejno wszystkie programy oraz przedstawione ich tabulogramy.

Opisu każdego programu dokonano według schematu:

1. Funkcja programu
2. Wykorzystanie urządzeń zewnętrznych
3. Budowa programu
4. Dane wejściowe programu
5. Przygotowanie nośników danych
6. Wyniki
7. Wykorzystanie pamięci.

Wszystkie programy napisane są w języku FORTRAN 1900.

Realizacja na maszynie cyfrowej serii ODRA-1300.

Program Z009

1. Funkcja programu

Program Z009 realizuje algorytm optymalnej selekcji względem ryzyka Bayesa.

W programie przyjęto N -niezależnych cech o rozkładach normalnych $N(m, \sigma)$

m - wartość oczekiwana

σ - odchylenie standardowe.

Program wyznacza spośród N cech, M cech których odrzucenie spowoduje najmniejszą stratę informacji w sensie ryzyka Bayes'a.

2. Wykorzystanie urządzeń zewnętrznych

1 czytnik kart

1 drukarka wierszowa

3. Budowa programu

Program zawiera 8 segmentów, 2 bloki wspólne.

a/ Segment główny MASTER SELEZW9 - jest to segment inicjujący, przejmuje częściowe sterowanie program, drukuje wyniki programu.

b/ Segment SUBROUTINE WSTĘPNY - czyta dane, dokonuje ich wstępnych przekształceń, drukuje dane oraz definiuje wspólne bloki.

c/ Segment SUBROUTINE SELEKTIONM1S - jest jednym z najbardziej istotnych segmentów programu. Wyznacza pierwsze rozwiązanie optymalne. Dokładne omówienie tej procedury zawarte jest w rozdziale 5.

- d/ Segment SUBROUTINE MISS jest segmentem pomocniczym dla segmentu SELECTIONM1S.
- e/ Segment SHELLMALR2 - dokonuje uszeregowania w ciąg malejący wartości tablicy jednowymiarowej oraz tworzy tablicę indeksów odpowiadającą kolejności ciągu malejącego.
- f/ Segment SELECTIONRM jest drugim z najbardziej istotnych segmentów programu. Wyznacza on rozwiązanie optymalne zadania.
- g/ Segment FUNCTION RYZYKO, oblicza ryzyko dla odpowiednich x, w oparciu o metodę Monte-Carlo.
- h/ Segment FUNCTION RNORMAL - oblicza wartości odpowiednich funkcji gęstości.

Wspólne bloki

- a) ALA
- b) BETA

4. Dane wejściowe programu

N - liczba cech w obrazie

M - liczba cech do odrzucenia

LC- liczba generowanych punktów w metodzie Monte-Carlo

P(1), P(2)- prawdopodobieństwo a priori wystąpienia 1, 2

klasy

m_{11}, m_{12} - wartość oczekiwana i-tej cechy odp. w klasie 1 i 2

σ_{11}, σ_{12} - odchylenie standardowe " " " " " "

a_{11}, a_{12} - dolna granica całkowania " " " " " "

b_{11}, b_{12} - górna " " " " " "

M - inne liczby odrzucania cech

$\emptyset$ - umieszczenie zera na końcowej karcie implikuje zakończenie programu.

5. Przygotowanie nośników danych

Dane należy przygotować na kartach perforowanych w kolejności

$N, M, LC, P(1), P(2)$

$m_{11}, \sigma_{11}, a_{11}, b_{11}, m_{12}, \sigma_{12}, a_{12}, b_{12}$

$m_{21}, \sigma_{21}, a_{21}, b_{21}, m_{22}, \sigma_{22}, a_{22}, b_{22}$

$\vdots$

$m_{N1}, \sigma_{N1}, a_{N1}, b_{N1}, m_{N2}, \sigma_{N2}, a_{N2}, b_{N2}$

M

M

$\vdots$

M

$\emptyset$

Wszystkie dane pisane są w formacie swobodnym. Dane $N, M, LC, M, M, \dots, \emptyset$ - są typu Integer, pozostałe typu Real.

Dane mogą być pisane na kartach w sposób dowolny, nie ma żadnych przeciwwskazań co do ilości danych na jednej karcie, dopuszcza się możliwość rozdzielenia danych na poszczególne karty według uznania użytkownika. Tylko dodatkowe inne niż zadane na początku wartości M muszą być pisane po jednej na każdej karcie. $\emptyset$ implikujące koniec algorytmu musi być na osobnej karcie.

6. Wyniki

Wyniki wyprowadzane są w postaci wydruków na LP i zawierają

0. Wartość M

1. Wczytane dane wejściowe /wszystkie/

2. Ryzyko przed selekcją

3. Numery cech odrzuconych - pierwsze rozwiązanie optymalne

4. Numery cech odrzuconych - rozwiązanie optymalne

5. Numery cech pozostawionych
6. Wartość ryzyka po selekcji
7. Czas obliczeń dla każdego M.
7. Wykorzystanie pamięci

Program wykorzystuje 7424 komórek pamięci, umożliwia rozwiązywanie problemów selekcji dla

$$N \leq 20, \quad 2 \leq M < N$$

/Można zwiększyć rozmiar zadań poprzez zadeklarowanie większych tablic/.

Ograniczenia z góry na LC wynikają wyłącznie z czasu obliczeń, a z dołu ze względu na dokładność obliczania ryzyk.

Postuluje się $LC = K^{N-M}$, $K \gg 20$.

Przedziały całkowania zaś $[a_{ij}, b_{ij}] = [m_{ij} - L_1 \sigma_{ij}, m_{ij} + L_1 \sigma_{ij}]$.

Dla K małych $L_1, L_2 \sim 2$, dla dużych $L_1, L_2 \gg 3$.

FORTRAN COMPIATION BY #XFAT MK 5A DATE 24/04/76 TIME 17/16/22

0001

LIST

0002	LIBRARY(SUBGROUPESCF)	2
0003	PROGRAM(700Q)	3
0004	CUMPRESS INTEGER AND LOGICAL	4
0005	INPUT1 =CR0	5
0006	OUTPUT2=IPO	6
0007	TRACE 2	7
0008	END	8

0009	MASTER SELEZWQ	9
0010	DIMENSION NC(20), NCA(20), NCO(20), K(20), ,R(20), K1(20), KOPT(20)	10
0011	CUMMON /ALA/ N,M,LC,P,A1,B1	11
0012	EXTERNAL RYZYKO	12
0013	WRITE(2,12)	13
0014	12 FORMAT(////)	14
0015	CALL USTEPNY	15
0016	4 DO 1 I=1,N	16

0017	1	NC(1)=1	17
0018		CALL ITIME(ITP)	
0019		CALL SELECTIONMIS(NC, NCO, NCA, K, N, M, RYZYKO, E2, RY, 7R)	18
0020		WRITE(2,11) (NC(1), I=1, M)	19
0021	11	FORMAT(/5X, 23HNUMERY ODZUCONYCH CECH/(6X, 2015))	20
0022		WRITE(2,14) PY, E2	21
0023	14	FORMAT(/5X, 8HRYZYKO, F20.11, 9HRLADRYZ, F20.11)	22
0024		DO 3 J=1, M	23
0025	3	ZK(NC(J)) =-10.	24
0026		CALL SHEILMALR2(N, ZR, NC)	27
0027		L1=M-1	29
0028		DO 4 J=1, M	30
0029	4	NC(L1+J)=NCO(J)	31
0030		DO 20 KK=1, N	
0031	20	KOPT(KK)=KK	
0032		CALL SELECTIONRM(N, M, L1, K1, NCA, NCO, KOPT, ZR, ROPT, RYZYKO)	33
0033		DO 5 J=1, L1	34
0034	5	NCA(J)=NC(KOPT(J))	35
0035		J1=0	36
0036		DO 7 I=1, N	38
0037		DO 6 J=1, L1	39
0038		IF(1.E9, NCA(J)) GO TO 7	40
0039	6	CONTINUE	44
0040		J1=J1+1	41
0041		NC(J1)=1	42
0042	7	CONTINUE	43
0043		WRITE(2,15) ROPT, (NCA(I), I=1, L1)	46
0044		WRITE(2,16) (NC(1), I=1, M)	47
0045	15	FORMAT(/5X, 4HROPT, F20.11/5X, 20HCECHY POZOSTAWIONE, 2014)	48
0046	16	FORMAT(/5X, 17HCECHY ODZUCONE, 2014)	49
0047		CALL ITIME(ITK)	
0048		IT=ITK-ITP	
0049		WRITE(2,100) IT	
0050	100	FORMAT(/7HCZAS = ,110)	
0051		WRITE(2,12)	50
0052		READ(1,13) M	51
0053		WRITE(2,101) M	
0054	101	FORMAT(/3HM, 15)	
0055	13	FORMAT(10)	52
0056		IF(M, NE, 0) GO TO 2	53
0057		STOP	54
0058		END	55

END OF SEGMENT, LENGTH 360, NAME SELEZWO

0059 SUBROUTINE USTEPNY

56

0060	DIMENSION P(2),RM(20,2),SIG(20,2),A(20,2),R(20,2),SI(20,2),SG(20,2)	257
0061	1) A1(20),R1(20)	58
0062	COMMON/AIA/ N,M,LC,P,A1,R1	59
0063	COMMON /BETA/ RM,SI,SG	60
0064	L=2	61
0065	READ(1,3) N,M,LC,(P(I),I=1,L),((RM(I,J),SIG(I,J),A(I,J),R(I,J),J=1,20	62
0066	1,L),I=1,N)	63
0067	5 FORMAT(3I0,50F0.0)	64
COMMENT 178	IS THIS LARGE A REPEAT COUNT INTENDED AT ABOUT COLUMN 19, LINE 0067	
0068	WRITE (2,4) N,M,LC,(P(I),I=1,L),((RM(I,J),SIG(I,J),A(I,J),R(I,J),J=1,20	65
0069	1,L),I=1,N)	66
0070	4 FORMAT(//10X,2H1=,15,2X,2H1=,15,2X,3H1C=,17,10X,8HPR,APR.=,2F12.3/	67
0071	10X,14H1RM, SIG, A, B=/(9X,4F12.3,10X,4F12.3))	68
0072	DO 1 I=1,N	69
0073	A1(I)=AMIN1(A(I,1),A(I,2))	70
0074	R1(I)=AMAX1(R(I,1),R(I,2))	71
0075	1 R1(I)=R1(I)-A1(I)	72
0076	DO 2 I=1,N	73
0077	DO 2 J=1,L	74
0078	SI(I,J)=.3989422804014/SIG(I,J)	75
0079	2 SG(I,J)=.5/(SIG(I,J)*SIG(I,J))	76
0080	RETURN	77
0081	END	78

END OF SEGMENT, LENGTH 283, NAME USTEPNV - COMMENTS

0082	SUBROUTINE SELECTIONM1S(NC,NC0,NCA,K,N,M,FUN,F2,U,ZR)	79
0083	DIMENSION NC0(N),NC(N),NCA(N),K(N),ZR(20)	80
0084	DO 1 I=1,N	81
0085	1 NCA(I)=NC(I)	82
0086	DO 2 J=1,M	83
0087	IN=IN+1-J	84
0088	W=10.	85
0089	DO 3 I=1,IN	86
0090	CALL MISS(IN,I,NCA,K)	87
0091	LAC=IN-1	88
0092	X=FUN(LAC,K,EPS)	89
0093	IF(J.EQ.1) ZR(I)=X	90
0094	IF(U.LE.X) GO TO 3	91
0095	U=X	92
0096	F2=EPS	93
0097	INDEX=I	94
0098	3 CONTINUE	95
0099	NC0(J)= NCA(INDEX)	96
0100	4 CALL MISS(IN,INDEX,NCA,NCA)	100
0101	RETURN	101

0102	END	105
------	-----	-----

END OF SEGMENT, LENGTH 107, NAME SELECTION15

0103	SUBROUTINE MISS(N,L,M,MM)	103
0104	DIMENSION M(N),MM(N)	104
0105	DO 1 I=1,L	105
0106	1 MM(I)=M(I)	106
0107	MM(L)=0	107
0108	IF(L.EQ.N) RETURN	108
0109	DO 2 I=L,N-1	109
0110	2 MM(I)=M(I+1)	110
0111	MM(N)=0	111
0112	RETURN	112
0113	END	113

END OF SEGMENT, LENGTH 113, NAME MISS

0114	SUBROUTINE SHELLMADR2(N,R,M)	114
0115	DIMENSION R(N),M(N)	115
0116	K=1	116
0117	X=R(1)	117
0118	DO 6 I=2,N	118
0119	IF(X.GE.R(I)) GO TO 6	119
0120	X=R(I)	120
0121	K=I	121
0122	6 CONTINUE	122
0123	R(K)=R(1)	123
0124	L=M(K)	124
0125	M(K)=M(1)	125
0126	R(1)=X	126
0127	M(1)=L	127
0128	DO 7 I=2,N	128
0129	J1=1	129
0130	J2=1	130
0131	X=R(1)	131
0132	1 J=(J1+J2)/2	132
0133	IF(J.EQ.J1) GO TO 3	133
0134	IF(X.GT.R(J)) GO TO 2	134
0135	J1=J	135
0136	GO TO 1	136
0137	2 J2=J	137
0138	GO TO 1	138
0139	3 K=1	139
0140	J2=J+2	140

0141	L=M(1)	141
0142	4 IF(K.LT.J2) GO TO 5	142
0143	R(K)=R(K-1)	143
0144	M(K)=M(K-1)	144
0145	K=K-1	145
0146	GO TO 4	146
0147	5 R(J+1)=X	147
0148	7 M(J+1)=L	148
0149	RETURN	149
0150	END	150

END OF SEGMENT, LENGTH 284, NAME SHELLMAIR2

0151	SUBROUTINE SELECTIONRM(N,M,L1,K1,K2,KA,KOPT,RR,ROPT,FUN)	151
0152	DIMENSION K1(M),K2(M),KOPT(L1),RR(L1),KA(N)	152
0153	K1(1)=1	153
0154	L=K2(1)=1+1	154
0155	1 IF(K1(1).GE.K2(1)) RETURN	155
0156	J=K2(1)=K2(1)-1	156
0157	IF(ROPT,1,RR(J)) RETURN	157
0158	NP=2	158
0159	2 IF(NP.EQ.M) GO TO 6	159
0160	3 K2(NP)=1+NP	160
0161	K1(NP)=K2(NP-1)+1	161
0162	4 IF(K1(NP).LT.K2(NP)) GO TO 5	162
0163	NP=NP-1	163
0164	IF(NP.EQ.1) GO TO 1	164
0165	GO TO 2	165
0166	5 K2(NP)=K2(NP-1)	166
0167	J,K,1,I1=1	167
0168	10 I=1+1	168
0169	IF(I.GT.N) GO TO 12	169
0170	IF(I.EQ.K2(K)) GO TO 11	170
0171	KA(J)=I	171
0172	J=J+1	172
0173	GO TO 10	173
0174	11 IF(K.EQ.NP) I1=0	174
0175	K=K+1	175
0176	GO TO 10	176
0177	12 K=M-NP	177
0178	R=FUN(K,KA,FPS)	178
0179	IF(ROPT,1,R) GO TO 4	179
0180	NP=NP+1	180
0181	GO TO 2	181
0182	6 K2(M)=L+M	182
0183	K1(M)=K2(M-1)+1	183
0184	7 IF(K1(M).GE.K2(M)) GO TO 8	184

0185		K2(I)=K2(I)-1	1
0186		J,K,I,I1=1	100
0187	13	I=I+1	100
0188		IF(I.GT.N) GO TO 15	101
0189		IF(I.EQ.K2(K)) GO TO 14	102
0190		KA(J)=1	103
0191		J=J+1	104
0192		GO TO 13	105
0193	14	IF(K.EQ.N) I1=0	106
0194		K=K+1	107
0195		GO TO 13	108
0196	15	R=FUN(L1,KA,EPS)	109
0197		IF(ROPT.IE.R) GO TO 7	201
0198		DO 16 I=1,L1	202
0199	16	RUDT(I)=KA(I)	203
0200		RUDT=R	204
0201		GO TO 7	205
0202	8	NP=M-1	206
0203		IF(NP.EQ.1) GO TO 1	207
0204		GO TO 3	208
0205		PETURN	209
0206		END	210

END OF SEGMENT, LENGTH 530, NAME SELECTIONRM

0207		FUNCTION RYZYKO(K,KA,EPS)	211
0208		DIMENSION KA(K),P(2),A1(20),A2(20),R1(20),R2(20),F1(2),X(20)	212
0209		COMMON /ALAZ, N,M,LC,P,A1,B1	213
0210		T=1./8388608.	214
0211		S,S2=0.	215
0212		OM=1.	216
0213		DO 6 I=1,K	217
0214		J=KA(I)	218
0215		R2(I)=R1(J)	219
0216		A2(I)=A1(J)	220
0217	6	OM=OM+R1(J)	221
0218		DO 1 J=1,LC	222
0219		DO 2 I=1,K	223
0220		T=EPHCRV(T)	224
0221	2	X(I)=R2(I)+T+A2(I)	225
0222		DO 3 J2=1,2	226
0223		F1(J2)=P(J2)	227
0224		DO 4 I=1,K	228
0225		IA=KA(I)	229
0226	4	F1(J2)=F1(J2)+RNDRMAI(X(I),IA,J2)	230
0227	5	CONTINUE	231

0228	F2= AHAX1(F1(1),F1(2))	232
0229	S =S+F2	233
0230	1 S2 =S2+F2+F2	234
0231	SLC =1./IC	235
0232	RYZYKO = 1-OM*S(C+S	236
0233	EPS =(S2-S*S*SLC)/(IC-1)	237
0234	EPS=SQRT(EPS)*OM	
0235	RETURN	241
0236	END	242

END OF SEGMENT, LENGTH 271, NAME RYZYKO

0237	FUNCTION RNORMAL(X,I,J)	243
0238	DIMENSION SI(20,2),SG(20,2),RM(20,2)	244
0239	COMMON /BETA/ RM,SI,SG	245
0240	V=X-RM(I,J)	246
0241	RNORMAL =SI(I,J)*EXP(-V*V*SG(I,J))	247
0242	RETURN	248
0243	END	249

END OF SEGMENT, LENGTH 64, NAME RNORMAL

0244	FINISH	250
------	--------	-----

END OF COMPILATION = NO ERRORS 1 COMMENT

S/C SURFILE: 48 BUCKETS USED

CONSOLIDATED BY XPCK 120 DATE 24/04/76 TIME 17/17/36

PROGRAM 2009
EXTENDED DATA (22AM)
COMPACT PROGRAM (DPM)
CORE 7424

SEG	SELEZNO
CUV	ALA
SEG	RYZYKO
SEG	WSTEPNY
SEG	ITIME
SEG	SELECTIONHIS
SEG	SELECTIONLR2
SEG	SELECTIONRM
CUV	BETA
SEG	AMIN1
SEG	AMAX1
SEG	MISS
SEG	FMCRV
SEG	RNORMAL

Program P002

1. Funkcja programu

Program P002 realizuje optymalny algorytm selekcji cech *w opowicie* o współczynnik Bhattacharyy'a ξ oraz ryzyko Bayes'a, dla dwu klas. W programie założono N cech niezależnych, każda o rozkładzie normalnym $N(m, \sigma)$,

m - wartość oczekiwana

σ - odchylenie standardowe

Program odrzuca spośród N cech, M cech wnoszących najmniej informacji w sensie ryzyka Bayes'a.

2. Wykorzystanie urządzeń zewnętrznych

1 czytnik kart

1 drukarka wierszowa

3. Budowa programu

Program składa się z 6 segmentów

a/ Segment - MASTER SELEKCJA-BHATTACHARYYA - segment ten spełnia rolę inicjującą, przejmuje częściowe sterowanie programem.

b/ Segment SUBROUTINE RYZYKO(R, EPS) - oblicza ryzyko Bayes'a dla zadanych cech.

c/ Segment SUBROUTINE SEG1 - wczytuje dane i dokonuje ich wstępnych przekształceń.

d/ Segment SUBROUTINE SEG3 - jest segmentem pomocniczym dla segmentu SUBROUTINE RYZYKO. Przyjmuje na siebie część funkcji koniecznych do obliczenia ryzyka Bayes'a.

e/ Segment SUBROUTINE SELECTION - jest najbardziej istotnym segmentem programu. Może być wykorzystany dla selekcji cech niezależnych o dowolnych rozkładach. Omówienie tej procedury w rozdziale V.

f/ Segment SUBROUTINE SHELLROS2 - jest segmentem zsregującym w ciąg rosnący wartości zawarte w innym ciągu oraz zapamiętuje tablicę indeksów odpowiadającą uporządkowanemu ciągowi wartości.

Program zawiera również bloki wspólne

a/ CUVR

b/ CUV3

c/ CUV11

d/ CUV MASTER

4. Dane wejściowe do programu

N - liczba cech w obrazie

M - liczba cech do odrzucenia

LC - liczba generowanych punktów w metodzie Monte-Carlo
obliczania całek - liczba ta implikuje dokładność
obliczania ryzyka

P1 - prawdopodobieństwo a priori wystąpienia pierwszej klasy.

Na tej podstawie program oblicza $P2 = 1 - P1$

m_{i1}, m_{i2} - wartość oczekiwana i-tej cechy odpowiednio
w klasie pierwszej i drugiej

σ_{i1}, σ_{i2} - odchylenie standardowe i-tej cechy odpowiednio
w klasie pierwszej i drugiej

a_{i1}, a_{i2} - górna granica całkowania i-tej cechy odpowiednio
w klasie pierwszej i drugiej

b_{i1}, b_{i2} - górna granica całkowania -i-tej cechy odpowiednio
w klasie pierwszej i drugiej.

5. Przygotowanie nośników danych

Dane należy przygotować na kartach perforowanych w kolejności
N, M, LC, P1

$$\begin{array}{cccccccc} m_{11}, & \sigma_{11}, & a_{11} & b_{11}, & m_{12}, & \sigma_{12}, & a_{12}, & b_{12} \\ m_{21}, & \sigma_{21}, & a_{21}, & b_{21}, & m_{22}, & \sigma_{22}, & a_{22}, & b_{22} \\ \vdots & & & & & & & \\ m_{N1}, & \sigma_{N1}, & a_{N1}, & b_{N1}, & m_{N2}, & \sigma_{N2}, & a_{N2}, & b_{N2} \end{array}$$

Wszystkie dane pisane są w formacie swobodnym

N, M, LC - są typu Integer, pozostałe typu Real.

Dane mogą być perforowane "ciurkiem" lub rozdzielone wg uznania użytkownika na poszczególne karty.

6. Wyniki

Wyniki wyprowadzane są w postaci wydruków na LP i zawierają

1. Numery odrzuconych cech
2. Czas obliczeń na m.c.

7. Wykorzystanie pamięci m.c. i ograniczenia

Program wykorzystuje 13120 komórek pamięci.

Umożliwia rozwiązanie problemów selekcji dla

$$N \leq 100, \quad 2 \leq M \leq 99$$

Ograniczenia z góry na LC wynikają wyłącznie z czasu obliczeń, a z dołu ze względu na dokładność obliczeń. Postuluje się ze względu na dokładność

$$LC = K^{N-M}, \quad K \geq 20$$

Przy małych $K \sim 20$ lub niewiele więcej postuluje się dobierać małe przedziały $[a_{ij}, b_{ij}]$ całkowania

$$[a_{ij}, b_{ij}] = [m_{ij} - L_1 \sigma_{ij}, m_{ij} + L_2 \sigma_{ij}], \quad L_1 L_2 \sim 2$$

Przy stosunkowo dużych K , $K = 100$ i więcej postuluje się

$$[a_{ij}, b_{ij}] = [m_{ij} - L_1 \sigma_{ij}, m_{ij} + L_2 \sigma_{ij}], \quad L_1, L_2 \geq 3.$$

Dokładniejsza analiza przyjętych L_1 , L_2 opierać się musi na znajomości rozkładów których mają dotyczyć.

0031	K=N-M	
FORTRAN COMPILATION BY #XFAT MK 5A DATE 15/07/76 TIME 17/16/20		
0001	LIST	
0002	LIBRARY(SUNGROUPFSCE)	2
0003	PROGRAM(P002)	
0004	COMPRESS INTEGER AND LOGICAL	Z0050040
0005	INPUT1=TR0	Z0050050
0006	INPUT2=CR0	Z0050060
0007	OUTPUT3=LP0	Z0050070
0008	USE10=MT1/F0RMATTED/35	Z0050080
0009	TRACE 2	Z0050090
0010	END	Z0050100
0011	MASTER SELEKCJA-BATTACHARYA	Z0050110
0012	INTEGER 0,OX	Z0050120
0013	COMMON /CUVMMASTER/ 0,OX	Z0050130
0014	N=2	Z0050160
0015	OX=3	Z0050170
0016	LP=3	Z0050180
0017	MT=10	Z0050190
0018	CALL SEG1	Z0050200
0019	CALL SELECTION	Z0050210
0020	PAUSE	Z0050220
0021	CALL WTLP(LP)	Z0050230
0022	STOP	Z0050240
0023	1 F0RMAT(I0)	Z0050250
0024	END	Z0050260

END OF SEGMENT, LENGTH 42, NAME SELEKCJA

[illegible]

END OF SEGMENT, LENGTH 196, NAME, RYZYKO

0083 END 20050830

END OF SEGMENT	LENGTH	NAME	COMMENTS
	290	SEG1	

END OF SEGMENT, LENGTH 165, NAME SEG3

0109		SUBROUTINE SELECTION	20051090
0110		INTEGER KUPT(100),KA(100),NRO(100),K(2707),O,OX	20051100
0111		REAL R1(101),R(100),RO(101)	20051110
0112		COMMON /CUV11/ N,M,LC,P1,P2,P	20051120
0113		COMMON /CUVR/ KA	20051130
0114		COMMON /CUVMASER/ O,OX,RO	20051140
0115		EQUIVALENCE (R1(1),K(1))	20051150
0116		CALL ITIME(ITP)	20051160
0117		IND=1	20051190
0118		DO 21 I=1,N	20051200
0119	21	NRO(I)=1	20051210
0120		CALL SHELLROSR2(N,RO,NRO)	
0121		K(3),J=N-M	20051520
0122		NC=J+1	20051530
0123		DO 1 I=1,J	20051540
0124	1	KA(I),KUPT(I)=NRO(I)	20051550
0125		DO 2 I=NC,N	20051560
0126	2	KA(I),KUPT(I)=0	20051570
0127		CALL RYZYKO(ROPT,EPS)	20051580
0128		CALL ITIME(ITK)	20051590
0129		ITP1=ITK-ITP	20051600
0130		K(2),LR,K2=NC+3	20051620
0131		K(M*LR+3)=0	20051630
0132		K(1)=5	20051640
0133		PI=1.	20051670
0134		K4=LR+1	20051680
0135		DO 3 I=1,NC	20051690
0136	3	PI=PI*RO(I)	20051700
0137		R(1)=PI	20051710
0138		DO 4 I=1,J	20051740
0139		IB=K4-I	20051750
0140		K(IB),KIR=NC-I	20051760
0141		F=PI/RO(KIB)	20051770
0142		IF((1.-SQRT(1.-P*F)).LT.ROPT) GO TO 4	20051780
0143		K(1),K1=IB+1	20051790
0144		IF(K1,GT.LR) GO TO 19	20051800
0145		GO TO 5	20051810
0146	4	CONTINUE	20051820
0147	5	NP=2	20051830
0148	6	IP=(NP-1)*LR	20051840
0149		K3L=K(IP-LR+3)	20051890
0150		L=J+NP-K3L	20051900
0151		K(IP+2),K2=IP+LR	20051910

0152	K(IP+1),K1=K2+1-L	Z0051920
0153	K(IP+3),K3=K3L+L	Z0051930
0154	NC=K3+1	Z0051940
0155	K4=K2+1	Z0051950
0156	P(HP),PI=R(NP-1)*RO(N-M+NP)/RO(K3L)	Z0051960
0157	IF(HP,EQ,M) GO TO 10	Z0051970
0158	DO 7 I=1,L	Z0051980
0159	IB=K4-I	Z0051990
0160	K(IB),KIB=NC-I	Z0052000
0161	F=PI/RO(KIB)	Z0052010
0162	IF((1.-SQRT(1.-P+F)).LT.ROPT) GO TO 9	Z0052020
0163	K(IP+1),K1=IB+1	Z0052030
0164	NP=NP+1	Z0052040
0165	IF(K1,LE,K2) GO TO 6	Z0052050
0166	NP=NP-1	Z0052060
0167	7 NP=NP-1	Z0052070
0168	I2=(NP-1)*LR+2	Z0052080
0169	K(I2),K2=K(I2)-1	Z0052090
0170	IF(K2,GE,K(I2-1)) GO TO 8	Z0052100
0171	IF(NP,EQ,1) GO TO 19	Z0052110
0172	GO TO 7	Z0052120
0173	8 K(I2+1)=K(K2)	Z0052130
0174	Np=Np+1	Z0052140
0175	GO TO 6	Z0052150
0176	9 CONTINUE	Z0052160
0177	Np=Np+1	Z0052170
0178	GO TO 6	Z0052180
0179	10 DO 11 I=1,L	Z0052190
0180	IB=K4-I	Z0052200
0181	K(IB),KIB=NC-I	Z0052210
0182	F=PI/RO(KIB)	Z0052220
0183	IF((1.-SQRT(1.-P+F)).GE.ROPT) GO TO 12	Z0052230
0184	11 CONTINUE	Z0052240
0185	GO TO 13	Z0052250
0186	12 K(IP+1),K1=IR+1	Z0052260
0187	IF(K1,GT,K2) GO TO 7	Z0052310

0188	13 INDEX=3	Z0052320
0189	DO 14 I=1,N	Z0052330
0190	KA(I)=NR0(I)	Z0052340
0191	IF(I.NE,K(INDEX)) GO TO 14	Z0052350
0192	INDEX=INDEX+LR	Z0052360
0193	KA(I)=0	Z0052370
0194	14 CONTINUE	Z0052380
0195	CALL RYZYKO(RA, EPS)	Z0052390
0196	IND=IND+1	Z0052400
0197	K(IP+2),K2,I2=K2-1	Z0052420
0198	K(IP+3),K3,I3=K3-1	Z0052430
0199	IF(RA.LT.ROPT) GO TO 15	Z0052440
0200	IF(K1.GT.K2) GO TO 7	Z0052450
0201	GO TO 13.	Z0052460
0202	15 ROPT=RA	Z0052470
0203	DO 16 I=1,N	Z0052480
0204	16 KOPT(I)=KA(I)	Z0052490
0205	IF(I2.LT.K1) GO TO 7	Z0052510
0206	17 F=P1/RO(I3)	Z0052520
0207	IF((1.-SQRT(1.-P*F)).GE.ROPT) GO TO 18	Z0052530
0208	I2=I2-1	Z0052540
0209	I3=I3-1	Z0052550
0210	IF(I2.GE.K1) GO TO 17	Z0052560
0211	GO TO 13	Z0052570
0212	18 K(IP+1),K1=I2+1	Z0052580
0213	IF(K1.GT.K(IP+2)) GO TO 7	Z0052590
0214	GO TO 13	Z0052600
0215	19 CALL ITIME(ITK)	Z0052610
0216	WRITE(3,130)(KOPT(I),I=1,M)	
0217	ITP=ITK-ITP	Z0052660
0218	WRITE(UX,33) ITP	Z0052670
0219	RETURN	Z0052680
0220	130 FORMAT(///10X,23HNUMERY ODRZUCONYCH CECH/(30I4))	
0221	33 FORMAT(10X,6HCZAS =,I5,100X)	Z0052740
0222	END	Z0052750

0223	TRACE 0	Z0052760
0224	SUBROUTINE HTLP(LP)	Z0052770
0225	INTEGER M(30)	Z0052780
0226	REWIND 10	Z0052790
0227	1 READ(10,2) M	Z0052800
0228	WRITE(LP,2) M	Z0052810
0229	GO TO 1	Z0052820
0230	2 FORMAT(1X,30A4)	Z0052830
0231	END	Z0052840

END OF SEGMENT, LENGTH 30, NAME HTLP

0232	SUBROUTINE SHELLROSR2(N,R,M)	27
0233	DIMENSION R(N),M(N)	115
0234	K=1	116
0235	X=R(1)	117
0236	DO 6 I=2,N	118
0237	IF(X.LE,R(I)) GO TO 6	119
0238	X=R(I)	120
0239	K=I	121
0240	6 CONTINUE	122
0241	R(K)=R(1)	123
0242	L=M(K)	124
0243	M(K)=M(1)	125
0244	R(1)=X	126
0245	M(1)=L	127
0246	DO 7 I=2,N	128
0247	J1=1	129
0248	J2=1	130
0249	X=R(I)	131
0250	1 J=(J1+J2)/2	132

0251	IF(J.EQ,J1) GO TO 3	133
0252	IF(X.LT,R(J)) GO TO 2	134
0253	J1=J	135
0254	GO TO 1	136
0255	2 J2=J	137
0256	GO TO 1	138
0257	3 K=1	139
0258	J2=J+2	140
0259	L=M(I)	141
0260	4 IF(K.LT,J2) GO TO 5	142
0261	R(K)=R(K-1)	143
0262	M(K)=M(K-1)	144
0263	K =K-1	145
0264	GO TO 4	146
0265	5 R(J+1)=X	147
0266	7 M(J+1)=L	148
0267	RETURN	149
0268	END	150

END OF SEGMENT, LENGTH 179, NAME. SHELLROSR2

0269	FINISH	Z0052850
------	--------	----------

END OF COMPILATION - NO ERRORS 1 COMMENT

S/C SUBFILE: 44 BUCKETS USED

CONSOLIDATED BY XPC 12D DATE 15/07/76 TIME 17/17/20

PROGRAM P002
EXTENDED DATA (22AM)
COMPACT PROGRAM (DAM)
CORE 13120

SEG	SELEKCJA
CUV	CUVMASTER
SEG	SEG1
SEG	SELECTION
SEG	MTLP
SEG	RYZYKO
CUV	CUVR
CUV	CUV3
CUV	CUV11
SEG	SEG3
SEG	FPMCRV
SEG	EXP
SEG	AMIN1
CUV	CUV12
SEG	SWRT
SEG	AMAX1
SEG	ITIME
SEG	SHELLROSR2

Program Z005

1. Funkcja programu

Program Z005 realizuje suboptymalny względem ryzyka, ale optymalny względem zdefiniowanego w pracy oszacowania ryzyka $M_K(J/X)$, algorytm selekcji. W programie założono N cech niezależnych, każda o rozkładzie normalnym $N(m, \sigma)$,

m - wartość oczekiwana

σ - odchylenie standardowe

Program odrzuca spośród N cech, M cech wnoszących najmniej informacji w sensie oszacowania ryzyka.

2. Wykorzystanie urządzeń zewnętrznych

1 czytnik kart

1 drukarka wierszowa

3. Budowa programu

Program składa się z 6 segmentów

a/ segment główny MASTER SELEZW5 - spełnia rolę inicjującą, przejmuje częściowe sterowanie programem oraz drukuje wyniki programu.

b/ Segment SUBROUTINE WSTĘPNY - wczytuje dane do całego programu, dokonuje ich wstępnych przekształceń, drukuje wczytane dane oraz definiuje wspólne bloki.

c/ Segment SUBROUTINE SELECTIONM1 jest najważniejszym segmentem programu. Może być wykorzystany do selekcji cech o dowolnych rozkładach i o N nawet powyżej 1000 /patrz rozdział V/.

d/ Segment SUBROUTINE MISS - spełnia rolę pomocniczą dla segmentu SELECTIONM1.

e/ Segment FUNCTION RMK - oblicza wartości miary $M_K (J/X)$ dla określonych, wygenerowanych X w oparciu o metodę Monte-Carlo.

f/ Segment FUNCTION RNORMAL - oblicza funkcję gęstości dla odpowiednich X_i oraz j.

Wspólne bloki programu to:

a/ ALA

b/ BETA - zawiera przekształcone parametry rozkładów.

4. Dane wejściowe programu

E - dowolna liczba rzeczywista

N - liczba cech w obrazie

M - liczba cech do odrzucenia

LC - liczba generowanych punktów w metodzie Monte-Carlo obliczania całek wielowymiarowych, liczba ta implikuje dokładność obliczania miary $M_K (J/X)$

NP- numer potęgi występującej w mierze $M_K (J/X)_K \equiv NP$

P(1), P(2)- prawdopodobieństwo a priori wystąpienia odpowiednio pierwszej i drugiej klasy

m_{11}, m_{12} - wartość oczekiwana i-tej cechy odp.w klasie 1 i 2

σ_{11}, σ_{12} - odchylenie standardowe " " " " " "

a_{11}, a_{12} - dolna granica całkowania " " " " " "

b_{11}, b_{12} - górna " " " " " "

M - inne proponowane liczby odrzucania cech

5. Przygotowanie nośników danych

Dane należy przygotować na kartach perforowanych w kolejności

E, N, M, LC, NP, P(1), P(2)

$m_{11}, \sigma_{11}, a_{11}, b_{11}, m_{12}, \sigma_{12}, a_{12}, b_{12}$

:

$$m_{N1}, \quad N1, \quad a_{N1}, \quad b_{N1}, \quad m_{N2}, \quad N2, \quad a_{N2}, \quad b_{N2}$$

M

M

•

M

Ø

Wszystkie dane pisane są w formie swobodnym. Dane N, M, IC, NP są typu Integer, pozostałe typu Real. Dane mogą być drukowane na kartach w sposób dowolny, nie ma szczególnych przeciwwskazań co do ilości danych na karcie lub też rozdzielone wg uznania użytkownika na poszczególne karty, za wyjątkiem innych proponowanych liczb cech do odrzucenia. Należy je pisać w formie IO każda na osobnej karcie. Ostatnia karta powinna zawierać $M = 0$, ta zerowa^{wartość} implikuje zatrzymanie programu.

6. Wyniki

Wyniki wyprowadzane są w formie wydruków na LP i zawierają

1. Wczytane wartości N , M , LC , NP , $P(1)$, $P(2)$ oraz parametry rozkładów
2. Liczbę M ,
3. Numery odrzuconych cech
4. Wartość oszacowania po selekcji
5. Czas obliczeń na m.c.

7. Wykorzystanie pamięci m.c. i ograniczenia

Program wykorzystuje 6500 komórek pamięci. Umożliwia rozwiązanie problemów selekcji dla

$$N \leq 20, \quad M < N$$

Ograniczenia z góry na IC wynikają wyłącznie z czasu obliczeń, a z dołu ze względu na dokładność obliczeń.

Postuluje się

$$LC = K^{N-M}, \quad K \geq 20$$

Przy małych K , postuluje się dobierać małe przedziały

$[a_{ij}, b_{ij}]$ całkowania, $[a_{ij}, b_{ij}] = [m_{ij} - L_1 \sigma_{ij}, m_{ij} + L_2 \sigma_{ij}]$,
 $L_1, L_2 \sim 2$.

Przy dużych K , $L_1, L_2 \geq 3$.

Dokładniejsza analiza przyjętych L_1, L_2 opierać się musi
na znajomości rozkładów.

FORTRAN COMPILATION BY #XFAT MK 5A DATE 20/04/76 TIME 08/46/34

```

0001      LIST
0002      LIBRARY(SUBGROUPESCE)
0003      PROGRAM (Z005)
0004      COMPRESS INTEGER AND LOGICAL
0005      INPUT1 = CR0
0006      OUTPUT2 = LP0
0007      TRACE 2
0008      END

0009      MASTER SFLEZWS
0010      DIMENSION NCO(20),NCA(20),K(20),NC(20),P(2),A1(20),B1(20)
0011      COMMON /ALA/ N,M,LC,ND,P,A1,B1,F,L
0012      EXTERNAL RMK
0013      WRITE(2,12)
0014      CALL WSTEPNY
0015      2 DO 11=1,N
0016      1 NC(1)=1

```

FORTRAN COMPILATION BY #XFAT MK 5A DATE 20/04/76 TIME 09/38/09

```

0001      LIST
0002      LIBRARY(SUBGROUPESCE)
0003      PROGRAM (Z005)
0004      COMPRESS INTEGER AND LOGICAL
0005      INPUT1 = CR0
0006      OUTPUT2 = LP0
0007      TRACE 2
0008      END

0009      MASTER SFLEZWS
0010      DIMENSION NCO(20),NCA(20),K(20),NC(20),P(2),A1(20),B1(20)
0011      COMMON /ALA/ N,M,LC,ND,P,A1,B1,F,L
0012      EXTERNAL RMK
0013      WRITE(2,12)
0014      CALL WSTEPNY
0015      2 DO 11=1,N

```

```

0016      1 NC(I)=1
0017      CALL ITIME(ITP)
0018      CALL SELECTIONM1(NC, NCO, NCA, X, N, M, RMK)
0019      WRITE(2,11) (NCO(I), I=1, M)
0020      11 FORMAT(///6X,23HNUMERY ODRZUCONYCH CEGH/(6X,20I5))
0021      CALL ITIME(ITK)
0022      IT=ITK-ITP
0023      WRITE(2,100) IT
0024      100 FORMAT(//7HC7AS = ,I10)
0025      12 FORMAT(////)
0026      WRITE(2,12)
0027      READ(1,13) M
0028      WRITE(2,101) M
0029      101 FORMAT(/3HM= ,I5)
0030      13 FORMAT(10)
0031      IF(M.NE.0) GO TO 2
0032      STOP
0033      END

```

END OF SEGMENT, LENGTH 123, NAME SELEZWS

```

0034      SUBROUTINE USTEPHY
0035      DIMENSION P(2), RM(20,2), SIG(20,2), A(20,2), R(20,2), SI(20,2),
0036      1 S(20,2), A1(20), R1(20)
0037      COMMON /ALA/ N,M,LC,NP,P,A1,R1,E,L
0038      COMMON /BETA/ RM,SI,S
0039      L=2
0040      READ(1,3) E,N,M,LC,NP,(P(I), I=1,1),((RM(I,J),SIG(I,J),A(I,J),R(I,J)
0041      1),J=1,L),I=1,N)
0042      3 FORMAT(FO,0,4I0,1000F0,0)
COMMENT 178 IS THIS LARGE A REPEAT COUNT INTENDED AT ABOUT COLUMN 26, LINE 0042
0043      WRITE(2,4) E,N,M,LC,NP,(P(I), I=1,L),((RM(I,J),SIG(I,J),A(I,J),
0044      1 R(I,J),J=1,L),I=1,N)
0045      4 FORMAT(/9X,2HE=,E20.11,5X,10HN,M,LC,NP=,4I7/9X,8HPR,APP.=,
0046      12F8.3/9X,14HRH, SIG, A, R=/(9X,4F12.3,10X,4F12.3))
0047      DU 1 I=1,N
0048      A1(I) = AMIN1(A(I,1),A(I,2))
0049      R1(I) = AMAX1(R(I,1),R(I,2))
0050      1 R1(I) = R1(I)-A1(I)
0051      DU 2 I=1,N
0052      DU 2 J=1,L
0053      SI(I,J) = .3089422804014/SIG(I,J)
0054      2 SI(I,J) = .5/(SIG(I,J)*SIG(I,J))
0055      WRITE(2,5) (I,A1(I),R1(I),I=1,N)
0056      5 FORMAT(/9X,7HA1,R1 =/(5(5X,12,2FR,3)))
0057      RETURN

```


0058 END

END OF SEGMENT, LENGTH 337, NAME WSTEPNY - COMMENTS

```

0059 SUBROUTINE SELECTIONM1(NC, NCO, NCA, K, N, M, EPS)
0060 DIMENSION NCO(N), NC(N), NCA(N), K(N)

0061 DO 1 I=1, N.
0062 1 NCA(I) = NC(I)

0063 DO 2 J=1, M
0064 IN = N+1-J

0065 W = 0.
0066 DO 3 I=1, IN

0067 CALL MISS(IN, I, NCA, K)
0068 LAC = IN-1

0069 X = FUN(LAC, W, EPS)
0070 IF(W.GT.X) GO TO 3

0071 W = X
0072 INDEX = I

0073 3 CONTINUE
0074 NCO(J) = NCA(INDEX)

0075 2 CALL MISS(IN, INDEX, NCA, NCA)
0076 WRITE(2,110) W, EPS

0077 110 FORMAT(/,5X,8HMK(J/X)=,F20,11,4HEPS=,E20,11)
0078 RETURN
0079 END

```

END OF SEGMENT, LENGTH 171, NAME SELECTIONM1

```

0080 FUNCTION RMK (LAC, K, EPS)
0081 DIMENSION P(2), A1(20), B1(20), K(LAC), F1(2), X(20)

0082 COMMON /ALA/ N, M, LC, NP, P, A1, B1, E, L
0083 T=1./8388608.

0084 S, S2 = 0.
0085 V1 = 1./I

0086 V2 = 1./NP
0087 NM = 1

0088 DO 1 J1=1, LC
0089 DO 2 I=1, LAC

0090 T = FPMCRV(T)
0091 J = K(I)

0092 2 X(I) = B1(J)*T + A1(J)
0093 F2 = 0.

0094 DO 3 J2=1, L
0095 F1(J2)=P(J2)

0096 DO 4 I=1, LAC

```

6.

```

0097      4 F1(J2)=F1(J2)*RNORMAL(X(I),K(I),J2)
0098      5 F2 = F2 + F1(J2)*.MP
0099      F2 = (V1,F2)*.V2

0100      S = S + F2
0101      1 S2 = S2+F2*F2

0102      DU 5 I=1,LAC
0103      J = K(I)

0104      5 OM = OM + B1(J)
0105      SLC = 1./LC

0106      RMK = OM*S*SLC
0107      EPS = (S2-S*S*SLC) / (LC-1)

0108      EPS=SQRT(EPS)*OM
0109      RETURN

0110      END

```

END OF SEGMENT, LENGTH 270, NAME RMK

```

0111      FUNCTION RNORMAL(X,I,J)
0112      DIMENSION RM(20,2),S1(20,2),SG(20,2)

0113      COMMON/BETA/ RM,S1,SG
0114      V = X-RM(I,J)

0115      RNORMAL = S1(I,J)*EXP(-V*V*SG(I,J))
0116      RETURN

0117      END

```

END OF SEGMENT, LENGTH 64, NAME RNORMAL

```

0118      SUBROUTINE MISS(N,L,M,MM)
0119      DIMENSION M(N),MM(N)

0120      DU 11 = 1,L
0121      1 MM(I) = M(I)

0122      MM(L) = 0
0123      IF (L.EQ.N) RETURN

0124      DU 21 = 1,N-1
0125      2 MM(I) = M(I+1)

0126      MM(N) = 0
0127      RETURN

0128      END

```

END OF SEGMENT, LENGTH 113, NAME MISS

0129 FINISH

END OF COMPILATION = NO ERRORS 1 COMMENT

S/C SURFILE1

31 BUCKETS USED

CONSOLIDATED BY XPCK 120 DATE 20/04/76 TIME 09/38/34

PROGRAM 2005
EXTENDED DATA (22AM)
COMPACT PROGRAM (DRM)
CORE 6208

SEG	SELFZWS
CUV	ALA
SEG	RHK
SEG	WSTEPNY
SEG	ITIME
SEG	SELECTIONM1
CUV	BETA
SEG	AMIN1
SEG	AMAX1
SEG	MISS
SEG	FPMCRV
SEG	RNORMAL
SEG	SHORT
SEG	EXP

Program Z006

1. Funkcja programu

Program Z006 realizuje suboptymalny względem ryzyka, ale optymalny względem zdefiniowanego w pracy oszacowania ryzyka $M_K(J/X)$, algorytm selekcji przy zadanej liczbie $\epsilon > 0$, o jaką może zmienić się wartość M_K . Program wyznacza maksymalną liczbę cech do odrzucenia. W programie założono N cech niezależnych, każda o rozkładzie normalnym $N(m, \sigma')$

m - wartość oczekiwana

σ' - odchylenie standardowe

2. Wykorzystanie urządzeń zewnętrznych

1 czytnik kart

1 drukarka wierszowa

3. Budowa programu

Program składa się z 6 segmentów

a/ Segment główny MASTER SELEEPSZW6 - spełnia on rolę inicjującą, przejmuje częściowe sterowanie programem oraz drukuje wyniki programu.

b/ Segment SUBROUTINE WSTĘPNY - wczytuje dane do całego programu, dokonuje ich wstępnych przekształceń, drukuje wczytane dane oraz definiuje wspólne bloki.

c/ Segment SUBROUTINE SELECTIONM2 - jest najistotniejszym segmentem programu. Może być wykorzystany do selekcji cech o dowolnych rozkładach i o N nawet powyżej 1000.
/patrz rozdział V/.

d/ Segment SUBROUTINE MISS - spełnia rolę pomocniczą dla segmentu SELECTIONM2.

e/ Segment FUNCTION RMK - oblicza wartości miary M_K

w oparciu o metodę Monte-Carlo

f/ Segment FUNCTION RNORMAL - oblicza funkcje gęstości dla

odpowiednich x oraz i, j /cech i klas/

Wspólne bloki programu to,

a/ ALA - inf. o $N, M, LC, NP, P, A1, B1, E, L$

b/ BETA - zawiera przekształcone parametry rozkładów.

4. Dane wejściowe programu

E - liczba o jaką może zmienić się wartość M_K

N - liczba cech w obrazie

M - pewna liczba typu Integer

LC - liczba generowanych punktów w metodzie Monte-Carlo

obliczania całek wielowymiarowych, liczba ta implikuje

dokładność obliczania miary M_K (J/X)

NP - wielkość potęgi występującej w wyrażeniu na M_K , $K = NP$

$P(1), P(2)$ - prawdopodobieństwo a priori wystąpienia 1 i 2 klasy

m_{i1}, m_{i2} - wartość oczekiwana i -tej cechy odpowiednio

w 1 i 2 klasie

σ_{i1}, σ_{i2} - odchylenie standardowe " " "

a_{i1}, a_{i2} - dolna granica całkowania " " "

b_{i1}, b_{i2} - górna " " " "

5. Przygotowanie nośników danych

Dane należy przygotować na kartach perforowanych w kolejności

$E, N, M, LC, P, P(1), P(2)$

$m_{11}, 11, a_{11}, b_{11}, m_{12}, \sigma_{12}, a_{12}, b_{12}$

$m_{21}, 21, a_{21}, b_{21}, m_{22}, \sigma_{22}, a_{22}, b_{22}$

⋮

$m_{N1}, N1, a_{N1}, b_{N1}, m_{N2}, \sigma_{N2}, a_{N2}, b_{N2}$

Wszystkie dane pisane są w formacie swobodnym.

Dane N, M, IC, NP - są typu Integer, pozostałe typu Real.

Dane mogą być drukowane na kartach w sposób dowolny.

Nie ma żadnych przeciwwskazań co do ilości danych na jednej karcie, dopuszcza się rozdzielenie danych na poszczególne karty wg uznania użytkownika.

6. Wyniki

Wyniki wyprowadzane są w postaci wydruków na LP i zawierają

1. Wczytane wartości E, N, M, IC, NP, P(1), P(2), parametry rozkładów
2. Numery odrzuconych cech oraz ich ilość
3. Czas

7. Wykorzystanie pamięci m.c. i ograniczenia

Program wykorzystuje 6500 komórek pamięci, umożliwia rozwiązanie problemów selekcji dla

$$N \leq 20, 1 \leq M < N$$

Ograniczenia z góry na IC wynikają wyłącznie z czasu obliczeń, a z dołu ze względu na dokładność obliczeń.

Postuluje się

$$IC = K^{N-M}, \quad K \leq 20$$

Przy małych K, postuluje się dobierać małe przedziały

$[a_{ij}, b_{ij}]$ całkowania,

$$[a_{ij}, b_{ij}] = [m_{ij} - L_1 \sigma_{ij}, m_{ij} + L_2 \sigma_{ij}]$$

$$L_1, L_2 \sim 2,$$

$$\text{przy dużych } K, \quad L_1, L_2 \gg 3$$

Dokładniejsza analiza przyjętych L_1, L_2 opierać się musi na znajomości odpowiednich rozkładów.

FORTRAN COMPILATION BY #XFAT MK 5A . DATE 20/04/76 TIME 08/40/31

```

0001      LIST
0002      LIBRARY(SUBGROUPFSCE)
0003      PROGRAM(Z006)
0004      COMPRESS INTEGER AND LOGICAL
0005      INPUT 1 = CR0
0006      OUTPUT 2 = LPO
0007      TRACE 2
0008      END

0009      MASTER SELEEPSZW6
0010      DIMENSION NCO(20),NCA(20),K(20),NC(20),P(2),A1(20),B1(20)
0011      COMMON/ALA/N,M,LC,NP,P,A1,B1,E,L
0012      EXTERNAL RMK
0013      WRITE (2,12)
0014      CALL WSTEPNY
0015      CALL ITIME(ITP)
0016      DO 11 = 1,N
0017      1 NC(1) = 1
0018      CALL SELECTIONM2(NC,NCO,NCA,K,N,E,RMK,K1)
0019      WRITE (2,11)K1,(NCO(I),I = 1,K1)
0020      11 FORMAT(///6X,18HLCZBA ODrZ, CECH:,15//5X,25HNUMBER ODrZUCONYCH
0021      1CECH:,6X,2015)
0022      CALL ITIME(ITK)
0023      IT=ITK-ITP
0024      WRITE(2,100) IT
0025      100 FORMAT(//7HCZAS = ,110)
0026      12 FORMAT (////)
0027      STOP
0028      END

```

END OF SEGMENT, LENGTH 89, NAME SELEEPSZW6

```

0029      SUBROUTINE WSTEPNY
0030      DIMENSION P(2),RM(20,2),SIG(20,2),A(20,2),B(20,2),SI(20,2),
0031      1SG(20,2),A1(20),B1(20)
0032      COMMON /ALA/ N,M,LC,NP,P,A1,B1,E,L
0033      COMMON /BETA/ RM,SI,SG

```

```

0034          L=2
0035          READ(1,5) E,N,M,LC,NP,(P(I),I=1,L),((RM(I,J),SIG(I,J),A(I,J),R(I,
0036          1),J=1,L),I=1,N)
0037          3 FORMAT (F0.0,4I0,1000F0.0)
COMMENT 178    IS THIS LARGE A REPEAT COUNT INTENDED AT ABOUT COLUMN 26, LINE 0037

0038          WRITE(2,4) E,N,M,LC,NP,(P(I),I=1,L),((RM(I,J),SIG(I,J),A(I,J),
0039          1B(I,J),J=1,L),I=1,N)
0040          4 FORMAT (/9X,2HE=E20.11,5X,10HM,M,LC,NP=,4I7/9X,8HPR,APR.=,
0041          12F8.3/9X,14HRM, SIG, A, B=/(9X,4F12.3,10X,4F12.3))
0042          DO 1 I=1,N
0043          A1(I) = AMIN1(A(I,1),A(I,2))
0044          B1(I) = AMAX1(B(I,1),B(I,2))
0045          1 B1(I) = B1(I)-A1(I)
0046          DO 2 J=1,N
0047          DO 2 J=1,L
0048          SI(I,J) =.3989422804014/SIG(I,J)
0049          2 SG(I,J) =.5/(SIG(I,J)*SIG(I,J))
0050          WRITE(2,5) (I,A1(I),B1(I),I=1,N)
0051          5 FORMAT (/9X,7HA1,B1 =/(5(5X,12,2F8.3)))
0052          RETURN
0053          END

```

END OF SEGMENT, LENGTH 337, NAME WSTEPNY - COMMENTS

```

0054          SUBROUTINE SELECTIONM2(NC,NCO,NCA,K,N,E,FUN,K1)
0055          DIMENSION NCO(N),NC(N),NCA(N),K(N)
0056          K1 = 0
0057          DO 1 I = 1,N
0058          1 NCA(I) = NC(I)
0059          W1 = FUN(N,NCA,EPS)
0060          WRITE (2,4) W1,EPS
0061          4 FORMAT (/5X,9HMK(JIX) = ,E24.11,5X,5HEPS = ,E24.11/)
0062          N1 = N-1
0063          DO 2 J = 1,N1
0064          IN = N+1-J
0065          U = 0
0066          DO 3 I = 1,IN
0067          CALL MISS(IN,I,NCA,K)
0068          LAC = IN-1
0069          X = FUN(LAC,K,EPS)
0070          IF (W,GT.X) GO TO3
0071          U = X
0072          INDEX = I
0073          3 CONTINUE
0074          IF((W1-W).GT.E) RETURN
0075          K1 = K1+1

```



```

0076      HCU(J) = NCA(INDEX)
0077      2 CALL MISS (IN,INDEX,NCA,NCA)
0078      RETURN
0079      END

```

END OF SEGMENT, LENGTH 212, NAME SELECTIONH2

```

0080      FUNCTION RMK (LAC,K,EPS)
0081      DIMENSION P(2),A1(20),B1(20),K(LAC), F1(2),X(20)
0082      COMMON /ALA/ N,M,LC,NP,P,A1,B1,E,L
0083      T=1./8388608.
0084      S,S2 = 0.
0085      V1 = 1./L
0086      V2 = 1./NP
0087      OM = 1
0088      DO 1 J1=1,LC
0089      DO 2 I=1,LAC
0090      T = EPOCHRV(T)
0091      J = K(I)
0092      2 X(1) = B1(J)*T + A1(J)
0093      F2 = 0.
0094      DO 3 J2=1,L
0095      F1(J2)=P(J2)
0096      DO 4 I=1,LAC
0097      4 F1(J2)=F1(J2)*RNDNORMAL(X(1),K(1),J2)
0098      3 F2 = F2 + F1(J2)**NP
0099      F2 = (V1+F2)+*V2
0100      S = S + F2
0101      1 S2 = S2+F2*F2
0102      DO 5 I=1,LAC
0103      J = K(I)
0104      5 OM = OM + B1(J)
0105      SLC = 1./LC
0106      RMK = OM*S*SLC
0107      EPS = (S2-S*S*SLC) / (LC-1)
0108      EPS=SQRT(EPS)*OM
0109      RETURN
0110      END

```

END OF SEGMENT, LENGTH 270, NAME RMK

```

0111      FUNCTION RNDNORMAL(X,I,J)
0112      DIMENSION R0(20,2),SI(20,2),SG(20,2)
0113      COMMON/BETA/ R0,SI,SG
0114      U = V-RNDT.IX

```

```

0115      RNORMAL = SI(I,J)*EXP(-V*V*SG(I,J))
0116      RETURN
0117      END

```

END OF SEGMENT, LENGTH 64, NAME RNORMAL

```

0118      SUBROUTINE MISS(N,L,M,MM)
0119      DIMENSION M(N),MM(N)
0120      DO 1 I = 1,L
0121      1 MM(I) = M(I)
0122      MM(L) = 0
0123      IF (L.EQ.N) RETURN
0124      DO 2 I = L,N-1
0125      2 MM(I) = M(I+1)
0126      MM(N) = 0
0127      RETURN
0128      END

```

END OF SEGMENT, LENGTH 113, NAME MISS

0129 FINISH

END OF COMPILATION = NO ERRORS 1 COMMENT

S/C SUBFILE: 32 BUCKETS USED

CONSOLIDATED BY XPC6 12D DATE 20/04/76 TIME 08/40/56

ICLFORTRAN (0) RENAMED ICL-DEFAULT(0)

PROGRAM Z006
 EXTENDED DATA (Z2AM)
 COMPACT PROGRAM (DBM)
 CORE 6272

```

SEG      SELEEPSZW6
CUV      ALA
SEG      RMK
SEG      WSTEPHY
SEG      ITIME
SEG      SELECTIONM2
CUV      BETA
SEG      AMIN1
SEG      AMAX1
SEG      MISS
SEG      FPMCRV
SEG      RNORMAL
SEG      SORT
SEG      EXP

```


Program Z007

1. Funkcja programu

Program Z007 realizuje suboptymalny algorytm selekcji w oparciu o oszacowanie $\bar{\delta}$. W programie przyjęto N cech niezależnych o rozkładach normalnych $N(m, \sigma)$

m - wartość oczekiwana

σ - odchylenie standardowe

2. Wykorzystanie urządzeń zewnętrznych

1 czytnik kart

1 drukarka wierszowa

3. Budowa programu

Program składa się z 4 segmentów i 3 bloków wspólnych.

a/ Segment główny MASTER SELEZW7 - spełnia on rolę inicjującą, przejmuje częściowe sterowanie programem oraz drukuje wyniki programu.

b/ Segment SUBROUTINE WSTEPNY - wczytuje dane do programu dokonuje ich wstępnych przekształceń, drukuje wczytane wartości oraz definiuje wspólne bloki.

c/ Segment SUBROUTINE WYBORNAJM - jest kluczowym segmentem programu, dokonuje on wyboru zadanych M cech do odrzucenia /patrz rozdział V/.

d/ Segment FUNCTION RNORMALMOD, dokonuje obliczeń modułu różnicy funkcji gęstości w klasie pierwszej i drugiej.

Ad.a/ Segment ten oblicza również za pomocą procedury bibliotecznej F4INTSMP wartości odpowiednich oszacowań dla poszczególnych cech.

Wspólne bloki

a/ ALA

b/ BETA

c/TOLA

4. Dane wejściowe programu

E - zadana dokładność obliczania całek metodą F4INTSMP

IND-początkowa liczba w F4INTSMP

N - liczba cech w obrazie

M - dodana liczba cech do odrzucenia

m_{11}, m_{12} - wartość oczekiwana i-tej cechy odp.w klasie 1 i 2

σ_{11}, σ_{12} - odchylenie standardowe " " " " " "

a_{11}, a_{12} - dolna granica całkowania " " " " " "

b_{11}, b_{12} - górna " " " " " "

EPS- dowolna liczba rzeczywista

5. Przygotowanie nośników danych

Dane należy przygotować na kartach perforowanych w kolejności

E, IND, N, M

$m_{11}, \sigma_{11}, a_{11}, b_{11}, m_{12}, \sigma_{12}, a_{12}, b_{12}$

$m_{21}, \sigma_{21}, a_{21}, b_{21}, m_{22}, \sigma_{22}, a_{22}, b_{22}$

⋮

$m_{N1}, \sigma_{N1}, a_{N1}, b_{N1}, m_{N2}, \sigma_{N2}, a_{N2}, b_{N2}$

EPS

Wszystkie dane pisane są na kartach w formacie swobodnym.

Dane IND, N, M - są typu Integer, pozostałe typu Real.

Dane mogą być drukowane na kartach w sposób dowolny.

Nie ma żadnych przeciwwskazań co do ilości danych na jednej karcie, dopuszcza się rozdzielenie danych na poszczególne karty wg uznania użytkownika.

6. Wyniki

Wyniki wyprowadzane są w postaci wydruków na LP i zawierają

1. Wczytane wartości /wszystkie/

2. Numery odrzuconych cech

3. Wartość oszacowania po selekcji

4. Czas.

7. Wykorzystanie pamięci m.c. i ograniczenia

Program wykorzystuje 6000 komórek pamięci, umożliwia
rozwiązanie problemów selekcji dla

$$N \leq 20, \quad 1 \leq M < N$$

/Można zwiększyć rozmiar zadań poprzez zadeklarowanie
większych tablic/

Dokładność wyznaczania odpowiednich całek zależy od
wartości wczytanego E.

FORTRAN COMPILATION BY #XFAT MK 5A DATE 12/06/76 TIME 11/00/51

```

0001      LIST
0002      LIBRARY(SUBGROUPFSCE)

0003      PROGRAM (Z007)
0004      COMPRESS INTEGER AND LOGICAL

0005      INPUT1=CR0
0006      OUTPUT2=LPO

0007      TRACE 2
0008      END


0009      MASTER SELEZW7
0010      DIMENSION K(20),R(20),A1(20),B1(20),S1(20),SI(20,2),SG(20,2),ST(2
0011      1),ST1(2),RM1(2),RM(20,2)
0012      COMMON /ALA/ N,M,A1,B1,E,IND,EPS

0013      COMMON /BETA/ RM,SI,SG
0014      COMMON /TOLA/ ST,ST1,RM1

0015      EXTERNAL RNORMALMOD
0016      CALL WSTEPNY

0017      5 CALL ITIME(ITP)
0018      DO 1 I=1,N

0019      DO 3 J=1,2
0020      ST1(J)=SG(I,J)

0021      ST(J)=SI(I,J)
0022      8 RM1(J)=RM(I,J)

0023      CALL F4 INTS MP(A1(I),B1(I),RNORMALMOD,E,IND,R1)
0024      1 R(I)=R1

0025      DO 4 I=1,N
0026      4 K(I)=I

0027      CALL SHELLROSR2(N,R,K)
0028      WRITE(2,2)(K(I),R(I),I=1,M)

0029      2 FORMAT(10X,39HNUMERY CECH ODRZUC,, WARTOSCI OSZACOWAN,(/5(17,E15.5
0030      1)))

0031      CALL ITIME(ITK)
0032      IT=ITK-ITP

0033      WRITE(2,100) IT
0034      100 FORMAT(/7HCZAS = ,I10)

0035      READ(1,102)M
0036      102 FORMAT(I0)

```



```

0037      WRITE(2,101)M
0038      101  FORMAT(//,5X,3HM=,15)

0039      IF(M.NE.0) GO TO 5
0040      STOP

0041      END

```

END OF SEGMENT, LENGTH 201, NAME SELEZW7

0042	SUBROUTINE SHELIROSR2(N,R,M)	27
0043	DIMENSION R(N),M(N)	28
0044	K=1	29
0045	X=R(1)	30
0046	DO 6 I=2,N	31
0047	IF(X.LE.R(I)) GO TO 6	32
0048	X=R(I)	33
0049	K=1	34
0050	6 CONTINUE	35
0051	M(K)=R(1)	36
0052	L=M(K)	37
0053	M(K)=M(1)	38
0054	R(1)=X	39
0055	M(1)=L	40
0056	DO 7 I=2,N	41
0057	J1=1	42
0058	J2=1	43
0059	X=R(1)	44
0060	1 J=(J1+J2)/2	45
0061	IF(J.E4.J1) GO TO 3	46
0062	IF(X.LT.R(J)) GO TO 2	47
0063	J1=J	48
0064	GO TO 1	49
0065	2 J2=J	50
0066	GO TO 1	51
0067	3 K=1	52
0068	J2=J+2	53
0069	L=M(1)	54
0070	4 IF(K.LT.J2) GO TO 5	55
0071	R(K)=R(K-1)	56
0072	M(K)=M(K-1)	57
0073	K=K-1	58
0074	GO TO 4	59
0075	5 R(J+1)=X	60
0076	7 M(J+1)=L	61
0077	RETURN	62
0078	END	63

END OF SEGMENT, LENGTH 284, NAME SHELLROSR2

```

0079      SUBROUTINE WSTEPNY
0080      DIMENSION RM(20,2),SIG(20,2),A(20,2),B(20,2),SI(20,2),SG(20,2),A1(
0081      120),B1(20)
0082      COMMON /ALA/N,M,A1,B1,E,IND,EPS
0083      COMMON /BETA/ RM,SI,SG
0084      READ(1,3) E,IND,N,M,((RM(I,J),SIG(I,J),A(I,J),B(I,J),J=1,2),I=1,N)
0085      1,EPS
0086      3 FORMAT(F0.0,3I0,10U0F0.0)

COMMENT 178      IS THIS LARGE A REPEAT COUNT INTENDED AT ABOUT COLUMN 25, LINE 0086

0087      WRITE(2,4) E,IND,N,M,((RM(I,J),SIG(I,J),A(I,J),B(I,J),J=1,2),I=1,N
0088      1)
0089      4 FORMAT(///5X,2HF=,E10.7,5X,5HIND= ,15,3X,3HN= ,15,3HM= ,15//5X,11H
0090      1RM,SIG,A, B/(9X,4F12.3,5X,4F12.3))
0091      WRITE(2,5) EPS
0092      5 FORMAT( 5X,5HEPS= ,F20.11)

0093      DO 1 I=1,N
0094      A1(I)=AMIN1(A(I,1),A(I,2))

0095      1 B1(I)=AMAX1(B(I,1),B(I,2))
0096      DO 2 I=1,N

0097      DO 2 J=1,2
0098      SI(I,J)=.3989422804014/SIG(I,J)

0099      2 SG(I,J)=.5/(SIG(I,J)* SIG(I,J))
0100      RETURN
0101      END

```

END OF SEGMENT, LENGTH 257, NAME WSTEPNY - COMMENT

```

0102      FUNCTION RNORMALMOD(X)
0103      DIMENSION ST(2),ST1(2),RM1(2),V(2),F(2)

0104      COMMON /TOLA/ ST,ST1,RM1
0105      DO 1 J=1,2

0106      V(J)= X-RM1(J)
0107      1 F(J)= ST(J)*EXP(-V(J)*V(J)*ST1(J))

0108      F1=F(1)-F(2)
0109      RNORMALMOD =ABS(F1)

0110      RETURN
0111      END

```

END OF SEGMENT, LENGTH 98, NAME RNORMALMOD

0112 FINISH

-47-

END OF COMPILATION - NO ERRORS 1 COMMENT

S/C SUBFILE1 25 BUCKETS USED

CONSOLIDATED BY XPCK 120 DATE 12/06/76 TIME 11/01/20

PROGRAM Z007
EXTENDED DATA (22AM).
COMPACT PROGRAM (DBM)
CORE 5888

SEG	SELEZH7
CUV	ALA
CUV	BETA
CUV	TOLA
SEG	RNORMALMOD
SEG	WSTEPNY
SEG	ITIME
SEG	F4INTSMP
SEG	SHELLROSR2
SEG	AMIN1
SEG	AMAX1
SEG	EXP
SEG	ABS

Program Z008

1. Funkcja programu

Program Z008 realizuje suboptymalny algorytm selekcji w oparciu o oszacowanie $\bar{\delta}$. W programie przyjęto N-nieza-
leżnych cech o rozkładach normalnych $N(m, \sigma)$

m - wartość oczekiwana,

σ - odchylenie standardowe.

Program wyznacza suboptymalną liczbę oraz numery cech, które można odrzucić, aby wzrost ryzyka nie przekroczył zadanej wartości $\epsilon > 0$.

2. Wykorzystanie urządzeń zewnętrznych

1 czytnik kart

1 drukarka wierszowa

3. Budowa programu

Program składa się z 7 segmentów oraz 3 bloków wspólnych

a/ Segment główny MASTER SELEZW8 - spełnia rolę inicjującą, przejmuje częściowe sterowanie programem, oblicza wartości oszacowań dla każdej cechy wykorzystując procedurę biblioteczną liczenia całek F4NTSMP, drukuje wyniki. Zadaje wartość IND- ilości początkowe punkty w F4INTSMP

b/ Segment SUBROUTINE WSTĘPNY - wczytuje dane do programu, dokonuje ich wstępnych przekształceń, drukuje te dane oraz definiuje wspólne bloki.

c/ Segment SUBROUTINE SHELLROSR2 - szereguje rosnąco tablicę oszacowań dla każdej z cech oraz wyznacza tablicę indeksów odpowiadającą rosnącemu ciągowi oszacowań.

- d/ Segment SUBROUTINE OBLRYZ - oblicza ryzyko Bayes'a w oparciu o metodę Monte-Carlo liczenia całek wielowymiarowych.
- e/ Segment SUBROUTINE SELEKCJAM3 - jest najistotniejszym segmentem programu, wyznacza liczbę oraz numery cech do odrzucenia. Dokładne omówienie tego segmentu przedstawiono w rozdziale V.
- f/ Segment FUNCTION RNORMAL - oblicza wartości poszczególnych funkcji gęstości w punkcie X_0
- g/ Segment FUNCTION RNORMALMOD - oblicza moduł różnicy odpowiednich funkcji gęstości w punkcie X .

Wspólne bloki

a/ ALA

b/ BETA

c/ TOLA

4. Dane wejściowe programu

E - zadana dokładność obliczania całki metodą F4INTSMP

EPS- zadana wartość o jaką może wzrosnąć ryzyko (ϵ)

N - liczba cech w obrazie

LC- liczba generowanych punktów w metodzie Monte-Carlo

P(1), P(2)- prawdopodobieństwo a priori wystąpienia klas 1 i 2

m_{11} , m_{12} - wartość oczekiwana i-tej cechy odp.w 1 i 2 klasie

σ_{11} , σ_{12} - odchylenie standardowe " " " " " "

a_{11} , a_{12} - dolna granica całkowania " " " " " "

b_{11} , b_{12} - górna " " " " " "

5. Przygotowanie nośników danych

Dane należy przygotować na kartach perforowanych w kolejności

E, EPS, N, LC, P(1), P(2)

m_{11}	σ_{11}	a_{11}	b_{11}	m_{12}	σ_{12}	a_{12}	b_{12}
m_{21}	σ_{21}	a_{21}	b_{21}	m_{22}	σ_{22}	a_{22}	b_{22}
$\vdots$							
m_{N1}	σ_{N1}	a_{N1}	b_{N1}	m_{N2}	σ_{N2}	a_{N2}	b_{N2}

Wszystkie dane pisane są na kartach w formacie swobodnym.

Dane N, LC - typu Integer, pozostałe typu Real.

Dane mogą być drukowane na kartach w sposób dowolny, nie ma żadnych przeciwwskazań co do ilości danych na jednej karcie.

Dopuszcza się rozdzielanie danych na poszczególne karty według uznania użytkownika.

6. Wyniki

Wyniki wyprowadzane są w postaci wydruków na LP i zawierają

1. Wczytane wartości /wszystkie/
2. Liczbę odrzuconych cech
3. Numery odrzuconych cech
4. Wartość ryzyka po selekcji
5. Czas obliczeń.

7. Wykorzystanie pamięci

Program wykorzystuje 6600 komórek pamięci, umożliwia rozwiązanie problemów selekcji dla

$$N \leq 20, \quad 1 \leq M < N$$

/Można zwiększyć rozmiar zadań poprzez zadeklarowanie większych odpowiednich tablic. Ograniczenia z góry na LC wynikają wyłącznie z czasu obliczeń a z dołu ze względu na dokładność obliczeń ryzyk. Postuluje się

$$LC = K^N - M, \quad K \geq 20$$

Przedziały całkowania zaś

$$[a_{ij}, b_{ij}] = [m_{ij} - L_1 \sigma_{ij}, m_{ij} + L_2 \sigma_{ij}] \quad , \text{ dla małych } k \sim 20, \\ L_1, L_2 \sim 2 \quad , \text{ dla dużych } k, \quad L_1, L_2 \geq 3$$

FORTRAN COMPILATION BY #XEAT MK 5A DATE 16/07/76 TIME 15/22/17

0001	LIST	1
0002	LIBRARY(SUBGROUPESCE)	
0003	PROGRAM(700R)	3
0004	COMPRESSION INTEGER AND LOGICAL	
0005	INPUT1=CR0	4
0006	OUTPUT2=,P0	5
0007	TRACE 2	6
0008	END	7
0009	MASTER SEIFZW8	8
0010	DIMENSION P(2),A1(20),B1(20),M(20),RM(20,2),S1(20,2),SG(20,2),ST(2	
0011	1),ST1(2),RM1(2),N2(20),R(20)	
0012	COMMON /TOLA/ ST,ST1,RM1	
0013	COMMON /BETA/ RM,S1,SG	
0014	COMMON /AIA/ EPS,N,IC,P,A1,B1,E	10
0015	EXTERNAL RNORMAIMOD	11
0016	CALL WSTEPNY	12
0017	IND=3	
0018	DO 5 I=1,N	
0019	5 M(I)=1	14
0020	CALL UNLBYZ(0,M,R0,F1)	15
0021	WRITE(2,4) R0,F1	

0022	4	FORMAT(///5X,12HRYZYKQ CAIK.,2X,F20.11,5HBIAD:,F20.11)	17
0023		DO 1 I=1,N	18
0024		DO 3 J=1,2	
0025		ST(I)= ST(I,I)	
0026		ST1(J)=SG(I,I)	
0027	3	RM1(J)=RM(I,I)	
0028		CALL F4INTSMP(A1(I),R1(I),RNORMALMOD,F,IND,P1)	19
0029	1	R(I)=R1	20
0030		CALL SHEIROS2(N,R,M)	21
0031		CALL SEIFKCIAM3(M,K,R0)	22
0032		WRITE(2,2) (M(I),I=1,K)	23
0033	2	FORMAT(///5X,16HODR7UCONE CFCHY:,2015)	24
0034		STOP	25
0035		END	26

END OF SEGMENT, LENGTH 169, NAME SEIFZWR

0036		SUBROUTINE SHEIROS2(N,R,M)	27
0037		DIMENSION R(N),M(N)	28
0038		K=1	29
0039		X=R(1)	30
0040		DO 4 I=2,N	31
0041		IF(X.LE.R(I)) GO TO 6	32
0042		X=R(I)	33
0043		K=I	34
0044	6	CONTINUE	35
0045		R(K)=R(1)	36
0046		I=M,K	37
0047		M(K)=M(1)	38
0048		R(1)=X	39
0049		M(1)=I	40
0050		DO 7 I=2,N	41
0051		J1=1	42
0052		J2=1	43
0053	1	X=(R(J1)+J2)/2	44
0055		IF(1.F0.J1) GO TO 3	46

056	IF(V.LT.R(J)) GO TO 2	47
057	J1=1	48
058	GO TO 1	49
059	2 J2=1	50
060	GO TO 1	51
061	3 K=1	52
062	J2=1+2	53
063	I=M(I)	54
064	4 IF(V.LT.J2) GO TO 5	55
065	R(K)=R(K-1)	56
066	M(K)=M(K-1)	57
067	K=K-1	58
068	GO TO 4	59
069	5 R(J1)=X	60
070	7 M(J1)=L	61
071	RETURN	62
072	END	63

END OF SEGMENT LENGTH 284 NAME SHEILROSR2

0073	SUBROUTINE WSTFNY	64
0074	DIMENSION P(2),RM(20,2),SIG(20,2),A(20,2),B(20,2),SI(20,2),SG(20,2),	65
0075	1),A1(20),R1(20)	66
0076	COMMON /A1 A/ EPS,N,LC,P,A1,R1,E	67
0077	COMMON /REF A/ RM,SI,SG	68
0078	I=2	69
0079	READ(1,3) E, EPS, N, LC, (P(I), I=1,1), ((RM(I,J), SIG(I,J), A(I,J), B(I,J),	70
0080	1), I=1,1), I=1,N)	71
0081	3 FORMAT(2E0.0,2I0,500E0.0)	72

COMMENT 178 IS THIS LARGE A REPEAT COUNT INTENDED AT ABOUT COLUMN 25, LINE 0081

0082	WRITE(2,4) E, EPS, N, LC, (P(I), I=1,1),	73
0083	1((RM(I,J), SIG(I,J), A(I,J), B(I,J), J=1,1), I=1,N)	74
0084	4 FORMAT(//9X,2HF=,F10.3,5X,4HFPS=,F10.3,5X,16HN: (E,2I7/8X,16HPR,ARR.	75
0085	1=,2FR,3/9X,14HRM, SIG, A, B=/(9X,4F12.3,10X,4F12.3))	76
0086	DO 1 I=1,N	77

0087	A1(I)=AHIN1(A(I,1),A(I,2))	78
0088	R1(I)=AMAX1(R(I,1),R(I,2))	79
0089	1 R1(I)=R1(I)-A1(I)	80
0090	DO 2 I=1,N	81
0091	DO 2 J=1,I	82
0092	S1(I,J)=.3989422804014/SIG(I,J)	83
0093	2 S2(I,J)=.5/(SIG(I,J)*SIG(I,J))	84
0094	RETURN	85
0095	END	86

END OF SEGMENT: LENGTH 289 NAME WSTEPNY - COMMENTS

0096	SUBROUTINE DRLV7(K,M,R2,F1)	87
0097	DIMENSION M(20),D(2),A1(20),R1(20),F1(2),X(20)	88
0098	COMMON /A1A/ EPS,N,IC,P,A1,R1,E	89
0099	T=1/R388608	90
0100	S+S2=0.	91
0101	DM=1	92
0102	DO 4 J1=1,LC	93
0103	DO 2 I=1,N-K	94
0104	T=EDMC RV(T)	95
0105	J=M/(K+1)	96
0106	2 X(I)=R1(J)*T+A1(J)	97
0107	F2=0.	98
0108	DO 3 J2=1,2	99
0109	F1(J2)=P(J2)	100
0110	DO 4 I=1,N-K	101
0111	1A=M/(K+1)	102
0112	4 F1(J2)=F1(J2)*RNORMAL(X(I),1A,J2)	103
0113	3 CONTINUE	104
0114	F2=AMAX1(F1(1),F1(2))	105
0115	S=S+F2	106
0116	1 S2=S2+F2+F2	107
0117	DO 5 I=1,N-K	108
0118	J=M/(K+1)	109
0119	5 DM=DM*R1(J)	110

0120	SIC = 1./IC	111
0121	W2 = 1 - OM*SIC*S	112
0122	F1 = (S2 - S + S*SIC)/(IC-1)	113
0123	F1 = SQRT(F1)*OM	114
0124	RETURN	115
0125	END	

END OF SEGMENT, LENGTH 289, NAME ORIRY2

0126	FUNCTION RNORMAL(X,I,J)	116
0127	DIMENSION SI(20,2),SG(20,2),RM(20,2)	117
0128	COMMON /RFTA/ RM,SI,SG	118
0129	V=X-RM(I,J)	119
0130	RNORMAL = SI(I,J)*EXP(-V*V*SG(I,J))	120
0131	RETURN	121
0132	END	122

END OF SEGMENT, LENGTH 64, NAME RNORMAL

0133	FUNCTION RNORMALMOD(X)	
0134	DIMENSION ST(2),ST1(2),RM1(2),V(2),F(2)	
0135	COMMON /TOLA/ ST,ST1,RM1	
0136	DO 1 J=1,2	126
0137	V(J) = X-RM1(J)	
0138	1 F(J) = ST(J)*EXP(-V(J)*V(J)*ST1(J))	
0139	F1 = F(1)-F(2)	129
0140	RNORMALMOD = ABS(F1)	130
0141	RETURN	131
0142	END	132

END OF SEGMENT, LENGTH 98, NAME RNORMALMOD

0143	SUBROUTINE SELEKJAM3(M,K,R0)	133
0144	DIMENSION M(20),D(2),A1(20),R1(20)	134
0145	COMMON /A1A/ EPS,N,IC,P,A1,R1,F	135

0146	K = 0	136
0147	1 K = K + 1	137
0148	CALL OBLRVZ(K,M,R2,F1)	138
0149	WRITE(2,4) K,R2,F1,M(K)	139
0150	4 FORMAT(/,5X,2HK=,15,5X,7HRV2YKO,,F20.11,5HRIAD,,F20.11,9HNR,CECHV,140	
0151	1,15)	141
0152	IF((R0-R).GT.EPS) GO TO 2	143
0153	IF(K.EQ.(N-1)) GO TO 3	144
0154	GO TO 1	145
0155	2 K = K - 1	146
0156	3 RETURN	147
0157	END	148

END OF SEGMENT: LENGTH 99, NAME SEIFKCJAM3

- 56 -

0158

FINISH

END OF COMPILATION - NO ERRORS 1 COMMENT

S/C SUBFILE:

35 BUCKETS USED

CONSOLIDATED BY XPC 120 DATE 16/07/76 TIME 15/22/52

PROGRAM 2008

C O M P A C T D A T A (15A M)

COMPACT PROGRAM (DRM)

CORE

6592

SFG	SEIFZWR
CIIV	TOI A
CIIV	RETA
CIIV	AIA
SFG	RNORMAIMON
SFG	WCTEPNY
SEG	OPIRYZ
SFG	FAINTSMP
SFG	SHEILROSR2
SFG	SEIFKCIAM3
SFG	AMTN1
SFG	AMAX1
SFG	EDMCRV
SFG	RNORMAI
SFG	SOBT
SFG	FYP
SFG	ARS

