AUTOMATYKA • 2008 • Tom 12 • Zeszyt 3

Paweł Kośla*, Marcin Raniszewski*

Nowe metody selekcji cech i redukcji zbiorów odniesienia dla klasyfikatora typu 1-NN

1. Wprowadzenie

W zadaniach *klasyfikacji wzorców* bardzo często stosuje się intuicyjną, prostą i w wielu wypadkach niezwykle efektywną *regułę k najbliższych sąsiadów* (*k*-NN*, k Nearest Neighbour*) [9, 11]. Konstrukcja klasyfikatora opartego na tej regule wymaga znajomości zbioru obiektów o znanej przynależności do klas, zwanego *zbiorem uczącym* lub *zbiorem treningowym*. Na podstawie tego zbioru można wyznaczyć parametr *k.* Reguła ta polega na zaklasyfikowaniu obiektu do klasy, która przeważa wśród *k* najbliższych sąsiadów tego obiektu, *w zbiorze odniesienia*. Zbiorem odniesienia jest zwykle cały zbiór uczący, ale może nim być wydzielony podzbiór tego zbioru lub inny sztuczny zbiór utworzony na podstawie zbioru uczącego. Dowodzi się, że ta metoda jest przy odpowiednich założeniach zbieżna do *klasyfikatora Bayesa*, tj. do klasyfikatora oferującego teoretycznie najmniejsze prawdopodobieństwo mylnej decyzji [9]. Szczególnym przypadkiem klasyfikatora *k*-NN jest klasyfikator 1-NN (*jednego najbliższego sąsiada*), w którym klasę obiektu ustala się na podstawie klasy obiektu mu najbliższego. O efektywności reguły 1-NN świadczy między innymi fakt, że dla dostatecznie liczebnego zbioru treningowego, oferowane przez nią prawdopodobieństwo mylnej decyzji nie przekracza podwojonej frakcji błędów klasyfikatora Bayesa [9].

Zarówno reguła 1-NN, jak i jej ogólniejsza wersja: *k*-NN, mają jednak wady. Dla dużych zbiorów odniesienia, rozpoznawanie nowych obiektów może wymagać zbyt wielu obliczeń (liczenie najmniejszej odległości do wszystkich sąsiadów). Aby usunąć wymienioną wadę próbuje się optymalizować proces liczenia odległości, albo zabiega się o to, by zbiór odniesienia był mniejszy. O ile w pierwszym przypadku uzyskujemy zwiększenie szybkości klasyfikacji o tej samej frakcji błędów co w standardowej regule 1-NN, o tyle w drugim, z powodu minimalizacji zbioru odniesienia, dodatkowo możliwe jest zwiększenie *jakości klasyfikacji* (udziału poprawnych klasyfikacji). Dzieje się tak w przypadku, gdy

* Katedra Informatyki Stosowanej, Politechnika Łódzka

odrzucimy nadmiarowe *cechy* (parametry opisujące obiekty), które utrudniały poprawną klasyfikację i cechy, które nie wnosiły żadnej dodatkowej informacji z punktu widzenia danego problemu (*selekcja cech*), bądź też usuniemy obiekty, które stanowiły *szum* lub ich istnienie nic nie wnosiło do procesu klasyfikacji (*redukcja zbioru odniesienia*). Są też przypadki, gdy selekcja cech lub redukcja zbioru odniesienia spowoduje obniżenie jakości klasyfikacji. Należy wtedy rozważyć, czy zależy nam na szybszej, czy na bardziej dokładnej klasyfikacji.

Oczywiście szybkość reguły 1-NN można optymalizować poprzez jednoczesną redukcję zbioru odniesienia i selekcję cech.

Pomimo identycznych celów, które przyświecają metodom selekcji cech i redukcji zbioru odniesienia, stanowią one odrębne zagadnienia badawcze i wykorzystują inne techniki oraz inne algorytmy. W związku z tym często prezentowane są niezależnie od siebie jako metody minimalizacji zbioru odniesienia. W niniejszym artykule przedstawiony został wpływ jednoczesnego działania obu tych metod na proces klasyfikacji. Dodatkowo zaprezentowane zostały dwa nowe algorytmy: selekcji cech (*algorytm wykorzystujący badanie zależności między cechami*), którego autorem jest Paweł Kośla, oraz redukcji zbioru odniesienia (*sekwencyjny algorytm wykorzystujący podwójne sortowanie*), którego autorem jest Marcin Raniszewski.

2. Selekcja cech

Zmniejszając rozmiar przestrzeni *atrybutów* (cech), obniżamy koszt ich pomiaru, przyspieszamy klasyfikację, a także możemy podnieść jej jakość. Wybór cech powinien być taki, by możliwa była klasyfikacja z jak najmniejszym prawdopodobieństwem wystąpienia błędnej decyzji, przy jak najmniejszej liczbie cech. Istnieje wiele metod selekcji cech, w niniejszym artykule wykorzystano dwie.

Pierwszą jest dobrze znana i wykorzystywana w wielu systemach rozpoznawania wzorców procedura *Forward Feature Selection* (w dalszej części określana jest skrótem FFS). Polega ona na wyborze cechy oferującej największą jakość klasyfikacji, a następnie kolejnym dołączaniu jednej z pozostałych cech, by uzyskać jeszcze wyższą jakość. Metoda ta została szczegółowo opisana w [4].

Drugą procedurą selekcji cech stosowaną w niniejszej pracy (po FFS) jest algorytm będący kombinacją wcześniejszych wyników prac jednego z autorów [10, 13]. Bazuje on na *analizie zależności między cechami*. Idea nowego algorytmu polega na zmniejszeniu stopnia wykorzystywania informacji o przynależności obiektów do klas. Dzięki temu podejściu, selekcja cech staje się metodą nienadzorowaną, a za tym – mniej podatną na przypadkowe dopasowanie zestawu cech do zbioru uczącego.

2.1. Analiza zależności między cechami – $corr(f_i, f_j)$

Analiza zależności między dwoma cechami realizowana jest z wykorzystaniem regresji nieliniowej, opartej na zmodyfikowanej regule k najbliższych sąsiadów (k-NN). Reguła k najbliższych sąsiadów, używana powszechnie jako reguła klasyfikacji, może być stoso-

wana również w regresji nieliniowej. Celem analizy regresji jest odtworzenie zależności wielkości skalarnej y od kilku zmiennych stanowiących składowe pewnego wektora $\mathbf{x}$. Jedyną informacją dostępną w tym celu jest zbiór uczący, jako zbiór par $\{(y_i, \boldsymbol{x}_i)\colon 1 = i = m\}$, gdzie m jest liczbą obiektów w zbiorze uczącym. Wnioskowanie wyniku następuje, podobnie jak w przypadku klasyfikacji, na podstawie informacji o k obiektach najbliższych obiektowi, dla którego chcemy wyznaczyć wartość y związaną z obiektem $\boldsymbol{x}$. Wartość y związana z obiektem $\boldsymbol{x}$ obliczana jest jako średnia lub mediana wartości tych k znanych obiektów ze zbioru uczącego będących najbliższymi sąsiadami obiektu $\boldsymbol{x}$. Przyjmijmy, że $\boldsymbol{x}_i$, $1 = i = k$, są najbliższymi sąsiadami obiektu $\boldsymbol{x}$ i niech:

$$K_{\mathbf{x}} = \left\{ \left(y_i, \mathbf{x}_i \right) : i = 1 \ldots k,\ d\left(\mathbf{x}, \mathbf{x}_1\right) < d\left(\mathbf{x}, \mathbf{x}_i\right) < d\left(\mathbf{x}, \mathbf{x}_k\right) \right\} \tag{1}$$

$$y_{\mathbf{x}} = f\left(y_1, \ldots, y_k\right) \tag{2}$$

gdzie:

K_x – zbiór k sąsiadów dla badanego punktu $\mathbf{x}$,
$f(\,)$ – średnia lub mediana,
d – odległość między dwoma punktami.

Dzięki temu wartość y przypisana badanemu obiektowi znajduje się w otoczeniu znanych wartości y_i odpowiadających k najbliższym sąsiadom.

Jakość regresji określana będzie *współczynnikiem determinacji* (kwadrat współczynnika korelacji) dla regresji nieliniowej. Określany jest on następującym wzorem [8]:

$$R^2 = 1 - \frac{\sum_{i=1}^{m} \left(y_i - \hat{y}_i \right)^2}{\sum_{i=1}^{m} \left(y_i - \overline{y} \right)^2} \tag{3}$$

gdzie:

y_i – znana wartość charakteryzująca obiekt,
$\hat{y}_i$ – wartość oszacowana metodą k-NN,
$\overline{y}$ – średnia z wartości obiektów w zbiorze odniesienia,
m – liczność zbioru odniesienia.

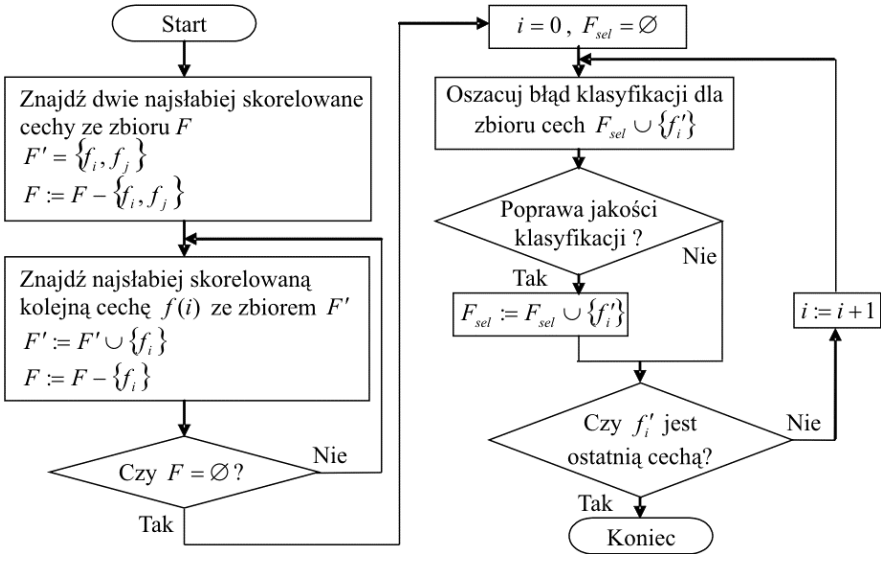
Załóżmy, że składową wektora $\boldsymbol{x}$ jest wartość cechy f_i, natomiast wartością y związaną z tym wektorem jest wartość cechy f_j. Wówczas dzięki regresji można odtworzyć zależność cechy f_j od cechy f_i. Dla tak postawionego problemu szacowanie jakości regresji (poprzez obliczenie współczynnika determinacji) określa, jak silna jest zależność między cechami. W dalszej części artykułu procedura badania zależności między cechami oznaczana będzie funkcją $corr(f_i, f_j)$. Wynikiem tej funkcji jest wartość współczynnika determinacji

dla regresji nieliniowej, zatem $corr(f_i, f_j) = 1$ oznaczać będzie silną zależność między cechami f_i i f_j, a wynik równy 0 oznaczać będzie brak takiej zależności.

2.2. Selekcja cech wykorzystująca zależności między cechami

Zaproponowana metoda selekcji cech polega na wstępnym uszeregowaniu atrybutów według kryterium, które nie wykorzystuje informacji o przynależności obiektów do klas. Następnie uruchamiana jest procedura dołączania kolejnych cech z tak uszeregowanego zbioru, badająca poprawę jakości klasyfikacji szacowanej *metodą minus jednego elementu* (*leave one out*) [9].

Dwie pierwsze pozycje w uszeregowanym zbiorze cech $F' = \{f_i, f_j\}$ zajmują cechy z oryginalnego zbioru cech (oznaczonego jako F i zawierającego l cech), które są najsłabiej ze sobą skorelowane. Współczynnik determinacji (funkcja $corr(f_i, f_j)$) dla tych dwóch cech jest najmniejszy spośród wszystkich dostępnych par atrybutów. Atrybut osiągający większą wartość odchylenia standardowego znajduje się w zbiorze F' przed cechą o mniejszym odchyleniu.



Rys. 1. Sieć działań zaproponowanego algorytmu selekcji cech. F – zbiór z oryginalnymi cechami, $F^{\smile}$ – zbiór uszeregowanych cech, F_{sel} – zbiór zawierający ostatecznie wybrane cechy

Pozostałe cechy znajdujące się w zbiorze $F := F - \{f_i, f_j\}$ są sekwencyjnie dołączane do zbioru F'. Cecha $f_i \in F$, która jest najsłabiej skorelowana z wybranym zestawem cech (min $corr(f_i F')$), jest dołączana do niego w następnej kolejności $F' := F' \cup \{f_i\}$. Jednocześnie i-ta cecha jest usuwana ze zbioru F. Procedura taka jest powtarzana, aż wszystkie cechy zostaną uszeregowane. Stopień zależności między cechą f_i a pewnym zestawem cech F' jest

określony jako maksymalna wartość spośród obliczonych współczynników determinacji dla wszystkich par f_i oraz $f_i \in F'$.

Drugi etap zaproponowanej procedury selekcji cech polega na kolejnym dołączaniu cech według porządku ustalonego w poprzednim etapie. Metoda rozpoczyna działanie od zestawu cech zawierającego jedną, pierwszą cechę w/w szeregu F'. Jeśli dodanie kolejnej cechy do zbioru cech poprawia jakość klasyfikacji, zostaje ona na stałe do niego dołączona, w przeciwnym razie analizowany jest kolejny atrybut z szeregu (rys. 1). W dalszej części artykułu ta metoda selekcji cech oznaczana będzie jako *RegFS*.

3. Redukcja zbioru odniesienia

Redukcja zbioru odniesienia polega na odrzuceniu jak największej liczby obiektów w taki sposób, aby na podstawie pozostałych można było nadal z odpowiednio wysokim prawdopodobieństwem klasyfikować poprawnie nowe obiekty. Zatem redukcja zbioru odniesienia jest bezpośrednio powiązana z regułą 1-NN, gdyż dla klasyfikatorów przeprowadzających rozpoznawanie na podstawie więcej niż jednego sąsiada, ze względu na rozrzedzenie zbioru odniesienia, głos mogłyby mieć obiekty mocno oddalone od klasyfikowanego obiektu, co nie jest wskazane. Jeśli jednak posiadamy klasyfikator k-NN ($k \neq 1$) i chcielibyśmy zredukować zbiór odniesienia, powinniśmy dokonać *reklasyfikacji* zbioru odniesienia regułą (k+1)-NN. Na otrzymanym w ten sposób zbiorze możemy działać regułą 1-NN, uzyskując przybliżoną frakcję poprawnych klasyfikacji jak w przypadku reguły k-NN. Ponieważ zbiór zreklasyfikowany dotyczy reguły 1-NN, to można dokonywać jego redukcji.

Powstało dużo różnych algorytmów redukcji zbioru odniesienia. Wiele z nich jako warunek zakończenia przyjmuje utrzymanie *zgodności* z redukowanym zbiorem odniesienia. Mówimy, że zredukowany zbiór odniesienia jest zgodny z oryginalnym, jeżeli klasyfikuje on poprawnie wszystkie obiekty z oryginalnego zbioru odniesienia (za pomocą reguły 1-NN). Innymi warunkami zakończenia mogą być z góry ustalona liczba obiektów w wyjściowym zbiorze zredukowanym lub znalezienie się w pewnym lokalnym minimum frakcji błędów, co jest równoważne z tym, że dowolne dołączenie lub odrzucenie punktu będzie powodowało wzrost liczby nieprawidłowych klasyfikacji.

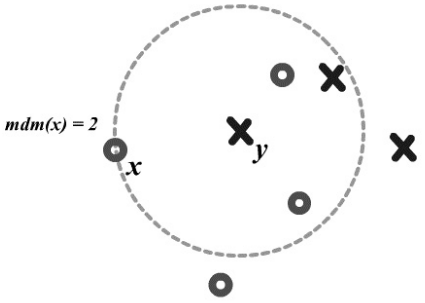
3.1. Algorytmy Harta i Gatesa

Algorytm Harta jest historycznie pierwszym algorytmem redukcji zbioru odniesienia [1]. Wykorzystuje on pojęcie zgodności jako kryterium zakończenia. Jego główną wadą jest zależność zbioru wynikowego od wstępnego uporządkowania zbioru odniesienia. Ponieważ początkowe obiekty w pierwszej fazie działania algorytmu są znacznie częściej dołączane do budowanego zbioru odniesienia, to wysoce prawdopodobne jest, że wynikowy zbiór będzie zawierał obiekty, które nie powinny się tam znaleźć. Dlatego często po algorytmie Harta stosuje się *algorytm Gatesa* [2], który usuwa po jednym punkcie ze zbioru

zredukowanego i sprawdza, czy w ten sposób pomniejszony zbiór nie stracił cechy zgodności. Dzięki temu obiekty dołączone niepotrzebnie w pierwszej fazie działania algorytmu Harta, mogą zostać usunięte.

3.2. Algorytm Gowdy–Krishny

Kolejnym z rodziny algorytmów zachowujących zgodność z oryginalnym zbiorem zredukowanym, jest *algorytm Gowdy–Krishny* [3]. Jest on pozbawiony po części głównej wady algorytmu Harta, sortuje wstępnie obiekty ze zbioru zredukowanego, używając tzw. *miary pozycyjnej*. Dla danego obiektu $\boldsymbol{x}$ znajdowany jest obiekt $\boldsymbol{y}$ z innej klasy, który leży w stosunku do niego najbliżej. Miara pozycyjna (oznaczmy ją jako *mdm – mutual distance measure*) określa liczbę obiektów tej samej klasy co $\boldsymbol{x}$, znajdujących się bliżej $\boldsymbol{y}$ niż $\boldsymbol{x}$ (rys. 2).



Rys. 2. Miara pozycyjna *mdm* dla obiektu $\boldsymbol{x}$ z klasy kółek

Obiekty posortowane rosnąco według miary pozycyjnej są poddawane algorytmowi Harta. Ponieważ obiekty znajdujące się blisko granic pomiędzy klasami mają zazwyczaj niską wartość miary pozycyjnej, to w algorytmie Gowdy–Krishny często obiekty te właśnie, jako pierwsze, trafiają do zbioru zredukowanego. Zgadza się to z intuicją, że między innymi takie obiekty powinny pozostać w zbiorze odniesienia.

Algorytm Gowdy–Krishny cechuje większy *stopień redukcji* (odsetek odrzuconych punktów ze zbioru odniesienia) i zazwyczaj wyższa jakość klasyfikacji z użyciem uzyskanych zbiorów zredukowanych, w porównaniu ze zbiorami uzyskanymi za pomocą standardowego algorytmu Harta. Niestety, w większości przypadków uzyskiwana jakość jest niższa niż jakość na pełnym zbiorze odniesienia. Wynika to przede wszystkim z warunku zatrzymania tych algorytmów, czyli z warunku zgodności.

3.3. Problem ze zgodnością

Zgodność powoduje, że silniejsza redukcja zbioru odniesienia jest ograniczona ze względu na utrzymanie poprawnej klasyfikacji wszystkich obiektów z oryginalnego zbioru odniesienia. Jeśli zbiór ten zawiera dużo szumu, to zredukowany zbiór odniesienia musi

zawierać także obiekty będące szumem, tak by spełniona była zgodność. Zatem, jest to warunek w pewnym sensie niewłaściwy i dla pewnych zbiorów nie prowadzi do zadowalających rezultatów.

Prezentowane poniżej algorytmy oparte są na innych kryteriach zakończenia.

3.4. Algorytm RMHC Skalaka

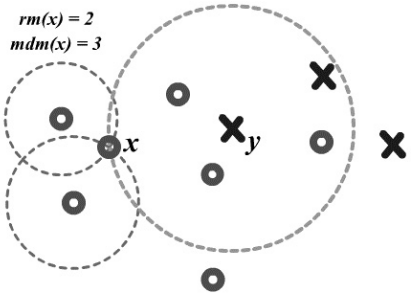
Algorytm RMHC Skalaka (*Random Mutation Hill Climbing*) [6] rozpoczyna się od zbudowania zredukowanego zbioru odniesienia z wylosowanych k obiektów z oryginalnego zbioru. Następnie przeprowadzanych jest m mutacji na zredukowanym zbiorze odniesienia, które polegają na wymianie losowo wybranego obiektu ze zbioru zredukowanego na również losowo wybrany obiekt ze zbioru oryginalnego, którego nie ma w zbiorze zredukowanym i sprawdzeniu jakości klasyfikacji zbioru oryginalnego za pomocą zbioru zredukowanego. Jeśli jakość po takiej wymianie się zwiększy, wymiana ta zostaje zaakceptowana, jeśli nie, powracamy do zbioru zredukowanego sprzed wymiany. Parametry k i m (odpowiednio: oczekiwana liczebność zbioru zredukowanego i liczba mutacji) są podawane przez użytkownika przed rozpoczęciem algorytmu.

Główną zaletą algorytmu RMHC Skalaka jest wysoka jakości klasyfikacji przy bardzo silnej redukcji zbioru odniesienia. W większości przypadków jakości otrzymane na zbiorach zredukowanych algorytmem RMHC Skalaka są wyższe od tych otrzymywanych na pełnym zbiorze odniesienia. Dodatkowymi zaletami są prostota algorytmu i łatwość implementacji.

Algorytm daje jednak za każdym razem różne wyniki. Wynika to z losowości algorytmu. Dodatkowo posiada on dwa parametry. Nie zawsze wiemy, do jakiej ilości obiektów powinien być zredukowany zbiór odniesienia, jaka liczebność jest odpowiednia, co więcej, nie wiadomo ile mutacji jest potrzebnych, by nie za mocno ani nie za słabo dopasować zbiór zredukowany do oryginalnego. Znalezienie tych wartości wiąże się z dodatkowymi testami dla szeregu par parametrów k i m, co jest utrudnione ze względu na wspomnianą niejednoznaczność wyniku.

3.5. Sekwencyjny algorytm redukcji zbioru odniesienia wykorzystujący podwójne sortowanie

Prezentowany w niniejszej pracy sekwencyjny algorytm wykorzystujący podwójne sortowanie *(Sequential Double Sort Algorithm – SeqDSA*) oparty jest na *algorytmie DSA* (*Double Sort Algorithm*) opisanym w [14] (wersja *rm-mdm*). Obiekty ze zbioru odniesienia są wstępnie sortowane według dwóch miar: *miary reprezentatywności* i miary pozycyjnej (zaczerpniętej z algorytmu Gowdy–Krishny). Miara reprezentatywności ma za zadanie odzwierciedlić przydatność obiektu jako dobrego reprezentanta swojej klasy. Miarą reprezentatywności obiektu $\boldsymbol{x}$ (oznaczmy ją symbolem *rm*) nazywamy liczbę obiektów z tej samej klasy co $\boldsymbol{x}$, które położone są bliżej $\boldsymbol{x}$ niż względem jakiegokolwiek obiektu z innej klasy (rys. 3).



Rys. 3. Miara reprezentatywności *rm* i miara pozycyjna *mdm* dla obiektu ***x*** z klasy kółek

Obiekty porządkowane są według malejącej miary reprezentatywności, tak by promować obiekty, które mają zdolność największej poprawnej klasyfikacji punktów ze swojego otoczenia. Następnie w ramach obiektów o tej samej wartości miary reprezentatywności są one sortowane względem rosnącej miary pozycyjnej (strategia *rm-mdm*). Dzięki temu pierwszeństwo wśród obiektów o tej samej sile reprezentatywności będą miały obiekty leżące blisko granic klas.

Standardowy *algorytm wykorzystujący podwójne sortowanie* (algorytm DSA) używał zmodyfikowanej wersji algorytmu Harta, by zbudować z posortowanych obiektów zredukowany zbiór odniesienia [14].

Algorytm SeqDSA polega na budowaniu zbioru zredukowanego poprzez dołączanie i odrzucanie obiektów posortowanych tak, by dzięki każdemu dołączeniu lub odłączeniu uzyskiwać coraz wyższą jakość klasyfikacji na pełnym zbiorze odniesienia.

Sekwencyjny algorytm wykorzystujący podwójne sortowanie składa się z następujących kroków:

1. Posortowanie obiektów ze zbioru odniesienia według strategii *rm-mdm*.
2. Dołączenie do wstępnie pustego zbioru X pierwszego obiektu z posortowanego zbioru odniesienia.
3. Poniższe kroki: 4. i 5. są wykonywane w pętli aż do sytuacji gdy żaden obiekt w punkcie 4., jak i w 5. nie zostanie odpowiednio dołączony lub odłączony do/z X.
4. Spośród niedołączonych jeszcze do zbioru X obiektów (według wyznaczonego podwójnym sortowaniem porządku) dołączane są pojedynczo obiekty i dla każdego dołączenia sprawdzana jest nowa jakość klasyfikacji pełnego zbioru odniesienia zbiorem X. Jeśli jakość się poprawia, to dołączony obiekt jest pozostawiany w X i przechodzimy do punktu 5., jeśli nie, obiekt jest odrzucany i sprawdzany jest kolejny obiekt. Jeśli dołączenie po kolei każdego z obiektów nie daje poprawy jakości klasyfikacji, przechodzimy do punktu 5.
5. Spośród dołączonych do X obiektów (według wyznaczonego podwójnym sortowaniem porządku, tylko, że od końca) odłączane są pojedynczo obiekty i dla każdego odłączenia sprawdzana jest nowa jakość klasyfikacji pełnego zbioru odniesienia zbiorem X. Jeśli jakość się poprawia, to odłączony obiekt nie wraca do zbioru X i przechodzimy do punktu 4., jeśli nie, obiekt jest z powrotem dołączany do X i sprawdzany jest

kolejny obiekt. Jeśli odłączenie po kolei każdego z obiektów nie daje poprawy jakości klasyfikacji, przechodzimy do punktu 4. (jeśli nie jest spełniony warunek wyjścia z pętli, czyli o ile wcześniej, w punkcie 4., nastąpiło dołączenie obiektu do zbioru X).

6. Otrzymany zbiór X jest zredukowanym zbiorem odniesienia.

4. Testy i wyniki

4.1. Zbiory

Testy zostały przeprowadzone na następujących zbiorach rzeczywistych i sztucznych: Liver Disorders (BUPA) [12], PIMA Indians Diabetes [12], WAVEFORM (wersja1) [12], Wisconsin Diagnostic Breast Cancer (WDBC) (Diagnostic) [12], YEAST [5, 12], SATIMAGE [15], FERRITES [7].

Każdy ze zbiorów został podzielony losowo 30 razy na dwie równe części: część treningową, która stanowiła zbiór odniesienia oraz część testującą, która służyła do uzyskania oceny klasyfikacji.

4.2. Opis testów i wyniki

We wszystkich testach użyto metryki euklidesowej i klasycznej standaryzacji. Wszystkie wynikowe zbiory zostały ocenione z wykorzystaniem zbiorów testujących, a średnie oceny jakości klasyfikacji z 30 podziałów przedstawione w tabelach 1–4.

Dla poszczególnych zbiorów treningowych zastosowano algorytmy selekcji cech a następnie redukcji zbiorów odniesienia. Wybór tej kolejności uzasadniony jest w podrozdziale 4.4.

Zbiory zostały poddane dwóm algorytmom selekcji cech:

1) FFS – algorytmowi *Forward Feature Selection*,
2) RegFS – algorytmowi opisanemu w podrozdziale 2.2.

Tabela 1 przedstawia rezultaty selekcji cech dla oryginalnych zbiorów odniesienia; w tabeli tej:

- kolumna *lcech* zawiera liczbę cech w oryginalnym zbiorze,
- kolumna *ucz* określa jakość klasyfikacji (w procentach) szacowaną na etapie uczenia z wykorzystaniem metody *leave one out*,
- kolumna *test* prezentuje średnią frakcję (w procentach) poprawnych klasyfikacji obiektów ze zbiorów testujących (na podstawie wyselekcjonowanych cech),
- kolumna *sel* prezentuje wynik selekcji cech rozumiany jako procentowy stosunek liczby wyselekcjonowanych atrybutów do oryginalnej liczby cech.

Uzyskane zbiory z wyselekcjonowanymi cechami oraz oryginalne zbiory odniesienia zostały poddane trzem algorytmom redukcji:

GK – algorytmowi Gowdy–Krishny,
RMHC – algorytmowi RMHC Skalaka,
SeqDSA – sekwencyjnemu algorytmowi wykorzystującemu podwójne sortowanie.

Tabela 1
Wyniki selekcji cech metodami *Forward Feature Selection* (FFS) i RegFS

Zbiory	lcech	1nn %	FFS			RegFS		
			ucz %	test %	sel %	ucz %	test %	sel %
BUPA	6	60,6	65,1	57,9	55,0	62,3	59,7	58,9
FERRITES	30	89,0	91,2	90,2	76,2	90,6	90,1	68,7
PIMA	8	69,8	72,5	67,6	52,1	71,1	67,8	26,3
SATIMAGE	36	89,6	90,7	89,5	75,0	89,9	89,2	60,1
WAVEFORM	21	76,1	78,9	76,8	65,7	77,0	76,5	80,0
WDBC	30	95,1	98,2	94,6	35,9	96,1	94,6	45,0
YEAST	8	51,1	51,8	50,7	88,8	48,6	47,7	79,2
Średnio	–	66,4	78,3	75,3	64,1	76,5	75,1	59,7

Tabela 2 przedstawia wyniki redukcji algorytmem GK pełnych zbiorów odniesienia (*oryg*), zbiorów odniesienia po selekcji cech algorytmem FFS oraz zbiorów odniesienia po selekcji cech algorytmem RegFS. Tabele 3 i 4 przedstawiają wyniki redukcji odpowiednio algorytmem RMHC i algorytmem SeqDSA dla analogicznych zbiorów odniesienia jak w tabeli 2.

W tabelach 2–4:

- kolumna *liczebność* zawiera oryginalne liczebności zbiorów,
- kolumna *test* prezentuje średnią frakcję (w procentach) poprawnych klasyfikacji na zbiorach testujących uzyskaną poprzez zastosowanie reguły 1-NN na zredukowanym zbiorze odniesienia,
- kolumna *red* prezentuje redukcję zbioru odniesienia (wyrażoną jako frakcja (również w procentach) liczebności zbioru zredukowanego w stosunku do liczebności pełnego zbioru odniesienia).

Parametr *k* algorytmu RMHC (patrz podrozdz. 3.4) każdorazowo był ustawiany na liczebność zbioru zredukowanego uzyskaną algorytmem SeqDSA dla danego zbioru, tak by możliwe było porównanie tych dwóch algorytmów na zbiorach o tej samej liczebności. Parametr *m* algorytmu RMHC (podrozdz. 3.4) został ustalony eksperymentalnie i wynosił odpowiednio:

300 – dla zbiorów BUPA, WDBC i YEAST,
500 – dla zbioru PIMA,
1000 – dla zbioru FERRITES,
2000 – dla zbioru SATIMAGE,
2500 – dla zbioru WAVEFORM.

Dodatkowo, dla celów porównawczych, w każdej tabeli znalazła się kolumna *1nn*, prezentująca średnie frakcje poprawnych klasyfikacji na pełnych zbiorach odniesienia.

Tabela 2
Wyniki redukcji algorytmu Gowdy–Krishny

Zbiory	Liczebność	1nn %	oryg		FFS		RegFS	
			test %	red %	test %	red %	test %	red %
BUPA	172	60,6	58,4	58,7	54,9	58,1	57,9	57,6
FERRITES	2949	89,0	86,6	21,0	86,6	20,1	87,1	20,2
PIMA	384	69,8	66,0	46,9	65,3	46,1	66,4	48,7
SATIMAGE	3216	89,6	87,2	20,1	86,9	20,1	86,6	20,6
WAVEFORM	2499	76,1	71,0	39,1	72,6	36,9	72,1	38,4
WDBC	284	95,1	93,2	14,1	92,9	12,7	93,0	14,8
YEAST	739	51,1	47,1	65,5	47,2	65,2	44,7	67,8
Średnio	–	75,9	72,8	37,9	72,3	37,0	72,5	38,3

Tabela 3
Wyniki redukcji algorytmu RMHC Skalaka

Zbiory	Liczebność	1nn %	oryg		FFS		RegFS	
			test %	red %	test %	red %	test %	red %
BUPA	172	60,6	59,8	13,4	59,1	7,6	59,8	8,7
FERRITES	2949	89,0	86,1	3,9	87,3	4,0	87,3	3,9
PIMA	384	69,8	71,9	7,3	71,9	4,2	73,1	2,9
SATIMAGE	3216	89,6	88,1	6,5	88,1	6,2	87,9	6,3
WAVEFORM	2499	76,1	80,0	3,6	81,2	2,5	80,8	3,1
WDBC	284	95,1	91,7	2,8	92,6	2,8	92,3	3,2
YEAST	739	51,1	52,6	10,6	52,6	10,4	50,0	10,7
Średnio	–	75,9	75,7	6,9	76,1	5,4	75,9	5,5

Tabela 4
Wyniki redukcji dla algorytmu SeqDSA

Zbiory	Liczebność	1nn %	oryg		FFS		RegFS	
			test %	red %	test %	red %	test %	red %
BUPA	172	60,6	60,8	13,4	59,3	7,6	60,9	8,7
FERRITES	2949	89,0	80,7	3,9	85,5	4,0	86,5	3,9
PIMA	384	69,8	72,2	7,3	71,8	4,2	72,2	2,9
SATIMAGE	3216	89,6	88,7	6,5	88,6	6,2	88,2	6,3
WAVEFORM	2499	76,1	83,6	3,6	83,2	2,5	83,8	3,1
WDBC	284	95,1	94,5	2,8	92,0	2,8	93,4	3,2
YEAST	739	51,1	55,7	10,6	56,2	10,4	52,2	10,7
Średnio	–	75,9	76,6	6,9	76,7	5,4	76,7	5,5

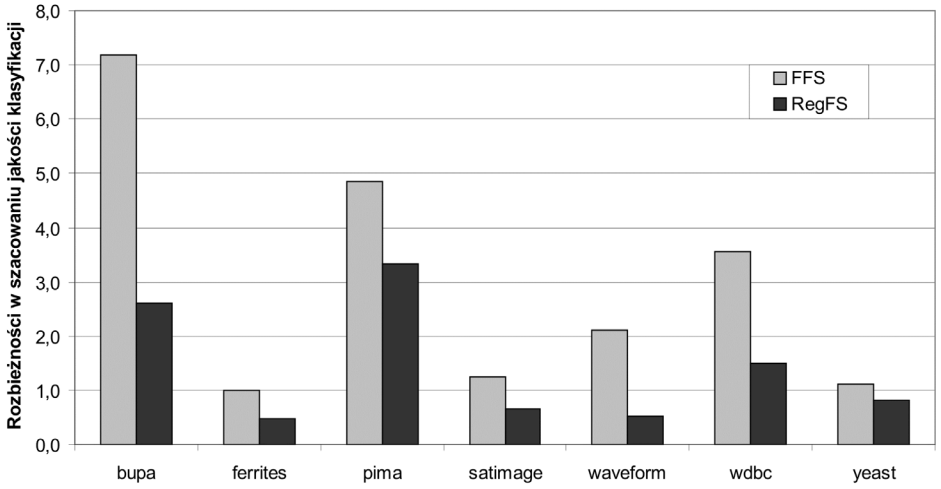
4.3. Wnioski

Algorytm selekcji cech wykorzystujący badanie zależności między cechami (RegFS) opracowywany był dla zadań selekcji cech na zbiorach o niedostatecznej liczebności obiektów w porównaniu z liczbą cech. Dla takich zadań metody nadzorowane powodują znaczny spadek jakości klasyfikacji obserwowanej na zbiorach testujących w porównaniu z oszacowaniem uzyskiwanym na podstawie zbiorów uczących. W niniejszych eksperymentach, gdzie liczba obiektów zbioru treningowego jest kilkanaście (kilkadziesiąt) razy większa niż liczba cech, metoda *Forward Feature Selection* sprawdza się bardzo dobrze. Mimo to, można zauważyć zalety nowego podejścia.

Pod względem jakości klasyfikacji szacowanej zbiorem testującym obie metody osiągają podobne rezultaty z wyjątkiem zbioru YEAST. Dla tego zbioru metoda FFS osiąga zdecydowanie lepszy rezultat pod względem jakości klasyfikacji, jednak należy zauważyć, że jest to zbiór słabo uwarunkowany, osiągający najgorszą jakość klasyfikacji spośród wszystkich analizowanych. Uwzględniając liczbę wyselekcjonowanych atrybutów, można stwierdzić, że w czterech przypadkach nastąpiła lepsza redukcja przestrzeni cech dla algorytmu RegFS. Dla zbioru PIMA różnica jest dwukrotna, a średnio metoda osiągnęła wynik lepszy o 4% od algorytmu FFS (patrz tab. 1).

Analizując jednocześnie jakość klasyfikacji na etapie uczenia i testowania można zauważyć zdecydowaną wyższość metody RegFS. Przewaga ta jest spowodowana ograniczonym stopniem wykorzystywania informacji o przynależności obiektów do klas. Różnice między tymi poziomami jakości dla poszczególnych zbiorów widoczne są na rysunku 4.

Pozornie lepsze wyniki szacowania jakości klasyfikacji na etapie uczenia dla metody FFS powodują ich nieprzydatność i konieczność stosowania fazy testowania. W przypadku metody RegFS, jakość klasyfikacji w fazie testowania jest bardziej przewidywalna (mniejsza różnica między jakością klasyfikacji szacowaną na etapie testowania i uczenia). Istnieją przypadki, kiedy nie jest możliwe wyodrębnienie zbioru testującego (na przykład spowodowane niewielką ilością danych), wówczas zastosowanie metod nienadzorowanych ma zdecydowanie lepsze uzasadnienie.



Rys. 4. Zestawienie różnic między jakością klasyfikacji szacowaną na etapie uczenia i testowania

Porównując wyniki w kolumnie *oryg* z tabel 2, 3 i 4 można wnioskować, że w porównaniu z algorytmem GK, zarówno algorytm SeqDSA, jak i RMHC prowadzi do zdecydowanie silniejszej redukcji (GK: średnia redukcja na poziomie 38%, SeqDSA i RMHC: średnio 7%, czyli poprawa o przeszło 81%) przy jednoczesnym zachowaniu wysokiej jakości klasyfikacji na większości zbiorów (GK: średnia jakość klasyfikacji niższa o około 4% od jakość klasyfikacji na pełnym zbiorze odniesienia, równa 72,8%, RMHC: średnia jakość prawie taka sama jak na pełnym zbiorze odniesienia, równa 75,7%, SeqDSA: średnia jakość minimalnie wyższa o około 1% od średniej jakości klasyfikacji na pełnym zbiorze odniesienia, która jest równa 76,6%).

Zastosowanie wstępnej selekcji cech zarówno algorytmem FFS, jak i algorytmem RegFS (kolumny *FFS* i *RegFS* w tab. 2, 3 i 4) nie powoduje wyraźnej poprawy podczas redukcji algorytmem GK w porównaniu z algorytmami RMHC i SeqDSA, gdzie widoczny jest minimalny wzrost średniej jakości klasyfikacji przy jednoczesnej silniejszej redukcji zbioru odniesienia (poprawa redukcji o przeszło 21%).

Dla zbiorów BUPA, FERRITES, PIMA oraz WDBC wstępne zastosowanie algorytmu RegFS przyczyniło się do uzyskania lepszych wyników w jakości klasyfikacji niż po zastosowaniu metody FFS.

Algorytm SeqDSA na podstawie analizowanych zbiorów dał średnio najlepsze wyniki pod względem jakości klasyfikacji.

4.4. Kolejność wykonywania procedur selekcji i redukcji

Niniejsza publikacja dotyczy jednoczesnego zastosowania procedur selekcji cech i redukcji zbioru odniesienia, czyli zastosowania obu metod w tym samym systemie rozpoznawania wzorców. W związku z tym wybór kolejności użycia tych procedur nie może być przypadkowy. Przeprowadzony został dodatkowy eksperyment determinujący ten porządek. W prezentacji wyników (tab. 5) ograniczono się do metod: selekcji cech – RegFS i redukcji zbioru odniesienia – SeqDSA. Kolumny *lcech*, *liczebność*, *red*, *sel* zostały opisane w podrozdziale 4.2. Kolumna *test* prezentuje średnią frakcję (w procentach) poprawnych klasyfikacji na zbiorach testujących uzyskaną poprzez zastosowanie reguły 1-NN na zredukowanym zbiorze odniesienia z wykorzystaniem tylko wyselekcjonowanych cech.

Tabela 5
Zestawienie rezultatów selekcji cech i redukcji zbiorów
w zależności od kolejności stosowania tych procedur

Zbiory	lcech	Liczebność	Selekcja + redukcja			Redukcja + selekcja		
			test %	red %	sel %	test %	red %	sel %
BUPA	6	172	60,9	8,9	58,9	57,3	13,2	60,6
FERRITES	30	2949	86,5	3,9	68,7	85,4	3,9	66,2
PIMA	8	384	72,2	2,7	26,3	70,9	7,2	42,1
SATIMAGE	36	3216	88,2	6,3	60,1	86,7	6,5	46,9
WAVE-FORM	21	2499	83,8	3,1	80,0	80,4	3,6	70,5
WDBC	30	284	93,4	3,1	45,0	94,3	3,0	87,0
YEAST	8	739	52,2	10,7	79,2	50,9	10,5	80,4
Średnio	–	–	76,7	5,5	59,7	75,1	6,8	64,8

Okazuje się, że zastosowanie w pierwszej kolejności selekcji cech a następnie redukcji zbioru odniesienia daje silniejszą redukcję rozmiaru danych, jak i wyższą jakości klasyfikacji. W takim podejściu zbiór odniesienia został średnio zredukowany o 97% danych (biorąc pod uwagę ilość obiektów i cech), w drugim podejściu średnio osiągnięto 95-procentową redukcję. Zwiększony błąd klasyfikacji dla drugiego podejścia związany może być z przeprowadzoną selekcją cech na zbiorze o małej liczności (po redukcji zbioru). Dla takich

zbiorów selekcja cech z większym prawdopodobieństwem prowadzi do przypadkowego dopasowania się zestawu cech do zbioru uczącego, co w rezultacie daje niekorzystne rezultaty klasyfikacji obiektów testowych. W przypadku redukcji zbioru odniesienia mniejszy rozmiar przestrzeni cech umożliwia silniejszą redukcję obiektów.

Stopień redukcji danych i jakość klasyfikacji zadecydowały, która kolejność została wybrana we właściwym procesie minimalizacji zbioru danych.

5. Podsumowanie

Zaprezentowany algorytm selekcji cech wykorzystując analizę zależności między cechami powoduje znacznie mniejsze rozbieżności między szacowanymi błędami na etapie trenowania i testowania. W związku z tym, w przypadkach, gdy nie ma możliwości wyodrębnienia zbioru testującego jego zastosowanie jest korzystniejsze niż zastosowanie metod nadzorowanych, których przykładem jest *Forward Feature Selection*.

Sekwencyjny algorytm redukcji zbioru odniesienia wykorzystujący podwójne sortowanie (SeqDSA) buduje zbiór zredukowany poprzez kolejne dołączanie i odrzucanie obiektów ze zbioru zredukowanego według kolejności wyznaczonej przez miary: reprezentatywności i pozycyjną. Jego zaletą w porównaniu do algorytmu RMHC Skalaka jest: brak parametrów i powtarzalny wynik działania, czyli brak losowości.

Przeprowadzone wyniki wskazują na ogromną przewagę algorytmów: RMHC Skalaka i SeqDSA nad algorytmem Gowdy–Krishny, zwracającym zgodny zbiór zredukowany, zarówno pod względem jakości klasyfikacji jak i stopnia redukcji.

Algorytm SeqDSA uzyskał średnio najlepsze wyniki na testowanych zbiorach.

Przeprowadzone testy wskazują na przydatność wstępnej selekcji cech, która powoduje spory wzrost stopnia redukcji zbiorów odniesienia, jak również poprawę jakości klasyfikacji. Zastosowanie takiej kolejności (najpierw selekcja cech, następnie redukcja zbioru odniesienia) skutkuje lepszą całkowitą redukcją ilości danych w zbiorze odniesienia niż w przypadku kolejności odwrotnej.

Literatura

[1] Hart P.E., *The condensed nearest neighbor rule*. IEEE Transactions on Information Theory, vol. IT-14, 3, 1968, 515–516.

[2] Gates G.W., *The reduced nearest neighbor rule*. IEEE Transactions on Information Theory, vol. 18, May, 1972, 431–433.

[3] Gowda K.C., Krishna G., *The condensed nearest neighbor rule using the concept of mutual nearest neighborhood*. IEEE Transaction on Information Theory, vol. IT-25, 4, 1979, 488–490.

[4] Devijver P., Kittler J., *Pattern Recognition: A Statistical Approach*. Englewood Clifs, Londyn, Prentice-Hall 1982.

[5] Nakai K., Kanehisa M., *Expert System for Predicting Protein Localization Sites in Gram-Negative Bacteria*. PROTEINS: Structure, Function, and Genetics, 11, 1991, 95–110.

[6] Skalak D.B., *Prototype and feature selection by sampling and random mutation hill climbing algorithms*. 11th International Conference on Machine Learning, New Brunswick, NJ, USA, 1994, 293–301.

[7] Jóźwik A., Chmielewski L., Skłodowski M., Cudny W., *A parallel net of (1-NN, k-NN) classifier for optical inspection of surface defects in ferrites*. Machine Graphics & Vision, 7, 1–2, 1998, 99–112.
[8] Starzyńska W., *Statystyka praktyczna*. Warszawa, PWN 2000.
[9] Duda R.O., Hart P.E., Stork D.G., *Pattern Classification – Second Edition*. John Wiley & Sons, Inc. 2001.
[10] Kośla P., *Zastosowanie regresji nieliniowej do selekcji cech*. XIV Konferencja „Sieci i systemy informatyczne", Łódź 2006, 217–220.
[11] Theodoridis S., Koutroumbas K., *Pattern Recognition – Third Edition*. Academic Press – Elsevier, USA, 2006.
[12] Asuncion A., Newman D.J., *UCI Machine Learning Repository* [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science 2007.
[13] Kośla P., *A feature selection approach in problems with a great number of feature*. Computer Recognition Systems 2, Advances in Soft Computing, vol. 45, Berlin/Heidelberg, Springer 2007, 394–401.
[14] Raniszewski M., *Reference set reduction algorithms based on double sorting*. Computer Recognition Systems 2, Advances in Soft Computing, vol. 45, Berlin/Heidelberg, Springer 2007, 258–265.
[15] The ELENA Project Real Databases [http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/REAL/].