

# Elementy sztucznej inteligencji

Sprawozdanie nr 2

## Analiza skupień

Maciej Ciupa

maciek.ciupa@gmail.com

## 1 Treść zadania

Implementacja dwóch algorytmów z analizy skupień:

- a) progowego
- b) min-max

które dla wybranych danych wejściowych z przestrzeni  $R^2$ , promienia i funkcji miary, utworzą zbiór klas do których zaklasyfikują wszystkie wejściowe obiekty.

## 2 Część teoretyczna

### 2.1 Algorytm progowy

Bardzo prosty algorytm, który opierając się na minimalnej odległości między klastrami, tzw. promień, tworzy kolejne środki klas dla punktów, których funkcja miary aktualnie rozpatrywanego punktu i wcześniej wybranych środków klas, jest większa od owego promienia.

*Dane wejściowe:*

Zbiór punktów (pomiarów)  $X = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$ , gdzie  $\bar{x}_i \in R^2$ ,  $n$  - ilość punktów

Promień klastra  $r \in R_+$

Funkcja miary  $d(\bar{x}_i, \bar{x}_j)$

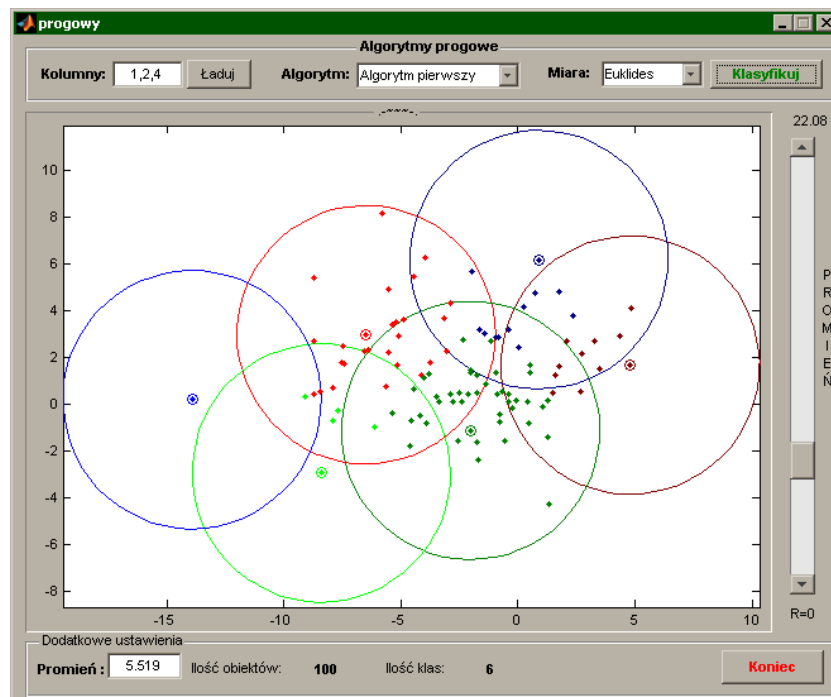
*Oznaczenia:*

Niech  $Z = \{z_1, z_2, \dots, z_j\}$  będzie zbiorem środków klas

*Algorytm*

1. Niech dowolny z punktów  $\bar{x}_i : i \in [n]$  będzie pierwszym środkiem klasy tj.:  
 $z_k = \bar{x}_i, k = 1, i = 1$
2. Obliczamy odległość między nie rozpatrywanymi dotąd punktami a środkami klas tj.:  $d(\bar{x}_i, z_j), j = 1, 2, \dots, k$ , gdzie  $\bar{x}_i$  to aktualnie rozpatrywany punkt

- Jeśli  $\forall_{j=1,2,\dots,k} d(\bar{x}_i, z_j) > r$  to otrzymujemy nowy środek klasy  $k+1$ -wszej tj.:  
 $k = k + 1, z_k = \bar{x}_i$
- Jeśli rozpatrzyliśmy wszystkie punkty tj.:  $i = n$  to idziemy do kroku 5. Jeśli nie - to idziemy do kroku 2 zwiększając  $i = i + 1$ .
- Środki klas mamy już ustalone, teraz przyporządkowujemy każdy z punktów (niebędący środkiem któreś z klas) do tej klasy, dla której funkcja miary od jej środka i danego punktu przyjmuje wartość najmniejszą



(rys. 1 - Implementacja algorytmu progowego)

## 2.2 Algorytm min-max

### Dane wejściowe:

Zbiór punktów (pomiarów)  $X = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$ , gdzie  $\bar{x}_i \in \mathbb{R}^2$ ,  $n$  - ilość punktów

Promień klastra  $\alpha \in (0;1)$

Funkcja miary  $d(\bar{x}_i, \bar{x}_j)$

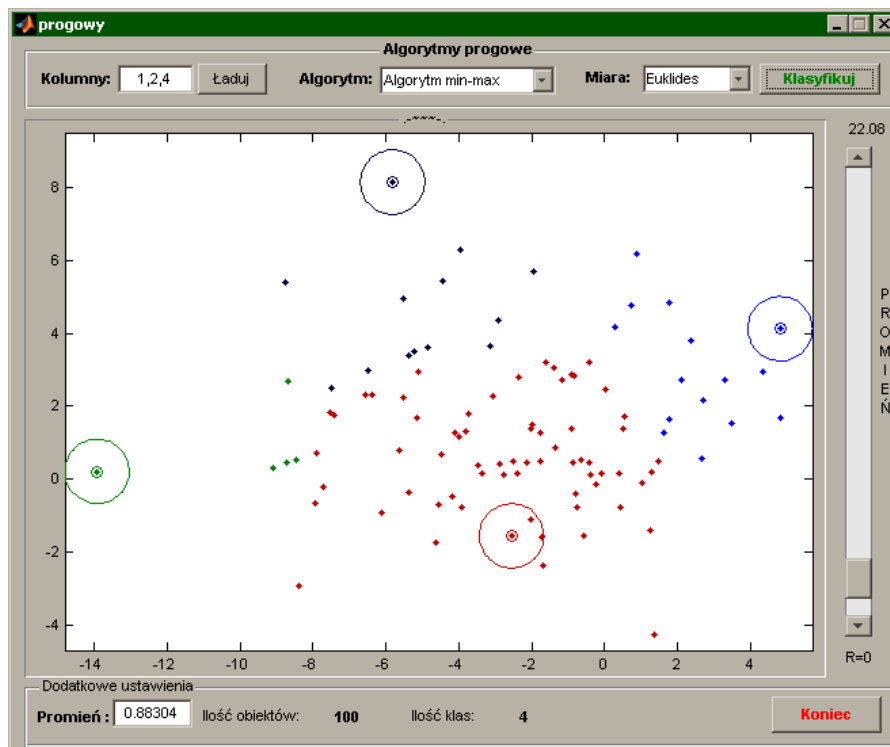
### Oznaczenia:

Niech  $Z = \{z_1, z_2, \dots, z_j\}$  będzie zbiorem środków klas

### Algorytm

- Niech dowolny z punktów  $\bar{x}_i : i \in [n]$  będzie pierwszym środkiem klasy tj.:  
 $z_k = \bar{x}_i, k = 1$

2. Obliczamy odległości między punktami a środkiem drugiej klasy tj.:  $d(\bar{x}_i, z_1)$ , środkiem drugiej klasy będzie taki  $z_2 = \bar{x}_i : i = \max\{d(x_j, z_1) : j = 1, 2, \dots, n\}$ ,  $k = 2$
3. Poszukujemy  $d_{k+1} = \max\{\min\{d_{i,j} : j = 1, 2, \dots, k\} : i = 1, 2, \dots, n\}$ , gdzie  $i$  przebiega przez wszystkie punkty (różne od ustalonych środków), a  $j$  przez wszystkie aktualnie ustalone środki klas
4. Jeżeli  $d_{k+1} > \alpha \cdot \min\{d_{i,j} : i = 1, 2, \dots, k \wedge j = 1, 2, \dots, i-1\}$ , to  $d_{k+1}$  staje się nowym środkiem nowej  $k+1$ -tej klasy. Niech  $k = k+1$  i idziemy do kroku 3.
5. Jeżeli nie, to mamy już ustalone wszystkie środki klas i teraz wystarczy tylko pozostałe punkty przyporządkować do tej klasy, od której funkcja miary danego punktu i środka tejże klasy przyjmuje wartość najmniejszą.



(rys. 2 – Implementacja algorytmu min-max)

### 3 Implementacja algorytmów

Algorytm został zaimplementowany w środowisku Matlab 7.2.0. (rys. 1,2) :

- *progowy.m* – główny plik programu
- *progowy.fig* – okno środowiska guide
- *dddane.txt* – dane wejściowe

Umożliwia on:

- załadowanie dwóch kolumn z pliku *dddane.txt*, które poddaje klasyfikacji w polu *Kolumny*, oddzielone przecinkiem
- wybranie jednego z dwóch omawianych algorytmów z listy *Algorytm*

```
function ilosc_klas = MetodaPierwsza(RADIUS, X, Y, MIARA)
function ilosc_klas = MetodaMinMax(Alpha, X, Y, MIARA);
```

- wybranie jednej z dwóch metryk (Euklidesowa, miejska) z listy *Miara*  

```
function length = miaraEuklidesa(p1, p2)
function length = miaraMiejska(p1, p2)
```
- zmianę promienia suwakiem, oraz wpisanie ręcznie w polu *Promień*

Kolejnym punktom nadaje odpowiedni kolor RGB danej klasy, wykorzystując funkcję

```
function [r g b] = getColor(nr, all)
```

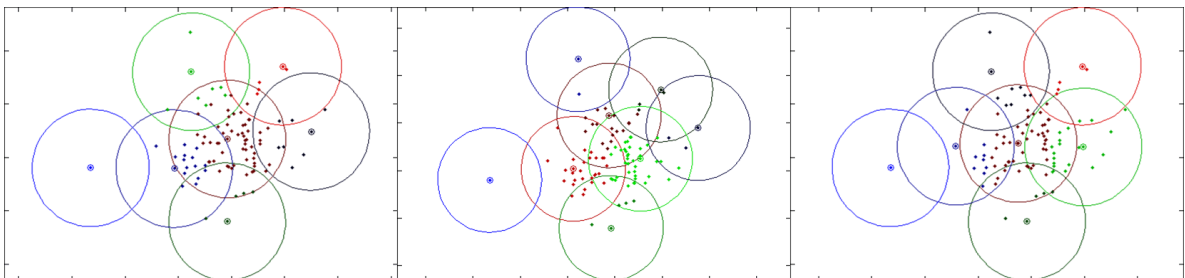
która nadaje każdej kolejnej klasie unikalny kolor. W przypadku algorytmu progowego dodatkowo rysuje okręgi o zadanym promieniu.

## 4 Wnioski i uwagi

Krótkie uwagi dotyczą danych z kolumn 3, 7 z pliku *dddane.txt*. Rozmiar próbki: 100.

### 4.1 Algorytm progowy

Przy promieniu  $r = 5.5$  i metryce euklidesowej, algorytm tworzy średnio 7 klas, w zależności od wylosowanego punktu startowego. Natomiast dla miary miejskiej, przy tym samym promieniu średnio 10 klas.



### 4.2 Algorytm min-max

Spore problemy sprawił algorytm min-max, który wydaje się bardzo „czuły” na zmianę parametru  $\alpha \in (0;1)$ . Ponieważ nawet małe zmniejszenie jego wartości powoduje, że algorytm klasyfikuje wszystkie punkty jako środki i tym samym tworzy tyle klas ile jest punktów.

Przy  $\alpha = 0.8$ , algorytm tworzy średnio 3 klasy, poniższe wyniki klasyfikacji zostały przeprowadzone na tych samych danych przy tej samej wartości parametru  $\alpha$ .

