Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Umelá Inteligencia

Zadanie 4

Klastrovanie

Matsveyeva Lada-Ivanna

ZS 2021/22

Obsah:

Matsveyeva Lada-Ivanna, ID: 99198

Zadanie

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 20000 bodov, avšak tieto body nebudú generované úplne náhodne, ale nasledovným spôsobom:

1. Náhodne vyberte jeden zo všetkých doteraz vytvorených bodov v 2D priestore.
   Ak je bod príliš blízko okraju, tak zredukujete príslušný interval v nasledujúcich dvoch krokoch.
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

Matsveyeva Lada-Ivanna, ID: 99198

Opis algoritmov:

Generujem náhodne body, podľa algoritmu, ktorý je uvedený v zadaní.

k-means

Najprv vyberiem *k* náhodných bodov, ktoré budú stredmi zhlukov. Ďalej pre každý bod určím najbližší zhluk. Po prvotnom ohodnotení vyrátam stred ešte raz.

Stred centroid – priemerná vzdialenosť všetkých x,y súradníc bodov.

Stred medoid – reálny bod, ktorý má najmenšiu euklidovskú vzdialenosť ku všetkým bodom.

V cykle určujem najbližší zhluk a aktualizujem stred pokiaľ sú zmeny v zhlukoch.

Aglomeratívne zhlukovanie

Každý vygenerovaný bod je zhluk. Vytvorím maticu susednosti, nájdem zhluky, ktoré sú najbližšie ku sebe a spojím ich do jedného zhluku. Aktualizujem maticu susednosti pre nový zhluk, ktorý vložím na koniec matice a zase nájdem najbližšie ku sebe zhluky. Tak sa cyklím kým nedosiahnem potrebný počet zhlukov.

Divízne zhlukovanie

Divízne zhlukovanie je opačne ako aglomeratívne. Všetky vygenerované body sú 1 zhluk. Tento zhluk pomocou k-means so stredom centroid rozdelím na 2 zhluky. Pôvodne centroidy, oproti tomu ako bolo v k-means, vyberiem nie je náhodne, ale podľa najväčšiej vzdialenosti dvoch bodov. Potom kým nebude dosiahnuty požadovaný počet zhlukov v cykle vyberiem zhluk ktorý v sebe má najväčší priemer vzdialenosti a rozdelím na 2 zhluky.

Reprezentácia údajov:

Vygenerované body mám uložené v dátovom type list. Jedna položka – jeden bod a zhŕňa v sebe x-ovú, y-ovú súradnice a zhluk ku ktorému patrí. Centroidy/medoidy mám v samotných list-och.

Pre maticu susednosti používam maticu z knižnice numpy, je lepšia pre reprezentáciu matice, ako vnorené list-y a taktiež má v sebe funkcie na nájdenie min, max prvkov.

Na výpočet vzdialenosti medzi dvoma bodmi/zhlukmi používam euklidovskú vzdialenosť z knižnice math, funkcia dist().

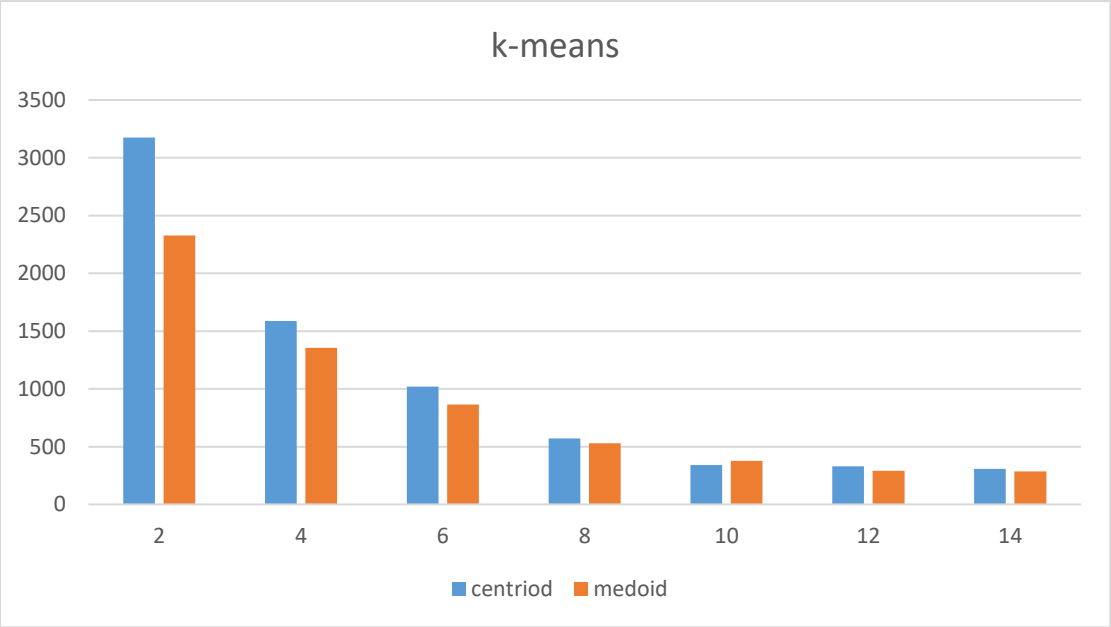
Pre vizualizáciu používam knižnicu matplotlib. Rôzne zhluky reprezentujem rôznymi farbami.

Testovanie a porovnanie výsledkov

Na zhodnotenie výsledkov algoritmov porovnávam vzdialenosť bodov od stredu zhlukou pre rôzny počet zhlukou.

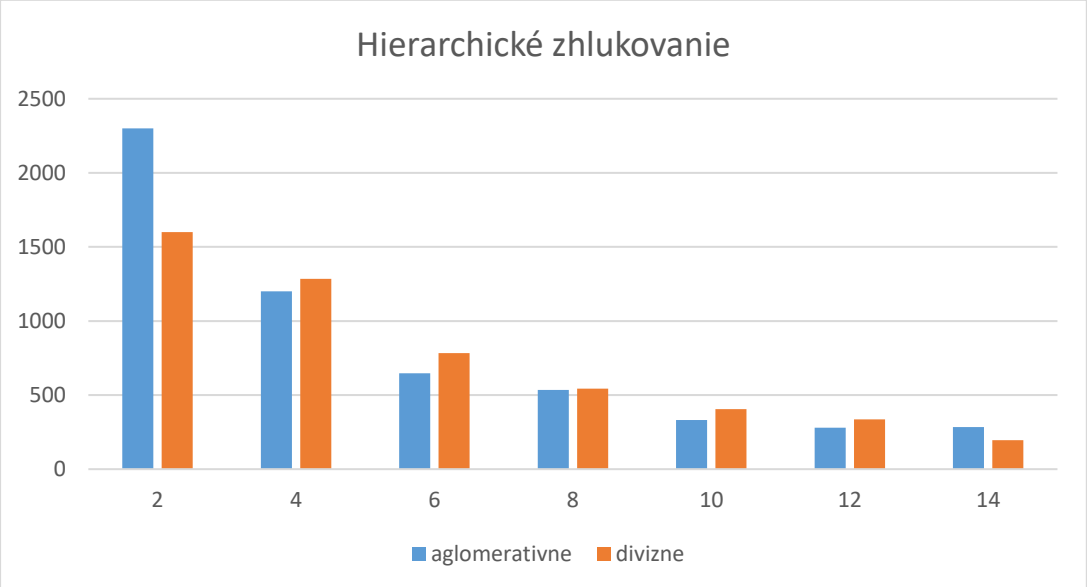
Podľa výsledkov je vidieť, že keď zvyšujem počet zhlukou, tak sa znižuje priemerná vzdialenosť v zhlukoch. Pre každý algoritmus po 8 zhlukov priemerná vzdialenosť sa neklesá. Podľa tohto je optimálny počet zhlukov 8 až 10.

Medzi k-means časovo rýchlejší je so stredom ako centroid, lebo len spočíta všetky súradnice a nájde z nich priemer. Vtedy keď pre medoid je potrebné prejsť každý bod, a nájsť pre neho vzdialenosť ku každému bodu v zhluku. Ale podľa efektívnosti je lepší algoritmus so stredom ako medoid, lebo je to už reálny bod v zhluku.



Medzi aglomeratívnym a divíznym je rýchlejší divízny ale majú približne rovnaké výsledky primernej vzdialenosti bodov od stredu. Aglomeratívne kvôli tomu že začína z toho, že každý bod je zhluk, tak matica susednosti pre počet bodov 20 000, je veľkosťou 20000*20000, čo je celkovo pamäťové náročné, a kvôli tomu je aj pomalšia.

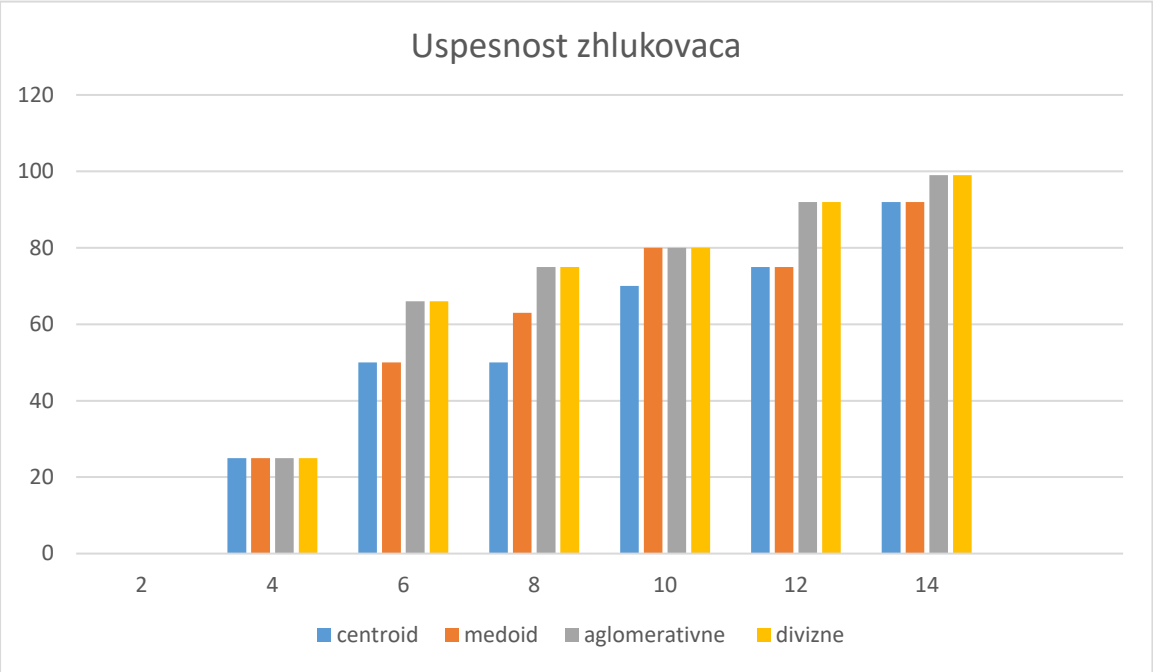
V aglomeratívnom zhlukovani môže nastať situácia, keď v jednom zhluku bude len pár bodov, vtedy keď pri divíznom toto nemôže nastať (len ak počet zhlukov sa takmer nerovná počtu bodov).



Matsveyeva Lada-Ivanna, ID: 99198

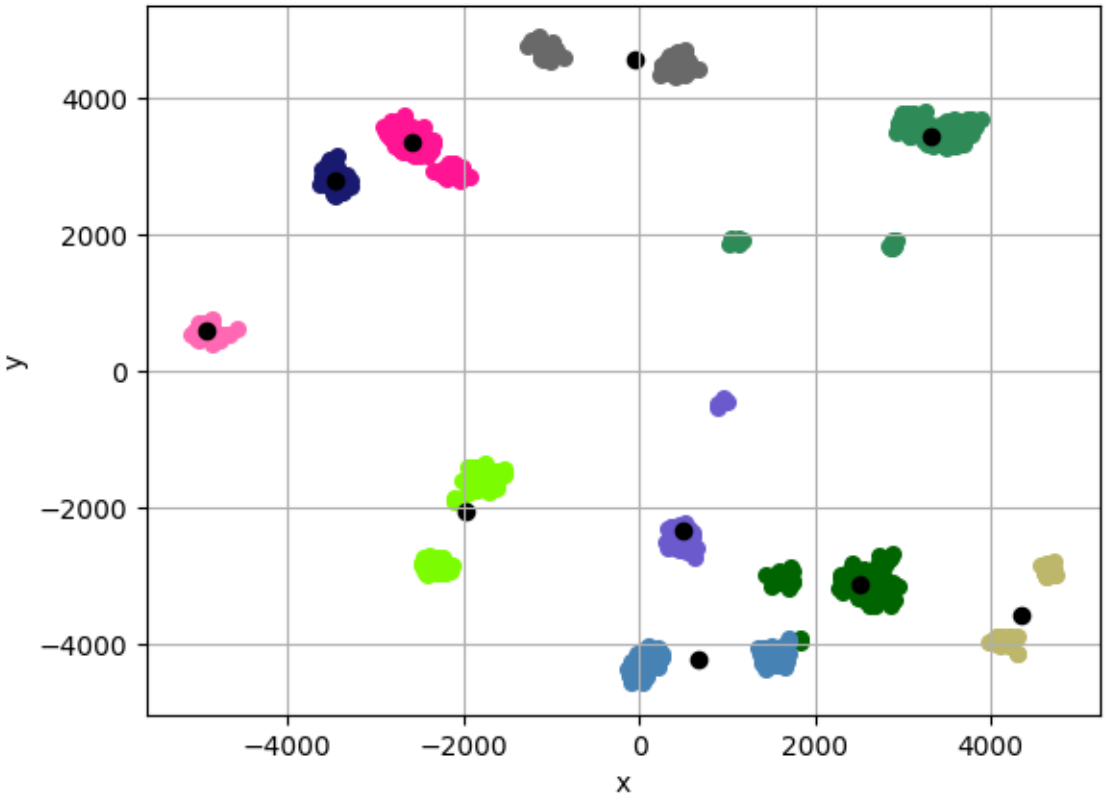
Porovnanie úspešnosti zhlukovačov.

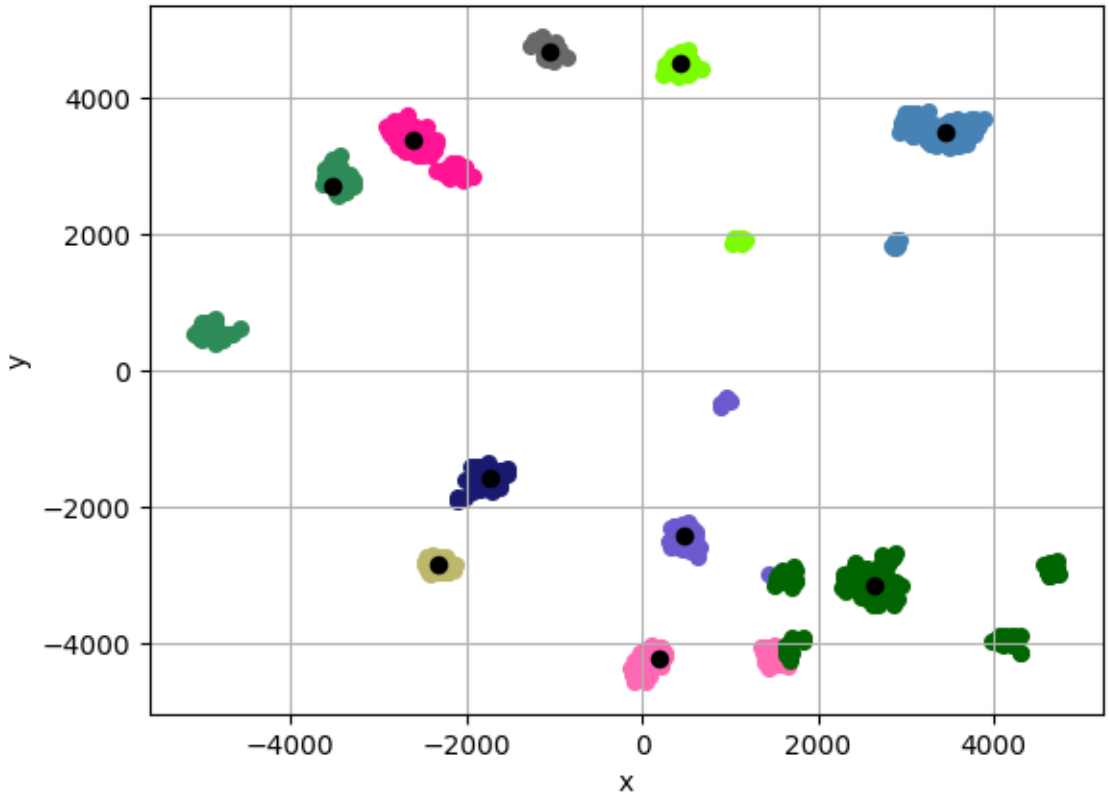
Podľa zadania úspešný zhlukovac vtedy keď každý zhluk má priemernú vzdialenosť bodov od stredu nie je viac ako 500. Porovnala som všetky algoritmy spolu, pre rôzny počet zhlukov. Pri menšom počte zhlukov od 2 po 4 úspešnosť je menšia ako 50%, je to spôsobené tým, že zhlukov je malo, a počet bodov a veľkosť plochy sú veľké. Pre počet zhlukov 4 a viac úspešnosť je viac ako 50%. Úspešnosť v hierarchyckom zhlukovani je väščia ako pre k-means.

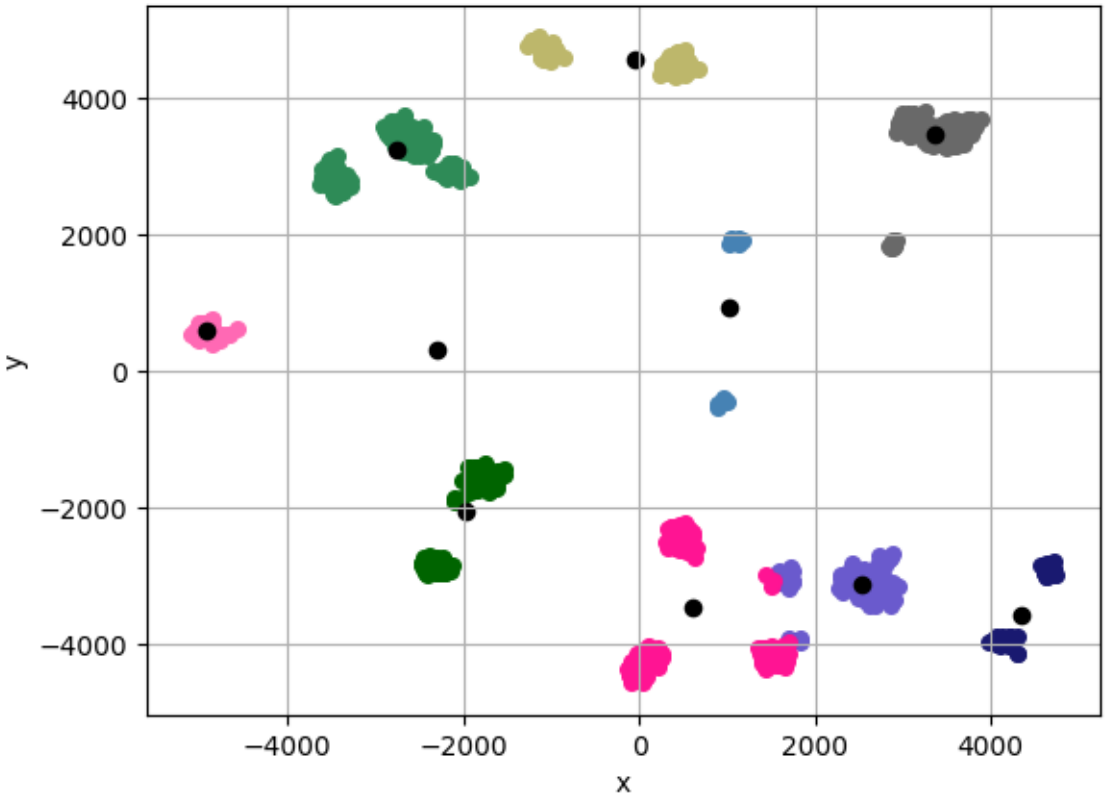


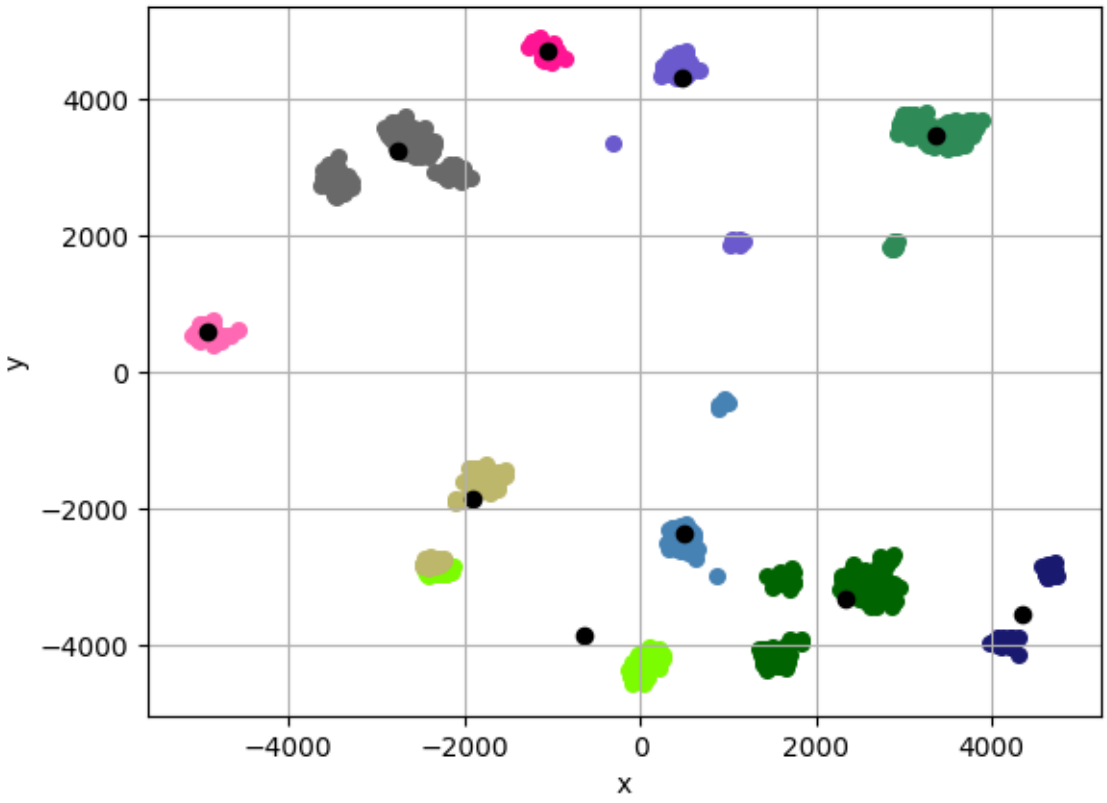
Príklady vizualizácie:

1. k-means centroid
2. k-means medoid
3. aglomeratívne zhlukovanie
4. divízne zhlukovanie


4000
2000
0
−2000
−4000
y
−4000
−2000
0
2000
4000
x


4000
2000
0
−2000
−4000
y
−4000
−2000
0
2000
4000
x


4000
2000
0
−2000
−4000
y
−4000
−2000
0
2000
4000
x


4000
2000
0
−2000
−4000
y
−4000
−2000
0
2000
4000
x

Zhrnutie a zhodnotenie riešenia

Pri k-means algoritmu veľkú rolu hrá to, ako sa vyberú stredné body zhluku, lebo sa vyberajú náhone. Je to časovo a pamäťové výhodnejšie, ako napríklad pri aglomeratívnom a divíznom algoritmoch.

Pri hierarchických algoritmoch rozdelenie do zhlukov je presnejšie, ale je časovo a pamäťové náročnejšie. Pri týchto algoritmoch je veľmi podstatné vedieť ich optimalizovať.