ZESZYTY NAUKOWE POLITECHNIKI ŁÓDZKIEJ
Nr 1078 ELEKTRYKA, z. 121 2010

**MARCIN RANISZEWSKI**
Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki
Politechniki Łódzkiej

# KLASYFIKACJA DANYCH ALGORYTMY REDUKCJI I EDYCJI ZBIORÓW WYKORZYSTUJĄCE MIARĘ REPREZENTATYWNOŚCI

Recenzent: **doc. dr hab. Adam Jóźwik**

Maszynopis dostarczono 1. 10. 2010

*Klasyfikacja danych to podejmowanie decyzji na podstawie informacji, które te dane przenoszą (tzw. cech danych). Prawidłowa i szybka klasyfikacja zależy od prawidłowego przygotowania zbioru danych, jak i doboru odpowiedniego algorytmu klasyfikacji. Jednym z takich algorytmów jest popularny algorytm najbliższego sąsiada (NN). Jego zaletami są prostota, intuicyjność i szerokie spektrum zastosowań. Jego wadą są duże wymagania pamięciowe i spadek szybkości działania dla ogromnych zbiorów danych. Algorytmy redukcji usuwają znaczną część elementów ze zbioru danych, co znacząco przyspiesza działanie algorytmu NN, jednocześnie pozostawiając te, na podstawie których nadal można z zadawalającą jakością klasyfikować dane. Algorytmy edycji oczyszczają zbiór danych z nadmiarowych i błędnych elementów. W artykule zaprezentowane zostaną algorytm redukcji i algorytm edycji zbiorów danych, obydwa wykorzystujące miarę reprezentatywności. Testy przeprowadzono na kilku dobrze znanych w literaturze zbiorach danych różnej wielkości. Otrzymane wyniki są obiecujące. Zestawiono je z wynikami innych popularnych algorytmów redukcji i edycji.*

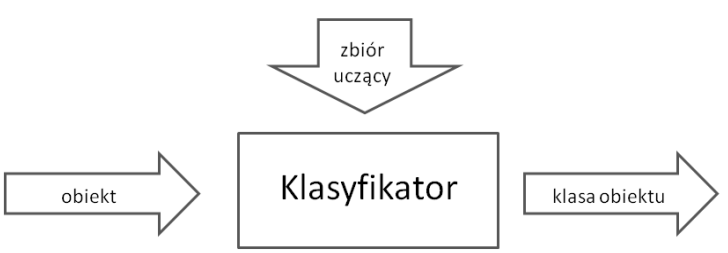
# 1. KLASYFIKACJA DANYCH

Klasyfikacja wzorców (ang. *Pattern Recognition*) to nauka, która może być stosowana wszędzie tam, gdzie istnieje potrzeba rozpoznawania (klasyfikacji) obiektów (wzorców, próbek) na podstawie ich cech [1,2,3]. Cechy (zarówno jakościowe jak i ilościowe) opisują obiekty, tworząc ich tzw. obrazy. Dlatego też często klasyfikację wzorców nazywa się rozpoznawaniem obrazów. Obiekty, a raczej ich obrazy są klasyfikowane do tzw. klas. Klasą nazywamy zbiór obiektów jednego rodzaju, zaś etykietą klasy – jej opis. Często etykietą klasy są po prostu liczby, gdzie każda liczbą oznacza pewną klasę. Przykładowo, klasyfikowanymi obiektami mogą być pacjenci, ich obrazami (cechami) wyniki badań nieinwazyjnych, zaś klasami: „*chory na chorobę x*"(klasa nr 1), „*nie chory, ale podatny na zachorowanie na chorobę x*" (klasa nr 2), „*nie chory i nie podatny na zachorowanie na chorobę x*" (klasa nr 3).

Klasyfikacja wzorców ma wspierać decyzyjność człowieka. Tam gdzie w stosunku do liczby rozpoznawanych obiektów, specjalistów mogących prawidłowo dokonać ich klasyfikacji jest niewystarczająco, systemy wspomagające rozpoznawanie mogą znacząco przyspieszyć i zwiększyć jakość procesów klasyfikacji. Także tam, gdzie chcemy dokonywać rozpoznawania obiektów na podstawie opisujących je cech, a nie jesteśmy w stanie wykryć zależności pomiędzy cechami a przypisanymi obiektom etykietami klas, klasyfikacja wzorców dostarcza nam takich metod.

## 1.1. Budowa klasyfikatora i jakość klasyfikacji

W klasyfikacji wzorców głównym zadaniem jest utworzenie klasyfikatora, który by z założoną pewnością rozpoznawał nowe obiekty na podstawie wcześniej dostępnego zbioru obiektów zwanego zbiorem uczącym. Od strony użytkownika klasyfikator można traktować jako „czarną skrzynkę", zbudowaną na podstawie informacji zawartej w zbiorze uczącym, która na wejściu przyjmuje „obraz" obiektu, natomiast na wyjściu zwraca decyzję, czyli etykietę klasy, do której według klasyfikatora należy zadany obiekt (rys. 1).




Rys. 1. Działanie klasyfikatora

Formalnie klasyfikator można traktować jako funkcję, sparametryzowaną informacjami ze zbioru uczącego, przyjmującą jako argument obiekt do zaklasyfikowania i zwracająca decyzję o przynależności tego obiektu do klasy. Do konstruującego klasyfikator należy wybór rodzaju tej funkcji oraz ustawienie wartości jej parametrów (o ile takie posiada). Powinna być ona dobrana w ten sposób by spełniała ograniczenia sprzętowe i czasowe klienta oraz z zadanym prawdopodobieństwem prawidłowo klasyfikowała nowe dane. To prawdopodobieństwo, z którym klasyfikator podejmuje prawidłową decyzję możemy jedynie przybliżać na podstawie próbek ze zbioru uczącego. Wartość przybliżona nazywana jest najczęściej jakością klasyfikacji, a popularną metodą jej znalezienia jest metoda zbioru testującego. W metodzie tej, ze zbioru uczącego wydziela się podzbiór obiektów (tzw. zbiór testujący), które następnie nie biorą udziału w budowaniu klasyfikatora. Obiekty te posłużą do określenia odsetka prawidłowo zaklasyfikowanych obiektów (jakości klasyfikacji) w ostatniej fazie, tzw. fazie oceny zbudowanego klasyfikatora. Jeśli odsetek ten jest za niski, proces budowy klasyfikatora musi zostać rozpoczęty ponownie, z uwzględnieniem zmiany typu klasyfikatora lub ponownego „przygotowania" danych ze zbioru uczącego.

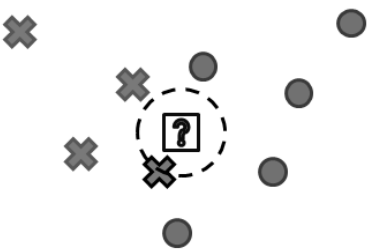
## 1.2. Selekcja obiektów i cech

Rozpoznawanie obrazów to nie tylko konstrukcja klasyfikatorów. To także przygotowanie danych (selekcja obiektów, selekcja cech). Zbiór uczący to zbiór, który otrzymujemy w celu zbudowania klasyfikatora. To na informacji zawartej w zbiorze uczącym ma być oparty wybór klasyfikatora, a także ustalenie jego parametrów. Wyróżniamy dwa główne typy zbiorów uczących i co za tym idzie dwa typy konstrukcji klasyfikatorów: zbiór uczący ze znanymi przynależnościami obiektów do klas – tzw. klasyfikacja nadzorowana (ang. *supervised learning*) oraz zbiór uczący bez takich przynależności – tzw. klasyfikacja nienadzorowana (ang. *unsupervised learning*) [3]. W ramach tego artykułu zajmować się będziemy klasyfikacją nadzorowaną.

Zbiór uczący często zawiera tzw. szum, czyli obiekty z błędnych pomiarów. Zakłócają one proces uczenia klasyfikatora, który korzystając z informacji zawartej w całym zbiorze uczącym, wykorzystuje także informacje błędną. Ich zlokalizowanie i usunięcie poprawia jakość klasyfikacji tworzonego klasyfikatora. Dodatkowo prawdopodobne jest, że niektóre cechy są nieistotne (nic nie wnoszą z punktu widzenia rozważanego problemu), nadmiarowe (ich wartości wynikają z wartości innych cech) lub obniżające jakość klasyfikacji. Prawidłowa selekcja cech może przyczynić się nie tylko wzrostu szybkości klasyfikatora (redukcja wymiaru przestrzeni cech), ale także do poprawy jego wyników.

## 2. REGUŁA NAJBLIŻSZEGO SĄSIADA

Reguła najbliższego sąsiada (NN - ang. *Nearest Neighbor Rule*) jest szczególnym przypadkiem reguły $k$ najbliższych sąsiadów ($k$-NN) dla $k = 1$ [1,2,3]. NN jest bardzo prostym, ale jednocześnie skutecznym i przez to popularnym klasyfikatorem. Nie posiada on żadnych parametrów, zaś faza uczenia ogranicza się do zapisania w pamięci wszystkich obiektów ze zbioru uczącego (które tworzą tzw. zbiór odniesienia). Obiekt jest klasyfikowany do klasy swego najbliższego (w sensie wybranej metryki; najczęściej jest to metryka euklidesowa) sąsiada-obiektu ze zbioru odniesienia (rys. 2).




Rys. 2. Klasyfikacja według reguły NN: nowy obiekt (oznaczony znakiem „?") jest zaklasyfikowany do klasy krzyżyków

Dowiedzione jest, że dla dostatecznie dużego zbioru odniesienia, prawdopodobieństwo błędu klasyfikacji reguły NN nigdy nie przekroczy podwojonej wartości prawdopodobieństwa błędu klasyfikatora Bayesa (teoretycznie najlepszego klasyfikatora) [1]. NN jest prosty w implementacji, intuicyjny, łatwo modyfikowalny w przypadku dynamicznie rozszerzającego się zbioru uczącego (wystarczy dodawać do zbioru odniesienia nowe wzorce), wystarczający dla większości zbiorów danych. Ma jednak wady. Wraz ze wzrostem liczby obiektów w zbiorze odniesienia maleje jego prędkość (konieczne jest znajdowanie najbliższego sąsiada z coraz większego zbioru odniesienia) oraz mogą pojawić się problemy z przechowywaniem w pamięci całego zbioru odniesienia. Ponadto NN jest podatny na negatywne działanie szumu w zbiorze odniesienia. Jeśli szum jest najbliższym sąsiadem klasyfikowanego obiektu, to zostanie on błędnie zaklasyfikowany. Alternatywą jest tu klasyfikator $k$-NN, gdzie wartość $k > 1$ ustala się na drodze dodatkowych testów, wykorzystując wydzielony ze zbioru uczącego tzw. zbiór walidacyjny. Klasyfikowany obiekt przypisuje się do klasy przeważającej wśród $k$ jego najbliższych sąsiadów-obiektów ze zbioru odniesienia. Nawet jeśli w jego pobliżu znajduje się obiekt będący szumem to już przy $k > 2$ jego głos nie będzie się liczył. W [1] znajduje się dowód zbieżności klasyfikatora $k$-NN do

klasyfikatora Bayesa przy $k$ i liczebności zbioru odniesienia zbieżnych do nieskończoności i jednoczesnej zbieżności ilorazu $k$ i liczebności zbioru odniesienia do zera. Jednak o ile NN wymagał znalezienia najbliższego sąsiada klasyfikowanego obiektu, o tyle $k$-NN wymaga znalezienia k najbliższych sąsiadów, co przy dużych zbiorach odniesienia jeszcze bardziej niż w NN spowalnia prędkość klasyfikacji.

Istnieje jednak szereg metod rozwiązujących problem szybkości NN przy dużych zbiorach odniesienia. Jednym z nich jest redukcja zbioru odniesienia.

## 2.1. Redukcja zbioru odniesienia

Redukcja zbioru odniesienia polega na usunięciu ze zbioru odniesienia jak największej liczby wzorców, tak by na podstawie pozostałych, z równie dobrą (lub niewiele gorszą) jakością klasyfikacji co na pełnym zbiorze odniesienia, rozpoznawać nowe obiekty. Algorytmy redukcji to metody optymalizacji reguł typu najbliższy sąsiad. Zaleca się aby po redukcji zbioru stosować regułę NN, a nie $k$-NN dla $k > 1$, gdyż zredukowany zbiór jest silnie rozrzedzony co może powodować, że przy większych $k$ prawo głosu dostaną wzorce dość mocno oddalone od klasyfikowanego obiektu.

Pierwszy algorytm redukcji zbioru odniesienia, algorytm CNN (and. *Condensed Nearest Neighbor*) Harta [4], powstał już pod koniec lat sześćdziesiątych XX wieku. Korzysta on z tzw. kryterium zgodności, które mówi, że aby zachować równie wysoką jakość klasyfikacji co na pełnym zbiorze odniesienia, reguła NN oparta na zbiorze zredukowanym powinna poprawnie klasyfikować pełny zbiór odniesienia (mówimy wtedy, że zbiór zredukowany jest zgodny ze zbiorem odniesienia). Kryterium to przez wiele lat było uważane za najważniejsze kryterium zakończenia budowania zbioru zredukowanego. Twórcy algorytmów redukcji próbowali uzyskiwać najmniejsze z możliwych zbiorów zredukowanych spełniających kryterium zgodności lub poprawiać i udoskonalać istniejące metody redukcji. Powstały algorytmy: Tomeka [5], Gowdy i Krishny [6], Dasarathy'ego [7]. Okazało się jednak, że kryterium zgodności nie sprawdza się przy zbiorach zaszumionych. Konieczność prawidłowej klasyfikacji pełnego zbioru odniesienia w przypadku zbiorów z szumem powoduje, że szum zostaje pobrany do zbioru zredukowanego, co nie tylko powoduje mniejszą redukcję zbioru odniesienia, ale także znaczne obniżenie jakości klasyfikacji reguły NN opartej na tak zredukowanym zbiorze.

Nowsze algorytmy redukcji, wykorzystują znane heurystyki i są w stanie nie tylko bardzo silnie zredukować zbiór odniesienia ale także w przypadku niektórych zbiorów poprawić jakość klasyfikacji reguły NN (poprzez prawidłowe odrzucenie szumu). Do algorytmów tych należą m.in. algorytmy Skalaka [8], Kunchevy [9], Cerverona i Ferriego [10]. Procedury te posiadają jednak wady: są losowe, mają kilka parametrów i długie czasy trwania dla większych zbiorów

odniesienia, posiadających powyżej 10 000 obiektów (algorytm Kunchevy i Cerverona-Ferriego). Prawidłowy dobór wartości parametrów wymaga dodatkowych testów, co przy większych zbiorach odniesienia jest przyczyną dużych kosztów czasowych. Losowość powoduje, że wyniki działania tych algorytmów nie są powtarzalne, a przez to nie ma się pewności, czy przy kolejnym uruchomieniu tego samego algorytmu nie otrzymałoby się zbioru lepszego.

Celem dalszych badań w tym obszarze było zaprojektowanie i zaimplementowanie algorytmu redukcji, który nie miałby powyższych wad, a oferowałby podobnie dobre wyniki co powyżej omawiane algorytmy wykorzystujące heurystyki. Wynikiem tych badań jest zaprezentowany w tej pracy algorytm redukcji wykorzystujący tzw. miarę reprezentatywności, dokładnie opisaną w rozdziale 3. Algorytm ten nie jest losowy i można przyjąć, że nie ma parametrów. Został on przetestowany na kilku znanych w literaturze i dostępnych w Internecie zbiorach uczących i porównany z niektórymi wskazanymi w tym podrozdziale algorytmami redukcji.

## 2.2. Edycja zbioru odniesienia

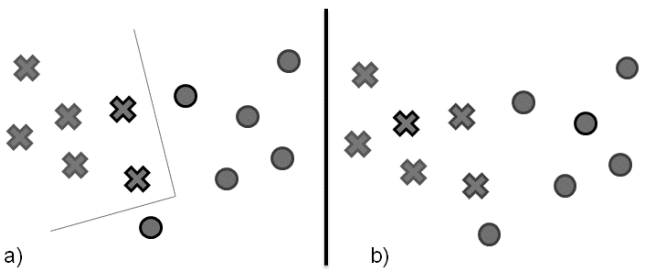
O ile głównym celem redukcji zbioru odniesienia jest odrzucenie jak największej liczby wzorców przy zachowaniu akceptowalnej jakości klasyfikacji, o tyle edycja zbioru odniesienia polega na odrzuceniu tylko tych wzorców, które stanowią szum. Edycja to odfiltrowanie zbioru z próbek niewłaściwych, czego konsekwencją ma być zwiększenie jakości klasyfikacji.

Podobnie i tutaj istnieje wiele znanych algorytmów edycji zbioru odniesienia. Wśród nich, najbardziej popularnymi są: EEN (ang. *Edited Nearest Neighbor*) Wilsona [11], RENN (ang. *Repeated EEN*) i All *k*-NN Tomeka [12], MULTIEDIT Devijvera i Kittlera [13] oraz algorytm genetyczny Kunchevy [14]. Z przeprowadzonych testów wynika jednak, że algorytmy te na rzeczywistych zbiorach danych tylko nieznacznie poprawiają jakość klasyfikacji reguły NN lub nieraz wręcz ją obniżają, odrzucając przy tym znaczną część próbek ze zbioru odniesienia.

Celem badań w tej dziedzinie było zaprojektowanie i zaimplementowanie algorytmu, który zwiększałby jakość klasyfikacji poprzez prawidłowe odfiltrowanie szumu i tylko szumu ze zbioru odniesienia. Wynikiem tych badań jest algorytm edycji wykorzystujący wynik algorytmu redukcji opartego na mierze reprezentatywności i kryterium zgodności [15]. Został on przetestowany na kilku znanych w literaturze i dostępnych w Internecie zbiorach uczących i porównany z niektórymi wskazanymi w tym podrozdziale algorytmami edycji.

## 3. MIARA REPREZENTATYWNOŚCI

Istnieją dwa różne podejścia przy redukcji zbiorów odniesienia. Pierwsze z nich zakłada, że warto zostawiać wzorce, które tworzą tzw. granice pomiędzy klasami, drugie natomiast zakłada, że warto zostawiać wzorce, które leżą w środku jednoklasowych obszarów (rys. 3).

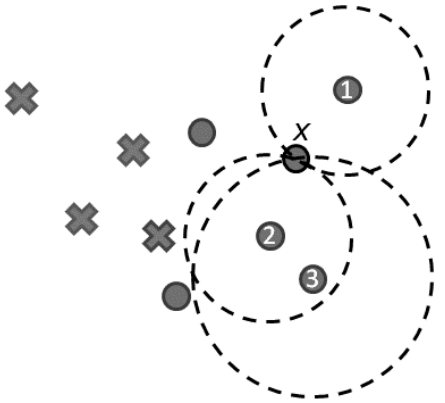



Rys. 3. Redukcja zbioru odniesienia (wybrane wzorce zaznaczone są czarnym konturem): a) wybór wzorców tworzących granice pomiędzy klasami, b) wybór wzorców leżących w środku jednoklasowych obszarów

Metoda prezentowana w tym artykule na pewno bliższa jest drugiemu z tych podejść, choć nie wyklucza włączania do zbioru również wzorców, tworzących granice pomiędzy klasami, o ile mają one wystarczająco dużą wartość miary reprezentatywności *rm* (ang. *representative measure*). Miara reprezentatywności jest próbą wyrażenia za pomocą liczby jak bardzo dany wzorzec jest reprezentatywny dla swojej klasy. Głównymi założeniami przy tworzeniu tej miary były:

- wzorzec *x*, który ma za swoich sąsiadów dużą liczbę wzorców z tej samej klasy co *x*, jest reprezentatywny, a więc powinien mieć przyporządkowaną wysoką wartość *rm*;
- wzorzec x, który jest szumem, czyli jest otoczony przez wzorce z innych klas, nie jest reprezentatywny, a więc powinien mieć przyporządkowaną niską wartość *rm*.

Przed zdefiniowaniem miary reprezentatywności zdefiniujmy pojęcie wyborcy. Wyborcą wzorca *x* będziemy nazywać każdy inny wzorzec z tej samej klasy co *x*, dla którego *x* jest bliższym sąsiadem niż najbliższy sąsiad z klasy przeciwnej. Miarą reprezentatywności *rm* wzorca *x* jest liczba jego wyborców (rys. 4).




Rys. 4. Miara reprezentatywności wzorca *x* jest równa 3. Wyborcy wzorca *x* zaznaczeni są kolejnymi cyframi

Do zbioru zredukowanego powinny trafiać w pierwszej kolejności wzorce najbardziej reprezentatywne dla swoich klas, czyli wzorce o wysokich wartościach miary reprezentatywności. Prezentowany w tym artykule algorytm redukcji dokonuje właśnie takiego wyboru.

## 4. ALGORYTM REDUKCJI ZBIORU ODNIESIENIA WYKORZYSTUJĄCY MIARĘ REPREZENTATYWNOŚCI

Algorytm redukcji zbioru odniesienia wykorzystujący miarę reprezentatywności (ang. *Reduction Algorithm based on Representative Measure* – RARM) wybiera do zbioru zredukowanego wzorce o aktualnie największej wartości miary reprezentatywności. Może się zdarzyć, że wzorców o największej wartości *rm* jest kilka. Wtedy wybierany jest z nich wzorzec o największej wartości tzw. priorytetu [16]. Wartość priorytetu *prior* liczona jest ze wzoru:

$$prior(x) = \sum_{j} \frac{1}{d(x, x_j)^2}, \tag{1}$$

gdzie $x_j$ jest wyborcą wzorca $x$, zaś $d(x,x_j)$ jest odległością (w prezentowanym algorytmie liczoną z użyciem metryki euklidesowej) pomiędzy wzorcami $x$ i $x_j$. Duże wartości priorytetu oznaczają, że wzorzec w swoim bliskim otoczeniu ma swoich wyborców, małe wartości natomiast, że od swoich wyborców dzieli go większa odległość (z większym prawdopodobieństwem jest wtedy wzorcem nietypowym lub szumem).

Po każdym dodaniu wzorca o największej mierze reprezentatywności do zbioru zredukowanego, miary reprezentatywności pozostałych w zbiorze

odniesienia wzorców są przeliczane, z tym że wyborcy dodanego wzorca nie mogą być już wyborcami innych wzorców, jak i dla nich nie są już liczone miary *rm*. W momencie gdy aktualnie największa wartość *rm* wzorców w zbiorze odniesienia jest mniejsza bądź równa wcześniej ustalonej tzw. minimalnej mierze reprezentatywności $rm_{min}$, algorytm jest przerywany a wybrane do tej pory wzorce tworzą wynikowy zbiór zredukowany. Czym większe wartości $rm_{min}$, tym zbiory zredukowane są mniejsze (warunek zakończenia algorytmu spełniony jest wcześniej).

Działanie algorytmu może być kontynuowane po zwróceniu zbioru zredukowanego dla wartości $rm_{min} > 0$. Kolejną wartością $rm_{min}$ jest wtedy poprzednia wartość $rm_{min}$ pomniejszona o jeden. Procedura ta może trwać aż osiągnie się $rm_{min} = 0$ i algorytm zwróci największy zbiór zredukowany. Ta właściwość algorytmu została wykorzystana w ten sposób, że ustala się tylko początkową wartość $rm_{min}$ nazywaną wartością inicjalizującą $rm_{init}$, a algorytm zwraca ciąg wstępujących zbiorów zredukowanych dla $rm_{min}$ równych kolejno: $rm_{init}$, $rm_{init} - 1$, …, 0. Przykładowo, dla $rm_{init} = 9$, RARM zwróci 10 zbiorów zredukowanych dla $rm_{min}$ równych odpowiednio: 9, 8, …, 0. Z tak otrzymanych zbiorów zredukowanych użytkownik może wybrać zbiór, który mu najbardziej odpowiada czy to pod kątem liczebności, czy też uzyskanej jakości klasyfikacji. W celu estymacji jakości klasyfikacji można posłużyć się popularną metodą walidacji krzyżowej (metoda *k-fold cross validation* w [3]). Jest to metoda stosowana przy walidacji wartości parametrów budowanego klasyfikatora (w naszym przypadku wyboru zbioru zredukowanego, na podstawie którego będzie działała reguła NN), w odróżnieniu od metody zbioru testującego, która jest stosowana na koniec, przy ocenie zbudowanego już klasyfikatora.

RARM można opisać w następujących krokach ($rm_{min} = rm_{init}$; zbiór zredukowany jest wstępnie pusty):

1. Usuń dublujące się wzorce ze zbioru odniesienia.
2. Oznacz wszystkie wzorce w zbiorze odniesienia jako “nie zawarte w zbiorze zredukowanym” i “dostępne”.
3. Policz miarę reprezentatywności dla wszystkich wzorców oznaczonych jako “dostępne” i “nie zawarte w zbiorze zredukowanym”. Wzorce oznaczone jako „niedostępne” nie mogą być wyborcami.
4. Jeśli największa miara reprezentatywności jest mniejsza lub równa $rm_{min}$, to zwróć zbiór zredukowany (tworzą go wszystkie wzorce oznaczone jako “zawarte w zredukowanym zbiorze), w przeciwnym przypadku przejdź do kroku 6.
5. Jeśli $rm_{min} = 0$, to zakończ algorytm; w przeciwnym przypadku zmniejsz wartość $rm_{min}$ o 1 i jeszcze raz wykonaj krok 4.
6. Oznacz wzorzec $x$ o największej mierze reprezentatywności jako “zawarty w zbiorze zredukowanym” (jeśli jest kilka wzorców o największej wartości *rm*, wybierz z nich wzorzec o największej wartości

priorytetu). Wszystkich wyborców wzorca $x$ oznacz jako „niedostępnych".

7. Jeśli w zbiorze odniesienia mamy co najmniej jeden wzorzec, który jest "nie zawarty w zbiorze zredukowanym" i "dostępny", przejdź do kroku 3.
8. Zwróć zbiór zredukowany złożony z wzorców oznaczonych jako "zawarte w zbiorze zredukowanym".
9. Jeśli $rm_{min} > 0$ to dla pozostałych wartości $rm_{min}$: $rm_{min} - 1$, $rm_{min} - 2$, …, 0 zwróć ten sam zbiór zredukowany co w punkcie 8.

## 5. ALGORYTM EDYCJI ZBIORU ODNIESIENIA WYKORZYSTUJĄCY MIARĘ REPREZENTATYWNOŚCI I KRYTERIUM ZGODNOŚCI

Zbiór zredukowany składa się ze wzorców reprezentatywnych dla swoich klas. Jeśli dołączymy do niego te wzorce ze zbioru odniesienia, które są prawidłowo klasyfikowane przez regułę NN opartą na tym zbiorze zredukowanym, to na pewno uzyskamy zbiór o „wzmocnionej" informacji o rozkładzie rozważanych klas. Szum, jako wzorce o nieprawidłowej klasie, zostanie w ten sposób odfiltrowany.

Opisany powyżej sposób wyboru wzorców można zawrzeć w następującym zdaniu: ze zbioru odniesienia należy wydzielić podzbiór, z którym zgodny (patrz rozdział 2) będzie zbiór zredukowany. Podzbiór ten będzie stanowił wynikowy zbiór edytowany.

Algorytm edycji zbioru odniesienia, wykorzystujący kryterium zgodności (ang. *Editing Algorithm based on Consistency* - EAC) można opisać w następujących krokach (oznaczmy jako $X_{ed}$ budowany zbiór edytowany, $X_o$ jako zbiór odniesienia, zaś jako $X_{red}$ uzyskany wcześniej algorytmem RARM zbiór zredukowany):

1. Niech $X_{ed} = X_{red}$.
2. Dla każdego wzorca $x$ z $X_o \setminus X_{red}$ sprawdź czy jest on poprawnie klasyfikowany przez $X_{red}$ z użyciem reguły NN. Jeśli tak, to dołącz $x$ do $X_{ed}$.
3. Zbiór $X_{ed}$ jest wynikowym zbiorem edytowanym ($X_{red}$ jest zgodny z $X_{ed}$).

Oczywiście nic nie stoi na przeszkodzie, aby EAC wykorzystywał zbiór zredukowany uzyskany za pomocą innego niż RARM algorytmu redukcji.

## 6. TESTY

Wszystkie testy wykonano na ośmiu znanych w literaturze i dostępnych w Internecie [17] zbiorach danych: Liver Disorders (BUPA), GLASS Identification, IRIS, PARKINSONS Disease Data Set, PIMA Indians Diabetes, WAVEFORM (version1), Wisconsin Diagnostic Breast Cancer (WDBC) (Diagnostic) oraz

YEAST. Liczby klas, cech i wzorców powyższych zbiorów danych są podane w tabeli 1.

Tabela 1. Testowe zbiory danych

| zbiór danych | liczba klas | liczba cech | liczba wzorców |
|---|---|---|---|
| BUPA | 2 | 6 | 345 |
| GLASS | 6 | 9 | 214 |
| IRIS | 3 | 4 | 150 |
| PARKINSONS | 2 | 22 | 195 |
| PIMA | 2 | 8 | 768 |
| WAVEFORM | 3 | 21 | 5000 |
| WDBC | 2 | 30 | 569 |
| YEAST | 10 | 8 | 1484 |

Wszystkie algorytmy zaimplementowano w języku Java. Testy przeprowadzono na komputerze Pentium Dual-Core CPU T4200 @ 2.00 Ghz z 4 GB pamięci RAM.

Do oceny jakości klasyfikacji (patrz podrozdział 1.1) w każdym przypadku użyto 10-krotnej walidacji krzyżowej (*10-fold cross validation*).

Stopniem redukcji (podanym w procentach) nazwano frakcję odrzuconych wzorców ze zbiorów odniesienia.

## 6.1. Testy algorytmów redukcji zbioru odniesienia

Testom zostały poddane następujące algorytmy redukcji zbioru odniesienia:

- algorytm CNN Harta;
- algorytm Gowdy i Krishny (GK);
- algorytm Dasarathy'ego (MCS – ang. *Minimal Consistent Set*);
- algorytm RMHC-P (ang. *Random Mutation Hill Climbing*) Skalaka – liczba iteracji [8] dobrana eksperymentalnie (200 dla IRIS, 300 dla GLASS i PARKINSONS, 400 dla BUPA, 600 dla WDBC, 700 dla PIMA, 1000 dla YEAST, 3000 dla WAVEFORM), liczebność zbiorów wynikowych taka sama jak liczebność zbiorów otrzymanych przez RARM (dla lepszego porównania uzyskiwanej jakości klasyfikacji), liczba najbliższych sąsiadów ustawiona na 1;
- algorytm genetyczny Kunchevy (GA) – zgodnie z implementacją podaną w [9]: liczba najbliższych sąsiadów ustawiona na 1, liczba iteracji na 200, poziom redukcji na 0.1, liczba chromosomów na 20,

współczynnik krzyżowania na 0.5, współczynnik mutacji na 0.025; w funkcji przystosowania (wzór (8) z pracy [19]) współczynnik α ustawiony na 0.01;

- algorytm *tabu search* Cerverona i Ferriego (TS) – funkcja celu i wartości parametrów ustawione na te zaproponowane w pracy [10] z jednym wyjątkiem: metoda inicjalizacji zgodnego rozwiązania początkowego typu *condensed* została zastosowana w przypadku zbioru WAVEFORM ze względu na bardzo długi czas generowania rozwiązania początkowego metodą typu *constructive* [10];
- prezentowany w tym artykule algorytm RARM – wartość parametru $rm_{init}$ ustawiona na 9; wybierany zbiór oferujący najwyższą jakość klasyfikacji (ocena z wykorzystaniem 10-krotnej walidacji krzyżowej).

Wyniki testów zostały przedstawione w tabelach 2a i 2b oraz 3a i 3b (wyniki reguły NN działającej na pełnym zbiorze odniesienia podane w wierszu NN). W tabeli 4 pokazano, który zbiór zredukowany (o jakim rmmin) został wybrany jako najlepszy w algorytmie RARM.

Tabela 2a. Wyniki (stopnie redukcji) algorytmów redukcji: dla każdego zbioru testowego stopień redukcji (w procentach) podany w pierwszej kolumnie i odchylenie standardowe stopnia redukcji podane w drugiej kolumnie. Najlepsze wyniki w ramach zbiorów testowych zostały pogrubione

| | BUPA | | GLASS | | IRIS | | PARKINSONS | |
|---|---|---|---|---|---|---|---|---|
| CNN | 40,97 | 2,16 | 51,87 | 1,34 | 87,7 | 1,12 | 66,44 | 1,47 |
| GK | 43,51 | 1,85 | 55,76 | 1,4 | 88,52 | 1,36 | 71,79 | 1,57 |
| MCS | 49,5 | 2 | 60,69 | 1,31 | 89,93 | 1,32 | 74,59 | 1,21 |
| RMHC-P | 92,72 | 0,79 | 79,5 | 1,42 | 95,7 | 0,58 | 89,46 | 0,74 |
| GA | **94,62** | **0,79** | **92,36** | **0,8** | **96,89** | **0,98** | **98,23** | **0,57** |
| TS | 87,57 | 1,21 | 77,05 | 2,26 | 93,7 | 1,12 | 82,79 | 2,67 |
| RARM | 92,72 | 0,79 | 79,5 | 1,42 | 95,7 | 0,58 | 89,46 | 0,74 |

Tabela 2b. Wyniki (stopnie redukcji) algorytmów redukcji: dla każdego zbioru testowego stopień redukcji (w procentach) podany w pierwszej kolumnie i odchylenie standardowe stopnia redukcji podane w drugiej kolumnie. Najlepsze wyniki w ramach zbiorów testowych zostały pogrubione

| | PIMA | | WAVEFORM | | WDBC | | YEAST | |
|---|---|---|---|---|---|---|---|---|
| CNN | 46,66 | 1,27 | 61,34 | 0,41 | 82,97 | 0,91 | 33,54 | 0,61 |
| GK | 53,3 | 1,17 | 65,59 | 0,48 | 85,98 | 0,53 | 37,2 | 0,6 |
| MCS | 57,47 | 0,59 | 71,8 | 0,44 | 86,27 | 0,65 | 40,9 | 0,64 |
| RMHC-P | **96,59** | **0,26** | 98,49 | 0,04 | 95,66 | 0,34 | 95,55 | 0,21 |
| GA | 92,51 | 0,33 | 90,92 | 0,17 | 94,06 | 0,41 | 91,55 | 0,49 |
| TS | 94,21 | 0,73 | **99,55** | **0,03** | **98,52** | **0,23** | **96,89** | **0,36** |
| RARM | **96,59** | **0,26** | 98,49 | 0,04 | 95,66 | 0,34 | 95,55 | 0,21 |

Tabela 3a. Wyniki (jakości klasyfikacji) algorytmów redukcji: dla każdego zbioru testowego jakość klasyfikacji (w procentach) podana w pierwszej kolumnie i odchylenie standardowe jakości klasyfikacji podane w drugiej kolumnie. Najlepsze wyniki w ramach zbiorów testowych zostały pogrubione

| | BUPA | | GLASS | | IRIS | | PARKINSONS | |
|---|---|---|---|---|---|---|---|---|
| NN | 62,61 | 7,3 | **71,56** | **9,86** | 96 | 4,66 | 84,49 | 6,49 |
| CNN | 60,01 | 8,62 | 69,36 | 9,67 | 93,33 | 5,44 | 83,01 | 7,51 |
| GK | 55,92 | 10,91 | 66,14 | 9,08 | 94 | 5,84 | 84,99 | 7,72 |
| MCS | 55,05 | 12,4 | 65,44 | 9,1 | 92,67 | 7,98 | 83,99 | 8,51 |
| RMHC-P | 64,31 | 5,22 | 67,69 | 5,94 | 94 | 8,58 | 81,82 | 9,66 |
| GA | 66,62 | 9,36 | 67,4 | 8,57 | 96 | 5,62 | 80,99 | 6,97 |
| TS | 68,13 | 6,49 | 68,83 | 8,91 | **96,67** | **4,71** | 81,35 | 7,17 |
| RARM | **68,4** | **8,66** | 66,89 | 6,45 | 96 | 4,66 | **85,51** | **7,29** |

Tabela 3b. Wyniki (jakości klasyfikacji) algorytmów redukcji: dla każdego zbioru testowego jakość klasyfikacji (w procentach) podana w pierwszej kolumnie i odchylenie standardowe jakości klasyfikacji podane w drugiej kolumnie. Najlepsze wyniki w ramach zbiorów testowych zostały pogrubione

| | PIMA | | WAVEFORM | | WDBC | | YEAST | |
|---|---|---|---|---|---|---|---|---|
| NN | 67,2 | 3,99 | 77,4 | 2,13 | 91,19 | 4,16 | 53,04 | 4,7 |
| CNN | 63,95 | 4,68 | 74,4 | 2,47 | 90,84 | 5,47 | 50,9 | 5,75 |
| GK | 63,69 | 5,03 | 73,7 | 1,9 | 91,55 | 3,15 | 50,29 | 5,3 |
| MCS | 64,73 | 6,13 | 74,66 | 2,21 | 91,36 | 5,5 | 49,13 | 6,09 |
| RMHC-P | 73,18 | 4,25 | 82,7 | 1,68 | 93,12 | 4,11 | 56,67 | 5,53 |
| GA | 68,5 | 3,82 | 77,42 | 2,51 | 91,03 | 4,63 | 48,27 | 3,72 |
| TS | 69,54 | 3,09 | **82,76** | **1,75** | 93,14 | 2,47 | **57,36** | **3,44** |
| RARM | **74,23** | **4,67** | 81,04 | 1,84 | **93,84** | **2,57** | 57,08 | 4,3 |

Tabela 4. Zestawienie wartości $rm_{min}$ wybranych zbiorów zredukowanych w algorytmie RARM (wybór pod kątem najwyższej jakości klasyfikacji)

| | $rm_{min}$ |
|---|---|
| BUPA | 2 |
| GLASS | 0 |
| IRIS | 2 |
| PARKINSONS | 1 |
| PIMA | 4 |
| WAVEFORM | 8 |
| WDBC | 1 |
| YEAST | 2 |

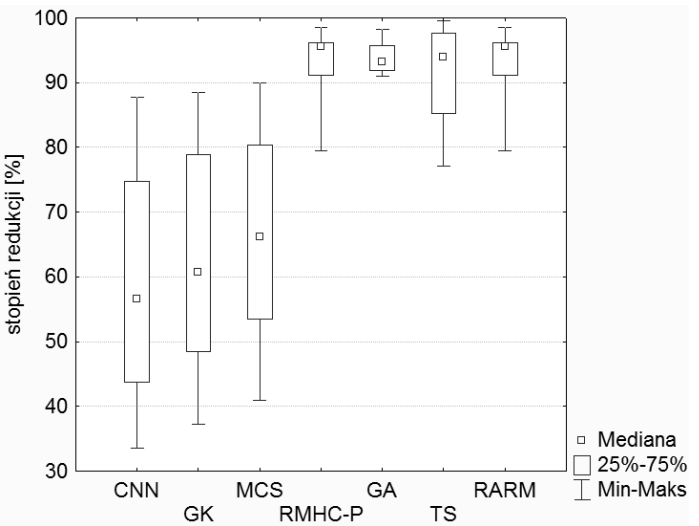
## 6.2. Omówienie wyników testów algorytmów redukcji zbioru odniesienia

Na rysunkach 5 i 6 przedstawiono wykresy pudełkowe wyników redukcji zbiorów danych, prezentujące odpowiednio: uzyskane stopnie redukcji i jakości klasyfikacji.

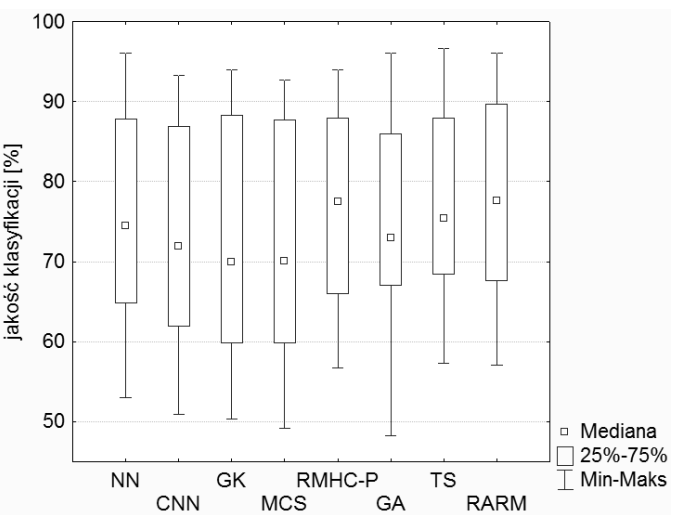
Nowsze algorytmy (RMHC-P, GA, TS i RARM) oferują znacznie silniejszą redukcję (średnio powyżej 90%) niż algorytmy starsze (CNN, GK, MCS), wykorzystujące kryterium zgodności (średnio około 60%) (rys. 5). Potwierdza to

omówione w podrozdziale 2.1 ograniczenia zbiorów zredukowanych zgodnych z pełnymi zbiorami odniesienia.

Jednocześnie (rys. 6) zbiory silniej zredukowane cechują się minimalnie lepszymi jakościami klasyfikacji reguły NN niż zbiory zgodne, co może być konsekwencją pozostawienia szumu w zbiorach zredukowanych spełniających kryterium zgodności (patrz podrozdział 2.1).




Rys. 5. Wykres pudełkowy stopni redukcji zredukowanych zbiorów danych




Rys. 6. Wykres pudełkowy jakości klasyfikacji zredukowanych zbiorów danych

Wydaje się, że z klasy testowanych algorytmów redukcji wykorzystujących kryterium zgodności najkorzystniej jest wybrać algorytm MCS Dasarathy'ego – uzyskał on w tej klasie najwyższe stopnie redukcji (tabele 2a i 2b). Jednocześnie wszystkie trzy algorytmy pod kątem jakości klasyfikacji reguły NN działającej na wynikowych zbiorach zredukowanych prawie się nie różnią (tabele 3a i 3b). MCS Dasarathy'ego jest jednak bardziej skomplikowany w implementacji niż algorytmy CNN Harta czy Gowdy i Krishny.

W algorytmach GA i RARM widoczna jest stabilność w uzyskanych stopniach redukcji (rys. 5) – ponad połowa zbiorów została zredukowana do mniej jak 1/10 początkowej liczebności, przy czym GA zredukował aż cztery zbiory (BUPA, GLASS, IRIS i PARKINSONS) najsilniej ze wszystkich algorytmów. RMHC-P jest sparametryzowany pod kątem wynikowej liczebności zbioru zredukowanego, więc nie można go w ten sposób porównywać. Trzy zbiory (WAVEFORM, WDBC i YEAST) zostały najsilniej zredukowane przez TS, jednak aż dwa zbiory (GLASS i PARKINSONS) zredukował on tylko do około 80% - tabele 2a i 2b.

Jakości klasyfikacji reguły NN działającej na zbiorach zredukowanych nowszymi algorytmami redukcji są minimalnie lepsze od jakości klasyfikacji na pełnym zbiorze odniesienia. Może być to konsekwencją silnej redukcji, która odrzuciła ze zbioru szum i wzorce nietypowe.

Algorytmy sparametryzowane, tj. GA, TS i RMHC-P mogą dawać lepsze jak i gorsze wyniki dla innych ustawień parametrów. Wyniki przez nie generowane nie są powtarzalne, co jeszcze bardziej utrudnia dobór odpowiednich wartości parametrów dla redukowanego zbioru.

Działanie algorytmów GA i TS na zbiorach większych, mających ok. 5000 elementów (takich jak WAVEFORM) zajmuje kilka godzin. Czasy działania reszty algorytmów są akceptowalne, ale oczywiście dla zbiorów danych powyżej 10000 elementów powinny być już stosowane metody podziału zbioru na mniejsze podzbiory w celu przyspieszenia redukcji.

Prezentowany algorytm RARM wypada bardzo dobrze na tle znanych algorytmów redukcji (silna redukcja i minimalna poprawa jakości klasyfikacji w stosunku do reguły NN działającej na pełnym zbiorze odniesienia). Jego jedyny parametr $rm_{init}$ może być na stałe ustawiony na wartość 9, gdyż testy pokazały, że zbiory zredukowane o najlepszej jakości klasyfikacji były zazwyczaj uzyskiwane dla małych wartości $rm_{min}$. Jednak w przypadku gdyby najlepszym zbiorem okazał się zbiór wygenerowany dla $rm_{min}$ równego 9, warto sprawdzić zbiory zredukowanego dla większych wartości $rm_{min}$.

Wymagana w RARM faza walidacji zbiorów wynikowych wydłuża jego czas działania.

## 6.3. Testy algorytmów edycji zbioru odniesienia

Testom zostały poddane następujące algorytmy edycji zbioru odniesienia:

- algorytm ENN Wilsona – liczba najbliższych sąsiadów [11] ustawiona na 1 (ENN1) i 3 (ENN3);
- algorytm RENN Tomeka – liczba najbliższych sąsiadów [12] podobnie jak w przypadku ENN ustawiona na 1 (RENN1) i 3 (RENN3);
- algorytm All *k*-NN Tomeka – wartość parametru *k* [12] ustawiona na 3 (All3NN);
- algorytm MULTIEDIT Devijvera i Kittlera – dla prawie wszystkich zbiorów wartości parametrów $q$ i $I$ [13] ustawiono na odpowiednio 3 i 5; jedynie dla zbioru WAVEFORM (większy zbiór) $q = 9$ i $I = 10$;
- algorytm genetyczny Kunchevy (GA1) - ze względu na to, że algorytm GA1 ma tworzyć edytowane zbiory, przyjęto zaproponowaną w [14] funkcję przystosowania; dla wszystkich zbiorów, zgodnie z [14], współczynnik redukcji i mutacji są odpowiednio równe: 0.8 i 0.05; liczba chromosomów i liczba iteracji [14] są odpowiednio równe: 50 i 200 dla prawie wszystkich zbiorów, z wyłączeniem zbioru WAVEFORM (większy zbiór), dla którego wartości tych parametrów zostały pomniejszone do odpowiednio 10 i 100, ze względu na bardzo długą fazę uczenia algorytmu dla wartości 50 i 200; algorytm przetestowano dla liczby najbliższych sąsiadów równej 1;
- prezentowany w tym artykule algorytm EAC – buduje zbiory edytowane na podstawie najlepszych wyników redukcji algorytmu RARM (patrz tabela 4).

Wyniki testów zostały przedstawione w tabelach 5a i 5b (wyniki reguły NN działającej na pełnym zbiorze odniesienia podane w wierszu NN) oraz 6a i 6b. Ponieważ algorytmy edycji powinny odfiltrowywać tylko szum, pozostawiając pozostałe wzorce w zbiorze odniesienia, w tabelach 6a i 6b jako najlepsze wyniki zostały pogrubione najmniejsze stopnie redukcji edytowanych zbiorów.

Tabela 5a. Wyniki (jakości klasyfikacji) algorytmów edycji: dla każdego zbioru testowego jakość klasyfikacji (w procentach) podana w pierwszej kolumnie i odchylenie standardowe jakości klasyfikacji podane w drugiej kolumnie. Najlepsze wyniki w ramach zbiorów testowych zostały pogrubione

| | BUPA | | GLASS | | IRIS | | PARKINSONS | |
|---|---|---|---|---|---|---|---|---|
| NN | 62.61 | 7.30 | **71.56** | **9.86** | 96.00 | 4.66 | 84.49 | 6.49 |
| ENN1 | 63.73 | 5.79 | 70.15 | 6.72 | **96.67** | **4.71** | 84.04 | 4.09 |
| ENN3 | 63.74 | 7.02 | 69.30 | 8.73 | **96.67** | **4.71** | 84.04 | 4.09 |
| RENN1 | 65.18 | 5.89 | 66.35 | 8.13 | **96.67** | **4.71** | 84.07 | 3.94 |
| RENN3 | 64.03 | 8.01 | 66.88 | 9.50 | **96.67** | **4.71** | 84.57 | 4.43 |
| All3NN | 64.32 | 6.86 | 69.22 | 7.99 | **96.67** | **4.71** | 83.54 | 4.26 |
| MULTIEDIT | 58.87 | 8.98 | 55.87 | 7.15 | 96.00 | 4.66 | 81.07 | 5.24 |
| GA1 | 63.49 | 8.63 | 70.40 | 11.95 | 95.33 | 4.50 | 84.51 | 5.76 |
| EAC | **68.66** | **7.71** | 70.05 | 6.81 | 96.00 | 4.66 | **86.56** | **6.05** |

Tabela 5b. Wyniki (jakości klasyfikacji) algorytmów edycji: dla każdego zbioru testowego jakość klasyfikacji (w procentach) podana w pierwszej kolumnie i odchylenie standardowe jakości klasyfikacji podane w drugiej kolumnie. Najlepsze wyniki w ramach zbiorów testowych zostały pogrubione

| | PIMA | | WAVEFORM | | WDBC | | YEAST | |
|---|---|---|---|---|---|---|---|---|
| NN | 67.20 | 3.99 | 77.40 | 2.13 | 91.19 | 4.16 | 53.04 | 4.70 |
| ENN1 | 68.49 | 3.83 | 79.92 | 2.31 | 92.25 | 2.14 | 56.06 | 3.96 |
| ENN3 | 71.37 | 4.72 | 80.18 | 2.55 | 92.61 | 1.68 | 56.11 | 5.44 |
| RENN1 | 68.10 | 4.17 | 80.38 | 2.22 | 92.77 | 2.35 | 56.80 | 4.34 |
| RENN3 | 72.67 | 4.04 | 80.70 | 2.46 | **93.49** | **1.72** | 56.38 | 4.78 |
| All3NN | 71.36 | 4.01 | 80.04 | 2.28 | 92.61 | 1.68 | 56.59 | 4.79 |
| MULTIEDIT | 70.71 | 5.19 | 80.22 | 1.71 | 90.50 | 3.78 | 52.91 | 4.20 |
| GA1 | 69.02 | 3.91 | 78.62 | 1.90 | 91.55 | 2.40 | 53.22 | 4.04 |
| EAC | **73.97** | **4.60** | **81.14** | **2.26** | 93.13 | 2.00 | **57.68** | **3.59** |

Tabela 6a. Wyniki (stopnie redukcji) algorytmów edycji: dla każdego zbioru testowego stopień redukcji (w procentach) podany w pierwszej kolumnie i odchylenie standardowe stopnia redukcji podane w drugiej kolumnie. Najlepsze wyniki (najniższe stopnie redukcji) w ramach zbiorów testowych zostały pogrubione

| | BUPA | | GLASS | | IRIS | | PARKINSONS | |
|---|---|---|---|---|---|---|---|---|
| ENN1 | 38.07 | 1.47 | 27.31 | 1.07 | 4.15 | 0.72 | 15.62 | 1.29 |
| ENN3 | 38.62 | 2.00 | 30.11 | 1.00 | 4.15 | 0.62 | 16.41 | 0.80 |
| RENN1 | 41.35 | 1.58 | 31.00 | 1.51 | 4.15 | 0.72 | 16.64 | 1.16 |
| RENN3 | 43.74 | 2.38 | 34.47 | 1.84 | 4.15 | 0.62 | 17.55 | 1.17 |
| All3NN | 41.84 | 1.96 | 32.04 | 1.22 | 4.22 | 0.70 | 17.27 | 0.95 |
| MULTIEDIT | 75.17 | 5.37 | 60.24 | 3.76 | 13.33 | 2.49 | 32.14 | 4.24 |
| GA1 | 48.80 | 3.72 | 48.24 | 3.27 | 52.59 | 5.11 | 46.72 | 2.99 |
| EAC | **28.92** | **1.76** | **16.46** | **1.90** | **3.48** | **0.76** | **10.37** | **1.20** |

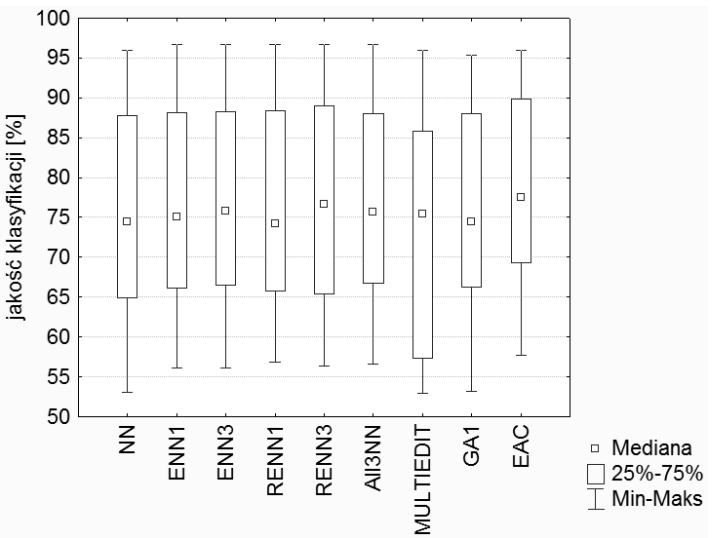
Tabela 6b. Wyniki (stopnie redukcji) algorytmów edycji: dla każdego zbioru testowego stopień redukcji (w procentach) podany w pierwszej kolumnie i odchylenie standardowe stopnia redukcji podane w drugiej kolumnie. Najlepsze wyniki (najniższe stopnie redukcji) w ramach zbiorów testowych zostały pogrubione

| | PIMA | | WAVEFORM | | WDBC | | YEAST | |
|---|---|---|---|---|---|---|---|---|
| ENN1 | 32.16 | 1.05 | 21.90 | 0.34 | 8.36 | 0.52 | 47.87 | 0.84 |
| ENN3 | 30.02 | 0.99 | 20.60 | 0.36 | 8.05 | 0.48 | 46.53 | 0.88 |
| RENN1 | 35.26 | 1.04 | 24.91 | 0.30 | 8.77 | 0.58 | 51.49 | 0.84 |
| RENN3 | 33.75 | 1.25 | 23.66 | 0.40 | 8.20 | 0.41 | 52.31 | 0.93 |
| All3NN | 36.37 | 1.11 | 23.98 | 0.36 | 9.26 | 0.50 | 50.58 | 1.07 |
| MULTIEDIT | 56.12 | 1.46 | 57.82 | 0.86 | 13.85 | 0.47 | 76.69 | 2.13 |
| GA1 | 49.06 | 1.75 | 46.94 | 1.31 | 48.76 | 1.65 | 48.68 | 1.95 |
| EAC | **25.43** | **0.90** | **18.47** | **0.46** | **6.05** | **0.21** | **40.87** | **1.27** |

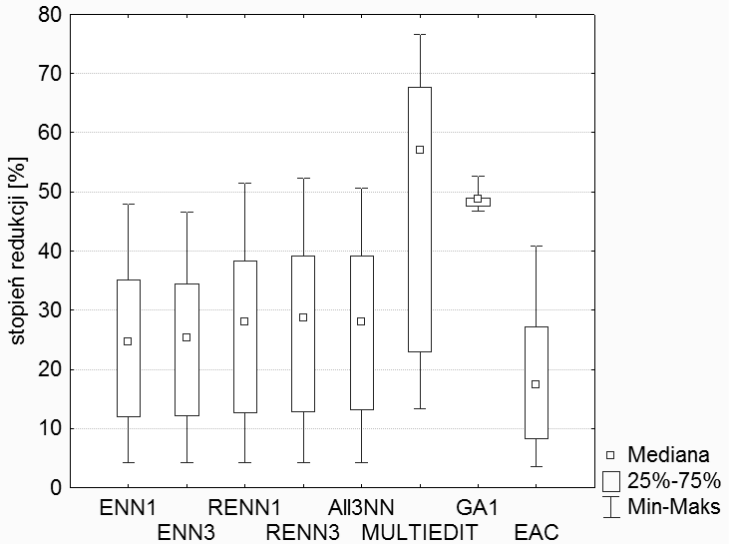
## 6.4. Omówienie wyników testów algorytmów edycji zbioru odniesienia

Na rysunkach 7 i 8 przedstawiono wykresy pudełkowe wyników edycji zbiorów danych, prezentujące odpowiednio: uzyskane jakości klasyfikacji i stopnie redukcji.

Algorytmy edycji tylko nieznacznie poprawiły jakość klasyfikacji reguły NN (rys. 7). Algorytm MULTIEDIT spowodował wręcz obniżenie jakości klasyfikacji. Algorytmem najlepszym pod kątem uzyskanej jakości wydaje się być zaproponowany algorytm EAC. Zwrócił on też największe zbiory edytowane (najmniejsze stopnie redukcji zbiorów) – patrz rys. 8 i tabele 6a i 6b. Można przypuszczać, że prawidłowo odfiltrował on szum, powodując wzrost jakości klasyfikacji reguły NN i zostawiając w zbiorze odniesienia wysoki odsetek wzorców prawidłowych.




Rys. 7. Wykres pudełkowy jakości klasyfikacji edytowanych zbiorów danych




Rys. 8. Wykres pudełkowy stopni redukcji edytowanych zbiorów danych

Na uwagę zasługuje algorytm GA1, który dość silnie jak na algorytm edycji i stabilnie (wszystkie wyniki w okolicach 50%) zredukował wynikowe zbiory edytowane (rys. 8 i tabele 6a i 6b). Możliwe, że potrzebna by była inna funkcja przystosowania, który w mniejszym stopniu redukowałaby zbiór wynikowy. Podobnie wysokich (jak na algorytm edycji) redukcji dokonał algorytm MULTIEDIT.

Tak jak w przypadku algorytmów redukcji, możliwe jest, że dla innych wartości parametrów algorytmy MULTIEDIT i GA1 będą zwracać lepsze jak i gorsze (pod względem oferowanej jakości klasyfikacji) zbiory edytowane. MULTIEDIT i GA1 są losowe, a więc nie zwracają powtarzalnych wyników, co utrudnia ustalenie prawidłowych wartości parametrów dla danego zbioru odniesienia.

Czasy działania wszystkich algorytmów, z wyjątkiem GA1, są znikome. GA1 edytował zbiór WAVEFORM i YEAST kilka godzin.

Należy wziąć pod uwagę, że do czasu działania algorytmu EAC należy doliczyć czas tworzenia zbioru zredukowanego.

## 7. PODSUMOWANIE

Skuteczną metodą przyspieszenia działania reguły NN na dużych zbiorach danych jest redukcja zbioru odniesienia. Nowe algorytmy redukcji są w stanie odrzucić ponad 90% wzorców ze zbioru odniesienia, jednocześnie powodując

minimalny wzrost jakości klasyfikacji reguły NN działającej na tak zredukowanym zbiorze.

Zaprezentowany algorytm RARM wybiera do zbioru zredukowanego wzorce o aktualnie najwyższej wartości miary reprezentatywności. Do jego cech należą:

- silna redukcja zbioru odniesienia, porównywalna z wynikami nowszych algorytmów redukcji zbioru odniesienia, opartych na znanych heurystykach;
- utrzymanie lub wręcz podwyższenie jakości klasyfikacji reguły NN;
- brak parametrów (jedyny parametr $rm_{init}$ można na stałe ustawić na wartość 9);
- brak losowości (powtarzalne wyniki);
- wymagana jest faza walidacji w celu wyboru najlepszego zbioru (z dziesięciu generowanych zbiorów).

Algorytmy edycji, w odróżnieniu od algorytmów redukcji, nie mają na celu maksymalnej redukcji zbioru odniesienia, ale takie odfiltrowanie szumu ze zbioru, aby spowodować wzrost jakości klasyfikacji reguły NN.

Zaprezentowany algorytm EAC wykorzystuje zbiór zredukowany, by na jego podstawie zbudować zbiór edytowany. Ze zbioru odniesienia wybiera on punkty, które są prawidłowo klasyfikowane przez zbiór zredukowany z wykorzystaniem reguły NN. Zbiór zredukowany jest zgodny z tak zbudowanym na jego podstawie zbiorem edytowanym (w sensie spełnienia kryterium zgodności).

W niniejszym artykule algorytm EAC wykorzystywał zbiory zredukowane algorytmem RARM, ale nic nie stoi na przeszkodzi by skorzystać ze zbiorów wynikowych innych algorytmów redukcji.

Do cech algorytmu edycji EAC należą:

- widoczne poprawienie jakości klasyfikacji reguły NN na większości zbiorów;
- niska redukcja zbioru odniesienia;
- prosta implementacja;
- wymagana wcześniejsza redukcja zbioru odniesienia wybranym algorytmem redukcji;
- dla ustalonego zbioru zredukowanego zwracanie powtarzalnych wyników;
- jedyne parametry to parametry (o ile istnieją) algorytmu redukcji, który zwraca zbiór zredukowany.

## PODZIĘKOWANIA

Autor publikacji jest stypendystą projektu „Innowacyjna dydaktyka bez ograniczeń – zintegrowany rozwój Politechniki Łódzkiej - zarządzanie uczelnią, nowoczesna oferta edukacyjna i wzmacnianie zdolności do zatrudniania, także osób niepełnosprawnych" współfinansowanego przez Unię Europejską w ramach Europejskiego Funduszu Społecznego.

## LITERATURA

[1] **Duda R.O., Hart P.E., Stork D.G.:** Pattern Classification – Second Edition. John Wiley & Sons, Inc., 2001.
[2] **Theodoridis S., Koutroumbas K.:** Pattern Recognition – Third Edition. Academic Press - Elsevier, USA, 2006.
[3] **Stąpor K.:** Automatyczna klasyfikacja obiektów. EXIT, Warszawa, 2005.
[4] **Hart P.E.:** The condensed nearest neighbor rule. IEEE Transactions on Information Theory, vol. IT-14, 3, 1968, pp. 515-516.
[5] **Gowda K. C., Krishna G.:** The condensed nearest neighbor rule using the concept of mutual nearest neighborhood. IEEE Transaction on Information Theory, v. IT-25, 4, 1979, pp. 488-490.
[6] **Tomek I.:** Two modifications of CNN. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-6, No. 11, 1976, pp. 769-772.
[7] **Dasarathy B.V.:** Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. IEEE Transactions on Systems, Man, and Cybernetics 24(3), 1994, pp. 511-517.
[8] **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, pp. 293-301.
[9] **Kuncheva L.I., Bezdek J.C.:** Nearest prototype classification: clustering, genetic algorithms, or random search? IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 28(1), 1998, pp. 160-164.
[10] **Cerveron V., Ferri F.J.:** Another move towards the minimum consistent subset: A tabu search approach to the condensed nearest neighbor rule. IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 31(3), 2001, pp. 408-413.
[11] **Wilson D.L.:** Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions On Systems, Man and Cybernetics, Vol. 2, 1972, pp. 408-421.
[12] **Tomek I.:** An experiment with the edited nearest-neighbor rule. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 6, 1976, pp. 448-452.
[13] **Devijver P.A., Kittler J.:** On the edited nearest neighbor rule. Proc. 5th Internat. Conf. on Pattern Recognition, 1980, pp. 72-80.
[14] **Kuncheva L.I.:** Editing for the k-nearest neighbors rule by a genetic algorithm. Pattern Recognition Letters, Vol. 16, 1995, pp. 809-814.
[15] **Raniszewski M.:** The Edited Nearest Neighbor Rule Based on the Reduced Reference Set and the Consistency Criterion. Biocybernetics and Biomedical Engineering, Vol. 30(1), 2010, pp. 31-40.

[16] **Xi X., Keogh E.J., Shelton C., Wei L., Ratanamahatana C.A.:** Fast Time Series Classification Using Numerosity Reduction. ICML, 2006, pp. 1033-1040.
[17] **Frank A., Asuncion A.:** UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.
[18] The ELENA Project Real Databases [http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/databases/REAL/].
[19] **Kuncheva L.I.:** Fitness functions in editing k-NN reference set by genetic algorithms. Pattern Recognition, Vol. 30, No. 6, 1997, pp. 1041-1049.

# DATA CLASSIFICATION DATA SET REDUCTION AND EDITING ALGORITHMS USING THE REPRESENTATIVE MEASURE

**Abstract**

In data classification we make decision based on data features. Proper and fast classification depends on a preparation of a data set and a selection of a suitable classification algorithm. One of these algorithms is popular Nearest Neighbor Rule (NN). Its advantages are simplicity, intuitiveness and wide range of applications. Its disadvantages are large memory requirements and decrease in speed for large data sets. Reduction algorithms remove much of data, which significantly speeds up NN. Simultaneously, they leave that data on the basis of which we can still make decisions with an acceptable classification quality. Editing algorithms remove redundant and atypical data from a data set. In this paper new reduction and editing algorithms, both using the representative measure, are presented. Tests were performed on several well-known in the literature data sets of different sizes. The results are promising. They were compared with the results of other popular reduction and editing procedures.

Politechnika Łódzka
Katedra Informatyki Stosowanej