Slovenská Technická Univerzita v Bratislave
Fakulta informatiky a informačných technológií

Umelá inteligencia

Zadanie 4b – Zhlukovanie

Riešený problém

Zadanie sa týka riešenia problému zhlukovania bodov v 2D priestore. Cieľom je vytvoriť zhlukovače, ktoré efektívne rozdelia body v 2D priestore do **k** zhlukov.

Veľkosť 2D priestoru predstavujú X a Y súradnice v rozmedzí od -5000 do +5000, pričom do priestoru sa umiestni 20020 vygenerovaných bodov. Súradnice prvých 20 bodov sú vygenerované náhodne. Ďalších 20000 bodov sa potom generuje v blízkosti už vygenerovaných bodov, z ktorých je vždy vybratý iba jeden, do ktorého blízkosti sa nový bod vygeneruje.

Po vygenerovaní všetkých 20020 bodov sa spustí prehľadávanie 2D priestoru príslušným zhlukovačom, kedy sa body v priestore rozdelia na požadovaný počet zhlukov. Za úspešný zhlukovač sa považuje taký, ktorého priradené body nemajú vzdialenosť od stredu zhluku väčšiu ako 500, pričom stred môže byť reprezentovaný ťažiskom zhluku ***(centroid)***, alebo bodom, ktorý sa nachádza najbližšie ku ťažisku zhluku ***(medoid).***

Opis riešenia

Úlohou je naprogramovať zhlukovač, ktorý prehľadá všetky vygenerované body v 2D priestore a následne rozdelí tento priestor na **k** zhlukov, pričom je potrebné implementovať verzie zhlukovača s použitím nasledujúcich algoritmov:

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

Priebeh algoritmov

Na začiatku sa vygeneruje do 2D priestoru prvých 20 náhodných bodov. Následne sa vygeneruje ďalších 20000 a to tak, že vždy sa z už doteraz vygenerovaných bodov vyberie náhodne jeden, k jeho X a Y súradnici sa vygeneruje x-ový a y-ový súradnicový ofset z intervalu -100 až +100, k pôvodným súradniciam sa priráta príslušný ofset a výsledkom sú súradnice nového bodu. Pokiaľ vzniknuté súradnice nového bodu presahujú niektorý okraj 2D priestoru, tak sa ofset zredukuje, aby sa novovzniknutý bod vmestil do rozmerov 2D priestoru. Po vygenerovaní všetkých 20020 bodov sa spustí príslušný zhlukovač.

K-means zhlukovanie (centroid/medoid)

Z vygenerovaných 20020 bodov sa náhodne vyberie **k** bodov, ktoré budú predstavovať jednotlivé zhluky, pričom tento prvý bod reprezentuje začiatočný centroid a medoid zhluku.

Následne sa v cykle prehľadávajú všetky ostatné body v 2D priestore, ktoré ešte nie sú priradené do žiadneho zhluku. Pre každý bod sa zo všetkých zhlukov vyberá najviac vyhovujúci zhluk, do ktorého bude prehľadávaný bod následne priradený.

Najviac vyhovujúcim zhlukom je taký, ktorého vzdialenosť od jeho centra (centroid/medoid) k prehľadávanému bodu je najmenšia. Po priradení príslušného bodu do najviac vyhovujúceho zhluku sa vyberie ďalší bod a cyklus sa opakuje.

Po prehľadaní všetkých bodov sa pre každý zhluk vypočíta jeho nové centrum (centroid/ medoid, všetky jeho priradené body sa odstránia a znovu sa spustí celý cyklus prehľadávania a priraďovania bodov. To sa opakuje až do vtedy, pokiaľ sa novovzniknuté centrum nelíši od starého, alebo sa líši len minimálne. Vtedy cyklus končí a jednotlivé zhluky sú finálne.

Aglomeratívne zhlukovanie (centroid)

Každý z vygenerovaných 20020 bodov na začiatku predstavuje samostatný zhluk, pričom tento prvý bod reprezentuje začiatočný centroid a medoid zhluku.

Následne sa v cykle hľadajú také 2 zhluky, ktoré sú od seba najmenej vzdialené. Po ich nájdení sa zlúčia do jedného spoločného zhluku a to tak, že body z jedného zhluku sa priradia druhému zhluku, vypočíta sa nové spoločného centrum zhluku a prázdny zhluk sa odstráni. Následne sa cyklus zlučovania zhlukov opakuje, až pokiaľ neostane iba požadovaných **k** zhlukov. Vtedy cyklus končí a jednotlivé zhluky sú finálne.

Divízne zhlukovanie (centroid)

Na začiatku sa body v 2D priestore rozdelia do dvoch zhlukov pomocou K-means algoritmu. Následne sa v cykle vždy vyberá z už vzniknutých zhlukov ten, ktorý má najhoršie ohodnotenie – najhoršia priemerná vzdialenosť od centra (centroid). Ten sa potom rozdelí na ďalšie dva zhluky pomocou K-means algoritmu. Cyklus sa opakuje do vtedy, pokiaľ nevzniklo požadovaných **k** zhlukov.

Reprezentácia údajov

Reprezentácia priestoru – trieda **Space**

```
public class Space {
    public ArrayList<Point> points;
    public boolean[][] spaceMap;
    public int spaceMapOffset;
    public Random random;
```

Reprezentácia bodu – trieda **Point**

```
public class Point {
    public int posX;
    public int posY;
```

Reprezentácia zhluku – trieda **Cluster**

```
public class Cluster {
    public Point centroid;
    public Point medoid;
    public ArrayList<Point> points;
    public int sumX;
    public int sumY;
```

Reprezentácia kMeans zhlukovania – trieda **KMeans**

```
public class KMeans {
    public int k;
    public ArrayList<Point> points;
    public ArrayList<Cluster> clusters;
    public Random random = new Random();
    public final double maxDistance = Integer.MAX_VALUE;
```

Reprezentácia aglomeratívneho zhlukovania– trieda **Agglomeration**

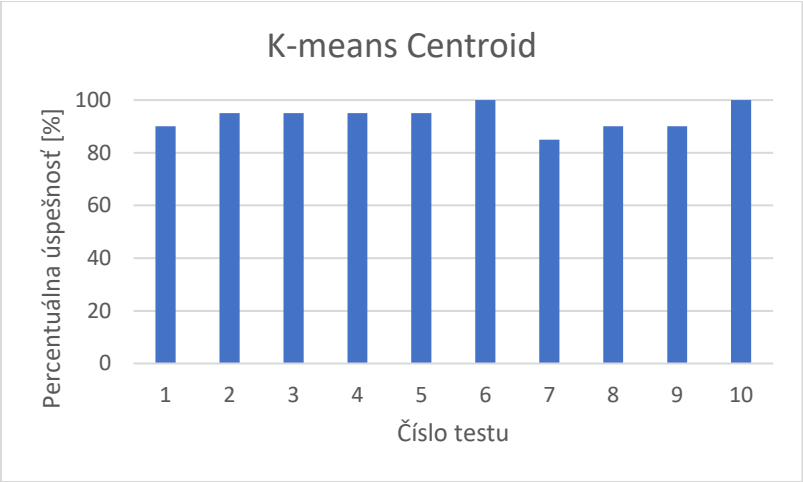
```
public class Agglomeration {
    public int k;
    public ArrayList<Point> points;
    public ArrayList<Cluster> clusters;
    public double[][] distanceMap;
    public final double maxDistance = Integer.MAX_VALUE;
    public Map<Integer, Cluster> dictionary;
    public ArrayList<Integer> list;
```

Reprezentácia divízneho zhlukovania – trieda **Division**

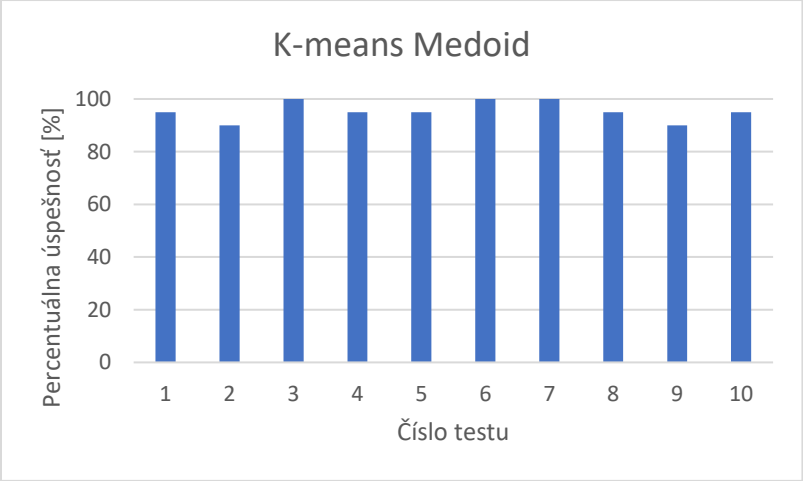
```
public class Division {
    public int k;
    public ArrayList<Point> points;
    public ArrayList<Cluster> clusters;
```

Testovanie (pre rovnaké rozmiestnenie – 20020 bodov a 20 klastrov)

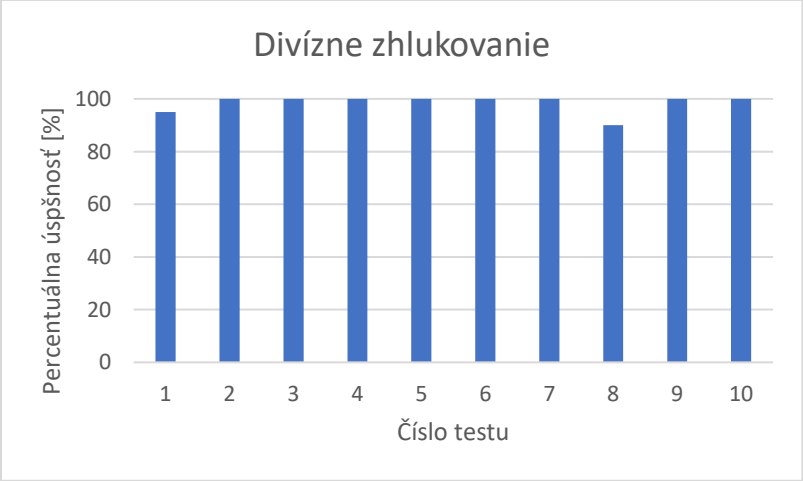
K-means Centroid (Priemerný čas **41ms**, akceptovanie **2/10**)



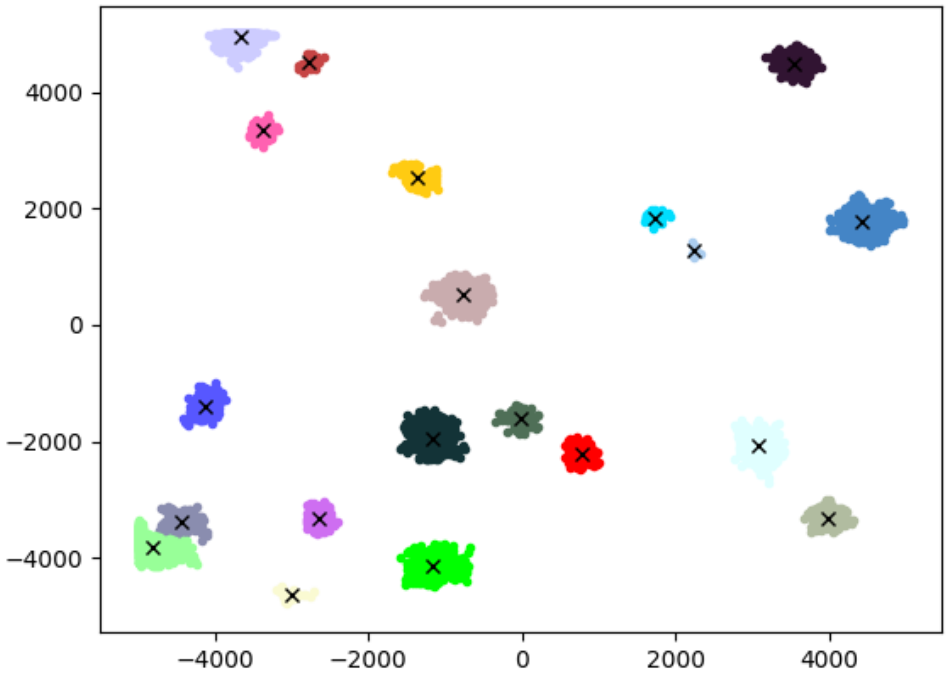
K-means Medoid (Priemerný čas **34ms**, akceptovanie **3/10**)



Divízne zhlukovanie (Priemerný čas **14ms**, akceptovanie **8/10**)



Aglomeratívne zhlukovanie (iba jeden beh, čas **145min**, akceptovanie **1/1**)



Všeobecné informácie

Program:

- Program bol vyhotovený v jazyku Java (SDK 16)
- Použité bolo IDE IntelliJ IDEA Ultimate 2021.2

Prerekvizity:

- **Python** vo verzii 3+
- Python knižnica **matplotlib**
- Java knižnica **matplotlib4j** (https://github.com/sh0nk/matplotlib4j)

Používateľské rozhranie:

- CLI navigácia

```
(0) Exit
(1) K-means - centroid
(2) K-means - medoid
(3) Agglomerative - centroid
(4) Divisive - centroid
Enter your choice:
```