SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

ZADANIE 4(b) - Klastrovanie

Martin Nemec
FIIT STU
Cvičenie: Štvrtok 12:00
9.12.2021

1 Zadanie úlohy

Najprv sme na 2D priestore mali vygenerovať 20 000 bodov podľa postup zo zadania. Následne našou úlohou bolo implementovať rôzne algoritmy na vytváranie zhlukov.

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

2 Opis riešenia

2.1 Generovanie bodov

Na začiatku som vygeneroval náhodných 20 bodov, kde x-ová aj y-ová súradnica bodu bola z intervalu -5000; 5000.
Následne som vždy vybral náhodný bod, z týchto 20 bodov a vygeneroval som k nemu další bod, s tým, že oproti zadaniu som nepoužil offset z intervalu -100; 100, pretože to vytváralo akési štvorce. Dočítal som sa, že na vytváranie takýchto bodov dobre slúži príkaz random.gauss(daný bod, odchýlka)

2.2 K-means, kde stred je centroid

Kedže centroid je iba fiktívny bod, vygeneroval som si toľko centroidov koľko zhlukov je potrebné vytvoriť. Následne som spravil prvú iteráciu, kde som priradil do zhlukov najbližšie body k daným centroidom.
Následuje cyklus, v ktorom v každom zhluku vypočítam novú centroidu tak, že spočítam x-ové súradnice všetkých bodov a y-ové súradnice všetkých bodov v danom zhluku a urobím priemer. Znovu príradím k centroidom najbližšie body. Takto sa tento cyklus opakuje pokiaľ už nenastane žiadna zmena v zhlukoch.

2.3 K-means, kde stred je medoid

Podobne ako pri centroide spravím prvú iteráciu. Hlavný cyklus je odlišný tým, že medoid je realny bod a je to taký bod, v ktorom v danom zhluku je vzdialenosť všetkých bodov najmenšia. Znovu príradim k medoidom v zhlukoch body a opakujem pokiaľ nenastane žiadna zmena.

2.4 Divízne zhlukovanie, kde stred je centroid

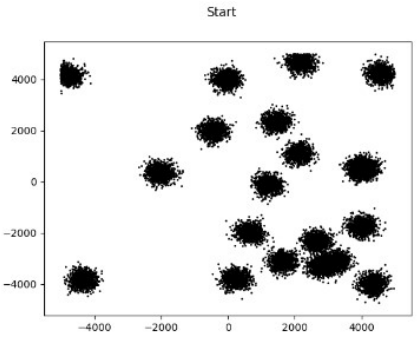
Divízne zhlukovanie bolo jednoduché na implementáciu, keďže sa tam využíva k-means s centroidom.
Najprv si vytvorím dané zhluky z bodov. To znamená že budem mať 2 zhluky, ktoré si rozdelili body podľa k-means. Následuje cyklus, v ktorom si zistím ktorý zhluk je najväčší podľa toho aký ma priemer vzdialeností bodov a ten rozdelím na 2 zhluky. Takto pokračujem pokiaľ nemám požadovaný počet zhlukov.

2.5 Aglomeratívne zhlukovanie, kde stred je centroid

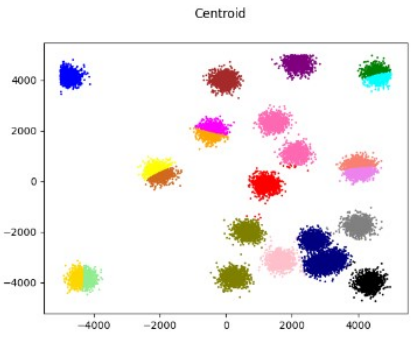
Aglomeratívne zhlukovanie funguje opačne ako divízne. Na začiatku, keď mám 20k bodov, tým pádom budem mať aj 20k zhlukov o veľkosti jedného bodu.
V tomto algoritme som využíval knižnicu numpy a pomáhal som si maticou vzdialeností každého zhluku s každým. V hlavnom cykle som vždy zistil podľa matice vzdialeností 2 najbližšie zhluky. Tieto zhluky som spojil do jedného, vypočítal som mu centroid a pôvodné 2 zhluky som vymazal. Tento cyklus sa takto opakoval až pokiaľ sa nedostal na požadované množštvo zhlukov.

3 Vykresľovanie grafov v programe

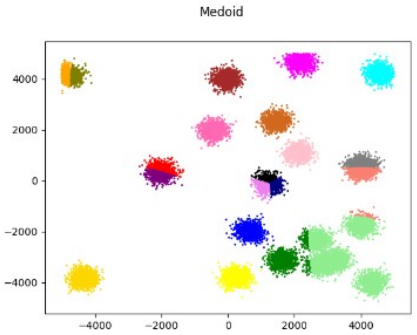
Na vykreslenie grafov som používal knižnicu matplotlib. Na týchto obrázkoch môžeme vidieť rôzne typy zhlukovača pri 20k bodoch a 20 zhlukoch.



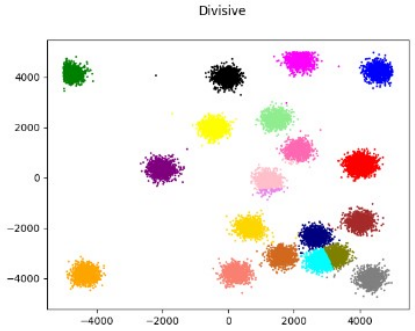
Obr. 1: Pred zhlukovaním



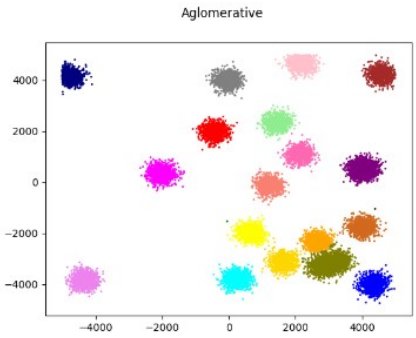
Obr. 2: K-means, centroid



Obr. 3: K-means, medoid



Obr. 4: Divízne zhlukovanie



Obr. 5: Aglomeratívne zhlukovanie

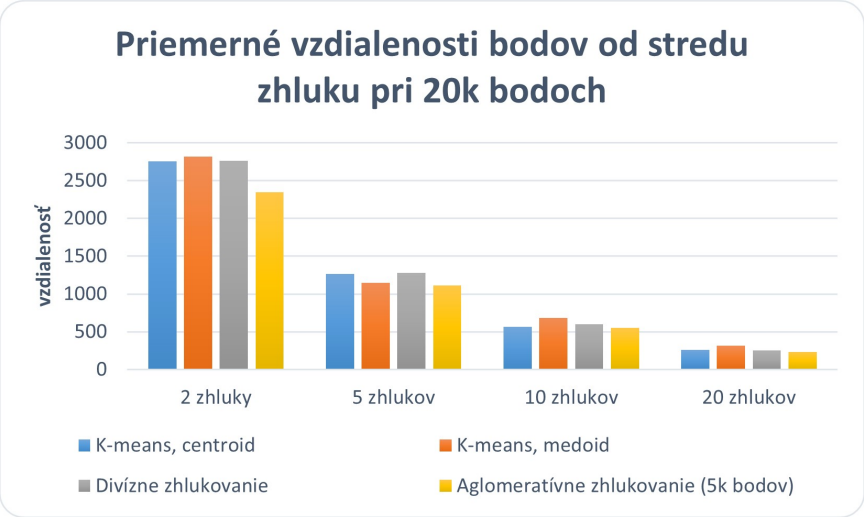
```
Aglomerativne zhlukovanie pri 20 zhlukoch zabral 53207.18304872513
```

Pri aglomeratívnom testovaní som netestoval s 20k bodmi pretože je to extrémne náročné na čas. Spustil som to iba raz s 20k bodmi pre 20 zhlukov. V prepočte aglomeratívne zhlukovanie pri 20k bodov zabralo 14,77 hodín. Podľa obrázkov môžeme vidieť že divízne a aglomeratívne merania sú dosť presné. Ďalšie info je v časti Testovanie.

4 Testovanie

Všetky údaje z testovania sú priemery po 5 spusteniach.

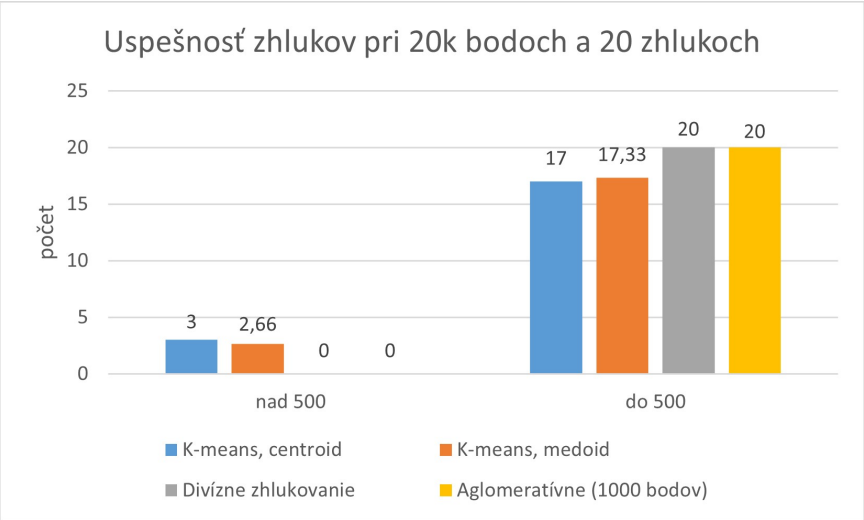
4.1 Priemerné vzdialenosti



Tento graf zobrazuje spriemerované vzdialenosti bodov od stredu zhluku pri rôznych algoritmoch. Pri všetkých algoritmoch okrem aglomeratívneho zhlukovania som použil 20k bodov, pre aglomeratívne som použil 5000 bodov. Ako môžeme z grafu vidieť pri zvyšovaní počtu zhlukov sa logicky zmenšujú priemerné vzdialenosti bodov od stredu zhluku. Všetky algoritmy sú priemerne rovnaké.

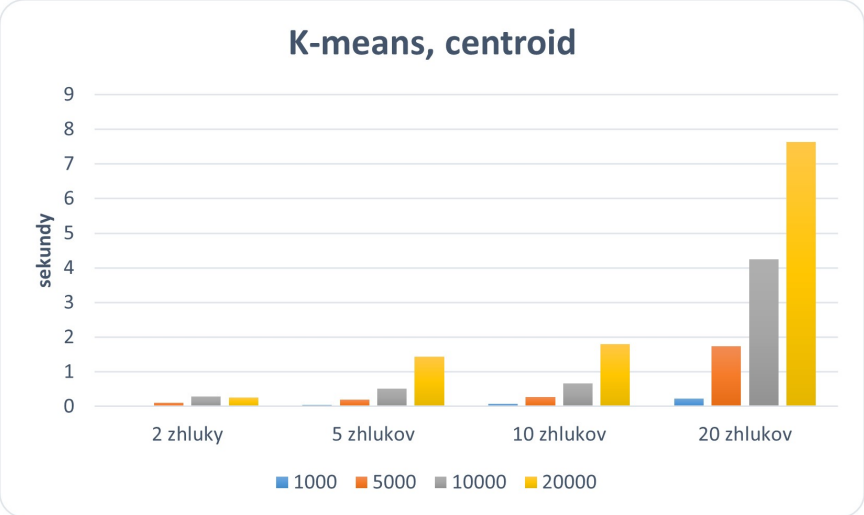
4.2 Vyhodnocovanie úspešnosti zhlukovačov

Tento graf znázorňuje aké úspešné sú dané zhlukovače. Za úspešný zhlukovač považujeme taký, v ktorom žiaden zo zhlukov nemá väčšiu priemernú vzdialenosť bodov od stredu zhluku ako 500. Na tomto grafe môžeme vidieť že aj po viacerých spusteniach pri divíznom a aglomeratívnom zhlukovaní je priemerná vzdialenosť všetkých zhlukov do 500. Zatial čo pri K-means pri centroide aj medoide vidíme že nie vždy sú vzdialenosti bodov od stredu zhluku do 500.



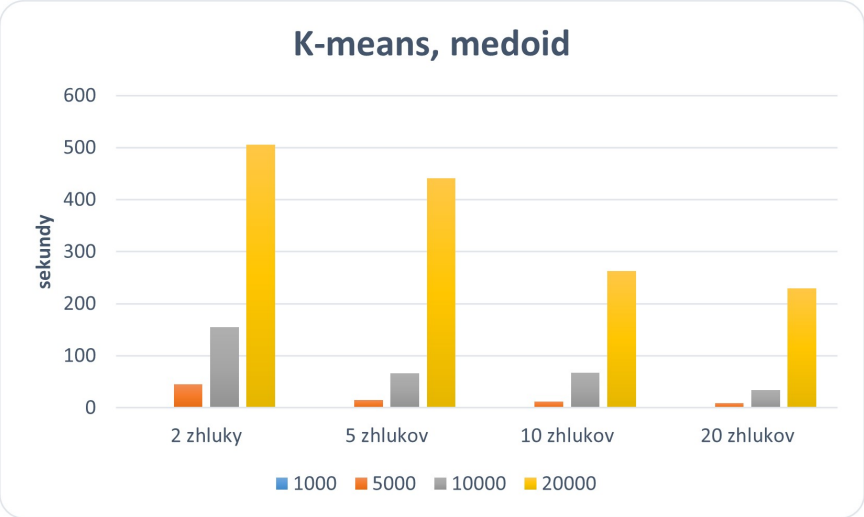
4.3 Časové testovanie

K-means, kde stred je centroid



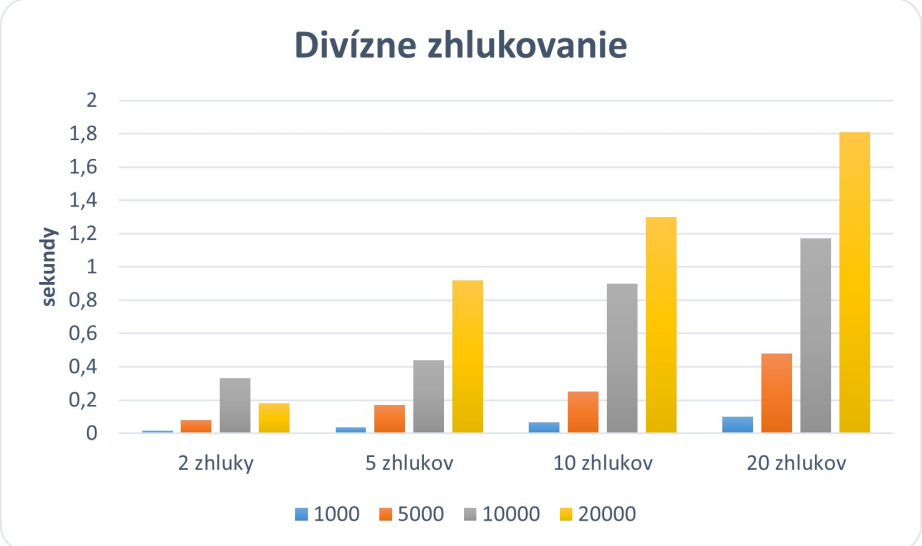
Pri časovom testovaní K-means, kde stred je centroid som nameral že, pri zväčšovaní počtu zhlukov a zväčšovaní údajov sa zvyšuje čas. Kde pri 1000 bodoch je jedno koľko je počet zhlukov, lebo je to veľmi rýchle. Pri 10k a 20k bodoch vidíme už väčšie rozdiely, kde pri 2 zhlukoch priemerný čas je niečo malo cez nad nulou. No pri 10 a 20 zhlukoch sa tie časy dosť predlžujú. Kontkrétne pri 20k bodoch a 20 zhlukoch je to takmer 8 sekúnd.

K-means, kde stred je medoid



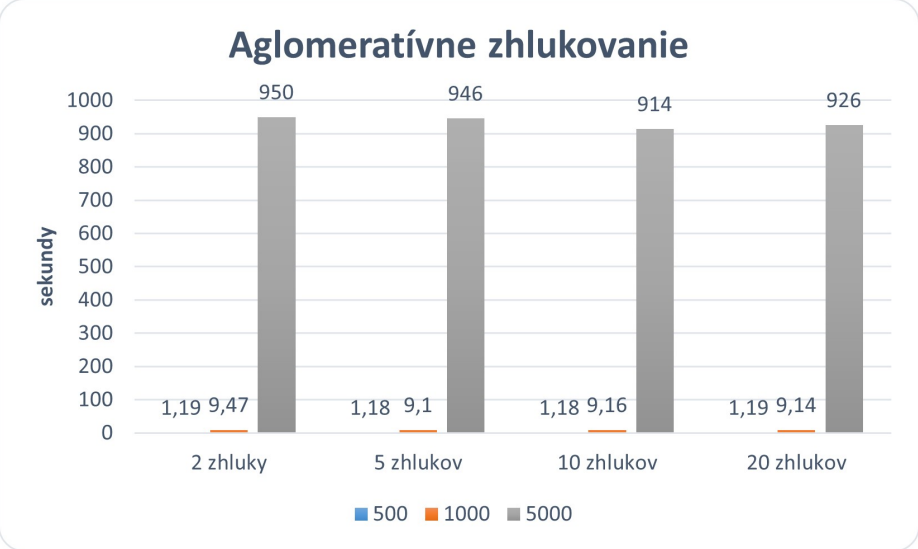
Pri časovom testovaní k-meansu, kde stred je medoid som zistil, že oproti k-means s centroidom pri zvyšovaní počtu zhlukov sa práveže znižuje čas. Ako môžeme vidieť pri 20k bodoch a 2 zhlukoch sa čas pohyboval priemerne okolo 500s a pri rovnakom počte bodov ale 20 zlukoch sa čas zmenšil viac než o polovicu. Ak by som porovnal K-means s centroidom a s medoidom tak k-means, kde stred je medoid je podstatne pomalší.

Divízne zhlukovanie



Divízne zhlukovanie má veľmi podobné rýchlosti ako k-means s centroidom, pričom pri zvyšovaní počtu bodov tie časy narastajú ešte menej ako pri K-means. Takéto dobré časy dosahuje preto lebo vždy delí iba jeden zhluk na 2. Pri zvyšovaní počtu zhlukov a počtu údajov čas klasicky rastie, ale pri 20 zhlukoch a 20k bodov bol priemerný čas len okolo 1,8 sekundy.

Aglomeratívne zhlukovanie



Pri aglomeratívnom zhlukovaní som robil testovanie len s 500, 1000 a 5000 bodmi pretože je to omnoho náročnejšie na čas. Pri meraní som zistil že vôbec nezáleží na počte zhlukov, vždy tie časy na daný počet bodov boli takmer rovnaké. Je to preto, lebo ak máme najprv 5k bodov tak je to 5k zhlukov na začiatku a ide to veľmi pomaly. Postupne ako ubúdajú počty zhlukov až na požadovaný počet, tak úprava zhlukov sa zrýchľuje. Ako z grafu môžeme vidieť pri 500 a 1000 bodoch sú časy extrémne malé oproti časom pri 5000 bodoch. Časová zložitosť v tomto algoritme by mala byť O(n^3)

5 Záver a porovnanie

Na záver by som porovnal a zhodnotil jednotlivé algoritmy. Najprv by som porovnal K-means, kde stred je centroid a K-means, kde stred je medoid. K-means s centroidom dosahuje celkom pekné výsledke, síce nie je úplne presný ako napríklad divízne alebo aglomeratívne zhlukovanie. Oproti medoidu sa mi podarilo centroid implementovať oveľa lepšie ako medoid, keďže tie časy sú pri centroide ďaleko lepšie a výsledok je takmer rovnaký.

Ďalej by som chcel porovnať aglomeratívne a divízne zhlukovanie. Pri divíznom zhlukovaní je čas veľmi rýchli a je naozaj veľmi presné, naopak kde aglomeratívne zhlukovanie je extrémne časovo náročne ale rovnako presné.
Ak by som mal porovnať K-means, kde stred je centroid a divízne zhlukovanie, osobne by som volil divízne zhlukovanie, pretože má najlepší pomer cena/výkon. Divízne zhlukovanie dokáže pri nízkom čase byť veľmi presné.