

ci2.15Metryka odległościsection.2.1

Politechnika Wrocławska
Wydział Elektroniki
Rok akad. 2015/2016
Kierunek Informatyka

KOMPUTEROWE WSPOMAGANIE DIAGNOZOWANIA CHOROBY NIEDOKRWIENNEJ U DZIECI Z WYKORZYSTANIEM ALGORYTMÓW MINIMALNO-ODLEGŁOŚCIOWYCH

Zastosowanie informatyki w medycynie: Projekt

Grupa projektowa:

Barłomiej Grzegorek, XXXXXX
Marcin Mantke, 200633

Prowadzący:

prof. dr hab. inż. Marek Kurzyński

Wrocław, 06.06.2016

Spis treści

Spis rysunków	2
Spis listingów	3
1 Charakterystyka analizowanego problemu	4
2 Opis stosowanych algorytmów	5
2.1 Metryka odległości	5
2.2 Algorytm nearest neighbour	6
2.3 Algorytm nearest mean	6
3 Implementacja	8
3.1 Informacja o środowisku implementacyjnym	8
3.2 Ranking cech	8
3.3 Implementacja algorytmów	9
4 Opis badań eksperymentalnych	11
5 Wyniki badań	12
6 Podsumowanie i wnioski	13
6.1 Wnioski płynące z analizy wyników	13
6.2 Ocena krytyczna i podsumowanie projektu	13
Bibliografia	14

Spis rysunków

3.1	Ranking cech.	9
-----	-----------------------	---

Spis listingów

3.1	Podział danych na zbiór uczący i testowy.	9
3.2	Etap treningu - wyznaczanie centroidów.	10
3.3	Wyznaczenie odległości próbki od centroida i przydzielenie do klasy.	10

Rozdział 1

Charakterystyka analizowanego problemu

Rozdział 2

Opis stosowanych algorytmów

2.1 Metryka odległości

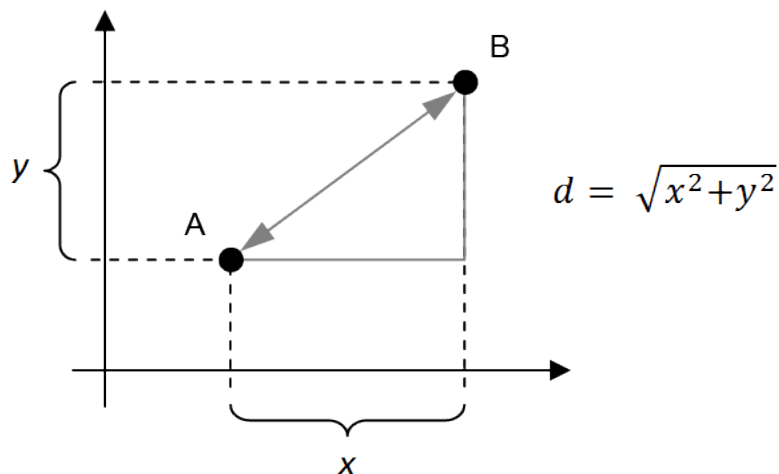
W algorytmach odległościowych istotną rolę odgrywa wybrana metryka wg której mierzone będzie odległość pomiędzy dwoma badanymi punktami. Spośród wielu wybrane zostały metryka euklidesowa i metryka Manhattan.

Metryka euklidesowa

Metryka euklidesowa, w której za odległość między dwoma punktami w przestrzeni przyjmuje się pierwiastek euklidesowego iloczynu skalarnego różnicy dwóch wektorów:

$$d_e(x, y) = \sqrt{(y - x, x - y)}$$

Przykład:

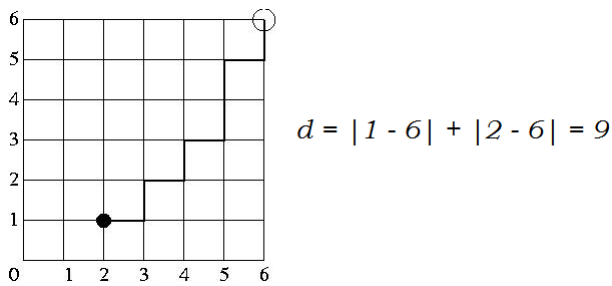


Metryka Manhattan

Metryka Manhattan, w której odległość dwóch punktów to suma wartości bezwzględnych różnic ich współrzędnych, zgodnie z poniższym wzorem:

$$d_m(x, y) = \sum_{k=1}^n |x_k - y_k|$$

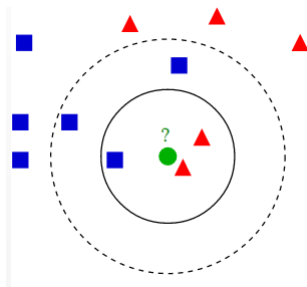
Przykład:



2.2 Algorytm nearest neighbour

Algorytm k-NN pozwala wyszukać w zadanym zbiorze punktów, k punktów znajdujących się najbliżej nowego punktu, korzystając przy tym z wybranej miary odległości.

W zagadnieniu klasyfikacji, algorytm k-NN przyjmuje jako argument tzw. zbiór uczący składający się z wielowymiarowych wektorów cech oraz przypisanych do nich klas. Faza klasyfikacji nowego obiektu polega na przypisaniu odpowiedniej klasy nieznanemu wektorowi cech. Nowy obiekt przypisywany jest do klasy, która występuje najczęściej wśród k najbliższych znajdujących się obiektów ze zbioru uczącego, zgodnie z wybraną metryką. Specjalnym przypadkiem algorytmu jest sytuacja, w której $k = 1$. Nazywa się algorytmem najbliższego sąsiada, w którym klasa nowego obiektu ustalana jest na podstawie najbliższej leżącej próbki ze zbioru danych uczących. Na poniższym rysunku przedstawiona została przykładowa przestrzeń zawierająca dane uczące (niebieskie kwadraty oraz czerwona trójkąty) oraz niezidentyfikowaną próbkę – zielone kółko):



Przyjmując liczbę najbliższych sąsiadów $k = 3$ (ciągła linia na rysunku), nowy obiekt zostanie przypisany do grupy czerwonych trójkątów, ponieważ jest ich więcej wśród 3 najbliższych próbek. Przyjmując $k = 5$, nowy obiekt zostanie przypisany do klasy niebieskich kwadratów, ponieważ w najbliższym otoczeniu znajdują się 3 kwadraty i tylko 2 trójkąty.

2.3 Algorytm nearest mean

Algorytm nearest mean jest jednym z algorytmów minimalnoodległościowych. Jest on wykorzystywany w statystyce to prognozowania wartości pewnej zmiennej losowej bądź do zadania klasyfikacji.

Założenia:

- dany jest zbiór uczący, który zawiera obserwacje. Każda z obserwacji ma przypisany wektor zmiennych objaśniających $X_1 \dots X_n$ oraz wartość zmiennej objaśnianej Y ,
- dana jest obserwacja C z przypisanym wektorem zmiennych objaśniających $X_1 \dots X_n$ dla której chcemy prognozować wartość zmiennej objaśnianej Y .

Przebieg algorytmu:

1. korzystając ze zbioru uczącego oblicz centroid dla każdej wartości zmiennej objaśnianej Y ,
2. zaklasyfikuj obserwację C do jednej z klas zmiennej objaśnianej Y poprzez minimalizację odległości pomiędzy wektorem wybranych cech, a centroidem.

Rozdział 3

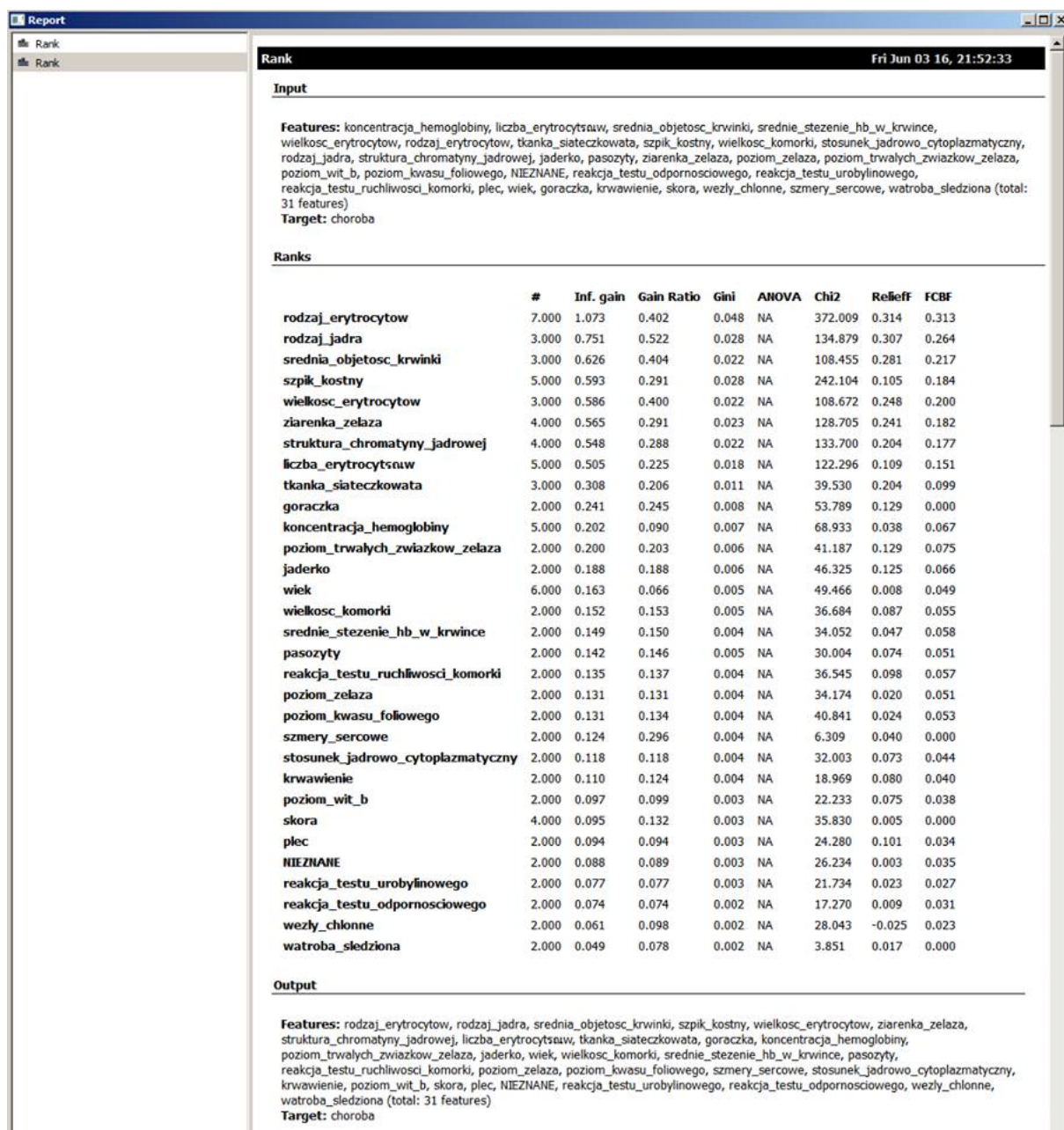
Implementacja

3.1 Informacja o środowisku implementacyjnym

Algorytmy zostały zaimplementowane w środowisku **MATLAB**. Ranking cech został uzyskany przy pomocy środowiska **Orange** (<http://orange.biolab.si/>). Jest to rozwiązanie open source, które umożliwia wizualizację oraz analizę danych.

3.2 Ranking cech

Korzystając z narzędzia **Orange** wygenerowaliśmy ranking cech przedstawiony na rysunku ??.



Rysunek 3.1: Ranking cech.

3.3 Implementacja algorytmów

Nearest mean

Na poniższych listingach przedstawione są najważniejsze części programów.

Listing 3.1: Podział danych na zbiór uczący i testowy.

```
data = data(randperm(end), :);
train = data(1:floor(0.5*size(data, 1)), :);
test = data(floor(0.5*size(data, 1))+1:end, :);
```

Listing 3.2: Etap treningu - wyznaczanie centroidów.

```
centroid = [unique(data(:, 1)) zeros(size(unique(data(:, 1)), 1), size(data, 2)-1)
];

for label = unique(train(:, 1))'
    % collect all the data of under the label
    train(train(:, 1) == label, 2:end)
    % compute the centroid for the label
    centroid(centroid(:, 1) == label, 2:end) = mean(train(train(:, 1) == label, 2:
        end));
end
```

Listing 3.3: Wyznaczenie odległości próbki od centroida i przydzielenie do klasy.

```
pre_result = zeros(size(test, 1), 1);
for i = 1:size(test, 1)
    dist = pdist2(test(i, feature_rank(1:k)), centroid(:, feature_rank(1:k)));
    [~, templabel] = min(dist);
    pre_result(i) = centroid(templabel, 1);
end
```

Rozdział 4

Opis badań eksperymentalnych

Rozdział 5

Wyniki badań

Rozdział 6

Podsumowanie i wnioski

6.1 Wnioski płynące z analizy wyników

6.2 Ocena krytyczna i podsumowanie projektu

Bibliografia

- [1] M.M. Sysło, N.Deo, J.S. Kowalik, *Algorytmy optymalizacji dyskretnej z programami w języku Pascal*, Wydawnictwo Naukowe PWN , 1999
- [2] R. Neapolitan, K. Naimipour, *Podstawy Algorytmów z przykładami w C++*, Helion, 2004
- [3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Wprowadzenie do algorytmów*, Wydawnictwo Naukowe PWN, 2013