Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

# Umelá Inteligencia Zadanie 3 – b) Zhlukovanie

Martin Rudolf

Cvičenie: Štvrtok 16.00

Cvičiaci: Ing. Boris Slíž

2020/2021

## Znenie Zadania:

Máme 2D priestor, ktorý má rozmery *X* a *Y*, v intervaloch od −5000 do +5000. Tento 2D priestor vyplňte 20 bodmi, pričom každý bod má náhodne zvolenú polohu pomocou súradníc *X* a *Y.* Každý bod má unikátne súradnice (t.j. nemalo by byť viacej bodov na presne tom istom mieste).
Po vygenerovaní 20 náhodných bodov vygenerujte ďalších 40000 bodov, avšak tieto body nebudú generované úplne náhodne, ale nasledovným spôsobom:

1. Náhodne vyberte jeden z existujúcich bodov v 2D priestore
2. Vygenerujte náhodné číslo *X_offset* v intervale od -100 do +100
3. Vygenerujte náhodné číslo *Y_offset* v intervale od -100 do +100
4. Pridajte nový bod do 2D priestoru, ktorý bude mať súradnice ako náhodne vybraný bod v kroku 1, pričom tieto súradnice budú posunuté o *X_offset* a *Y_offset*

Vašou úlohou je naprogramovať zhlukovač pre 2D priestor, ktorý zanalyzuje 2D priestor so všetkými jeho bodmi a rozdelí tento priestor na *k* zhlukov (klastrov). Implementujte rôzne verzie zhlukovača, konkrétne týmito algoritmami:

- • k-means, kde stred je centroid
- • k-means, kde stred je medoid
- • aglomeratívne zhlukovanie, kde stred je centroid
- • divízne zhlukovanie, kde stred je centroid

Vyhodnocujte úspešnosť/chybovosť vášho zhlukovača. Za úspešný zhlukovač považujeme taký, v ktorom žiaden klaster nemá priemernú vzdialenosť bodov od stredu viac ako 500.
Vizualizácia: pre každý z týchto experimentov vykreslite výslednú 2D plochu tak, že označkujete (napr. vyfarbíte, očíslujete, zakrúžkujete) výsledné klastre.

## Riešenie Zadania:

Najprv je potrebné získať dataset, to znamená je potrebné vygenerovať body podľa zadania s ktorými budeme pracovať. O to sa stará funkcia **create_points().** Po vygenerovaní bodov môžeme riešiť problém príslušnými algoritmami.

## K-means Centroid

V projkte funkcia **k_means_centroid()**. Na začiatku hodíme do nášho 2D priestoru náhodných k bodov (centroidov), k volíme podľa toho koľko chceme zhlukov, pomocou funkcie **getLabel()** priradím každému centroidu príslušné body ktoré mu patria. Funkcia getCentroids() vycentruje súradnice centroidov na aritmetický priemer všetkých bodov ktoré im patria. Toto prebieha v cykle ktorý skončí ak ani jeden bod v zhluku nie je vzdialený od centroidu viac ako 500, alebo súradnice každého centroidu ostanú rovnaké.

## K-means Medoid

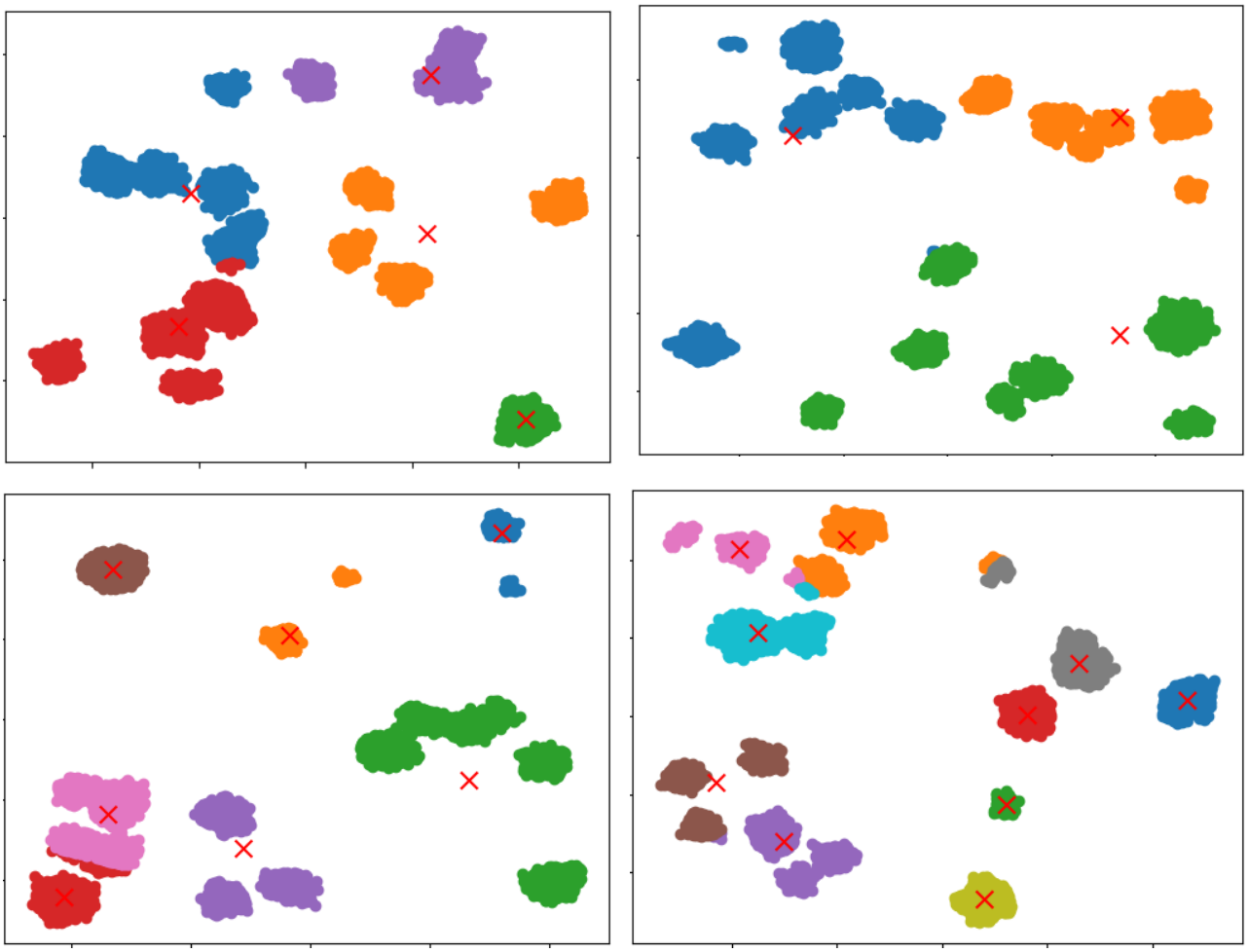
V projekte funkcia **k_means_medoid()**. Podobne ako pri k-means centroid hodíme do priestoru k náhodných medoidov random_medoids(), a priradíme im príslušne body. Po priradení bodov ohodnotíme medoid tak že sčítame vydialenosti všetkých bodov od medoidu na to slúži funkcia **getMedoidCost()**. Túto hodnotu si uložíme a vyberieme si nový náhodný medoid v zhluku opäť ho ohodnotíme a porovnáme nove ohodnotenie so starým to ktoré je menšie vyhráva a stáva sa medoidom daného zhluku. Tento proces prebieha v cykle ktorý sa zastaví ak medoidy ostanú niekoľko krát nezmenené alebo vzdialenosť každého bodu k príslušnému medoidu nie je väčšia ako 500.

## Divizívne zhlukovanie

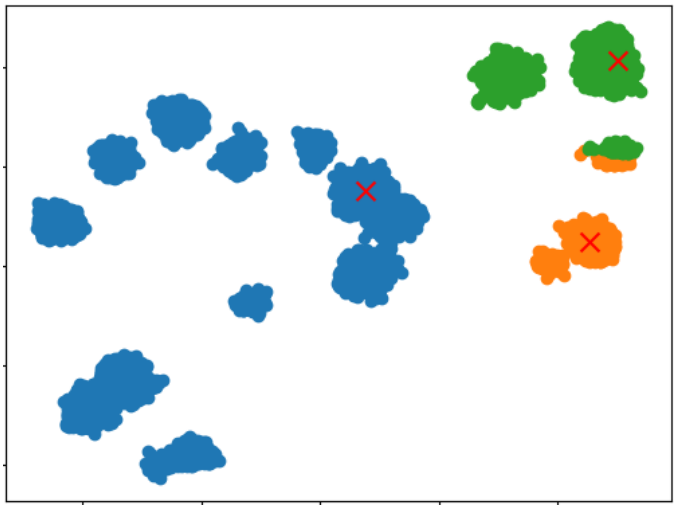
V projekte funkcia **divisive(),** táto metóda na začiatku vytvorí pomocou k-means prvý počiatočný klaster. V nekonečnom cykle funkcia začne deliť klastre na „polovicu“ pomocou k-means do ktorého posielam body patriace klastru ktorý rozdeľujem a, n = 2. Cyklus sa zastaví ak máme k klastrov čo je vstupným argumentom funkcie.

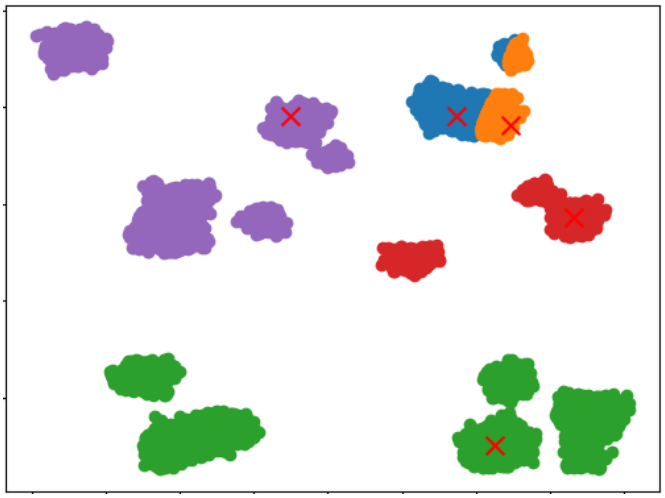
## Pozorovanie a záver:

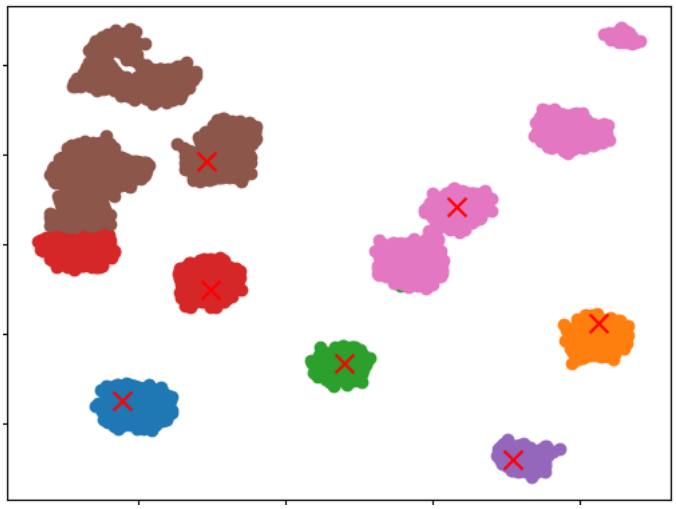
### K-means-centroid pre K = 3, 5, 7, 10 pre 40020 bodov v priestore

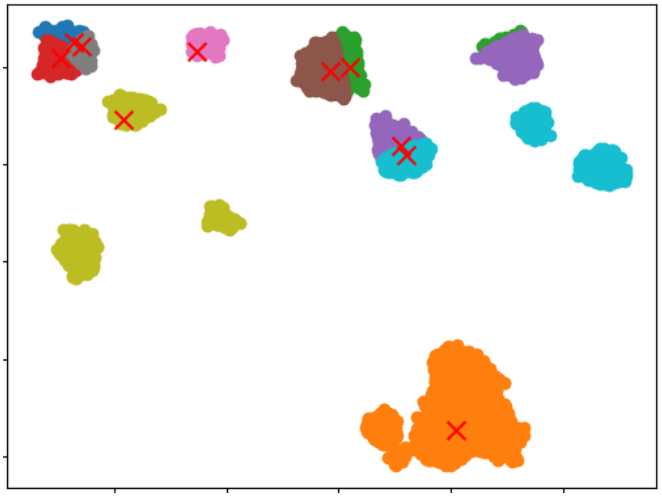


## K-means-medoid pre K = 3, 5, 7, 10 pre 40020 bodov v priestore









## Divízne zhlukovanie pre K = 3, 5, 7, 10 pre 40020 bodov v priestore