Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
13. 12. 2022

ZADANIE 3 –
Klastrovanie

Filip Remšík

Zadanie

Zadaním úlohy je rozdeliť body do klastrov. Budeme mať mapu s danými rozmermi a do nej si najskôr vygenerujeme 20 bodov. Následne budeme generovať ďalšie body tak, že vyberieme náhodne už vygenerovaný bod a podľa toho či sa bod nachádza blízko okraju alebo ďalej od neho vygenerujeme

Typ úlohy:
Použite na riešenie úlohy algoritmus aglomeratívneho zhlukovania, kde stredy klastrov môžu byť centroid alebo medoid.

Opis riešenia

Aglomeratívne zhlukovanie

1. Zo všetkých vstupných bodov si vytvorím klastre
2. Klastre, ktoré majú najmenšiu vzdialenosť stredov od seba zlúčim do jedného
3. Prerátam si nový stred klastru
4. Body 2-3 opakujem dokiaľ nedosiahnem zadaný počet klastrov
5. Zistím ktoré klastre boli úspešné (priemerná vzdialenosť od stredu<500)

Centroid

Stred klastru, ktorý sa vyráta zo všetkých bodov v danom klastri na základe priemeru súradníc. Tento stred je akoby “nereálny“ -> zväčša sa jedná o nový bod, ktorý ale nepatrí ku kontrolovanej skupine bodov.

Medoid

Stred klastru, ktorý sa vyráta na základe vzdialeností. Pre každý bod v klastri si vyrátame súčet jeho vzdialeností od ostatných bodov a bod s najmenším súčtom sa stane novým stredom. Tento typ stredu je náročnejší na čas výpočtu.

Reprezentácia údajov

Klastre

Jednotlivé klastre si ukladám do triedy.

```
class Cluster:
    def __init__(self):
        self.center=[]
        self.points=[]
```

Matica so vzdialenosťami

Vzdialenosti medzi všetkými klastrami sa nachádzajú v dvojrozmernom poli. Aby som ušetril čas a nenapĺňal zbytočne celú maticu tak používam vrchnú časť nad diagonálou (dáta uložené pod ňou by boli rovnaké, preto ju nevyužívam).

Testovanie

Pre každé testované dáta bolo urobených 10 testov a z nich vyrátaný priemer

Centroid

Počet všetkých bodov	500		1000		1500		2000	
Počet konečných klastrov	Presnosť (%)	Čas (s)	Presnosť (%)	Čas (s)	Presnosť (%)	Čas (s)	Presnosť (%)	Čas (s)
5	24	11,61	30	98	30	332	32	1157
10	71	11,45	77	98,63	71	336,98	65	800
15	96,6	12,34	96,6	100,34	98	338,27	97	935

Medoid

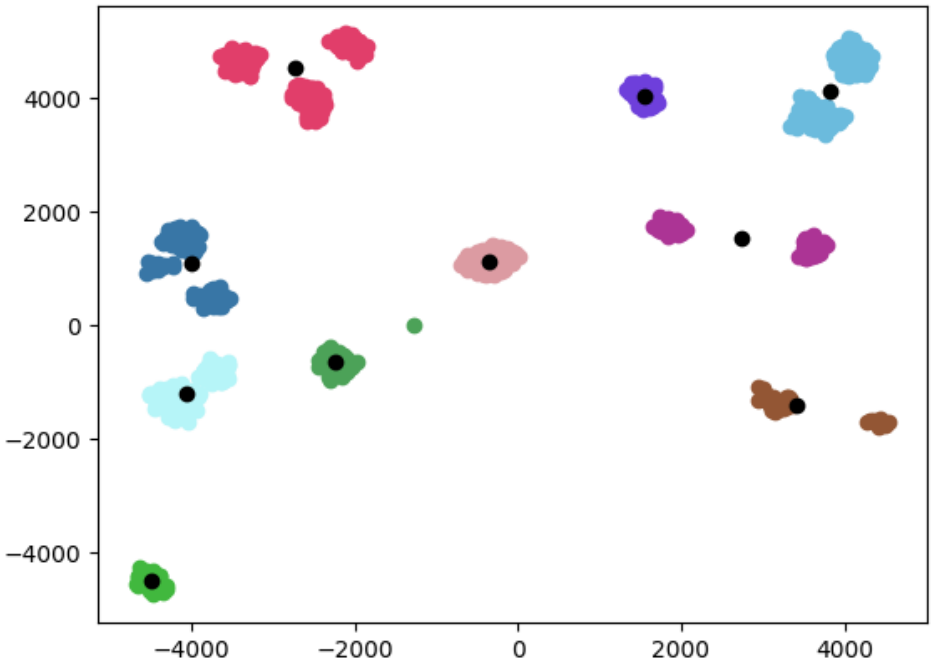
Počet všetkých bodov	500		1000		1500	
Počet konečných klastrov	Presnosť (%)	Čas (s)	Presnosť (%)	Čas (s)	Presnosť (%)	Čas (s)
5	40	14,09	28	114	27	365
10	77	12,86	75	99,55	68	387,51
15	98	11,4	97,3	95	98	331,4

Výstup do konzoly

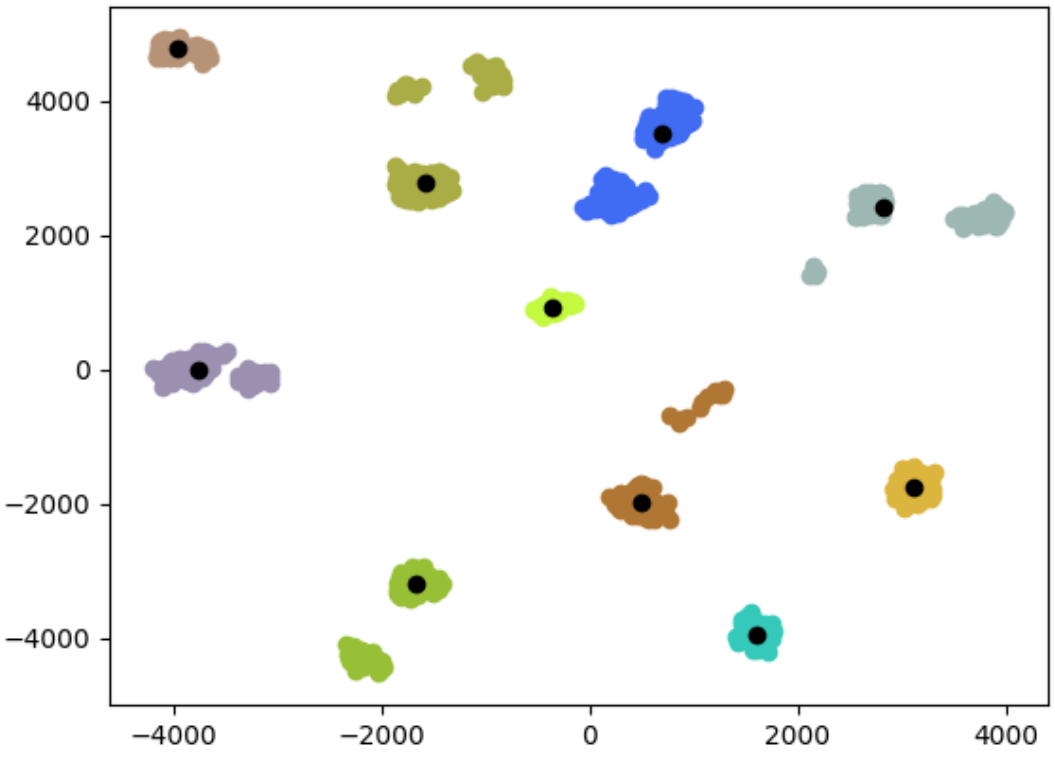
```
Neúspešné klastre:  5
 Úspešné klastre:  5
Celkový čas (bez grafickej časti):  346.78218817710876
```

Grafický výstup

Centroid


4000
2000
0
−2000
−4000
−4000
−2000
0
2000
4000

Medoid


4000
2000
0
−2000
−4000
−4000
−2000
0
2000
4000

Zložitosť

Zložitosť programu je exponenciálna

N^3 -> generujem dvojrozmernú maticu a vkladám/pracujem s jej dátami, čo zároveň znamená že mám aj veľkú pamäťovú zložitosť ale nemusím si po každej úprave klastrov znova prerátavať všetky vzdialenosti.

Zhodnotenie

Program podľa výsledkov funguje správne. Program som skúšal spúšťať aj cez interpreter pypy, avšak výsledné hodnoty boli ešte horšie oproti tým ktoré som dostal spustením programu cez python.

Používateľská príručka

Program je písaný v jazyku Python vo verzii 3.10.2 Po spustení je možné si navoliť vstupné údaje : hlavné body, ostatné body, typ stredu klastru, veľkosť mapy a celkový počet klastrov.

Ukážka zadaných vstupných údajov

```
Zadaj hlavné body: 20
Zadaj ostatné body: 2000
Zadaj 1 pre centroid a 2 pre monoid: 1
Zadaj veľkosť mapy: 5000
Zadaj počet klastrov: 10
```