Grzegorz Stawski

Probabilistyczne miary odległości, sieci neuronowe i cechy symboliczne.

Praca magisterska
wykonana pod kierunkiem Prof. Włodzisława Ducha

Katedra Metod Komputerowych
Uniwersytetu Mikołaja Kopernika
Toruń 2000

Spis treści.

1.Wstęp.

Proces klasyfikacji jest podstawową czynnością poznawczą wykonywaną przez człowieka. Polega on na podziale obiektów na grupy i kategorie. Klasyfikację można traktować jako wstęp do myślenia, uczenia się i podejmowania decyzji. Można powiedzieć, że dzięki procesowi klasyfikacji dokonujemy odkryć naukowych, konstruujemy teorie naukowe i hipotezy.

Obecnie proste komputery o stosunkowo małej mocy obliczeniowej są używane do sterowania różnymi urządzeniami (lodówka, silnik samochodu, kuchenka mikrofalowa itp.). Do takiego sterowania wystarczy kilka prostych reguł, jednakże od komputerów sterujących przyszłych generacji (np. komputer kierowca samochodu) będziemy wymagali „pseudo myślenia", tj. odpowiedniej reakcji na różne sytuacje, w których może znaleźć się urządzenie sterowane przez nasz komputer. W oprogramowaniu tego typu komputerów będą musiały się znaleźć wysoko wydajne algorytmy klasyfikacyjne.

Jednym z problemów, jaki stoi przed zastosowaniem sprawnych klasyfikatorów wywodzących się ze statystyki, sieci neuronowych czy uczenia maszynowego jest konieczność podawania danych w postaci numerycznej. Wady tej nie mają oparte na podobieństwie klasyfikatory typu najbliższych sąsiadów. Można w nich używać cech symbolicznych w sposób bezpośredni, korzystając z probabilistycznie określonych miar odległości.

Głównym celem tej pracy jest zbadanie możliwości zastosowania takich miar odległości dla dowolnych klasyfikatorów. Wymaga to określenia takich wartości numerycznych cech, które dają odległości obliczane za pomocą miary Euklidesowej identyczne jak odległości obliczane za pomocą miar probabilistycznych. Opracowano program komputerowy zamieniający symboliczne wartości atrybutów na wartości numeryczne i przetestowano go na szeregu bazach danych. Wstępne wyniki opublikowane zostały w pracy [1]

W poniższej pracy przedstawimy różne typy algorytmów klasyfikacyjnych, ich zalety oraz wady. W pierwszej części zaprezentowane zostaną klasyczne algorytmy klasyfikacyjne oraz próby usprawnienia tych algorytmów tak, aby mogły pracować ze zmiennymi opisanymi przy pomocy symboli. Kolejna część pracy poświęcona została różnym typom algorytmów nieklasycznych, potrafiących „obsługiwać" cechy symboliczne. W ostatniej części poniższej pracy zajmiemy się problemem przygotowania danych dla algorytmów klasyfikacyjnych, które mogą działać tylko na bazach danych, w których obiekty są opisane wartościami numerycznymi.

2. Podstawowe pojęcia.

Pojęcie klasyfikacji zostało po raz pierwszy wprowadzone przez biologów dla potrzeb semantyki roślin i zwierząt (pierwszą systematykę opracował Linneusz). Obecnie mówi się o całej dziedzinie nazwanej *taksonomią,* która zajmuje się zagadnieniami klasyfikacyjnymi. Termin *taksonomia* pochodzi od greckich słów: *taksis* (porządek) oraz *nomos* (prawo, zasada). Dziś metody klasyfikacyjne stosuje się w zupełnie innych dziedzinach niż biologia takich jak między innymi: nauki techniczne, fizyka, chemia, ekonomia, antropologia, psychologia, socjologia, lingwistyka i wiele innych.

Początkowo klasyfikacja opierała się na opisie cech przedstawicieli roślin i zwierząt i miała ona charakter wyłącznie opisowy (obiekty odróżniało się na podstawie ich jakościowego opisu). Obecnie algorytmy klasyfikacyjne są oparte wyłącznie na metodach ilościowych rozróżniania obiektów.

Jakie mamy rodzaje algorytmów klasyfikacyjnych? Są dwa rodzaje metod: metody tzw. klasyczne (ilościowe), oparte na statystyce oraz tzw. metody symboliczne (jakościowe). Są to zupełnie nowe metody, większość algorytmów tego typu powstała w ciągu ostatnich 10 lat i jest obecnie przedmiotem intensywnych badań. Następuje dalszy burzliwy rozwój znanych już metod jakościowych i dalsze poszukiwania nowych bardziej wydajnych metod.

W ramach samej klasyfikacji możemy w zależności od dostępności informacji wyróżnić dwa zagadnienia:

a) *Klasyfikację wykorzystującą znane wzorce*, nazywaną w statystyce analizą dyskryminacyjną – w tym przypadku znamy klasy, do których chcemy przydzielać obiekty z bazy treningowej. W terminologii cybernetycznej zagadnienie to nazywane jest *rozpoznawaniem z nauczycielem.* Algorytmowi klasyfikacyjnemu w procesie „uczenia" podawane są przykłady obiektów z jednoczesnym podaniem informacji, do której klasy należy dany obiekt przydzielić.

b) *Klasyfikację bezwzorcową* (zwaną w statystyce analizą skupień) – nic nie wiemy o strukturze klas i chcemy ją dopiero odkryć. Algorytm nie dysponuje żadnymi informacjami o klasach, do których należy przydzielić obiekty. Przydatne klasy muszą zostać przez algorytm dopiero skonstruowane.

W zależności od sposobu grupowania obiektów algorytmy klasyfikacyjne dzielimy na (podany poniżej podział jest prawdziwy zarówno do metod klasycznych jak i nieklasycznych):

a) *Optymalizacyjno-iteracyjne,* w których iteracyjnie dzieli się zbiór obiektów na grupy, przenosi się obiekty z jednej grupy do drugiej aż do uzyskania optymalnego podziału.

b) *Hierarchiczne,* w ramach których skupienia tworzą binarne drzewa, gdzie liście reprezentują poszczególne obiekty, a węzły - ich grupy. Skupienia wyższego poziomu zawierają w sobie skupienia niższego poziomu. Ta metoda jest bardzo często stosowana w algorytmach symbolicznych. Wśród hierarchicznych metod wyróżnia się dodatkowo:

- *Metody aglomeracyjne,* polegające na sukcesywnym łączeniu skupień (zakłada się, że początkowo każdy obiekt tworzy oddzielną klasę).

- *Metody podziałowe,* w ramach których początkowy zbiór obiektów (jedno skupienie) jest dzielony kolejno na dwie części aż do momentu, gdy każdy obiekt znajdzie się w oddzielnej klasie.

-*Drzewa klasyfikacyjne* budowane przez symboliczne algorytmy klasyfikacji wzorcowej.

c) Tworzące *skupienia nierozłączne* – w przypadku niektórych baz danych obiekty mogą należeć do więcej niż jednej grupy (klasy). Metoda ta jest stosowana przeważnie przez algorytmy symboliczne do problemów lingwistycznych (algorytmy do rozpoznawania tekstów i mowy).

Co uzyskujemy dzięki zastosowaniu algorytmów klasyfikacyjnych do baz danych? Algorytmy tego typu pozwalają odkryć „strukturę" baz danych. Wykryć zależności i korelacje występujące pomiędzy obiektami oraz cechami opisującymi te obiekty. Po wykonaniu takiej wstępnej analizy, dane są łatwe do interpretacji przez człowieka, łatwo zapisać prawa rządzące danymi w postaci prostej reguły lub wzoru np. $E = mc^2$. Algorytmy tego typu dają nam także możliwość łatwego i szybkiego budowania *systemów eksperckich.* Na podstawie już zebranych danych np. o klientach banku algorytmy klasyfikacyjne są w stanie odpowiedzieć na pytanie czy danemu klientowi warto udzielać pożyczki czy też nie.

Co jest przyczyną odchodzenia od metod klasycznych? Metody klasyczne dają dobre wyniki o ile operują wyłącznie na zmiennych określonych przy pomocy tzw. skal mocnych a większość baz danych, z jakimi mamy do czynienia w realnych przypadkach, zawiera dane różnych typów tzn. określonych przy pomocy różnych skal. Niektóre cechy opisujące obiekty mają charakter ciągły opisany przy pomocy liczb rzeczywistych (skala mocna, zmienna ilościowa) np. dla samochodów zużycie paliwa, przyspieszenie, itp. Inne z kolei będą opisywane za pomocą tzw. symboli (skala słaba, zmienna jakościowa) takich jak kolor, kształt itp. Użyta skala pomiarowa ma wpływ na ilość informacji dostarczaną przez zmienną. Dokładniejszy podział skal pomiarowych to:
-skale słabe:(a) nominalna, (b) porządkowa,
-skale mocne: (c) przedziałowa, (d) ilorazowa.

(a) Skala ta pozwala na najprostszy sposób rozróżniania obiektów. Do zmiennych wyrażonych w tej skali możemy stosować wyłącznie operatory $=, \neq$, czyli możemy stwierdzać czy dwa obiekty są sobie równe czy też nie. Zmienne nominalne pozwalają jedynie na zaliczenie poszczególnych obiektów (jednostek, osobników itd.) do jednej z rozróżnialnych kategorii, których nie możemy uporządkować. Typowymi przykładami zmiennych nominalnych są płeć, rasa, kolor itp. Zmienne tego typu są zapisywane najczęściej przy pomocy symboli.

(b) Zmienne porządkowe pozwalają na rangowanie (ustawianie w określonym porządku dzięki operatorom < , >) elementów, które mierzymy. Element z wyższą rangą posiada cechę reprezentowaną przez mierzoną zmienną w większym stopniu, lecz ciągle nie można powiedzieć o ile większym.

(c) Zmienne przedziałowe z kolei pozwalają nie tylko szeregować (rangować) mierzone elementy, lecz również mierzyć różnice wielkości pomiędzy nimi. Na przykład temperatura mierzona w stopniach Celsjusza jest zmienną przedziałową. Możemy powiedzieć, że temperatura 40 stopni jest wyższa niż temperatura 30 stopni a wzrost temperatury od 20 do 40 stopni jest dwa razy większy niż od 30 do 40 stopni.

(d) Zmienne ilorazowe są podobne do zmiennych przedziałowych, lecz oprócz wszystkich cech skali przedziałowej, charakteryzuje je istnienie punktu absolutnego zera skali, dzięki czemu prawdziwe są stwierdzenia typu x jest dwa razy większe niż y. Typowymi przykładami skal ilorazowych są skale przestrzeni i czasu, również skala Kelvina pomiaru temperatury jest skalą ilorazową.

W przypadku danych mieszanych (różne skale pomiarowe, zmienne wyrażone przy pomocy symboli i liczb), aby móc zastosować klasyczny algorytm musimy dokonać ujednolicenia danych, wykonać zamianę symboli (skala słaba) na wartości numeryczne. W tym momencie pojawia się następujący problem. W jaki sposób dokonać takiej zamiany, aby wynik klasyfikacji był jak najlepszy? Dokonując zastąpienia symboli cyframi możemy zniszczyć związki występujące pomiędzy cechami opisującymi dane, przez co uzyskany wynik klasyfikacji może być błędny. Metody klasyczne są rozwijane od wielu lat, algorytmy opracowane w ramach tych metod są stosunkowo wydajne i stabilne. Na przeszkodzie stosowania tych metod stoją cechy opisane za pomocą symboli, które występują w większości obecnie gromadzonych baz danych.

Zamiana danych symbolicznych powinna uwzględniać globalne własności zbioru i korelacje występujące pomiędzy cechami opisującymi obiekty. Większość algorytmów klasycznych do odróżniana obiektów stosuje odległościowe miary podobieństwa obiektów, tak więc w przypadku nie uwzględnienia dwóch powyższych czynników otrzymujemy fałszywe odległości obiektów a co za tym idzie błędną klasyfikację lub słaby jej wynik. Metody klasyfikacji symbolicznej nie mają problemu z zmiennymi symbolicznymi, ponadto większość algorytmów tego typu dopuszcza nieznajomość wartości części atrybutów. Brak znajomości wartości niektórych atrybutów jest stosunkowo częstym zjawiskiem w rzeczywistych bazach danych (np. różne dane medyczne i psychiatryczne).

3. Klasyczne algorytmy klasyfikacyjne.

Co jest zadaniem algorytmu klasyfikacyjnego? Dokonanie procesu klasyfikacji oznacza znalezienie różnic pomiędzy obiektami i na tej podstawie, przydzielenie obiektów do poszczególnych kategorii i klas. Zadanie to nie jest wcale łatwe, nie wystarczy dokonać wszystkich możliwych podziałów zbioru obiektów na grupy, a następnie porównać je i wybrać najlepszy z nich. Liczba możliwych podziałów n obiektów na m klas jest bardzo duża i wyraża ją liczba Stirlinga drugiego rzędu:

$$S_n^{(m)} = \frac{1}{m!}\sum_{k=0}^{m}(-1)^{m-k}\binom{m}{k}k^n$$

Uwzględniając dodatkowo, że optymalna liczba klas może nie być znana, należy brać pod uwagę odpowiednią sumę liczb Stirlinga: $S_n^{(1)} + S_n^{(2)} + S_n^{(3)} + ... + S_n^{(m)}$. Już dla 25 obiektów liczba możliwych podziałów jest rzędu $\sim 10^{18}$.

3.1 Odróżnianie obiektów

Klasyczne algorytmy klasyfikacyjne do odróżniania obiektów stosują najczęściej różne geometryczne miary podobieństwa obiektów. Miary odległości są funkcjami obu rozważanych obiektów i charakteryzują się tym, że wzrost ich wartości oznacza zwiększenie stopnia zróżnicowania obiektów. Oprócz odległościowych miar podobieństwa można stosować tzw. współczynniki korelacyjne oraz miary asocjacyjne.

Miara odległości między dwoma obiektami x oraz y musi mieć następujące własności:

1) $d(x,y) = 0$ gdy $x = y$
2) $d(x,y) \geq 0$
3) $d(x,y) = d(y,x)$
4) $d(x,z) \leq d(x,y) + d(y,z)$

oraz powinna spełniać warunek: $d{:}\Omega \times \Omega \rightarrow R^+$. Zamiast miar odległościowych można także stosować tzw. *miary podobieństwa* (bliskości, zgodności). Miarą podobieństwa obiektów nazwiemy taką funkcję, której wartość rośnie, gdy maleją różnice między obiektami:

1) $0 \leq p(x,y) < 1$ dla $x \neq y$
2) $p(x,x) = 1$
3) $p(x,y) = p(y,x)$

Ponadto musi zachodzić: $p{:}\Omega \times \Omega \rightarrow R^+$

Miary odległościowe można stosować zamiennie z miarami podobieństwa, gdyż są one wzajemnie równoważne:

1) $p(x,y) = \dfrac{1}{1 + d(x,y)}$
2) $p(x,y) = \dfrac{1}{1 + d(x,y)^2}$

3) $p(x,y) = e^{-d(x,y)}$

4) $p(x,y) = c - d(x,y)$

$p(x,y)$ oznacza podobieństwo, $d(x,y)$ odległość. Zastąpienie miary odległości miarą podobieństwa nie zmienia istoty zagadnienia klasyfikacji i nie wpływa na jego wyniki. Stąd terminy „odległość" i „podobieństwo" można używać zamiennie, pamiętając o ich przeciwnym znaczeniu.

Najczęściej wykorzystywaną miarą odległości jest tzw. metryka *Minkowskiego*:

$$D(x,y;\alpha) = (\sum_{i=1}^{N} |x_i - y_i|^{\alpha})^{\frac{1}{\alpha}}$$

Gdzie x, y to obiekty (wektory), pomiędzy którymi wyznaczmy odległość, N jest liczbą cech opisujących obiekty, α to stała spełniająca warunek $\alpha > 0$.

Miara Euklidesowa jest szczególnym przypadkiem miary Minkowskiego dla $\alpha = 2$, inny często stosowany przypadek szczególny to miara Manhattan dla $\alpha = 1$, dla $\alpha \to \infty$ dostajemy funkcję Czebyszewa. W niektórych przypadkach, aby uzyskać dodatkowe parametry do optymalizacji klasyfikacji, do miary Minkowskiego wprowadza się czynniki skalujące.

Przykłady czynników korelacyjnych:

-funkcja Camberra $D_{Ca}(x,y) = \sum_{i=1}^{N} \frac{x_i - y_i}{x_i + y_i}$

-funkcja Czebyszewa $D_{Ch}(x,y) = \max_{i=1,..,N} |x_i - y_i|$

-odległość χ^2 $D_{\chi}(x,y) = \sum_{i=1}^{N} \frac{1}{s_i} \left(\frac{x_i}{s_x} - \frac{y_i}{s_y} \right)^2$ gdzie s_i jest sumą wartości cechy i w zbiorze trenującym, s_x i s_y są sumami wszystkich składników wektorów x i y

- korelacyjna miara odległości $D_K(x,y) = \frac{\sum_{i=1}^{N} (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^{N} (x_i - \bar{x}_i)^2 \sum_{i=1}^{N} (y_i - \bar{y}_i)^2}}$

W zależności od doboru funkcji określającej podobieństwo obiektów uzyskamy różne wyniki klasyfikacji obiektów. Po określeniu miary podobieństwa trzeba dla naszego problemu klasyfikacyjnego wybrać jedną z metod grupowania obiektów, to znaczy wybieramy procedurę, która nasz zbiór wejściowy podzieli na podzbiory obiektów najbardziej podobnych do siebie.

3.2 Grupowanie obiektów.

Wykorzystując określony sposób pomiaru odległości (podobieństwa) obiektów w wielowymiarowej przestrzeni cech, można podzielić klasyfikowany zbiór na podzbiory (grupy) tak, by zawierały one obiekty najbardziej do siebie podobne. W ramach klasycznych metod numerycznych i statystycznych liczba algorytmów realizujących podział obiektów jest duża.

Część algorytmów klasycznych w procesie grupowania optymalizuje pewną funkcję jakości podziału obiektów. Dąży się do tego, by zróżnicowanie (mierzone np. za pomocą wariancji) obiektów w grupach było jak najmniejsze, a między grupami - jak największe. Takie kryterium można zapisać w postaci formuły minimalizującej ślad macierzy wariancji wewnątrzgrupowej (W): $\min\{Tr(W)\}$ lub maksymalizującej ślad macierzy wariancji międzygrupowej (M): $\max\{Tr(M)\}$. Do najbardziej znanych metod optymalizacyjnych należą: metoda *K* średnich, metoda *K* centroidów, metoda Wisharta, metoda Thorndike' a [2].

Inne algorytmy budują w procesie grupowania pewną hierarchię zgrupowań. Z tej grupy najbardziej popularne są hierarchiczne metody aglomeracyjne, chociaż mogą być stosowane jedynie do stosunkowo małego zbioru obiektów, gdyż proces łączenia odbywa się w *N-1* krokach (*N* jest liczbą obiektów). Wśród metod aglomeracyjnych można wymienić np.

- Metodę najbliższego sąsiedztwa, w której odległość między skupieniami *A* i *B* jest równa odległości między dwoma najbliższymi obiektami należącymi do tych skupień:

$$d_{AB} = \min_{i,j}\{d(O_{A_i}, O_{B_j})\} \text{ gdzie } i = 1,..., n_A,\ j = 1,..., n_B \text{ są}.$$

O_{A_i} to obiekty ze skupienia A, O_{B_j} są obiektami skupienia *B*.

- Metodę najdalszego sąsiedztwa, odległość między skupieniami *A* i *B* to dystans między najbardziej odległymi obiektami należącymi do nich:

$$d_{AB} = \max_{i,j}\{d(O_{A_i}, O_{B_j})\}$$

- Metodę Warda, odległość pomiędzy skupieniami *A* i *B* jest sumą odległości poszczególnych obiektów od środków ciężkości skupień *A* i *B*:

$$d_{AB} = \frac{n_A n_B}{n_A + n_B}\sum_{i,j}(\overline{O}_{A_i} - \overline{O}_{B_j})^2$$

W wyniku stosowania większości metod grupowania powstają skupienia rozłączne, czasami jednak (np. w badaniach lingwistycznych) obiekty mogą należeć do więcej niż jednego skupienia. Metody tego typu produkują dwa podzbiory: skupienie zawierające obiekty podobne oraz reszta (tj. obiekty odległe od tych, które znajdują się we wspomnianej klasie). Funkcja spójności obiektów, która wyznacza podział, ma postać:

$$S(A,B) = \frac{S_{AB}}{S_{AA}S_{BB}}$$

gdzie S_{AB} jest sumą wartości pewnej miary podobieństwa *p* obiektów należących do *A* i *B*:

$$S_{AB} = \sum_{i\in A}\sum_{j\in B} p_{ij}$$

Algorytmy tego typu są zwykle mało efektywne, tj. ten sam podział obiektów można uzyskać w oparciu o różne skupienia wyjściowe.

Oprócz metod posługujących się odległością do wyróżniania skupień obiektów mamy także tzw. stochastyczne metody klasyfikacyjne. Metody te wymagają losowego rozkładu klasyfikowanych obiektów. Ponadto, aby móc je zastosować trzeba znać rozkłady cech w poszczególnych klasach, co powoduje, że użyteczność tych metod w przypadku rzeczywistych baz danych jest znikoma. Użyteczne rozszerzenie tych metod odchodzi od założenia o stochastycznym charakterze danych i wprowadza pojęcie zbioru rozmytego do określania stopnia przynależności obiektów do różnych klas.

Metody stochastyczne opierają się na założeniu, że zbiór obiektów badanej bazy danych reprezentowany przez wektory cech x_1, ... x_n, jest próbą losową pochodzącą *z k* podpopulacji P_i, ... , P_k a funkcje gęstości dla każdej klasy są znane. Funkcje te mają postać: $f_i(x \mid w_i)$, gdzie *i = 1, ... , k*, w_i to nieznany wektor parametrów. Ponadto zakłada się istnienie wektora $\vec{q} = [q_1,...,q_n]$ zawierającego numery klas, do których należą poszczególne obiekty. Z założenia, że zbiór obiektów jest próbą losową wynika, iż funkcja wiarygodności ma postać:

$$L(x_1,...,x_n \mid w,q) = \prod_{i=1}^{n} f_q(x_i \mid w_q)$$

Maksymalizacja tej funkcji oznacza estymację wektorów $\vec{w}$ oraz $\vec{q}$. Na podstawie wektora $\vec{q}$ możemy określić przynależność poszczególnych obiektów do klas.

Jakość klasyfikacji obiektów zależy od poprawności pomiaru podobieństwa obiektów, prawidłowe określenie podobieństwa obiektów może z kolei zależeć od globalnych własności zbioru, których metody klasyczne mogą nie uwzględniać.

3.4 Wady i ograniczenia klasycznych metod klasyfikacyjnych.

W metodach klasycznych nie bierze się pod uwagę intuicyjnych sposobów klasyfikacji obiektów, jakimi posługują się ludzie. Ważnym ograniczeniem jest konieczność posiadania pełnej i precyzyjnej informacji o grupowanych obiektach. Trzeba dysponować danymi statystycznymi będącymi wartościami tych samych cech dla wszystkich rozważanych obiektów. Brak możliwości pomiaru cechy w przypadku pewnego obiektu powoduje, że nie da się obliczyć odległości między nim a pozostałymi obiektami. Wspomniana pełność informacji oznacza także konieczność posiadania charakterystyk wszystkich obiektów jednocześnie. Nie jest możliwa klasyfikacja obiektów na podstawie napływających danych, ponieważ uzyskany zbiór skupień odzwierciedla jedynie strukturę aktualnego zbioru obiektów. Pojawienie się nowego obiektu wymaga powtórzenia całego procesu klasyfikacji, co pociąga za sobą często wysoki koszt obliczeniowy. Kolejnym problemem jest interpretacja wyników podziału dokonanego za pomocą klasycznych algorytmów numerycznych. Jedyny sposób opisu klas to wymienienie należących do nich obiektów. Trudno jednak przewidzieć, do której z klas byłyby zaklasyfikowane nowe obiekty, gdyby się pojawiły. Nie są bowiem znane żadne reguły określające przynależność do klas. W sytuacji, gdy część cech charakteryzujących klasyfikowane obiekty ma charakter ilościowy, a część – jakościowy. Żadna z omawianych dotąd metod nie radzi sobie

zadawalająco z tym zadaniem. W takiej sytuacji proponuje się najczęściej wykonanie jednej z następujących czynności:
- Dokonać klasyfikacji obiektów oddzielnie dla każdej grupy cech. Jeżeli w ich wyniku powstaną podobne struktury klas, można uznać je za rozwiązanie problemu. W przeciwnym przypadku nie można wyciągać żadnych wniosków.
- Wybrać tylko cechy jednego typu i dokonać klasyfikacji, godząc się z tym, że część informacji zostanie utracona.
- Przekształcić cechy w ten sposób, by wszystkie były mierzone na skali tego samego typu.

Pierwsze dwa wymienione rozwiązania dają w większości przypadków słabe wyniki, rezultaty w trzecim przypadku zależą od charakteru danych oraz zastosowanych metod „ujednolicania" danych. W takiej sytuacji można także próbować zastosować jeden z nowych typów algorytmów nieklasycznych, które potrafią radzić sobie z wspomnianymi problemami. Przykłady algorytmów „radzących" sobie z danymi symbolicznymi zostaną przedstawione w kolejnych rozdziałach.

3.5 Nowe typy miar i algorytmów.

Przykładem miary określającej podobieństwo obiektów, uwzględniającej dodatkowe własności zbioru, jest *miara wzajemnego sąsiedztwa* (*Mutual Neighborhood Value*), zaproponowana przez Gowde i Krishne [3]. Bierze się w niej pod uwagę także własności pozostałych obiektów podlegających grupowaniu, tj. w celu określenia podobieństwa obiekty ze zbioru Ω są porządkowane ze względu na odległość euklidesową wybranego obiektu i pozostałych obiektów (kilku najbliższych sąsiadów). Wzajemny dystans jest sumą rang, czyli zależy od stosunku porównywanych obiektów do pozostałych elementów zbioru. Miara ta powstała w oparciu o obserwacje zachowania ludzi. Uczucie przyjaźni między dwiema osobami ma zwykle charakter wzajemny. Jeśli X czuje, że Y jest jego najbliższym przyjacielem, a Y czuje, że X jest jego najbliższym przyjacielem, to istnieje poczucie wzajemnej bliskości obu osób. Gdy jednak Y nie podziela uczuć X-a, to przyjaźń między nimi jest znacznie słabsza. Siła przyjaźni między dwiema osobami jest funkcją wzajemnych, a nie jednostronnych uczuć.
Stąd analogia: stopień podobieństwa obiektów należy określać biorąc pod uwagę ich wzajemną odległość. Punktem odniesienia jego oceny są pozostałe obiekty podlegające grupowaniu, czyli podobieństwo traktowane jest jako suma rang obiektów według zasady najbliższego sąsiedztwa.

Niech $\Omega = \{o_1, o_2, ..., o_n\}$ będzie zbiorem obiektów podlegających grupowaniu, z których każdy jest opisany przez s cech. Jeśli teraz o_i jest m-tym najbliższym sąsiadem o_j, natomiast o_j jest n-tym najbliższym sąsiadem o_i, to wartością *miary wzajemnego sąsiedztwa* jest liczba:

$$MNV(o_i, o_j) = m + n$$

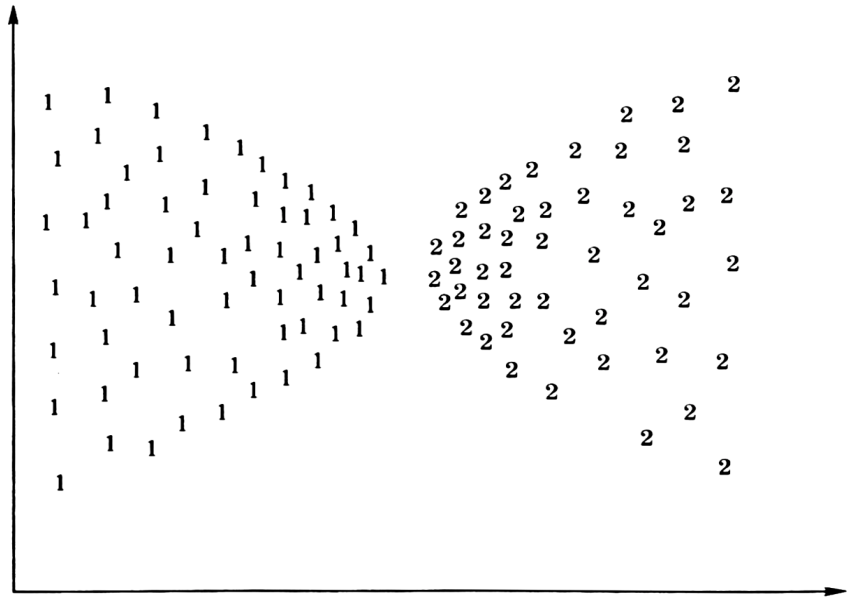
gdzie $m,n \in \{0,1,2,...,N-1\}$.

Miara ta jest semimetryką, przyjmującą następujące wartości:

$$MNV(o_i, o_j) = \begin{cases} \{2,3,...,2N-2\} & \text{gdy } i \neq j \\ 0 & \text{gdy } i = j \end{cases}$$

W oparciu o tak określoną miarę podobieństwa obiektów Gowda i Krishna zbudowali algorytm grupowania: nieparametryczny, hierarchiczny i podziałowy, ponadto nieiteracyjny i mający niewielkie wymagania co do pamięci operacyjnej komputera. „Siła" powyższego algorytmu jest zależna od parametru *k* oznaczającego liczbę sąsiednich obiektów rozważanych w celu określenia wartości MNV. Testy algorytmu [4] wykazały, że w większości przypadków wystarczająca jest wartość *k=5*. W zależności od tego, czy zależy nam odkryciu grup słabiej lub silniej zróżnicowanych, jego wartość może być mniejsza lub większa.

Jedną z ważniejszych własności algorytmów wykorzystujących miarę MNV jest to, że mogą być one wykorzystywane do rozpoznawania skupień niepodzielnych liniowo. Przykład wyników grupowania dla skupień o niejednorodnej gęstości przedstawia rysunek.



Dwa skupienia znalezione przez algorytm wykorzystujący miarę MNV

Ważnym nowym typem miar, które pozwalają unikać wyżej wymienionych problemów (Rozdział 3.4) są miary oparte na tzw. *mierze VDM* (*Value Difference Metric*). Stosowanie tych miar w klasycznych algorytmach numerycznych umożliwia tym algorytmom klasyfikację baz danych z symbolicznymi wartościami atrybutów. W przeciwieństwie do bezkontekstowych (zależnych tylko od bieżąco rozważanych obiektów) miar opartych na metryce Minkowskiego ten nowy typ miar jest częściowo kontekstowy (uwzględnia własności bazy danych). Może określać odległość (podobieństwo) obiektów opisanych za pomocą cech wyrażonych w różnych skalach pomiarowych.

Podstawowym typem miary, którą można zastosować do danych symbolicznych (lub danych o charakterze mieszanym) opartej na VDM jest MVDM (*Modified Value Difference Metric*) Odległość pomiędzy dwoma wektorami o składowych symbolicznych oblicza się jako różnicę prawdopodobieństw:

$$D_{VDM}(x,y) = \sum_{i=1}^{N}\sum_{j=1}^{C}\left|p(C_j \mid x_i) - p(C_j \mid y_i)\right|^q$$

gdzie N określa liczbę obiektów w bazie danych, C z kolei jest liczbą cech charakteryzujących obiekty. Prawdopodobieństwa oblicza się na podstawie wzorów $p(C_j \mid x_i) = \frac{N_j(x_i)}{N(x_i)}$ oraz $p(C_j \mid y_i) = \frac{N_j(y_i)}{N(y_i)}$. W liczniku mamy liczbę wystąpień cechy i w klasie j, w mianowniku zaś liczbę wszystkich wystąpień cechy i. Wartość q przyjmuje się najczęściej 1 lub 2.
Miara ta ma kilka wersji które pozwalają lepiej dobrać charakter miary do charakteru danych:

- HVDM (*Heterogeneous Value Difference Metric*) - niejednorodna VDM

$$D_{HDM}(x,y) = \sqrt{\sum_{i=1}^{N} (dh_i(x_i,y_i))^2} \text{ gdzie } dh_i(x_i,y_i) = \begin{cases} 1 & (1) \\ Nvdm(x_i,y_i) & (2) \\ Ndif(x_i,y_i) & (3) \end{cases}$$

Przypadek (1) stosujemy, gdy x lub y są nieznane, (2) stosujemy jeżeli x i y są dyskretne:

$$Nvdm(x_i,y_i) = \sqrt{\sum_{j=1}^{C} \left(\frac{N_j(x_i)}{N(x_i)} - \frac{N_j(y_i)}{N(y_i)} \right)^2},$$

a w przypadku gdy mają charakter ciągły (3) stosujemy wzór:

$$Ndif(x_i,y_i) = \frac{|x_i - y_i|}{4\sigma_i}$$

- DVDM (*Discrete Value Difference Metric*) - dyskretna odmiana VDM

$$D_{DVDM}(x,y) = \sum_{i=1}^{N} vdm_i(d_i(x_i),d_i(y_i))^2, \; d_i(x_i) = \begin{cases} \frac{x - \min_i}{w_i} + 1 & (1) \\ x & (2) \end{cases}$$

$\min_i$ jest najmniejszą wartością i-tej cechy, w_i w powyższym wzorze jest parametrem. Ta odmiana VDM w pierwszym kroku wykonuje dyskretyzację cechy o charakterze ciągłym (1), cechy dyskretne (2) pozostawia się bez zmian. Następnie stosuje się „klasyczny" wzór na obliczanie odległości VDM.

- HOEM (*Heterogeneous Euclidean-Overlap Metric*) - uproszczona wersja niejednorodnego VDM

$$D_{HOEM}(x,y) = \sqrt{\sum_{i=1}^{N} d_i(x_i,y_i)}, \; d_i(x_i,y_i) = \begin{cases} 1 & (1) \\ \begin{cases} 1 \text{ gdy } x_i = y_i \\ 0 \text{ gdy } x_i \neq y_i \end{cases} & (2) \\ \frac{|x_i - y_i|}{x_i^{\max} - y_i^{\min}} & (3) \end{cases}$$

Pierwszy przypadek (1) stosujemy, gdy x_i lub y_i są nieznane (brakująca wartość), drugi (2), gdy wartości atrybutów są nominalne. Ostatnią możliwość (3) stosujemy w pozostałych przypadkach, $x_i^{\max}$ i $y_i^{\min}$ w tym wzorze są odpowiednio maksymalną i minimalną wartością i-tego atrybutu.

-. IVDM (*Interpolated Value Difference Metric*) - interpolowana VDM

$$D_{IVDM}(x,y) = \sum_{i=1}^{N} ivdm_i(x_i, y_i)^2$$

$$ivdm_i(x_i,y_i) = \begin{cases} vdm_i(x_i,y_i) & (1) \\ \sum_{j=1}^{C} (p(C_j|x_i) - p(C_j|y_i))^2 & (2) \end{cases}$$

Jeżeli cechy mają charakter dyskretny (1) stosujemy standardową procedurę obliczania odległości VDM, w przeciwnym przypadku (2) (dla cech ciągłych) wyznaczamy prawdopodobieństwa potrzebne do wyznaczenia odległości VDM na podstawie wzoru interpolacyjnego:

$$p(C_j \mid x_i) = p(C_j \mid x_i; u) + \frac{x_i - x_{i,u}^{mid}}{x_{i,u+1}^{mid} - x_{i,u}^{mid}} (p(C_j \mid x_i; u+1) - p(C_j \mid x_i; u))$$

gdzie $x_{i,u}^{mid}$ i $x_{i,u+1}^{mid}$ są środkami dwóch kolejnych dyskretyzowanych podziałów, spełniających nierówność $x_{i,u}^{mid} \le x_i \le x_{i,u+1}^{mid}$.

Zastosowanie miar odległości tego typu zamiast zwykłej miary euklidesowej poprawia wyniki klasyfikacji dla danych mieszanych (dane z cechami o charakterze ciągłym i symbolicznym) oraz dla danych o charakterze czysto symbolicznym. Poniższa tabelka zaczerpnięta z pracy [4], przedstawia wyniki badań przeprowadzonych przez Wilsona i Martineza. Prezentowane wyniki uzyskano przy pomocy systemu kNN, testowane zbiory poddano dziesięciokrotnej cross validation, uzyskane uśrednione wyniki w procentach są podane poniżej.

Baza danych	Funkcje określania odległości pomiędzy obiektami					Charakter cech		
	Euclid	HOEM	HVDM	DVDM	IVDM	Co	In	No
Annealing	94.99	94.61	94.61	94.99	**96.11**	6	3	29
Audiology	60.50	72.00	**77.50**	**77.50**	**77.50**	0	0	69
Australian	80.58	81.16	81.45	**83.04**	80.38	6	0	8
Breast Cancer	94.99	95.28	94.99	**95.57**	**95.57**	0	9	0
Bridges	58.64	53.73	59.64	56.73	**60.55**	1	3	7
Credit Screening	78.99	81.01	80.87	80.14	80.14	6	0	9
Echocardiogram	94.82	94.82	94.82	**100.00**	**100.00**	7	0	2
Flag	48.95	48.84	55.82	**58.76**	57.66	3	7	18
Glass	**72.36**	70.52	**72.36**	56.06	70.54	9	0	0
Heart Disease	72.22	75.56	78.52	80.37	81.85	5	2	6
Heart (Cleveland)	73.94	74.96	76.56	79.86	78.90	5	2	6
Heart (Hungarian)	73.45	74.47	76.85	**81.30**	80.98	5	2	6
Heart (More)	72.09	71.90	72.09	72.29	**73.33**	5	2	6
Heart (Swiss)	**93.5**	91.86	89.49	88.59	87.88	5	2	6
Hepatitis	77.50	77.50	76.67	80.58	**82.58**	5	0	13
Horse-Colic	65.77	60.82	60.53	76.75	**76.78**	7	0	16
Hoose-Votes-84	93.12	93.12	**95.17**	**95.17**	**95.17**	0	0	16
Ionosphere	86.32	86.33	86.32	**92.60**	91.17	34	0	0
Iris	94.67	95.33	94.67	92.00	94.67	4	0	0
LED+17 noise	42.90	42.90	**60.70**	**60.70**	**60.70**	0	0	24
LED	**57.20**	**57.20**	56.40	56.40	56.40	0	0	7

Liver Disorders	62.92	**63.47**	62.92	55.04	58.23	6	0	0
Monks-1	**77.08**	69.43	68.09	68.09	68.09	0	0	6
Monks-2	59.04	54.65	**97.50**	**97.50**	**97.50**	0	0	6
Monks-3	87.26	78.49	**100.00**	**100.00**	**100.00**	0	0	6
Promoters	73.73	82.09	**92.36**	**92.36**	**92.36**	0	0	57
Satellite Image	90.21	**90.24**	90.21	87.06	89.79	36	0	0
Sonar	**87.02**	86.60	**87.02**	78.45	84.17	60	0	0
Soybean (Large)	87.26	89.20	90.88	**92.18**	**92.18**	0	6	29
Thyroid (Allb)	94.89	94.89	95.00	94.86	**95.32**	6	0	22
Thyroid (Allhyper)	97.00	97.00	**96.86**	96.93	**97.86**	6	0	22
Thyroid (Allhypo)	**90.39**	**90.39**	90.29	89.36	96.07	6	0	22
Thyroid (Allrep)	96.14	96.14	96.11	96.86	**98.43**	6	0	22
Thyroid (Dis)	98.21	98.21	98.21	**98.29**	98.04	6	0	22
Thyroid (Hypothyroid)	93.42	93.42	93.36	93.01	**98.09**	7	0	18
Thyroid (Sick-Euthyroid)	68.23	68.23	68.23	88.24	**95.07**	7	0	18
Thyroid (Sick)	86.93	86.89	86.61	88.82	96.86	6	0	22
Wine	95.46	95.46	95.46	94.38	**97.78**	13	0	0
Zoo	97.78	94.44	**98.89**	**98.89**	**98.89**	0	0	16

Opis tabeli:

Co – cechy o charakterze ciągłym.
In – cechy dyskretne liniowe.
No – cechy o charakterze symbolicznym.

Pogrubioną czcionką oznaczono najlepszy uzyskany wyniki klasyfikacji.

Na podstawie zamieszczonej tabeli można stwierdzić, że zastosowanie miar opartych na VDM do danych o charakterze symbolicznym lub z przewagą tego typu cech w większości przypadków poprawia wyniki klasyfikacji. W niektórych przypadkach (np. Echocardiogram, Flags, LED+17 noise) poprawa wyniku klasyfikacji jest bardzo wyraźna.

3.6 Klasyczna analiza dyskryminacyjna.

Analiza dyskryminacyjna (*discriminant analysis*) inaczej *klasyfikacja wzorcowa* (rozpoznawanie z nauczycielem), polega na przydzielaniu obiektów do klas, których charakterystyka jest znana.
Dyskryminacja obiektów opiera się na dwóch podstawowych podejściach:

- *Stochastyczne:* zakładamy, że posiadany zbiór obiektów jest próbą losową pobraną z k różnych podpopulacji $P_1, P_2, ..., P_k$. Naszym celem jest taki jego podział, aby każda klasa odpowiadała jednej podpopulacji. Podejście stochastyczne stosuje się np. w fizyce lub chemii gdzie dane można wielokrotnie pobierać powtarzając eksperymenty. Nasz zbiór obiektów podlegających klasyfikacji traktuje się jako próbę losową pobraną z nieskończonego lub skończonego, lecz zawsze bardzo licznego zbioru. Cechy obiektów muszą być reprezentowane przez zmienne losowe.

- *Opisowe:* nie mówimy nic o losowości próby, (bo jest on nie znana), obiekty pochodzą z *k* klas, do których należy je poprawnie przydzielić. Podejście opisowe wykorzystuje się w badaniach ekonomiczno-społecznych, gdy dostępne są jedynie dane statystyczne opisujące zachowanie analizowanego zjawiska. Wartości cech obiektów reprezentują zmienne, których rozkłady nie są badane. Szuka się jedynie ich podstawowych charakterystyk opisowych.

Klasyczna analiza dyskryminacyjna stosuje następujące metody do dzielna zbioru wejściowego danych na poszczególne kategorie:
- Funkcje liniowe R.A. Fisher a,
- Metody probabilistyczne,
- Statystyczne metody podejmowania decyzji.

Pierwszy rodzaj klasyfikacji wykorzystuje do podziału zbioru wejściowego danych tzw. funkcje dyskryminacyjne. To podejście jest obecnie szeroko stosowane praktycznie i stanowi istotną część wielu pakietów komputerowych do analizy statystycznej, np. STATISTICA.

Podział zbioru jest realizowany za pomocą *k* funkcji w postaci:

$$g_i : R^m \rightarrow R$$

gdzie g_i oznacza funkcję dyskryminacyjną dla klasy K_i tzn.

$$x \in K_i \rightarrow g_i(x) = \max_k \{g_k(x)\}$$

Obiekt *x* przydzielamy do tej klasy, dla której funkcja dyskryminacyjna przyjmuje największą wartość. Postać funkcji dyskryminacyjnej może być dowolna, jednak najczęściej stosuje się funkcje liniowe:

$$g_i(x) = a_0 + a_1 x_1 + a_2 x_2 + ... + a_m x_m$$

albo funkcje kwadratowe:

$$g_i(x) = a_0 + a_1 x_1 + a_2 x_2 + ... + a_m x_m + b_1 x_1^2 + ... + b_m x_m^2 + b_{12} x_1 x_2 + ...$$

Wyznaczenie wartości elementów wektora a_i dla powyższej funkcji odbywa się na podstawie próby uczącej. W tym celu stosowane są zwykle dwie metody:
- perceptronowa,
- najmniejszych kwadratów.

Metoda perceptronowa, zaproponowana przez M. Rosenblatta, ma charakter iteracyjny. Działanie rozpoczyna się od wektora a_1 W kolejnych krokach sprawdzana jest przynależność obiektu (jednego w każdej iteracji) do danej klasy. Badanie czy obiekt *x* został poprawnie sklasyfikowany polega na sprawdzeniu, czy zachodzi nierówność:

$$a_i^T x_i > 0$$

Jeśli wektor współczynników a_i funkcji dyskryminacyjnej poprawnie klasyfikuje obiekt *x*, to w następnej iteracji nie zmieniamy go: $a_{i+1} = a_i$.W przeciwnym przypadku wartości jego elementów są modyfikowane zgodnie z formułą: $a_{i+1} = a_i + s_i x_i$ gdzie s_i określa długość kroku. Długość tę można znajdować dwoma sposobami:
- przyjmujemy s_i stałe w każdym kroku iteracji, np. s_i =1,
- s_i wyznaczamy tak, by dany obiekt został poprawnie sklasyfikowany:

$$s_i = \min\left[\frac{a_i^T x_i}{x_i^T x_i}\right]$$

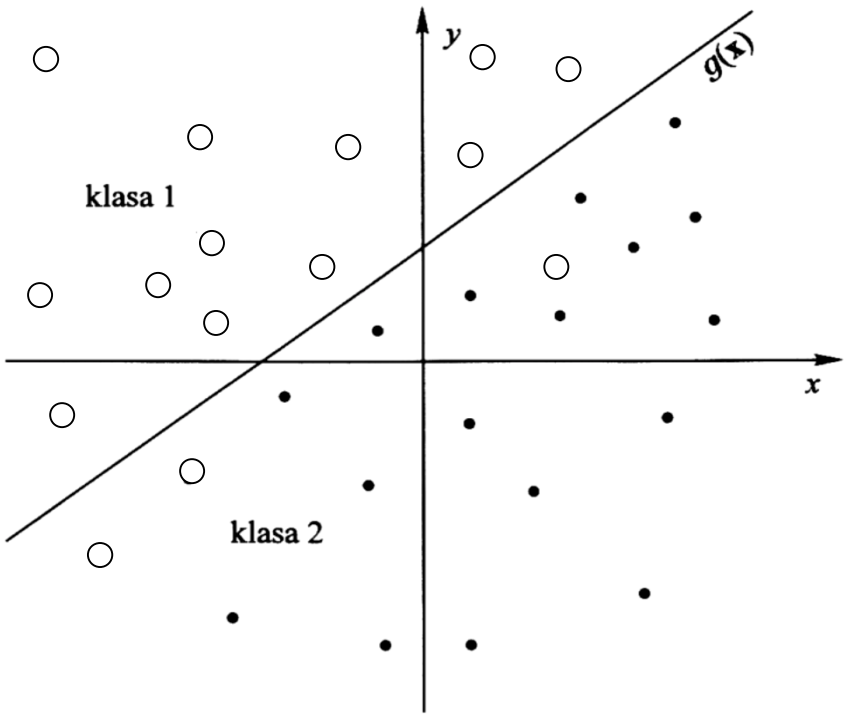
Metoda najmniejszych kwadratów: Nierówność $a_i^T x_i > 0$ zastępujemy układem równań liniowych: $XA = B$ gdzie B jest wektorem o dodatnich elementach $b_i > 0$. Ponieważ ten układ jest najczęściej sprzeczny, będziemy szukali takiego wektora A, który daje najmniejszą różnicę między XA oraz B:

$$\min_A \{(XA - B)^T (XA - B)\}$$

gdzie $A^T = [a_1, ...a_n]$, $B^T = [b_1, ...b_k]$. Wykorzystując minimalizację sumy kwadratów otrzymujemy rozwiązanie, którym jest wektor:

$$A = (X^T X)^{-1} X^T B$$

Wadą metody najmniejszych kwadratów jest brak gwarancji, w odróżnieniu od metody perceptronowej, że wektor A zapewni poprawną klasyfikację obiektów. W przypadku problemów nieseparowalnych liniowo obie metody mogą wyznaczać tylko przybliżone rozwiązania (przykład na rysunku poniżej).



Graficzna prezentacja liniowej funkcji dyskryminacjnej dla dwóch klas. W klasie 2 znajdują się 3 błędnie sklasyfikowane obiekty z klasy 1.

Metody stochastyczne: Podstawowym założeniem tego typu metod jest to, że podział populacji na klasy jest opisany rozkładem zmiennej losowej, określonej jako numer podpopulacji, z której pochodzi dany obiekt. Każda podpopulacja jest opisana przez wielowymiarowy rozkład warunkowy wektora x, którego realizacjami są nasze obiekty. Gęstość rozkładu w podpopulacji P_i można oznaczyć jako $f(x(P_i))$. Przydzielenie obiektu do klasy wykonuje się po zaobserwowaniu x. Prawdopodobieństwo, że obiekt należy do

danej podpopulacji P_j jest prawdopodobieństwem a posteriori, które oblicza się zgodnie ze wzorem Bayesa:

$$p(P_j \mid x) = \frac{p_j f(x \mid P_j)}{\sum_{i=1}^{k} p_i f(x \mid P_i)}$$

gdzie p_i to prawdopodobieństwo a priori, że obiekt należy do podpopulacji P_i. Obiekty są klasyfikowane na podstawie reguły *bayesowskiej*. Przydzielamy obiekt *x* do klasy K_j, dla której spełniony jest warunek:

$$j = \arg(\max_k \{p_k f(x \mid P_k)\})$$

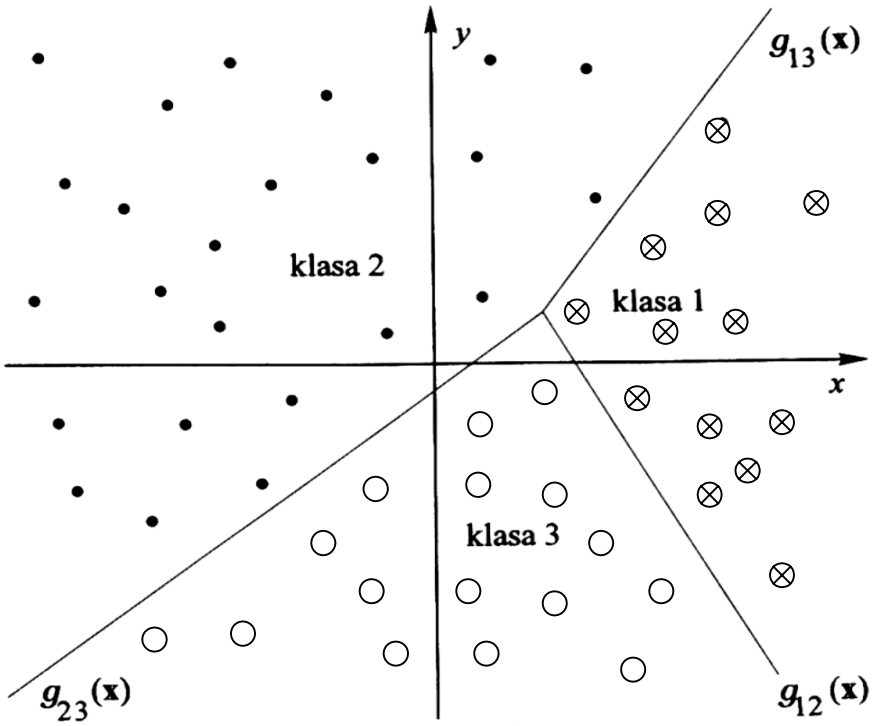
Obiekt jest przydzielany do tej klasy, dla której prawdopodobieństwo a priori jest największe.

Praktyczne zastosowanie metody wymaga znajomość parametrów rozkładu wektora x w podpopulacjach i prawdopodobieństw a priori, określających wielkość podpopulacji. Potrzebne parametry są szacowane na podstawie zbioru uczącego.

Zamiast reguły bayesowskiej czasami stosowana jest *reguła największej wiarygodności,* w której pomija się prawdopodobieństwa a priori. Obiekt *x* jest przydzielany do klasy K_j, dla której zachodzi:

$$f(x \mid P_j) = \max_k \{f(x \mid P_k)\}$$

Funkcja wiarygodności jest w tym przypadku równa funkcji gęstości. W praktyce porównuje się funkcje gęstości dla wszystkich klas i wybiera się tę klasę, dla której funkcja ta osiąga największą wartość.



Przykład funkcji dyskryminacjnej dla trzech klas.

Inna propozycja oparta jest na *minimaksowej regule klasyfikacyjnej*. O ile reguła bayesowska minimalizuje prawdopodobieństwo błędu, o tyle reguła minimaksowa minimalizuje maksymalne prawdopodobieństwo błędu. Na przykład dla dwóch klas ma ona postać:

$$\max\{p(x_1 \mid K_2), p(x_2 \mid K_1)\}$$

gdzie $p(x_i \mid K_i)$ jest prawdopodobieństwem, że obiekt x_i przydzielono błędnie do klasy K_i, podczas gdy w rzeczywistości należy on do klasy K_i.

Jeszcze inną metodą wykorzystującą podejście stochastyczne jest *metoda aproksymacji stochastycznej* H. Robbinsa i S. Monro. Metoda polega na wyznaczaniu współczynników funkcji dyskryminacyjnej dla *i*-tej klasy: $g_i(x) = a_i^T x$, za pomocą kryterium stochastycznego:

$$J_i(a_i, x) = E[f(a_i, x)]$$

gdzie E jest wartością oczekiwaną funkcji oceny jakości aproksymacji f(a_i, x). Wartości wektora a_i znajduje się iteracyjnie, na podstawie wzorów:

$$a_i^{(n+1)} = a_i^{(n)} - \delta_n f^{(n)}(a_i, x)$$

gdzie

$$f^{(n)}(a_i, x) = \frac{\partial f^{(n)}(a_i, x)}{\partial a_i^{(n)}}$$

natomiast δ_n jest dodatnią liczbą, spełniającą warunki:

$$\lim_{n\to\infty} \delta_n = 0$$

$$\sum_{n=1}^{\infty} \delta_n = \infty$$

$$\sum_{n=1}^{\infty} \delta_n^2 < \infty$$

Metody wyznaczania współczynników funkcji dyskryminacyjnych różnią się postacią kryterium aproksymacji. Najczęściej stosowane są dwa kryteria:
- średniego błędu absolutnego:

$$f(a_i, x) = | c_i(x) - a_i^T x |$$

- średniego błędu kwadratowego:

$$f(a_i, x) = 1/2 \cdot | c_i(x) - a_i^T x |^2$$

W powyższych wzorach wielkość $c_i(x)$ przyjmuje następujące wartości:

$$c_i(x) = \begin{cases} 1 & \text{gdy } x \in K_i \\ 0 & \text{gdy } x \notin K_i \end{cases}$$

Słabością wyżej wymienionych klasycznych metod dyskryminacyjnych jest wymaganie posiadania przez zmienne opisujące obiekty rozkładu normalnego. Wymóg normalności jest równoważny temu, że cechy obiektów muszą być opisywane przy pomocy skal mocnych, czyli metody te nie nadają się do klasyfikacji obiektów o cechach jakościowych. Drugim założeniem utrudniającym stosowanie algorytmów tego typu, jest wymaganie równości macierzy wariancji i kowariancji poszczególnych klas.

4. Algorytmy klasyfikacyjne taksonomii symbolicznej.

Termin *taksonomia symboliczna* określa grupę metod klasyfikacji zajmujących się głównie *obiektami symbolicznymi,* których cechy mają charakter jakościowy (nie koniecznie wszystkie cechy opisujące obiekt muszą mieć taki charakter). Podobieństwo tych obiektów jest ujmowane przez miary o charakterze heurystycznym, wykorzystujące teorię informacji, i różne metody statystyczne.

Tabela poniżej [5] prezentuje zestawienie najbardziej znanych algorytmów taksonomii symbolicznej Algorytmy te obecnie zostały ulepszone lub zainspirowały do powstania całych grup nowych algorytmów wykorzystujących dany rodzaj metod. Tabela prezentuje także wykorzystywane przez algorytmy techniki grupowania.

Nazwa algorytmu	Rok	Autorzy	Technika Grupowania	Sekwencje obiektów
EPAM	1961	Feigenbaum	hierarchiczna	tak
MK10	1980	Wolff	hierarchiczna	nie
IPP	1982	Lebowitz	skupienia nierozłączne	tak
UNIMEM	1983	Lebowitz	skupienia nierozłączne	tak
CYRUS	1983	Kolodner	skupienia nierozłączne	tak
CLUSTER	1983	Michalski, Stepp	iteracyjno-optymalizacyjna	nie
DISCON	1984	Langley, Sage	hierarchiczna	nie
RUMMAGE	1984	Fisher	hierarchiczna	nie
MERGE	1985	Wasserman	hierarchiczna	tak
GLAUBER	1985	Langley, Zytkow, Simon	skupienia nierozłączne	nie
RESEARCHER	1986	Lebowitz	hierarchiczna	tak
COBWEB	1986	Fisher	hierarchiczna	tak
OCCAM	1987	Fisher, Pazzani	skupienia nierozłączne	tak
AUTOCLASS	1988	Cheesman, Kelly,Self	iteracyjno-optymalizacyjna	nie
Bez nazwy	1988	Diday	hierarchiczna	nie
CLASSIT	1989	Gennari, Langley,Fisher	hierarchiczna	tak
WITT	1989	Hanson, Bauer	iteracyjno-optymalizacyjna	nie
ADECLU	1989	Decaestecker	hierarchiczna	tak
HIERARCH	1990	Nevins	skupienia nierozłączne	tak
LABIRYNTH	1991	Thompson, Langley	hierarchiczna	tak
OXBOW	1991	Iba, Gennari	hierarchiczna	tak
ITERATE	1992	Biswas, Weinberg	hierarchiczna	tak

Najważniejsze cechy algorytmów zawartych w tabelce to:
- metody te powstały w wyniku inspiracji rezultatami badań nad sposobami kategoryzacji dokonywanej przez ludzi prowadzonych w psychologii;
- podobieństwo pary obiektów jest rozważane przy uwzględnieniu własności pozostałych obiektów w zbiorze.

- miary jakości podziału zbioru obiektów na klasy wykorzystują reguły heurystyczne, teorię informacji, metody statystyczne itd.
- możliwa jest klasyfikacja danych niepełnych
- poszczególne cechy obiektów mogą mieć różny charakter
- wszystkie obiekty nie muszą być dostępne przed rozpoczęciem procesu grupowania, mogą pojawiać się w jego trakcie.
- utworzone kategorie można interpretować za pomocą pojęć;
- mając symboliczny opis klas łatwo można przewidzieć, do której klasy będzie należał nowy obiekt.

Obecnie są prowadzone badania nad nowymi typami algorytmów lub modyfikacjami już istniejących algorytmów, tak, aby były one bardziej wydajne i stabilne oraz aby miały jak najwięcej korzystnych własności wymienionych powyżej. Podstawowe typy algorytmów symbolicznych (tj. te, które dały początek poszczególnym typom algorytmów) zostaną omówione w kolejnych rozdziałach.

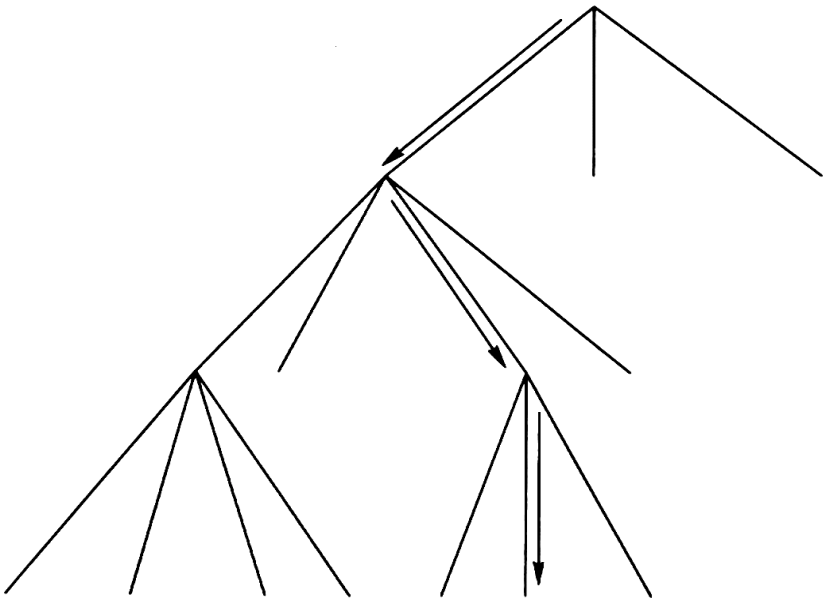
4.0 Przykłady symbolicznych algorytmów klasyfikacyjnych.

4.1 Metody podziałowe. Algorytm CLUSTER.

Pierwszym algorytmem iteracyjno-optymalizacyjnym, stosującym metody podziałowe, który zainspirował powstanie kolejnych algorytmów tego typu, był CLUSTER Michalskiego i Steppa [6][7]. Wykorzystywał on technikę iteracyjno-optymalizacyjną do tworzenia struktury klas opisywanych za pomocą pojęć koniunkcyjnych. We wcześniejszych pracach, prowadzonych w ramach psychologii, również zaproponowano kilka metod klasyfikacji bezwzorcowej tworzącej hierarchiczną strukturę klas (np. EPAM), jednak metody te nie były szerzej znane i wykorzystywane, ze względu na brak ich uniwersalności.

Algorytm CLUSTER powstał w 1980 roku i był rozwijany w ciągu następnych lat. Jego pierwsza wersja nazywała się CLUSTER/PAF 3 i była przykładem bezpośredniego zastosowania techniki iteracyjno-optymalizacyjnej.

W metodach iteracyjno-optymalizacyjnych stosuje się *strategię wspinaczki* (*hill-climbing),* tj. z każdym następnym krokiem staramy się uzyskiwać lepszą jakość podziału aż do osiągnięcia optimum. Nazwa strategii nawiązuje do zachowania się człowieka wchodzącego na szczyt góry. Wspinający się, chcąc jak najszybciej dotrzeć na szczyt, wybiera aktualnie najlepszą drogę, chociaż biorąc pod uwagę całość wyprawy, może okazać się, że decyzja ta była błędna. Z kolei droga w danej chwili gorsza może być w perspektywie całej wyprawy lepsza.



Sposób działania strategii wspinaczki.

Strategia wspinaczki gwarantuje szybkie i efektywne przeszukiwanie, nie gwarantuje jednak osiągnięcia dobrego rozwiązania (minimum globalnego). Jej podstawową wadą jest brak możliwości powrotu do tych kierunków przeszukiwania, które na danym etapie

okazały się gorsze. Jeśli jednak zastosowana funkcja kryterium ma dobre własności, strategia ta może zapewnić osiągnięcie sukcesu. Wady omawianej strategii równoważy jej prostota obliczeniowa sprawiająca, że jest ona popularna i szeroko stosowana.

Obiekty, które będą zalążkami skupień w ramach algorytmu CLUSTER wybiera się losowo tak, aby były one maksymalnie oddalone w sensie odległości syntaktycznej. Odległość tę oblicza się jako sumę poszczególnych odległości syntaktycznych atrybutów. Dla cech nominalnych odległość tę oblicza się na podstawie wzoru:

$$d(x_i, y_i) = \begin{cases} 1 \; gdy \; x_i = y_i \\ 0 \; gdy \; x_i \neq y_i \end{cases}$$

x_i oraz y_i we wzorze to kolejne atrybuty, dla cech ciągłych stosuje się zależność $d(x_i, y_i) = | x_i - y_i | / \bar{x}$, $\bar{x}$ oznacza wartość średnią cechy, z której pochodzą atrybuty *x* i *y*.

Jakość podziału obiektów mierzy się za pomocą LEF *tj.: leksykograficznego funkcjonału oceny* (*Lexicographical Evaluation Functional),* który jest sekwencją par:
< kryterium, próg tolerancji >, gdzie *próg tolerancji* przyjmuje wartości z przedziału [0, 100%], *a kryterium* to kolejne kryteria elementarne:
- uniwersalność opisu skupienia - określa się na podstawie liczby obiektów, które są opisywane przez reprezentujący je kompleks[1], a które nie znajdują się w klasyfikowanym zbiorze.
- prostota opisów klas - liczba składowych w selektorach[2] lub liczba selektorów występujących w kompleksach. Im jest ich mniej, tym większa wartość tego kryterium.
- podobieństwo obiektów - łączna liczba wspólnych własności obiektów należących do tego samego skupienia. Zwykle liczbę tę liczy się w ten sposób, że zlicza się łączną liczbę wspólnych selektorów występujących w kompleksach.
- rozłączność klas - suma stopni rozłączności między każdą parą kompleksów. Stopień rozłączności dla pary klas to liczba występujących w ich opisach selektorów, które wykorzystują tę samą zmienną, lecz o różnych wartościach.
- jakość dyskryminacji klas - liczba zmiennych, które przyjmują różne wartości w każdym z kompleksów.

Kolejność stosowania kryteriów określa prowadzący klasyfikację, ustalenie pewnej kolejności wśród kryteriów jest równoznaczne z przypisaniem im pewnych „wag".
W kolejnych krokach każdą strukturę skupień ocenia się pod kątem poszczególnych kryteriów. Proces ten jest kontynuowany aż do chwili, gdy pozostaje tylko jeden, „najlepszy" zbiór klas lub, gdy zbiór kryteriów zostanie wyczerpany. Jeśli istnieje więcej niż jedna struktura skupień o tej samej „jakości", wyboru dokonuje się arbitralnie.

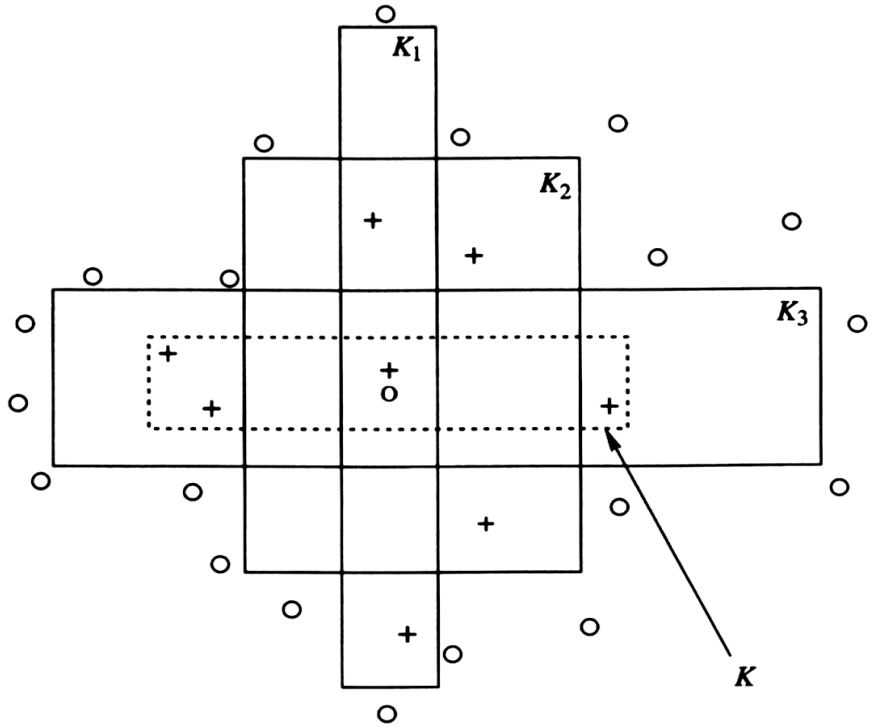
Wartości progu tolerancji funkcjonału LEF leżą między 0 a 100%. Wartość 100% jest sygnałem, że wszystkie struktury klas będą, ze względu na to kryterium, traktowane jako jednakowo dobre, 0% oznacza, że zbiory klas mają jednakową jakość ze względu na

---

[1] *Kompleksem nazywamy wyrażenie:* $\bigwedge_{i \in I} [x \# W_i]$ gdzie *I={1,...,n}* oraz $W_i \subseteq D_i$ *Przykład kompleksu opisującego piłkę do koszykówki [kolor = pomarańczowy][wielkość = średnia][kształt = okrągły].*

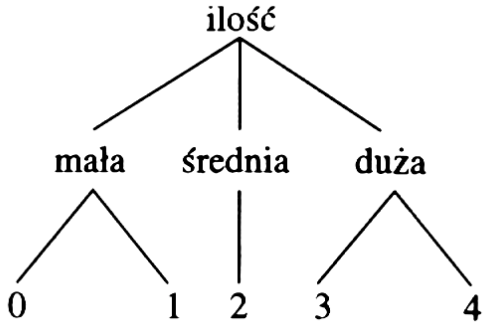
[2] *Wyrażenie* $[x \# W_i]$ *nazywamy selektorem;* $W_i \subseteq D_i$ *symbol # oznacza jeden z operatorów relacyjnych:* $=, \neq, <, \leq, >, \geq$ *. Przykład selektora: [kolor = zielony* $\vee$ *czerwony].*

rozważane kryterium elementarne tylko wtedy, gdy ma ono dla każdego z nich identyczną wartość.



Przykład redukcji pojęć odróżniających, *K* jest kompleksem najbardziej ogólnym.

Zasadnicza dla algorytmu CLUSTER jest operacja tworzenia kolejnych kompleksów najlepiej dopasowanych do posiadanych obiektów i istniejących kompleksów. Dla każdej ze zmiennych staramy się utworzyć kompleks o minimalnej *rzadkości,* tj. opisujący jak najmniej obiektów nie należących do reprezentowanej klasy. Budować takie kompleksy pozwala procedura upraszczania kompleksów, tj. uogólniania reprezentowanych przez nie pojęć. W zależności od rodzaju cechy wykorzystuje się różne techniki. Dla zmiennych przyjmujących wartości ze zbioru, w którym występuje porządek liniowy, stosuje się „regułę domykania przedziału" np. zbiór $[x_i = 1 \vee 2 \vee 3 \vee 7 \vee 9]$ upraszcza się do postaci $[x_i = 1..9]$. Ma to jednak sens tylko wtedy, gdy rzadkość takiego selektora jest mała. Zwykle ustala się pewną wartość graniczną, powyżej której powstaje jeden przedział: 1..9, a poniżej – dwa: $[x_i = 1..3 \vee 7..9]$. W przypadku zmiennych strukturalnych przechodzi się na wyższy poziom w hierarchii wartości. Na przykład dla cechy strukturalnej *ilość ,*



Wartości cechy strukturalnej *ilość*

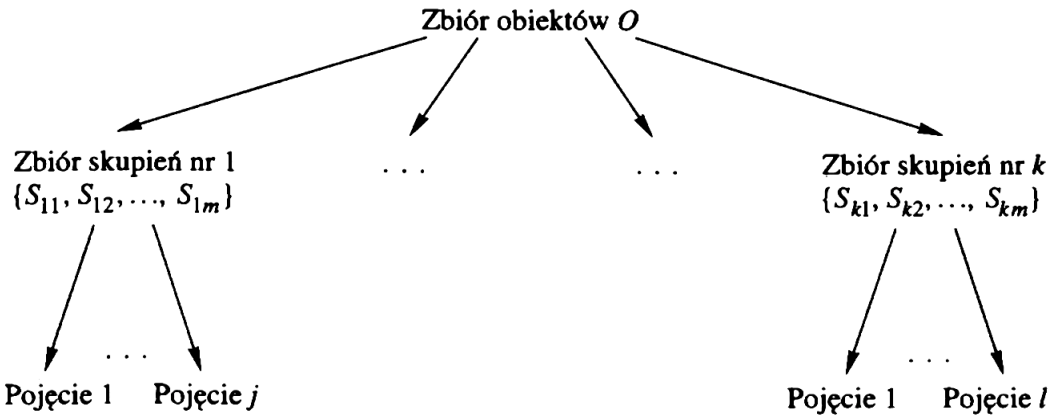
opisanej selektorem [ilość = 0 v 1], można dokonać jego uproszczenia zastępując go selektorem: [ilość = mała].

Kolejnym krokiem istotnym dla działania algorytmu jest usuwanie zbędnych selektorów z kompleksów opisujących klasy. Jeśli stosunek liczby brakujących wartości cechy w selektorze do liczebności jej dziedziny nie przekracza ustalonej wartości granicznej, selektor usuwa się z kompleksu. Na przykład cecha nominalna *wykształcenie* przyjmująca wartości: podstawowe, średnie, wyższe, (selektor ma postać: [wykształcenie = średnie v wyższe]), przy ustalonym progu np. 0,5 zostaje usunięta z kompleksu. Każdy z otrzymanych po zastosowaniu wyżej wymienionych procedur kompleksów ocenia się ze względu na kryterium jakości podziału LEF i wybiera się najlepszy z nich.

Wadą algorytmu jest długi czas obliczeń. Algorytm CLUSTER jest mało efektywny, ponieważ musi dokonywać przeszukiwania dużej przestrzeni, np. dla n zmiennych, z których każda przyjmuje k różnych wartości, istnieje $N = (2^k - 1)^n$ różnych pojęć (kompleksów). W celu przyspieszenia przeszukiwania w kolejnych wersjach algorytmu wprowadzono kilka reguł heurystycznych.

4.2 Metody hierarchiczne.

Metody hierarchiczne były pierwszymi, jakie powstały w ramach metod klasycznych. Po raz pierwszy zostały zastosowane przez biologów tworzących hierarchię gromad, rzędów, rodzin, rodzajów oraz gatunków zwierząt. Znaczna część algorytmów symbolicznych w procesie klasyfikacji buduje drzewo, którego struktura (gałęzie drzewa) ma reprezentować strukturę klasyfikowanego zbioru. W celu utworzenia drzewa o optymalnej strukturze stosowane są dwa procesy: *agregacji i specyfikacji.*



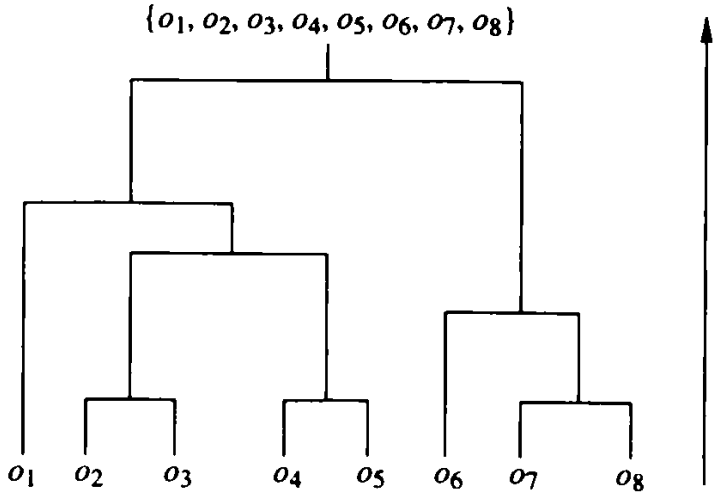
Schemat procesu agregacji

Naturalne podejście do definiowania skupień polega na rozwiązaniu problemu *agregacji,* tj. określeniu możliwych podziałów zbioru obiektów. Dla każdego z nich następnie wykonuje się *specyfikację,* tj. znajduje się możliwe opisy skupień za pomocą pojęć przez uruchomienie procedury „uczenia się na podstawie przykładów". Procedura ta ocenia jakość każdego z tych opisów pojęciowych i wybiera najlepszy z nich.

Z problemem podziału zbioru obiektów i tworzenia drzewa skupień związany jest problem przeszukiwania tworzonego drzewa. W taksonomii symbolicznej algorytmy przeszukiwania drzew zostały po raz pierwszy wprowadzone przez D. Fishera i P. Langley`a.

Z punktu widzenia sterowania przeszukiwaniem drzew zadanie taksonomii symbolicznej można podzielić na dwie grupy:
- utworzone drzewo klasyfikacji buduje się stopniowo,
- drzewo to jest nieustannie przekształcane (np. w algorytmie DISCON).

Graficzną prezentacją hierarchii obiektów tworzonej przez algorytmy hierarchiczne jest dendrogram przedstawiony na rysunku poniżej.



W zależności od sposobu konstruowania hierarchii klas można wyodrębnić:
- metody aglomeracyjne. W ramach tych metod początkowo każdy obiekt traktuje jako oddzielną, jednoelementowa klasę. Klasy te następnie stopniowo są łączone aż do chwili, gdy wszystkie obiekty znajdą się w jednym zbiorze. Rozważany proces odbywa się w oparciu o pewną miarę podobieństwa (odległości), za pomocą której można wybrać te skupienia, które w następnym kroku zostaną połączone.

- metody podziałowe (deglomeracyjne). Proces klasyfikacji w przypadku metod podziałowych przebiega w odwrotnym kierunku. Zbiór wszystkich obiektów jest dzielony w kolejnych krokach na podzbiory aż do uzyskania zbioru skupień zawierających pojedyncze obiekty. Problem określenia kryterium podziału zbioru obiektów na podzbiory jest znacznie trudniejszy niż przy ich łączeniu. Trzeba bowiem rozważać wszystkie możliwe podziały i wybrać najlepszy ze względu na przyjęte kryterium. Metody podziałowe są raczej rzadko stosowane wśród algorytmów symbolicznych.

Algorytmy symboliczne oparte na wymienionych powyżej rozwiązaniach np. EPAM, RUMMAGE czy DISCON mają tą wadę, że stawiają duże wymagania pod względem wielkości pamięci komputera i czasu obliczeń (algorytmy te powstały już ponad 10 lat temu, ale nawet dla najnowszych obecnie komputerów stają się one zupełnie bezużyteczne, gdy liczba klasyfikowanych obiektów przekroczy kilka tysięcy). Ograniczenia te spowodowały, że w ramach taksonomii symbolicznej powstała bardzo liczna grupa algorytmów wykorzystujących nieco inne podejście do tworzenia hierarchii klas, tj. metody *sekwencyjne* (*incremental methods*).

Metody sekwencyjne, w odróżnieniu od metod aglomeracyjnych i podziałowych, mogą być stosowane do budowy hierarchicznej struktury klas w sytuacji, gdy zbiór obiektów nie jest skończony. Podczas klasyfikacji nowego obiektu strukturę i charakterystykę klas (pojęć) modyfikuje się tak, by dostosować ją do aktualnie reprezentowanego zbioru. Proces ten jest wzorowany na przebiegu uczenia się człowieka: wiedza w postaci faktów i doświadczeń jest przyswajana stopniowo, zmieniając strukturę pojęć, system wartości itd. Najważniejszym zagadnieniem w przypadku metod sekwencyjnych jest wybór kryterium podziału zbioru w węźle na podzbiory.

Algorytmy będące implementacjami metody sekwencyjnej, podobnie jak metody iteracyjno-optymalizacyjne, wykorzystują strategię wspinaczki (możliwość utknięcia algorytmu w lokalnym minimum). Do najważniejszych zalet algorytmów tego typu należą:
- niskie koszty gromadzenia i przechowywania danych,
- możliwość klasyfikacji bardzo dużych zbiorów danych,
- krótki czas obliczeń,
- możliwość uzyskania w rezultacie klasyfikacji hierarchicznej struktury klas,
-dynamiczne modyfikowanie charakterystyki klas tak, by dopasować ją do aktualnie posiadanych informacji.

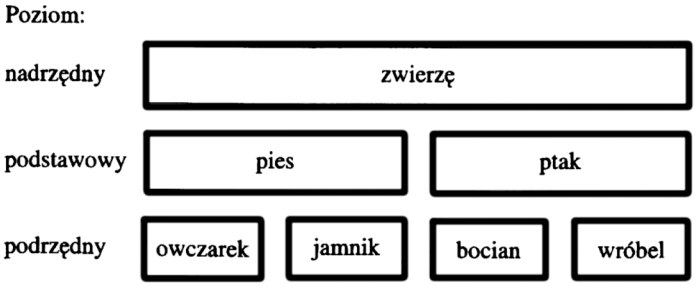
4.2.1 Algorytm COWEB

Najważniejszym algorytmem taksonomii symbolicznej opartym na hierarchicznych metodach sekwencyjnych jest COBWEB. Został on napisany przez D. Fishera w 1986 roku [8]. Wykonuje on podział zbioru obiektów tak, aby maksymalizować przydatność struktury skupień do przewidzenia klasy, do której należy przydzielić nowy obiekt.

COBWEB jest algorytmem hierarchicznym, tworzy bowiem drzewo klas, które następnie przeszukuje w dwu kierunkach za pomocą strategii wspinaczki. Ponieważ stosuje zarówno dzielenie, jak i łączenie klas, nie można jednoznacznie powiedzieć czy jest to algorytm aglomeracyjny, czy podziałowy. Zastosowanie takiej strategii pozwala algorytmowi „omijać" minima lokalne. COBWEB identyfikuje przynależność obiektu do danej klasy lub skupienia na podstawie wartości tzw. *miary użyteczności kategorii.* Miara ta odwołuje się do istotności cechy oraz istotności kategorii. Została skonstruowana przez psychologów M. Glucka i J. Cortera w połowie lat osiemdziesiątych do budowy takiej struktury kategorii, która pozwala uzyskać największą ilość informacji o obiektach na podstawie znajomości klasy, do której należą.

Użyteczność kategorii jest ważonym układem cech, może być postrzegana jako odpowiednik miar jakości podziału stosowanych w algorytmie CLUSTER, tj. *prostoty klas* oraz *dopasowania obiektów.* Uzyskane w wyniku klasyfikacji hierarchiczne drzewo skupień ma tę właściwość, że zbiór klas na pierwszym poziomie jest optymalny ze względu na tę miarę. Użyteczność kategorii pozwala uzyskać zbiór skupień charakteryzujących się jednorodnością poszczególnych klas przy ich jednoczesnym zróżnicowaniu. Podobieństwo obiektów należących do tej samej klasy mierzy się za pomocą prawdopodobieństwa warunkowego: $P(C_i = W_{ij} \mid K_k)$ gdzie K_k to rozważana klasa, *a* W_{ij} jest *j*-tą wartością cechy C_i. Prawdopodobieństwo to zostało nazywane *przewidywalnością* wartości cechy, tj. możliwością określenia wartości cechy na podstawie znajomości klasy, do której należy obiekt. Im większe jest to prawdopodobieństwo, tym większa jest liczba obiektów w klasie K_k mających tę samą wartość cechy C_i. Heterogeniczność skupień określa się za pomocą prawdopodobieństwa: $P(K_k \mid C_i = W_{ij})$ nazywanego *predyktywnością,* tj. możliwością określenia klasy, do której należy obiekt, na podstawie znajomości wartości cechy. Im ta wartość jest większa, tym mniej obiektów przypisanych do różnych klas ma tę samą wartość cechy C_i.

Przykład: jeśli wiadomo, że pewien obiekt należy do klasy "ptaki", to można przyjąć, że ma cechę „skrzydła”. Jednak cecha ta nie ma dużej predyktywności, ponieważ istnieją inne obiekty, np. owady, samoloty, które także mają skrzydła. Czyli na podstawie znajomości tej cechy nie można przewidzieć, czy klasyfikowany obiekt z całą pewnością jest ptakiem. Tak więc cecha ta ma dużą predyktywność, ale małą przewidywalność, ponieważ nie wszystkie obiekty posiadające skrzydła są ptakami.



Przykład poziomów pojęć

Połączenie obu tych prawdopodobieństw daje nam ogólną miarę jakości podziału obiektów na skupienia $K_1, K_2, ..., K_n$:

$$\sum_{k=1}^{n}\sum_{i}\sum_{j} P(C_i = W_{ij})P(K_k \mid C_i = W_{ij})P(C_i = W_{ij} \mid K_k)$$

gdzie indeksy *i, j, k* oznaczają kolejno: numer cechy, numer wartości z dziedziny danej cechy oraz numer klasy. Prawdopodobieństwo $P(C_i = W_{ij})$ jest wagą „nagradzającą” wartości często występujące (typowe) i „karzącą” wartości rzadko występujące. Stosując wzór Bayesa, powyższą miarę można zapisać jako:

$$\sum_{k=1}^{n} P(K_k)\sum_{i}\sum_{j}[P(C_i = W_{ij} \mid K_k)]^2$$

Drugi składnik iloczynu we wzorze jest spodziewaną liczbą wartości cech, które można dokładnie określić dla obiektu należącego do klasy K_k.

Ostatecznie wzór pozwalający obliczyć użyteczność kategorii przyjmie postać:

$$U = \sum_{k=1}^{n} P(K_k)\left[\sum_{i}\sum_{j} P(C_i = W_{ij} \mid K_k)^2 - \sum_{i}\sum_{j} P(C_i = W_{ij})^2\right] / n$$

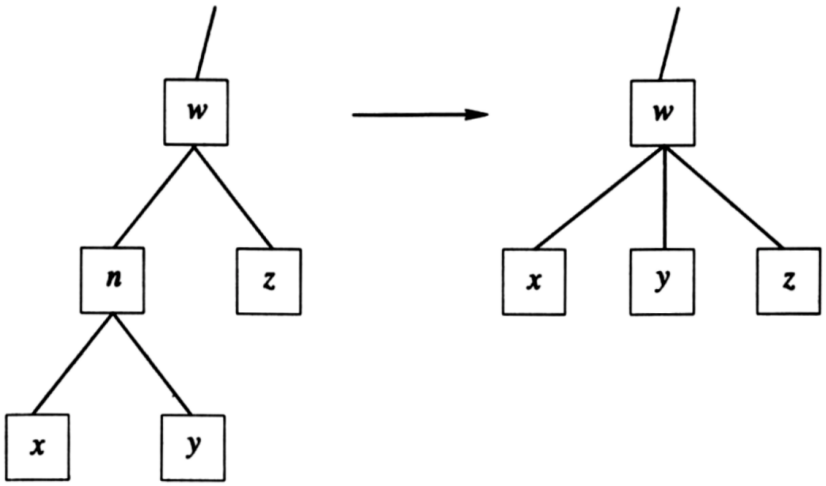
gdzie *n* to liczba klas, na które podzielono wejściowy zbiór obiektów. W przypadku występowania w danych brakujących atrybutów wzór na użyteczność kategorii przyjmuje postać:

$$U = \sum_{k=1}^{n} P(K_k)\left[\sum_{i}\sum_{j} P(C_i = W_{ij} \mid K_k)^2 - \sum_{i}\sum_{j} P(C_i = W_{ij})^2\right] / ln$$

gdzie l to faktyczna liczba wartości cechy C_i .

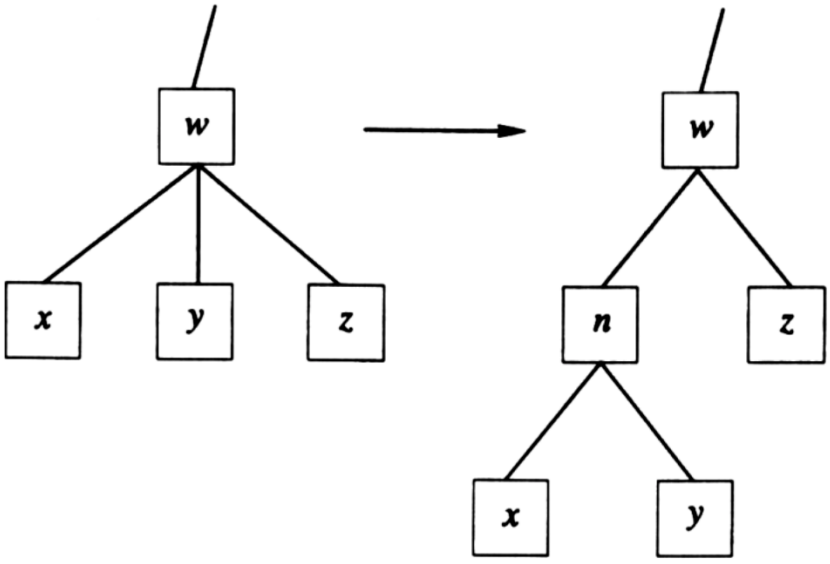
COBWEB szukając odpowiedniej klasy dla nowego obiektu przydziela go kolejno do każdego z istniejących skupień, a następnie ocenia jakość dokonanej klasyfikacji w oparciu o wartość funkcji użyteczności kategorii. Obiekt będzie przydzielony do tej klasy, dla której wartość funkcji jest największa. Jednocześnie jednak ten najlepszy z podziałów jest porównywany z sytuacją, gdy dla nowego obiektu zostanie utworzona odrębna klasa. Również dla takiej struktury skupień oblicza się wartość użyteczności i jeśli jest ona większa od wspomnianej poprzednio, to klasę tę faktycznie się tworzy.

Operacje te są bardzo wrażliwe na kolejność pojawiających się obiektów, stąd w celu podniesienia stabilności algorytmu realizuje się czynności *łączenia* i *dzielenia* skupień (węzłów drzewa). Połączenie dwóch węzłów na pewnym poziomie drzewa może dawać lepszy podział obiektów na klasy ze względu na wartość użyteczności. Aby wyeliminować konieczność oceny jakości podziału przy łączeniu każdych dwóch węzłów, sprawdza się tylko dwa "najlepsze" warianty.



Łączenie klas w algorytmie COBWEB

Również dzielenie skupienia (węzła) na kilka innych może także dawać lepszą strukturę klas. Operacji tej dokonuje się tylko dla najlepszego z istniejących skupień.



Dzielenie klas w algorytmie COBWEB

Obie omówione wyżej operacje równoważą się wzajemnie, powodując przeszukiwanie przestrzeni drzew w dwóch kierunkach.

4.2.2 Wady algorytmu COWEB

Algorytm COBWEB oprócz swych licznych zalet ma także kilka istotnych wad:
-jeśli wartość W_{ij} cechy C_i nie jest związana z przynależnością obiektów do klasy, wówczas:

$$P(C_i = W_{ij} \mid K_k) \approx P(C_i = W_{ij})$$

a co za tym idzie: $P(C_i = W_{ij} \mid K_k)^2 - P(C_i = W_{ij})^2 \approx 0$.

Oznacza to, *że* W_{ij} nie powinno być brane pod uwagę przy obliczaniu *n* użyteczności kategorii.
- użyteczność jest miarą syntaktyczną, nie uwzględniającą wiedzy użytkownika o dziedzinie, w której prowadzona jest klasyfikacja.
- COBWEB może klasyfikować tylko obiekty o cechach jakościowych, a w praktycznych zastosowaniach obiekty mają zarówno cechy jakościowe, jak i ilościowe.
- struktura klas tworzona przez COBWEB jest silnie uzależniona od kolejności pojawiania się obiektów.

4.3 Metody tworzące skupienia nierozłączne.

Większość algorytmów klasycznych jak i symbolicznych daje w wyniku grupowania zbiór klas rozłącznych. W pewnych zastosowaniach istnieje jednak konieczność przydzielenia tego samego obiektu do więcej niż jednej klasy. Dzieje się tak np. w badaniach lingwistycznych. Klasyfikacja słów ze względu na ich znaczenie sprawia, że niektóre z nich mogą znaleźć się w kilku klasach, bo mają kilka znaczeń. Metody tego typu, podobnie jak metody hierarchiczne, budują drzewo klas. Różnica polega na tym, że ten sam obiekt może należeć do więcej niż jednego skupienia. Wynika to z faktu, iż nie jest on przydzielany do klasy „najbliższej", lecz do tych wszystkich klas, od których dzieli go jednakowo mała odległość.

Algorytmy budujące strukturę klas nierozłącznych, są stosunkowo dość nieliczne, mimo że były jednymi z pierwszych, jakie powstały w ramach taksonomii symbolicznej. Opierają się one na rozwiązaniach zastosowanych w pierwszym algorytmie tego typu – EPAM [10]. Najważniejszym algorytmem tej grupy jest UNIMEM M. Lebowitza zbudowany w 1983 roku [11]. Mimo braku formalnych zasad klasyfikacji obiektów np. braku miary jakości podziału, był on podstawą skonstruowania kilku innych algorytmów. Odległości pomiędzy obiektami (podobieństwo obiektów) w tym algorytmie określa się sprawdzając jedynie zgodność wartości cech obiektu z wartościami znajdującymi się w opisie klasy, tj. czy są takie same, czy inne. Podobne rozwiązania, niezależnie od Lebowitza, zastosował J.L. Kolodner w algorytmie CYRUS [12].

4.3.1 Algorytm UNIMEM

Algorytm o nazwie UNIMEM [11] jest modyfikacją wcześniejszego systemu IPP, który był przeznaczony do rozpoznawania i zapamiętywania aktów terroryzmu międzynarodowego na podstawie publikacji prasowych. Metody klasyfikacji stosowane przez ten algorytm były modyfikowane i wykorzystane m.in. w algorytmach RESEARCHER, HIERARCH, OCCAM i EXOR. UNIMEM jest często postrzegany jako rozwinięcie algorytmu EPAM, tworzy bowiem hierarchię klas w oparciu o strategię wspinaczki. Jest on algorytmem sekwencyjnym, dokonującym klasyfikacji obiektów w kolejności ich pojawiania się. Wybór klasy następuje w oparciu o wartości kilku cech. Ma możliwość klasyfikacji obiektów, dla których wartości pewnych cech mogą nie być znane. UNIMEM nie tworzy, w ścisłym tego słowa znaczeniu, hierarchii klas *z góry* w dół, ponieważ stosuje operacje łączenia i usuwania klas. Szczególnie ta ostatnia operacja powoduje poprawę jakości struktury skupień.

UNIMEM tworzy zbiór skupień nierozłącznych, co jest rezultatem niezależnej oceny każdej z klas pod kątem możliwości przydzielenia do niej obiektu. W pewnym sensie jest to jego zaleta, gdyż umożliwia różnorodne interpretacje klasyfikowanych obiektów, np. „pszenica" może jednocześnie należeć do klasy zboża oraz do klasy rośliny jednoliścienne. UNIMEM buduje hierarchię, w której każdy węzeł tworzy klasę w ten sposób, że na wyższym poziomie drzewa występują pojęcia bardziej ogólne, a na niższym - bardziej szczegółowe. Na najniższym poziomie są liście, tj. jednoelementowe klasy, przy czym ten sam obiekt może pojawić się w kilku z nich.

Elastyczność tego algorytmu przejawia się w tym, że pozwala on na dołączanie obiektu do danego skupienia, jeżeli obiekt posiada określoną liczbę (jest to parametr podawany z zewnątrz) wspólnych cech dla obiektów w skupieniu. Mając na myśli cechy wspólne chodzi nam o tzw. cechy normatywne, inaczej nazywane przewidywalnymi (cechy wspólne dla odpowiednio dużej części obiektów w klasie). Każda wartość cechy charakteryzującej skupienie ma swoją wagę (pewna liczba całkowita), która mierzy zgodnie z terminologią Lebowitza pewność jej wystąpienia, czyli możliwość jej określenia na podstawie wiedzy o tym, do której klasy należy obiekt. Jeśli klasyfikowany obiekt ma te same wartości cech, które charakteryzują skupienie, to ich wagi powiększą się o 1, w przeciwnym przypadku wagi zmniejszą się o 1. Jeśli waga pewnej cechy przekroczy ustalony minimalny poziom (jest to zewnętrzny parametr, np. -6), to cechę tę usuwa się z opisu klasy. W przypadku, gdy opis klasy stał się już zbyt prosty, tj. ma zbyt mało cech (liczba ta jest kolejnym parametrem), klasę usuwa się ze struktury, a znajdujące się w niej obiekty klasyfikuje się na nowo. Podobnie, jeśli wartość pewnej cechy jest wspólna dla większej liczby klas (parametr zewnętrzny np. dla 3) to zostaje ona usunięta z ich opisów jako mało przydatna do dyskryminacji tych klas. Każda wartość cechy w algorytmie UNIMEM ma jeszcze jedną wagę. Jest nią ocena jej predyktywności mówiąca o tym, z jaką pewnością można na jej podstawie wskazać klasę dla obiektu.

Klasyfikacja jest sterowana przewidywalnością i predyktywnością cech obiektów, tj. cechy o wysokiej predyktywności wskazują odpowiednią klasę. Jeżeli obiekt spełnia dodatkowo charakterystykę klasy opartą na cechach przewidywalnych to jest do niej przydzielany. Klasyfikacja obiektu polega na przydzielaniu go do kolejnych węzłów drzewa (klas), począwszy od korzenia aż do liścia, zgodnie ze strategią wspinaczki. Jeśli obiekt „pasuje” do opisu klasy znajdującej się w węźle, to jest do niej przydzielany. Zarówno liczba cech wymaganych do oceny dopasowania, jak i sposób oceny są określane przez użytkownika w postaci parametrów. Jeśli obiekt spełnia opis klasy, ale nie pasuje do żadnego z jej węzłów podrzędnych, to UNIMEM porównuje wszystkie obiekty należące do klasy z klasyfikowanym obiektem. Jeśli są podobne, tworzy się dla obiektu węzeł podrzędny. Jeżeli obiekt nie pasuje do żadnego z już utworzonych węzłów, tworzy się nową klasę obiektów, do której początkowo będzie należał jedynie nasz obiekt. Powyższe postępowanie dołączania nowych węzłów jest kontynuowane tak długo, aż każdy z klasyfikowanych obiektów trafi do skupienia jednoelementowego, będącego liściem drzewa.

4.3.2 Wady algorytmu UNIMEM

Algorytm UNIMEM ma kilka istotnych wad, które mogą utrudniać jego praktyczne zastosowanie. Jest on, podobnie jak inne algorytmy sekwencyjne, wrażliwy na kolejność pojawiania się obiektów. Zgodnie z założeniami jego autora, „naturalna” struktura występująca w zbiorze danych powinna się ujawnić po sklasyfikowaniu odpowiednio dużej liczby obiektów (kilkuset). Nie są znane żadne formalne uzasadnienia tego stwierdzenia. Działanie algorytmu UNIMEM w dużym stopniu zależy od wartości dużej liczby parametrów zewnętrznych podawanych przez prowadzącego klasyfikację. Brakuje formalnych podstaw działania algorytmu, precyzyjnego znaczenia wag nadawanych wartościom cech. Podobieństwo wartości cech obiektu i klasy jest funkcją zerojedynkową, przyjmującą 1, gdy są identyczne, oraz 0 gdy się różnią.

4.4 Drzewa klasyfikacyjne.

Algorytmy oparte na drzewach klasyfikacyjnych należą do grupy symbolicznych algorytmów klasyfikacji wzorcowej. Pojęcie symboliczna klasyfikacja wzorcowa oznacza grupę metod klasyfikacji obiektów symbolicznych stosowanych w przypadku, gdy znana jest charakterystyka klas, do których przydzielane są obiekty. Opisy klas znajduje się na podstawie zbioru uczącego, zawierającego poprawnie sklasyfikowane obiekty. Posługując się terminologią cybernetyczną można powiedzieć, że omawiane metody to uczenie z nauczycielem, przy czym rolę „nauczyciela" pełni zbiór uczący. W procesie tym nauczyciel przedstawia uczącemu się systemowi pozytywne oraz negatywne przykłady rozpoznawanych klas.

Chociaż prace nad algorytmami tworzącymi drzewa klasyfikacyjne rozpoczęły się już w latach sześćdziesiątych, ich rozwój nastąpił dopiero w ostatnich dziesięciu latach. Przykłady algorytmów tworzących drzewa klasyfikacyjne prezentuje poniższa tabela:

Nazwa algorytmu	Rok	Autorzy	Rodzaj drzewa	Sekwencje obiektów
CLS	1966	Hunt, Marin, Stone	Binarne	Nie
ACLS	1982	Paterson, Niblett	Binarne	Nie
ID3	1983	Quinlan	Dowolne	Nie
CART	1984	Breiman, Fiedman, Olshen, Stone	Binarne	Nie
ASSISTANT	1985	Kononenko	Binarne	Nie
ID4	1986	Schlimmer, Fisher	Dowolne	Tak
PLS	1986	Rendell	Dowolne	Nie
C4	1987	Quinlan	Dowolne	Nie
GID3	1988	Cheng, Fayyad, bani	Dowolne	Nie
ID5	1989	Utgoff	Dowolne	Tak
LMDT	1991	Brodley, Utgoff	binarne, wielowymiarowe	Nie
CHAID	1993	SPSS Inc.	Dowolne	Nie
IND	1993	Buntine, Caruana	Dowolne	Nie
SADT	1993	Heat, Kasif, Salzberg	binarne, wielowymiarowe	Nie
SE-LEARN	1993	Rymon	Dowolne	Nie
OC1	1994	Murthy	binarne, wielowymiarowe	Nie

Niektóre z prezentowanych w tablicy algorytmów łączą w sobie kilka różnych rozwiązań, np. IND oraz SE-LEARN zawierają procedury bezpośrednio zaczerpnięte z CLS czy CART, z ich funkcjami oceny jakości podziału. Część algorytmów wymienionych w tablicy są algorytmami *sekwencyjnymi* np. ID4 oraz IDS. Poza algorytmami przedstawionymi wyżej istnieją także komercyjne systemy wykorzystujące drzewa klasyfikacyjne np. ACLS, Expert-Ease, EX-TRAN itd.

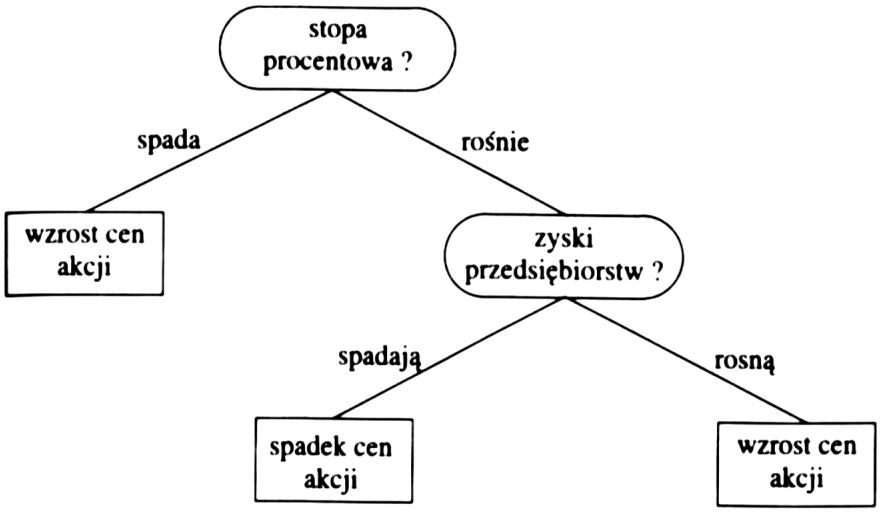
Podczas „uczenia się" algorytmu konstruowane są reguły klasyfikacji (budowane są tzw. drzewa klasyfikacyjne). Konstrukcja polega na stopniowym podziale zbioru obiektów na

podzbiory tak długo, aż zostanie osiągnięta ich jednorodność ze względu na przynależność do klas. Drzewa klasyfikacyjne powstały na początku lat osiemdziesiątych w wyniku poszukiwania metod naśladujących uczenie się i rozwiązywanie problemów przez ludzi. Główne idee pochodzą jednak z lat sześćdziesiątych, kiedy to powstał pomysł wykorzystania konstrukcji typu drzewo (nazywano je drzewami decyzyjnymi) do reprezentowania procesu tworzenia pojęć. Powstał wtedy algorytm CLS (*Concept Learning System*) zbudowany przez E.B. Hunt'a, J. Marin'a i P.J. Stone'a. Zgodnie z podejściem wykorzystanym w tym algorytmie, pojęcie to reguła decyzyjna, która zastosowana do charakterystyki obiektu mówi o tym, czy należy on do określonej klasy. Algorytm CLS stał się inspiracją prowadzenia dalszych badań w tym kierunku, nie tylko na gruncie psychologii.

Duże zainteresowanie stosowaniem tego typu metod zapoczątkowało dopiero pojawienie się algorytmu ID3 Quinlan'a [13], którego udane zastosowania praktyczne zwróciły uwagę na drzewa klasyfikacyjne jako wygodne narzędzie klasyfikacji danych. Drzewo klasyfikacyjne jest pewnym rodzajem grafu bez cykli (pętli), w którym istnieje tylko jedna ścieżka między dwoma różnymi węzłami tzn. drzewa są grafami spójnymi. Drzewa klasyfikacyjne składają się *z korzenia,* z którego wychodzą co najmniej dwie krawędzie do węzłów leżących na niższym poziomie. Z każdym węzłem związane jest pytanie o wartości cech. Jeśli obiekt je posiada, to przenosi się go w dół odpowiednią krawędzią. Węzły, z których nie wychodzą już żadne krawędzie nazywamy *liśćmi*. Krawędzie drzewa reprezentują wartości cech, na podstawie których dokonano podziału.

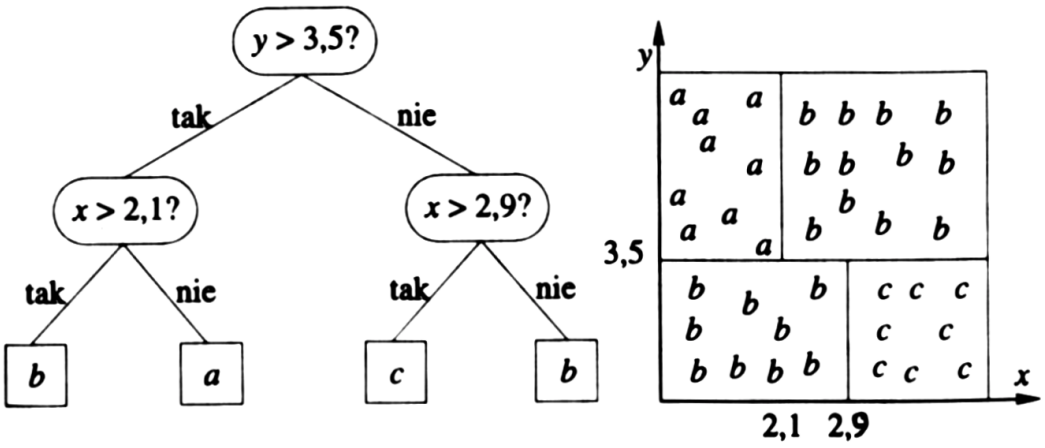
Na podstawie drzewa klasyfikacyjnego można łatwo sformułować reguły przynależności obiektów do klas, np. drzewo pokazane poniżej reprezentuje dwie takie reguły (po jednej dla każdej klasy).

Przykład drzewa klasyfikacyjnego: Ceny akcji.



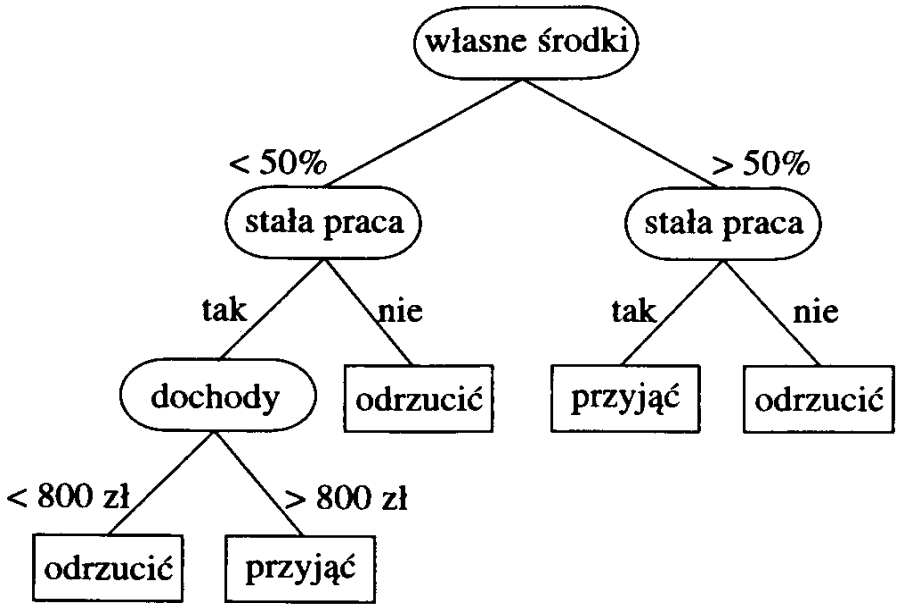
- w przypadku spadku cen akcji „jeżeli stopa procentowa rośnie i zyski przedsiębiorstw spadają, to ceny akcji spadają".
- w przypadku wzrostu cen akcji „jeżeli stopa procentowa spada lub jeżeli stopa procentowa rośnie i jednocześnie rosną zyski przedsiębiorstw, to ceny akcji rosną".

Z geometrycznego punktu widzenia drzewa klasyfikacyjne są podobne do liniowych funkcji dyskryminacyjnych, ponieważ także dzielą wielowymiarową przestrzeń cech hiperpłaszczyznami po to, by wyodrębnić jednorodne skupiska obiektów. Rysunek poniżej przedstawia jednowymiarowe drzewo klasyfikacyjne zbudowane dla obiektów charakteryzowanych przez dwie cechy ilościowe *x* oraz *y*, które należą do jednej z trzech klas: *a, b, c.* Na rysunku zaznaczono także odpowiednie linie oddzielające klasy w przestrzeni cech.



Geometryczna interpretacja drzewa klasyfikacyjnego

Drzewo klasyfikacyjne generuje zbiór reguł przynależności obiektów do każdej z klas. Ścieżka prowadząca od korzenia do liścia reprezentuje sumę testów, które należy wykonać, aby móc sklasyfikować obiekt. Na przykład mając drzewo klasyfikacyjne poniżej oraz wniosek o kredyt, można łatwo sprawdzić czy nasz wniosek znajdzie się w klasie „przyjąć" czy „odrzucić", wykonując testy w kolejnych węzłach drzewa. W tym celu należy poruszać się począwszy od korzenia, wzdłuż krawędzi drzewa, aż do jednego z liści.



Przykład drzewa klasyfikacyjnego dla wniosków kredytowych

4.4.1 Metody tworzące drzewa klasyfikacyjne.

Wszystkie metody budujące drzewa klasyfikacyjne mają bardzo podobną konstrukcję. Można powiedzieć, że oparte są na rozwiązaniach, jakie zawierały pierwsze algorytmy. Różnice dotyczą postaci funkcji oceniającej jakość podziału, sposobu klasyfikacji obiektów o brakujących wartościach cech.

Metody, w oparciu o które tworzy się drzewa klasyfikacyjne, można dzielić na kilka sposobów. Najbardziej podstawowy podział to podział na drzewa *binarne* i *niebinarne*. Binarne drzewo klasyfikacyjne to drzewo, w którym z każdego wewnętrznego węzła wychodzą jedynie dwie krawędzie, czyli zbiór obiektów jest dzielony na dwa rozłączne podzbiory. Pierwsze algorytmy, tj. CLS oraz CART tworzyły wyłącznie drzewa tego typu.
Drzewa binarne najczęściej występują w przypadku klasyfikacji obiektów o cechach ilościowych. W takim przypadku wykonuje się dyskretyzację zbioru wartości cechy przez jego podział na dwie części. Testy w węzłach drzewa mają postać nierówności $x \leq C$, gdzie C jest pewną ustaloną liczbą. Przykłady drzew binarnych znajdują się na rysunkach powyżej.

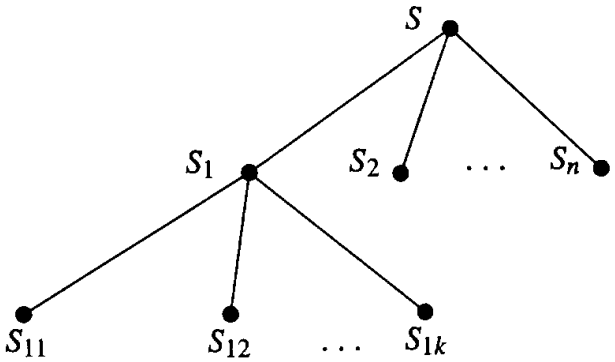
Drzewa mające przynajmniej jeden węzeł, z którego wychodzą więcej niż dwie krawędzie (czyli zbiór jest dzielony na więcej niż dwa rozłączne podzbiory), to drzewa niebinarne. Najczęściej występują one w przypadku klasyfikacji obiektów o cechach jakościowych, które mają odpowiednio liczne zbiory wartości.

Kolejne rozróżnienie związane jest z postacią testu w węźle drzewa, w oparciu o który dokonywany jest podział zbioru obiektów. Dotyczy ono jednak jedynie drzew binarnych oraz obiektów charakteryzowanych przez cechy ilościowe. W zależności od tego czy wspomniany podział oparty jest na wartości jednej cech, czy kilku można wyróżnić:
- drzewa jednowymiarowe
- drzewa wielowymiarowe.

Jednowymiarowe drzewa klasyfikacyjne dzielą zbiory obiektów na podstawie pojedynczych cech, tj. w oparciu o testy w postaci: $x \leq C$, a wielowymiarowe na podstawie ich kombinacji liniowych: $a_0 + a_1 x_1 + ... + a_n x_n > 0$.

4.4.2 Proces tworzenia drzewa klasyfikacyjnego

Tworzenie drzew klasyfikacyjnych odbywa się przez rekurencyjny podział zbioru uczącego na podzbiory aż do uzyskania ich jednorodności ze względu na przynależność obiektów do klas. Uzyskane drzewo powinno być jak najmniejsze i otrzymane reguły klasyfikacji powinny być jak najprostsze. Tworzenie drzewa klasyfikacyjnego odbywa się na podstawie zbioru uczącego, tj. zbioru poprawnie sklasyfikowanych obiektów, zgodnie ze strategią wspinaczki.



Tworzenie drzewa klasyfikacyjnego

Podział jest wykonywany w oparciu o charakterystykę obiektów (wartości ich cech). Cecha będąca podstawą podziału nie może być wybierana losowo, do tego celu stosowane są różne miary. Wszystkie miary opierają się na założeniu, że istnieje związek między wartościami cech obiektów a ich przynależnością do określonej klasy.

Większość metod dokonujących klasyfikacji na podstawie drzewa może klasyfikować obiekty o nieznanych wartościach pewnych cech (przypadek, gdy dysponuje się danymi z brakującymi atrybutami). W takiej sytuacji stosuje się zwykle jedno z trzech rozwiązań:
- brak obserwacji traktuje się jako jedną z dozwolonych wartości, np. „brak”
- zastępuje się brakującą wartość cechy jakąś inną, np. najczęstszą
- przydziela się takiej cesze wartości zaobserwowane u innych obiektów podczas obliczania *funkcji jakości podziału*.

Ponieważ budowane drzewo powinno być jak najmniejsze, większość algorytmów dodatkowo dokonuje *porządkowania drzewa* (*tree pruning),* polegającego na usuwaniu tych jego fragmentów, które mają niewielkie znaczenie dla jakości rezultatów klasyfikacji.
Każdy algorytm tworzący drzewa klasyfikacyjne musi rozwiązać trzy problemy:
- Jak wybrać jedną lub kilka cech, w oparciu o które nastąpi podział zbioru obiektów?
- Kiedy zakończyć dzielenie powstałego podzbioru obiektów?
- W jaki sposób przydzielić obiekty znajdujące się w liściu drzewa do pewnej klasy?

4.4.3 Miary jakości podziału

Efektywność algorytmów tworzących drzewa klasyfikacyjne zależy od wyboru sposobu podziału obiektów w węzłach drzewa. Wybór ten jest dokonywany w oparciu o miarę jakości podziału. Stosowane miary realizują dwie odmienne strategie: w pierwszym przypadku wybierany jest podział, który maksymalizuje wartość stosowanej miary, w drugim - podział, który minimalizuje jej wartość. Ponieważ miary jednorodności można traktować jako odwrotność miar oceniających heterogeniczność podzbiorów, wystarczy dokładnie omówić tylko jedną grupę.

Liczebność pewnej klasy K_i oznaczymy jako l_i, wszystkie klasy są podzbiorami zbioru uczącego zawierającego n obiektów. Dla każdego zbioru obiektów będziemy budować *wektor prawdopodobieństw przynależności do klas* w postaci:

$$\vec{p} = (p_1, p_2, ..., p_k) = \left(\frac{l_1}{n}, \frac{l_2}{n}, ..., \frac{l_k}{n} \right)$$

Zbiór obiektów jest jednorodny, jeśli $\exists i = 1,...,k \; p_i = 1$. Natomiast jego maksymalne zróżnicowanie występuje wtedy, gdy $\forall i = 1,...,k \; p_i = 1/n$.

Funkcją zróżnicowania nazwiemy funkcję o następujących własnościach: $\varphi : [0,1]^k \rightarrow R$ taka, że $\varphi(\vec{p}) \geq 0$ dla każdego wektora $\vec{p}$ .Własności funkcji zróżnicowania:

1) $\varphi(\vec{p}) = \max \Rightarrow \exists i = 1,...,k \; p_i = 1$
2) $\varphi(\vec{p}) = \min \Rightarrow \forall i = 1,...,k \; p_i = 1/n$
3) jest funkcją symetryczną
4) jest różniczkowalna w całej dziedzinie.

Najczęściej wykorzystywaną w algorytmach tworzących drzewa klasyfikacyjne funkcją zróżnicowania jest *funkcja entropii:*

$$E(\vec{p}) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

lub *wskaźnik zróżnicowania Giniego:*

$$G(\vec{p}) = \sum_{\substack{i=1 \\ i \neq j}}^{k} p_i p_j$$

Podział zbioru następuje w oparciu o wartości cech obiektów, należy więc wybrać taką cechę, która daje najbardziej jednorodne podzbiory. Kryterium wyboru jest miarą porównującą stopień zróżnicowania zbioru przed podziałem i po nim. Jeśli cecha X o wartościach $w_1, w_2, ..., w_m$ dzieli zbiór S na podzbiory $S_1, S_2, ..., S_m$, z których każdy zawiera odpowiednio $n_1, n_2, ..., n_m$ obiektów, to zróżnicowanie tych podzbiorów można oszacować stosując średnią ważoną:

$$z(S, X) = \frac{1}{n} \sum_{j=1}^{m} n_j \varphi(\vec{p}_j)$$

gdzie p_j to wektor prawdopodobieństw w zbiorze S_j. Jakość podziału dokonanego na podstawie cechy X mierzy się za pomocą funkcji *przyrostu informacji*:

$$J(S, X) = \varphi(\vec{p}) - z(S, X)$$

W oryginalnej postaci w algorytmie ID3 stosowano do wyboru cechy decydującej o podziale bezpośrednio funkcję *J(S, X).* Faworyzuje ona cechy mające liczniejszy zbiór wartości, więc dokonano jej normalizacji:

$$I(S, X) = \frac{J(S, X)}{E(X)}$$

uzyskując w ten sposób miarę, którą nazywano *ilorazem przyrostu informacji.* Funkcja *E(X)* pozwala określić ilość informacji, jaką niesie cecha *X.*

$$E(X) = -\sum_{i=1}^{m} p(w_i) \log_2(p(w_i))$$

gdzie $p(w_i)$ to prawdopodobieństwo, że cecha X przyjmuje wartość w_i .

Do oceny jakości podziału można zastosować także np. *regułę podziału na 2 części,* która zastosowana została po raz pierwszy w algorytmie CART:

$$T(S,S_A,S_B) = p_A p_B \left(\sum_{i=1}^{k} |p_i^A - p_i^B| \right)^2$$

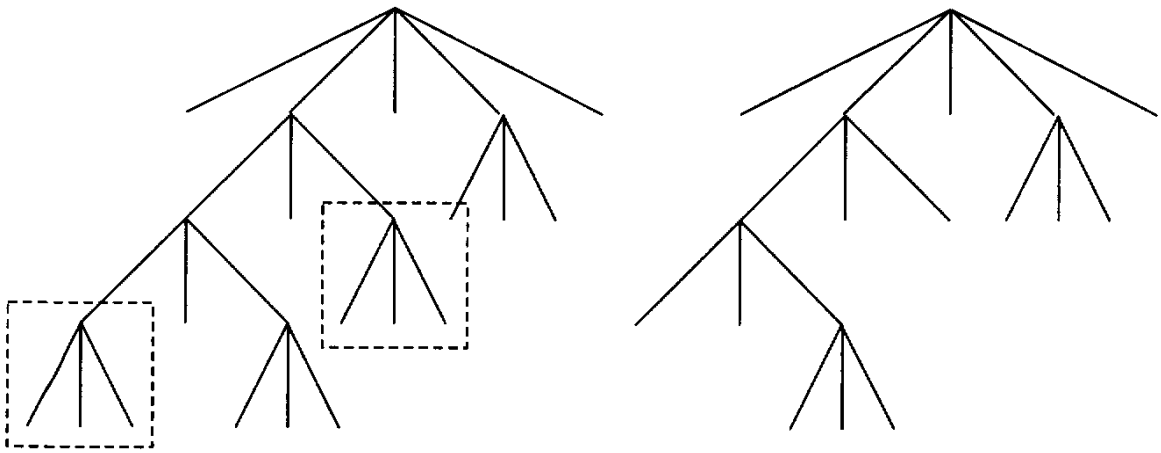
gdzie p_A oraz p_B to prawdopodobieństwa, że obiekt należy do podzbioru S_A *i* S_B $(S = S_A \cup S_B)$; p_i^A, p_i^B - prawdopodobieństwa należenia obiektów znajdujących się w S_A i S_B do klasy K_i.Wiele eksperymentów wskazuje, że jest to miara najbardziej uniwersalna, niezależna od charakteru zmiennych.

4.4.4 Porządkowanie drzewa

Kolejnym problem związanym z konstruowaniem drzew klasyfikacyjnych jest podjęcie decyzji, kiedy należy zaprzestać dalszego dzielenia podzbiorów. Staramy się przy tym uzyskać drzewo o minimalnej liczbie węzłów, nie obniżając „jakości" reguł klasyfikacji obiektów. W tym celu stosuje się jedną z poniższych metod:.

Pierwsza metoda tzw. *porządkowanie wstępne,* polega na ustaleniu granicznej wartości miary jakości podziału *J**, której przekroczenie oznacza zakończenie podziału i utworzenie liścia: $\max_i \{J(S, X_i)\} < J^*$. Metoda ta stosowana jest m.in. w algorytmie ASSISTANT. Brak reguł pozwalających ustalić wartość *J** powoduje, że podział kończy się zwykle zbyt wcześnie lub zbyt późno.

Do określenia końca podziału można wykorzystać tzw. reguły stopu. Według najprostszej takiej reguły należy zakończyć podział zbioru obiektów S, gdy podzbiory obiektów (klasy) są odpowiednio jednorodne: $c = l_{\max} / n$ gdzie n to liczba obiektów w zbiorze S, $l_{\max}$ jest liczebnością najliczniejszej klasy w S.
Inne rozwiązanie polega na uniknięciu tworzenia dalszych podzbiorów w sytuacji, gdy badany podzbiór jest mało liczny. Do oceny tego stosuje się parametr: $d = n_i / n$ gdzie n_i jest liczbą obiektów w danym podzbiorze a n liczbą obiektów w całym zbiorze uczącym. Jeśli przyjmie się np. wartość $d = 0.05$, oznacza, że podzbiór nie będzie dzielony, gdy liczebność obiektów w tym zbiorze spadnie poniżej 5% wszystkich obiektów.



Przykład porządkowanie drzewa klasyfikacyjnego

Odmienne rozwiązanie tego samego problemu zastosowano w algorytmie CART. Algorytm ten nie stosuje kryteriów stopu, tworzy pełne drzewo klasyfikacyjne (nawet gdy to drzewo jest bardzo duże) a następnie wykonuje tzw. *porządkowanie* utworzonego drzewa. Polega to na „obcinaniu" pewnych fragmentów drzewa (czyli zamianie węzłów wewnętrznych na liście) tak, by wraz ze zmniejszaniem się wielkości drzewa nie wzrastał błąd klasyfikacji. Błąd klasyfikacji jest liczony jako stosunek liczby błędnie przydzielonych obiektów b_i do klasy $S_i(b_i)$ do liczby wszystkich należących do tej klasy obiektów n_i:

$$B(S_i) = b_i / n_i$$

Aby zbadać błąd klasyfikacji buduje się pełne drzewo klasyfikacyjne *D*, a następnie zastępuje jego kolejne fragmenty (drzewa podrzędne) liśćmi, otrzymując sekwencję coraz mniejszych drzew: $D_1, D_2, ..., D_s$, gdzie D_{i+1} jest drzewem uzyskanym przez redukcję drzewa D_i. Decyzja o tym, który węzeł w drzewie D_i zostanie zredukowany do liścia, odbywa się na podstawie dodatkowego zbioru testowego zawierającego obiekty sklasyfikowane przy pomocy drzewa *D*. Porównywana jest wielkość błędu klasyfikacji w sytuacji, gdy każdy z wewnętrznych węzłów jest kolejno redukowany do liścia. Jeśli otrzymane drzewo daje błąd mniejszy lub równy poprzedniemu, to staje się drzewem D_{i+1}.

Chociaż metoda ta generuje sekwencję drzew oraz wymaga utworzenia specjalnego zbioru testowego ma zaletę: ostatnie z utworzonych drzew jest najbardziej dokładne, a przy tym jest najmniejszym możliwym drzewem o wysokiej dokładności klasyfikacji.

Inna metoda tego typu polega na minimalizacji pewnej funkcji *F*, będącej kombinacją liniową kosztu klasyfikacji *K* oraz złożoności drzewa *Z*. Koszt klasyfikacji obiektów (patrz wartość $B(S_i)$) liczony jest jako $K(D_i) = \frac{b_i}{n_i}$, $Z(D_i)$ jest funkcją liczby liści w drzewie $Z(D_i) = l$. Funkcja *F* ma postać:

$$F(D) = \alpha K(D) + \beta Z(D)$$

Porządkowanie drzewa odbywa się w dwóch kolejnych etapach: najpierw tworzymy sekwencję drzew $D_1, D_2, ..., D_s$ z których każde zostaje ocenione, za pomocą funkcji *F(D)*. Procedura ta jest przerywana po znalezieniu najlepszego drzewa, tj. dającego najmniejszą wartość funkcji *F(D)*. Ponieważ w praktyce podczas obliczania wartości funkcji *F* stosuje się wartość α = 1, trzeba określić jedynie wartość drugiego parametru. Jeśli pewien węzeł *T* drzewa *D* wraz ze wszystkimi węzłami podrzędnymi (których jest *w*) zostanie zredukowany do liścia, to takie drzewo *D′* będzie błędnie klasyfikowało dodatkowo *m* obiektów, mając jednak o *w-1* węzłów mniej. *D′* dawałoby taką samą wartość funkcji kryterium jak *D*, gdyby:

$$\beta = \frac{m}{n(w-1)}$$

Tworząc drzewo D_{i+1} na podstawie D_i należy badać wszystkie drzewa podrzędne wchodzące w skład D_i, szukając drzew, które dają najmniejszą wartość β, aby zastąpić je liśćmi. W drugim etapie wybieramy takie drzewo spośród $D_1, D_2, ..., D_s$, które daje najmniejszy błąd klasyfikacji. Do tego celu wykorzystujemy dodatkowy zbiór testowy

zawierający k obiektów, za pomocą którego testujemy wszystkie tworzone drzewa. Jeśli b oznacza najmniejszą liczbę błędnie sklasyfikowanych obiektów przez każde drzewo, to należy wybrać najmniejsze drzewo D_i, dla którego liczba błędów jest mniejsza niż $b + S(b)$, gdzie S oznacza błąd standardowy:

$$S(b) = \sqrt{\frac{b(k-b)}{k}}$$

Główną wadą tej metody jest wymaganie posiadania odrębnego zbioru testowego (zbioru walidacyjnego).

Metoda pesymistyczna do oceny błędu klasyfikacji wykorzystuje modyfikację rozkładu dwumianowego. Do wzoru na błąd klasyfikacji wprowadzamy poprawkę: $B(S_i) = (b_i + 0{,}5)/n_i$. Jeżeli T jest fragmentem drzewa D (drzewo podrzędne), którego l liści reprezentuje zbiory zawierające łącznie $\sum n_i$ obiektów, z których $\sum b_i$ zostało błędnie sklasyfikowanych, to pesymistyczne założenie głosi, że drzewo T może błędnie sklasyfikować $\sum b_i + l/2$ obiektów. Jeżeli T zastąpimy liściem, to będzie dokonywać błędnej klasyfikacji m obiektów ze zbioru uczącego. Metoda ta redukuje drzewo podrzędne do liścia, gdy spełniony jest warunek:

$$m + 1/2 < S\left(\sum b_i + l/2\right)$$

W kolejnych krokach sprawdzone zostają wszystkie drzewa podrzędne. Sprawdzaniu nie podlegają usuwane elementy. Zaletą metody jest jej szybkość wynikająca z faktu, że każdy fragment drzewa T jest sprawdzany co najwyżej raz. Nie wymaga ona także oddzielnego zbioru testowego, jak w przypadku poprzednich metod.

4.4.5 Przykłady algorytmów: ID3 oraz C4.

Najbardziej znana metoda budowy drzew klasyfikacyjnych to ID3 (*Induction of Decision Trees)*[13]. Powstała na przełomie lat siedemdziesiątych i osiemdziesiątych, inspirowana systemem CLS oraz pracami nad indukcyjnymi metodami uczenia się. Do wyboru cechy decydującej o podziale zbioru obiektów w tym algorytmie wykorzystano funkcję entropii. W pierwszej wersji algorytm akceptował jedynie dwie klasy obiektów, tj. zbiory zawierające przykłady pozytywne i przykłady negatywne (kontrprzykłady). Kolejne modyfikacje usunęły jednak to ograniczenie, a jego następca - algorytm C4 dodatkowo umożliwia klasyfikację obiektów, których cechy reprezentowane są za pomocą zmiennych ilościowych. Algorytm ten jest w dalszym ciągu rozwijany, obecnie najnowsza wersja tego algorytmu ma już numer 5. Najczęściej obecnie stosowaną wersją tego algorytmu jest wersja 4.5a [15].

Procedura wyboru cechy będącej podstawą podziału wymaga wielokrotnego przeszukiwania zbioru uczącego, który musiałby mieścić się w pamięci operacyjnej komputera. Utrudniałoby to stosowanie algorytmu ID3 do dużych zbiorów obiektów. Aby to przezwyciężyć, Quinlan zastosował tzw. *okno*. Polega to na losowym wyborze pewnego podzbioru obiektów i zastosowaniu do niego algorytmu ID3. Utworzone w ten sposób

drzewo jest następnie modyfikowane przez dodanie do okna obiektów, których skonstruowane drzewo nie klasyfikuje.

Algorytm ID3 doskonale radzi sobie z obiektami symbolicznymi, kolejne wersje algorytmu dobrze radzą sobie także z obiektami o cechach ilościowych i jakościowych. W przypadku gdy przedmiotem klasyfikacji są obiekty o cechach reprezentowanych także przez zmienne ilościowe, algorytm C4 (a także jego kolejne rozszerzenia), stosuje testy w postaci $x \leq C$ lub $x \geq C$, których wynikiem są dwa podzbiory. Oznacza to dyskretyzację cechy ilościowej, co odbywa się jednocześnie z obliczaniem wartości funkcji jakości podziału. Znajduje się taką wartość C, by funkcja $J(S,X)$ dawała największą wartość. Najpierw wartości analizowanej cechy ilościowej dla wszystkich obiektów w zbiorze uczącym porządkuje się rosnąco, a następnie sprawdza wartość funkcji jakości podziału $J(S,X)$ dla wszystkich możliwych podziałów tego przedziału na dwie części. Wybiera się ten, który daje największą wartość $J(S,X)$.

4.4.6 Kryterium SSV.

Sposobem na proste poradzenie sobie z problemem porządkowania drzewa jest zastosowanie kryterium nowego typu tzw. SSV (Separability of Split Value). Kryterium to pozwala budować drzewa a także przeprowadzać ich porządkowanie. Algorytm rozwijany w Katedrze Metod Komputerowych oparty tym kryterium nosi nazwę *SSV Tree* [14]. Do zalet jego należy zaliczyć:
- świetnie radzi sobie z obiektami opisanymi przy pomocy zmiennych różnych typów, brakującymi wartościami atrybutów.
- daje dobre rezultaty klasyfikacji w porównaniu z innymi algorytmami.
- drzewa produkowane przez algorytm są proste do analizy przez człowieka, algorytm ten nadaje się bardzo dobrze do ekstrakcji reguł logicznych

Kryterium SSV wprowadza bardzo prostą zasadę, która pozwala budować jak najmniejsze drzewa o najlepszym poziomie klasyfikacji. Zasada ta mówi: kryterium (reguła), które będzie wybrane do rozróżniania obiektów (dla cech ciągłych jest to pewna liczba rzeczywista a dla dyskretnych pewien podzbiór możliwych wartości cechy) w danej bazie danych, musi być dobrane w ten sposób, aby pozwalało ono odróżniać jak najwięcej obiektów z różnych klas. Do oceny reguł na podstawie powyższej zasady trzeba wprowadzić pojęcia tzw. lewe i prawe zbiory podziałów, które są zdefiniowane następująco (reguła s dla cechy f w bazie danych D):

$$LS(s,f,D) = \begin{cases} \{x \in D \,:\, f(\mathrm{x}) < \mathrm{s}\} & (1) \\ \{x \in D \,:\, f(x) \notin s\} & (2) \end{cases}$$

Pierwszą możliwość (1) stosujemy jeżeli f jest cechą ciągłą, drugą (2) w pozostałych przypadkach.

$$RS(s,f,D) = D - LS(s,f,D)$$

Wartość kryterium SSV dla cechy s oblicza się na podstawie wzoru:

$$SSV(s) = 2\sum_{c \in C} | LS(s,f,D) \cap D_c | \cdot | RS(s,f,D) \cap (D - D_c | - \sum_{c \in C} \min(| LS(s,f,D) \cap D_c |, | RS(s,f,D) \cap D_c |)$$

w tym wzorze C oznacza liczbę klas, D_c to część bazy danych należąca do klasy c. Pierwszy człon osiąga maksimum dla największej liczby prawidłowo rozdzielonych par, drugi wybiera taki podział, dla którego najmniejsza liczba wektorów z tej samej klasy znajduje się w różnych zbiorach *LS* i *RS.* Im wyższą wartość kryterium uzyska dana reguła tym lepiej nadaje się ona do dyskryminacji badanego zbioru danych.

Podane wyżej kryterium może służyć także do dyskretyzacji wartości cech ciągłych. Ponieważ SSV bada testy postaci $x \leq C$ (x –atrybut, c –pewna liczba rzeczywista), wystarczy wybrać kilka najlepszych takich testów aby przypisać odpowiednie wartości (dla dalszego procesu klasyfikacji) dyskretyzowanym atrybutom ciągłym.
Innym zastosowaniem może być określanie „ważności" poszczególnych cech służących do opisu obiektów. Dla baz wielowymiarowych (bardzo duża liczba cech opisujących obiekty) SSV pozwala zmniejszyć liczbę cech, pozostawiając w opisie obiektów cechy najbardziej istotne.

5 Zamiana symboli wartościami numerycznymi.

Zajmiemy się teraz problemem przejścia od zmiennych symbolicznych do ich odpowiedników numerycznych. Transformacja danych zastępująca symbole liczbami powinna zachowywać „odległości” (tj. odległości w sensie metryki Minkowskiego lub np. odległości stosowane przez algorytm UNIMEM) pomiędzy obiektami, związki występujące pomiędzy cechami.

Wykonamy kilka testów przy pomocy algorytmów dostępnych w Katedrze Metod Komputerowych. Dostępne algorytmy to: kNN (k Nearest Neighbors), FSM (Features Space Mapping) i C4.5.

Algorytm kNN stosuje do klasyfikacji obiektów bardzo prostą regułę: jeżeli najbliższy sąsiad klasyfikowanego obiektu należy do pewnej klasy to dany obiekt jest przydzielany do tej samej klasy. Stwierdzenie „najbliższy sąsiad” oznacza obliczenie odległości pomiędzy tymi obiektami za pomocą miar opartych na metryce Minkowskiego lub przy pomocy miar opartych o VDM (Rozdział 3.5). Algorytm FSM do klasyfikacji obiektów stosuje odmienną koncepcję. W procesie uczenia algorytm ten buduje sieć neuronów (węzłów). Klasyfikowany obiekt jest podawany na wejście takiej sieci i zostaje przydzielony do klasy, w której znajduje się najsilniej wzbudzający się węzeł tej sieci.

W tych systemach przedmiotem grupowania są obiekty, których cechy są reprezentowane przy pomocy atrybutów o wartościach rzeczywistych. Każdy taki obiekt traktuje się jako punkt w wielowymiarowej przestrzeni cech, w której należy w optymalny sposób wydzielić podzbiory klas lub w przypadku analizy wzorcowej poprawnie przydzielać nowe obiekty do istniejących podzbiorów.

Wszystkie trzy systemy nie mają możliwości ładowania baz danych zawierających symbole. Większość nowo tworzonych baz danych zawiera obiekty o cechach symbolicznych, ponadto opisy obiektów często są niepełne i nieprecyzyjne. Jeżeli chcemy pracować z danymi tego typu posługując się wyżej wymienionymi algorytmami, musimy zastąpić symboliczny opis cech liczbami. Najprostszy sposób zamiany to przypisanie kolejnym symbolom kolejnych liczb całkowitych np. mamy dwa symbole „kobieta” i „mężczyzna” numerujemy je kolejno np. 1 i 2. Rozpatrzmy prosty przykład danych symbolicznych:

	Cecha 1	Cecha 2
Klasa 1	Aa	Ab
Klasa 2	Ba	Bb
Klasa 3	Ca	Cb

Wykonajmy wspomnianą prostą zamianę symboli, zastępując je dla każdej z cech kolejnymi liczbami całkowitymi:

	Cecha 1	Cecha 2
Klasa 1	1	1
Klasa 2	2	2
Klasa 3	3	3

Jak widać na podstawie tabelki obiekty takie dla systemu kNN są doskonale odróżnialne. Kolejne liczby są przypisywane atrybutom w sposób losowy i równie często możemy mieć do czynienia z sytuacją odwrotną do powyższej. Po zastąpieniu symboli liczbami w innej kolejności, uzyskamy tabelkę:

	Cecha 1	Cecha 2
Klasa 1	1	3
Klasa 2	2	2
Klasa 3	3	1

Dla tych samych danych wejściowych (pierwsza tabelka) obiekty z klasy 1 i klasy 3 w tym przypadku przestają być odróżnialne dla algorytmu kNN.

Bez wykonywania dokładnej analizy danych nie można stwierdzić czy zastąpienie np. symbolu „a" jedynką jest lepsze od wstawiania zamiast „a" liczby pięć. Stosowanie tego typu zamiany symboli w większości przypadków prowadzi do fałszowania danych. Wprowadzamy do danych zależności, których wcześniej w danych nie było lub niszczymy pewne zależności pomiędzy symbolami. W obu tych przypadkach uzyskujmy błędny wynik klasyfikacji, w pierwszym przypadku zbyt „dobry", w drugim przypadku zbyt „słaby".

Wyjściem z tej sytuacji może być zamiana naszych symboli na prawdopodobieństwa ich występowania w danych. Sposób obliczania naszych prawdopodobieństw będziemy wzorować na mierze VDM, która może działać bezpośrednio na symbolach dając przy tym dobre rezultaty (patrz tabela w rozdziale „Nowe typy miar")

5.1 Bazy danych użyte do testów.

Bazy danych użyte do testów są dostępne pod adresem internetowym: www.ics.uci.edu/pub/machine-learning-databases.

Nazwa bazy	L. wektorów trening/test	Liczba klas	Liczba cech		Brakujące atrybuty	Zbiór testowy
			Dyskretne	Ciągłe		
Annealing	798/100	6	32	6	+	+
Australian Credit	690	2	8	6	-	-
Flags	194	8	25	3	-	-
Horse Colic	299/67	2	19	7	+	+
House-votes-84	435	2	16	0	+	-
Monks 1	124/432	2	6	0	-	+
Monks 3	122/432	2	6	0	-	+
Soybean-small	47	4	35	0	-	-
Soybean-large	290/340	15	35	0	+	+
Ttic-tac-toe	958	2	9	0	-	-

Cechy określone jako dyskretne w powyższej tabeli to cechy o charakterze symbolicznym i cechy zapisane przy pomocy liczb naturalnych, dla których można stosować poniżej

podane wzory zastępujące wartości atrybutów prawdopodobieństwami warunkowymi. Cechy o charakterze ciągłym były pozostawianie bez żadnych zmian.

W przypadku, gdy baza nie posiadała zbioru testowego wykonywano dziesięć razy dziesięciokrotną cross validation tj. „uczymy" nasz system na 9/10 wszystkich dostępnych obiektów w danej bazie danych, pozostałe obiekty używamy do testowania „poziomu nauczenia" systemu. Uzyskane w ten sposób wyniki zostały uśrednione.

5.2 Test 1

W pierwszym teście zamiast symbolu będziemy wstawiali wektor którego składowe będą prawdopodobieństwami warunkowymi określanymi na podstawie wzoru $p(C_j \mid x) = N_j(x) / N(x)$. Indeks j przebiega po wszystkich klasach, $N(x)$ jest liczbą wszystkich wystąpień atrybutu x dla danej cechy, $N_j(x)$ jest liczbą wystąpień atrybutu x w j-tej klasie.

Wyniki klasyfikacji dla systemu FSM podane są w procentach. Dla każdego zbioru przeprowadzano dziesięć prób i ich średni wynik wraz z błędem standardowym zamieszczono w tabelce. Na danych nie przeprowadzano standaryzacji. Jako funkcje realizowane przez neurony systemu FSM wybrano funkcje Gaussowskie.

	Wynik na zbiorze	
	przed zamianą	zamienionym
Annealing	76,60±1,26	**77,80**±1,81
Australian Credit	84,93±0,95	**85,31**±1,00
Flags	51,34±2,42	**61,54**±2,57
Horse Colic	**71,57**±2,58	69,63±2,91
House-votes-84	81,56±0,45	**89,40**±0,79
Monks 1	**94,24**±3,88	**94,39**±1,66
Monks 3	96,28±1,04	**97,24**±0,54
Soybean-small	**99,20**±1,69	97,90±0,21
Soybean-large	**86,51**±1,59	**86,75**±1,94
Ttic-tac-toe	**94,80**±0,78	83,04±0,48
Zoo	**93,76**±0,96	92,75±1,07

Hasło *„przed zamianą"* oznacza, że zbiór nie był modyfikowany jeżeli wszystkie cechy były zapisane przy pomocy liczb. Jeżeli zbiór zawierał symbole, przypisywano symbolom kolejne liczby naturalne 1,2,3...

Wyniki dla kNN: Do określania odległości pomiędzy obiektami użyto miary euklidesowej. Test przeprowadzono identycznie jak powyżej (te same zbiory, tak samo przygotowane). Wartość parametru k (liczba najbliższych sąsiadów) dla wszystkich testowanych baz danych przyjęto równe jeden. Wyniki podane są w procentach.

W systemie kNN można stosować miary VDM bezpośrednio. Wyniki uzyskane w tym teście są identyczne (potwierdziły to badania) z wynikami uzyskanymi w kNN-ie po

zastosowaniu do obliczania odległości miary VDM dla cech dyskretnych i miary euklidesowej dla ciągłych.

	Wynik na zbiorze	
	Przed zamianą	Zamienionym
Annealing	69,00	**75,00**
Australian Credit	65,29±0,77	**66,12**±0,80
Flags	**41,37**±1,67	40,39±1,34
Horse Colic	43,28±0,00	**65,67**±0,00
House-votes-84	**81,72**±1,39	79,78±0,49
Monks 1	85,88	**87,96**
Monks 3	90,97	**93,75**
Soybean-small	97,95±0,16	**100,00**±0,00
Soybean-large	77,94	**89,12**
Ttic-tac-toe	89,53±0,50	**94,22**±0,41
Zoo	**97,41**±0,51	96,45±1,06

Brak błędu standartowego oznacza, że dana baza danych posiada zbiór testowy. W takim przypadku algorytmy kNN i C4.5 są algorytmami deterministycznymi w przeciwieństwie do systemu FSM.

Wyniki dla C4.5.
Wszystkie cechy w C4.5 zostały potraktowane przez algorytm jako ciągłe tj. zastosowane zostały testy w postaci $x \leq C$ lub $x \geq C$. Zastosowany algorytm może także stosować testy w postaci $x \in [0.1 \ldots 0.3]$ dla cech dyskretnych, jednakże dostępna wersja programu nie działa stabilnie przy takim traktowaniu cech dyskretnych. Rubryka „Reguły" podaje średnią liczbę reguł produkowanych przez algorytm do rozróżniania obiektów w danej bazie danych. Wyniki podane są w procentach.

	Wynik na zbiorze			
	przed zamianą	Reguły	zamienionym	Reguły
Flags	57,51±9,46	11,40	**65,20**±6,93	8,90
Monks 3	**92,60**	10,00	**92,60**	7,00
Soybean-small	98,00±6,32	5,00	**98,50**±4,74	5,00
Soybean-large	89,40	25,00	**92,60**	23,00
Zoo	**92,96**±8,51	8,90	91,14±8,09	7,90

Stosowanie naszej transformacji do danych w przypadku algorytmu C4.5 nie zawsze poprawia wyniki klasyfikacji, można jednakże zauważyć spadek liczby reguł potrzebnych systemowi do rozróżniania obiektów przy zachowaniu identycznego poziomu klasyfikacji jak dla danych „przed zamianą". Dodatkowo obserwuje się spadek błędu standardowego zarówno w C4.5 jak i FSM oraz KNN w większości testowanych baz danych.

5.3 Test 2

W tym teście zamiast symbolu będziemy wstawiali wektor o składowych: $p(x \mid C_j) = N_j(x) / N_j$ gdzie N_j jest liczbą wszystkich wystąpień j klasy, $N_j(x)$ tak jak powyżej jest liczbą wystąpień atrybutu *x* w *j*-tej klasie.

Wszystkie uwagi dotyczące poszczególnych systemów z Testu 1 są aktualne także w tym teście. Na danych nie przeprowadzano standaryzacji w badanych systemach. Wyniki podawane są w procentach.

Wyniki dla systemu FSM.

	Wynik na zbiorze	
	przed zamianą	zamienionym
Annealing	**76,60**±1,26	**76,60**±1,07
Australian Credit	84,93±0,95	**85,31**±1,00
Flags	51,34±2,42	**62,31**±2,50
Horse Colic	71,57±2,58	**77,37**±5,24
House-votes-84	81,56±0,45	**90,18**±0,33
Monks 1	**94,24**±3,88	**95,06**±0,86
Monks 3	96,28±1,04	**97,20**±0,00
Soybean-small	**99,20**±1,69	95,65±0,24
Soybean-large	**86,51**±1,59	**86,71**±0,75
Ttic-tac-toe	94,80±0,78	**95,31**±0,88
Zoo	93,76±0,96	**95,59**±1,43

Wyniki dla kNN.

	Wynik na zbiorze	
	przed zamianą	zamienionym
Annealing	69,00	**73,00**
Australian Credit	65,29±0,77	**66,70**±0,70
Flags	**41,37**±1,67	39,96±1,44
Horse Colic	43,28	**74,63**
House-votes-84	**81,72**±1,39	79,90±0,76
Monks 1	**85,88**	84,72
Monks 3	90,97	**93,06**
Soybean-small	97,95±0,16	**100,00**±0,00
Soybean-large	77,94	**89,17**
Ttic-tac-toe	**89,53**±0,50	77,10±0,73
Zoo	**97,41**±0,51	96,09±1,31

Wyniki dla C4.5

	Wynik na zbiorze			
	przed zamianą	Reguły	Zamienionym	Reguły
Flags	57,51±9,46	11,40	**64,00**±12,42	8,10
Monks 3	**92,60**	10,00	**92,60**	10,00
Soybean-small	98,00±6,32	5,00	**100,00**±0,00	5,00
Soybean-large	89,40	25,00	**92,60**	23,00
Zoo	**92,80**±8,51	8,90	92,19±6,03	8,10

Problemem jaki pojawia się podczas stosowania obu typów transformacji (Test1 i Test2) jest znaczny w niektórych przypadkach (bazy danych z dużą liczbą klas) wzrost liczby cech. Pojedynczy atrybut symboliczny jest zamieniany na *C* wartości numerycznych (*C* jest liczbą klas). W przypadku dwóch klas można uniknąć tego problemu, korzystając z zależności:

$$d(x,y) = |P(1 \mid a = x) - P(1 \mid a = y)|^q + |P(2 \mid a = x) - P(2 \mid a = y)|^q =$$
$$2\,|P(1 \mid a = x) - P(1 \mid a = y)|^q$$

Wzrost liczby cech opisujących obiekty jest szczególnie dotkliwy w małych bazach danych, gdy stosunek liczby cech do liczby obiektów w danej bazie zbliża się do jedności, dane stają się trudne do analizy za pomocą algorytmów klasyfikacyjnych, następuje spadek wyników klasyfikacji.

5.4 Test procedury PCA

Aby uniknąć wzrostu liczby cech po zastosowaniu powyższych wzorów transformacyjnych zbadamy w poniższym rozdziale przydatność do tego celu procedury PCA (Principal Component Analysis).

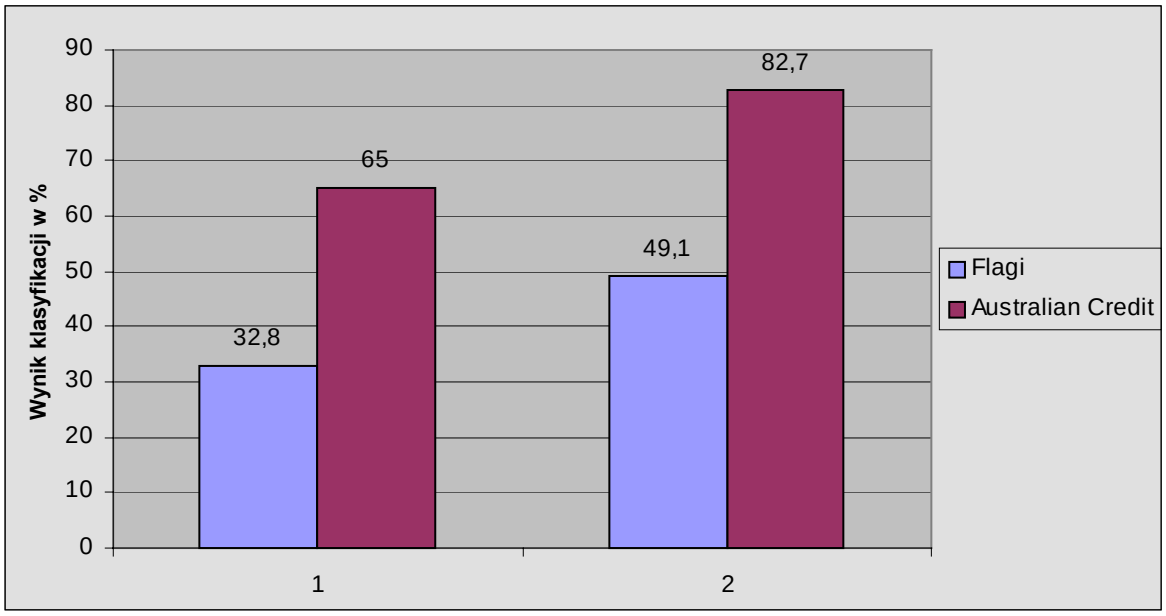
Kolejne kroki procedury analizy czynników głównych to :
- wyznaczenie macierzy kowariancji dla naszego zbioru danych. Kolejne elementy tej macierzy oblicza się na podstawie wzoru: $\sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})$ gdzie $\bar{x}$ i $\bar{y}$ są wartościami średnimi cech w kolejnych kolumnach, *N* jest liczbą wektorów w naszej bazie danych.
- wyznaczenie wartości własnych macierzy kowariancji. Wybieramy *n* największych wartości własnych i wyznaczamy odpowiadające im wektory własne.
- utworzenie kombinacji liniowej wyznaczonych wektorów własnych macierzy kowariancji z naszymi danymi wejściowymi. Taką samą kombinację liniową tworzymy również dla zbioru testowego jeżeli dana baza danych go posiada.

Po wykonaniu PCA obiekty są w danej bazie danych opisane przy pomocy tylko *n* cech.

W pierwszym teście jaki wykonamy zbadamy wpływ standaryzacji cech o charakterze ciągłym na wyniki klasyfikacji uzyskiwane po zastosowaniu procedury PCA.

Na poniższym wykresie jedynką oznaczono wyniki klasyfikacji po zredukowaniu danych wejściowych do 3 cech bez przeprowadzania standaryzacji (w danych występowały cechy

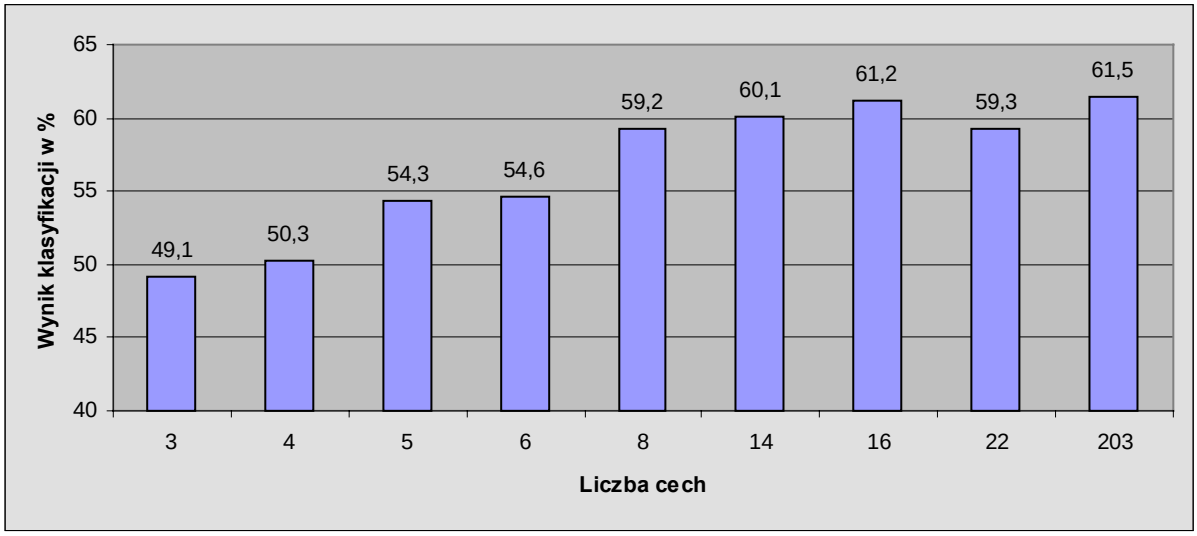
niesymboliczne o charakterze ciągłym). W przypadku drugim przeprowadzono identyczną redukcję liczby cech z uprzednią standaryzacją cech ciągłych. Wszystkie testy procedury PCA przeprowadzano w systemie FSM. Do standaryzacji cech ciągłych użyto najprostszego typu standaryzacji: dzielenie przez największą wartość.



Jeżeli chcemy redukować liczbę cech danych w których część cech ma charakter ciągły o wartościach nie leżących w przedziale [0,1] trzeba dla tych cech wykonać standaryzację. Nie wykonanie takiej standaryzacji powoduje zaburzenie wyznaczanych wartości własnych i odpowiadających im wektorów własnych, co powoduje pogorszenie końcowych wyników klasyfikacji.

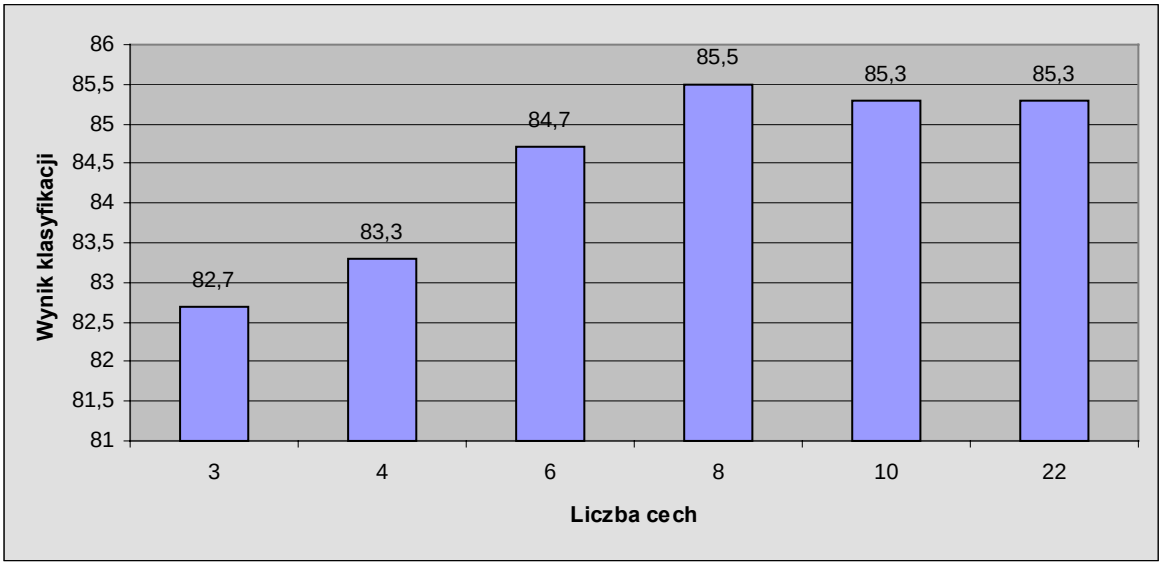
Wyniki klasyfikacji w zależności od stopnia redukcji ilości danych.

Poniższy wykres przedstawia wyniki klasyfikacji dla zbioru *Flagi.* Ostatni słupek przedstawia wynik dla zbioru „oryginalnego", na którym nie przeprowadzano redukcji danych za pomocą PCA.



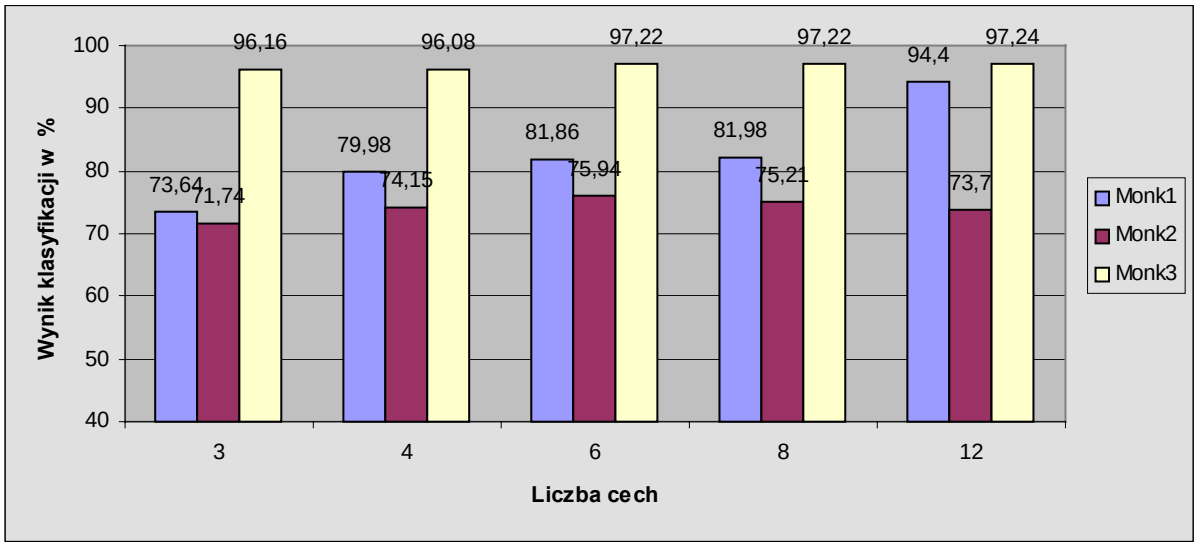
Jak widać z wykresu, jednakowo dobry wynik klasyfikacji w stosunku do zbioru nie modyfikowanego (który zawierał aż 203 cechy) uzyskuje się dla danych zredukowanych do 16 cech.

Podobny test został wykonany dla zbioru *Australian Credit.*



W przypadku tego zbioru dobre wyniki uzyskujemy przy redukcji danych do 8 cech z wejściowych 22.

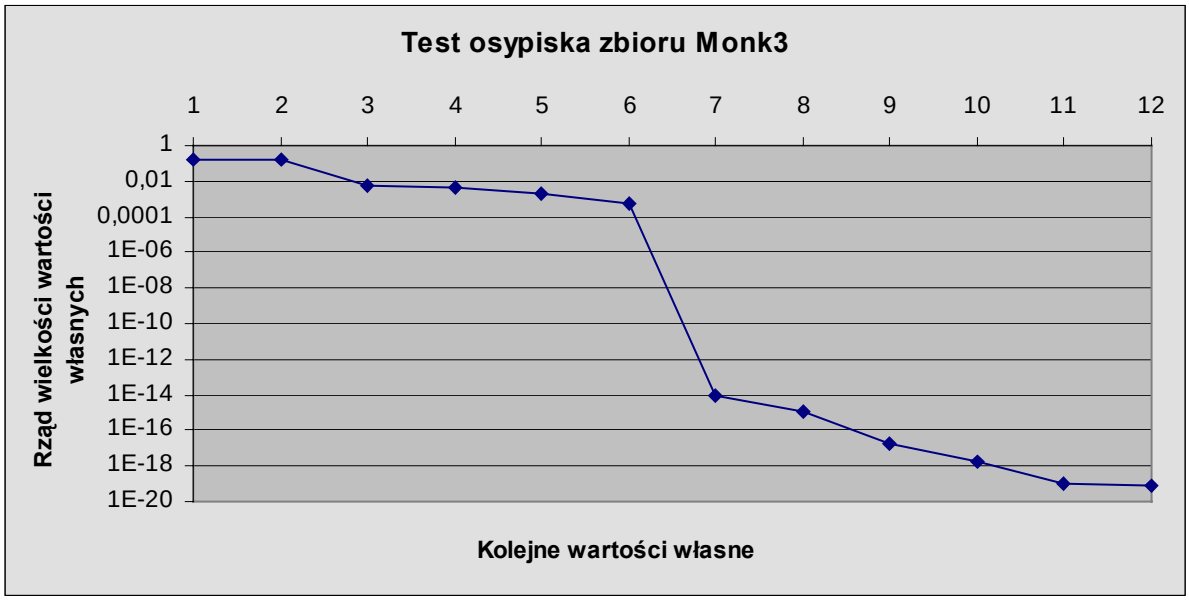
Rezultaty dla zbiorów: *Monk1*, *Monk2* i *Monk3*. po zastosowaniu naszej transformacji zastępującej symbole prawdopodobieństwami (wzrost liczby cech z 6 do 12).



Uzyskujemy następujące wyniki:
- Dla zbioru *Monk1* każda próba redukcji danych powoduje pogorszenie wyniku klasyfikacji.
- Dla zbioru *Monk2* przy redukcji do 6 cech poprawiamy wynik klasyfikacji w porównaniu z danymi wejściowymi o 2,24%.

- Dla zbioru *Monk3* redukcja do 6 cech daje identyczne rezultaty jak dla danych wejściowych (12 cech)

Stosowanie tej procedury jest kosztowne obliczeniowo a wykonanie dodatkowo kilku prób do określenia optymalnej liczby cech dodatkowo zwiększa ten koszt kilkakrotnie. Wykonywania kilku prób można uniknąć używając tzw. *testu osypiska.*



Test ten został po raz pierwszy zaproponowany przez Cattella w kontekście problemu liczby czynników w analizie PCA. Cattell sugerował odnalezienie na wykresie miejsca od którego na prawo występuje gwałtowny spadek modułu wartości własnych.

Dla zbioru *Monk3* taki spadek następuje po szóstej w kolejności co do wartości modułu wartości własnej. Dla zbiorów z większą liczbą wartości własnych np. *Flagi* (203 wartości własne) powyższy wykres byłby liną prostą (w przybliżeniu) i test osypiska nie pozwoliłby wytypować optymalnej liczby cech. Jednakże na podstawie zbiorów *Monk3* można zauważyć, że nie warto brać pod uwagę (typując optymalną liczbę cech) wartości własnych mniejszych niż $\sim 10^{-5}$ Reguła ta sprawdza się np. w przypadku zbioru *Australian Credit* gdzie test osypiska typuje aż 14 wartości własnych. Z przeprowadzonych testów wynika, że najlepsze wyniki uzyskuje się biorąc pod uwagę 8 cech.

6. Podsumowanie

Znalezienie wydajnych metod konwersji zmiennych symbolicznych do postaci numerycznej jest ważne dla wielu algorytmów klasyfikacyjnych. Metody konwersji prezentowane w tej pracy wzorowane są na mierze VDM. Na podstawie przeprowadzonych badań można stwierdzić, że prezentowane metody w wielu przypadkach dają poprawę wyników klasyfikacji oraz spadek błędu standardowego. Ponadto w przypadku FSM oraz C4.5 daje się także obserwować odpowiednio spadek liczby neuronów oraz liczby reguł potrzebnych do rozróżniania obiektów w testowanych bazach danych. Zastosowane metody zamiany wartości symbolicznych wartościami numerycznymi, pozwalają stosować dowolne klasyfikatory (oparte na statystyce, neuronowe i inne) na tak przygotowanych danych.

Wszystkie testowane bazy danych wraz z programem służącym do wstępnego przygotowywania danych znajdują się na dołączonej do pracy płycie CD-ROM.

7. Literatura

[1] Duch W, Grudziński K, Stawski G (2000) *Symbolic features in neural networks*. Fifth Conference on Neural Networks and Soft Computing, Zakopane, 6/2000, pp. 180-185.

[2] Walesiak M. *Metody analizy danych marketingowych* Wydawnictwo Naukowe PWN 1996.

[3] Gowda C K, Krishna G (1978) *Disaggregative clustering using the concept of mutual nearest neighborhood* IEEE Transactions on System, Man and Cybernetics no 12. 1978

[4] Smith S P *Threshold validity for mutual neighborhood clustering* IEEE Transactions on Pattern Analysis and Machine Intelligence vol 15 no 1, 1993.

[5] Gatnar E *Symboliczne metody klasyfikacji danych* Wydawnictwo Naukowe PWN 1998.

[6] Michalski R S, Stepp R E *Concept based clustering versus numerical analysis.* Technical report 1073 University of Illinois 1981

[7] Michalski R S, Stepp R E *Learning from observations: conceptual clustering.* Machine learning an artificial intelligence approach Tioga Palo Alto 1983a

[8] Fisher D *Knowledge acquisition via incremental conceptual clustering.* Machine learning no 2, 1987a

[9] Fisher D *Knowledge acquisition via incremental conceptual clustering.* Technical report 87-22 University of California, Irvine 1987b.

[10] Feigenbaum E A, Simon H A *EPAM - like models of recognition and learning.* Cognitive Science no 8, 1984.

[11] Lebowitz M. *Experiments results with incremental concept formation UNIMEM.* Machine learning no 2 1987.

[12] Kolodner J L *Reconstructive memory: a computer model.* Cognitive Science 7, 1983.

[13] Quinlan J R *Learning efficient classification procedures.* Machine learning an artificial intelligence approach Tioga Palo Alto 1983

[14] Duch W, Grąbczewski K *The separability of split value criterion.* 5th Conference on Neural Networks and Soft Computing, Zakopane, June 2000, pp. 201-208

[15] Quinlan J R *C4.5 programs for machine learning.* Morgan Kaufmann, San Mateo 1993.

[16] Wilson D. Randall, Tony R. Martinez (1996). *Heterogeneous Radial Basis Functions*, Proceedings of the International Conference on Neural Networks(ICNN'96), vol. 2, pp. 1263-1267, 1996.

[17] Wilson D. Randall, Tony R. Martinez (1996). *Value Difference Metrics for Continuously Valued Attributes*, International Conference on Artificial Intelligence, Expert Systems and Neural Networks (AIE'96), pp. 74-78 1996.

[18] Wilson D. Randall, Tony R. Martinez (1997a). *Improved heterogeneous distance functions* Journal of Artificial Intelligence Research, vol. 6, no. 1, pp. 1-34, 1997.

[19] Friedman J. H. (1994) *Flexible metric nearest neighbor classification,* praca dostępna na anonymous FTP playfair.stanford.edu.

[20] Short R D, Fukunaga K, (1981). *The optimal distance measure for nearest neighbor classification* IEEE Transactions on Information Theory 27 pp 622-627, 1981.

[21] Blanzieri E, Ricci F (1999). *Advanced Metrics for Class-Driven Similarity Search*. International Workshop on Similarity Search, Firenze, Italy, September 1999.