ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE FAKULTA STAVEBNÍ KATEDRA GEOMATIKY					
Název předmětu: **Geoinformatika**					
Úloha: U4	**Název úlohy:** Clusterizační algoritmy				
Akademický rok: 2024/2025	**Semestr:** zimní	**Studijní skupina:** 102C	**Vypracoval:** Michal Kovář Filip Roučka Magdaléna Soukupová	**Datum:** 6. 12. 2024	**Klasifikace:**

1 Zadání

1. **Generování bodů/klastrů**: Vygenerujte alespoň 3 množiny bodů/klastry (např. funkce randn) ve 2D.
2. **Implementace algoritmu k-means**: Vytvořte vlastní implementaci algoritmu shlukování k-means.
3. **Porovnání s funkcí kmeans**: Porovnejte svůj výsledek s výsledkem shlukování nad totožnou množinou s využitím funkce kmeans. Případné rozdíly komentujte.

2 Bonus

- *Rozšíření řešení shlukování k-means do n-dimenzionálního prostoru.*
- *Vlastní implementace hierarchického shlukování (Hierarchical Clustering).*
- *Vlastní implementace DBSCAN (Density-Based Spatial Clustering of Applications with Noise).*

3 Popis problému

Shlukování je metoda neřízené klasifikace, která se používá k identifikaci podobných skupin dat. Cílem je rozdělit data do shluků, přičemž data v každém shluku jsou si co nejvíce podobná a shluky navzájem jsou co nejvíce odlišné. Shlukování se používá například v DPZ pro neříznou klasifikaci satelitních snímků.[1]

3.1 Dílčí kroky shlukové analýzy

Shlukování zahrnuje několik klíčových kroků. Nejprve je třeba zjistit, zda jsou data vhodná pro shlukovou analýzu. Pokud data nemají tendenci vytvářet shluky v daném příznakovém prostoru, nebude shluková analýza účinná. Z tohoto důvodu je důležité provést následující kroky:

- **Výběr příznaků**: Je nutné vybrat relevantní příznaky a minimalizovat redundanci nebo korelaci mezi nimi.
- **Míry podobnosti/rozdílnosti**: Určení, jak měřit "blízkost"nebo "rozdílnost"mezi jednotlivými body.
- **Rozhodovací kritérium**: Výběr metody pro optimalizaci výsledků, např. pomocí nákladové funkce.
- **Výběr algoritmu shlukování**: Volí se na základě charakteristik dat.
- **Validace**: Ověření kvality výsledků.
- **Interpretace výsledků**: Posledním krokem je interpretace výsledků získaných ze shlukování.

3.2 Přístupy ke shlukové analýze

- **Sekvenční přístup**: Tento přístup nevyžaduje předem stanovený počet shluků a algoritmus postupně přiřazuje data k shlukům na základě definovaných parametrů, jako je prahová hodnota vzdálenosti.
- **Hierarchický přístup**: Algoritmus spojuje (Aglomerativní hierarchické algoritmy) nebo dělí (Dělící hierarchické algoritmy) shluky na základě podobnosti mezi body. Vzniká struktura podobná binárnímu strmu.
- **Optimalizace nákladové funkce**: Založen na optimalizaci nákladové funkce J (funkce vektorů datové sady X) parametrizované neznámým vektorem Θ.

3.3 Běžně používané algoritmy

- **K-means**: Algoritmus, který přiřazuje data k centroidům shluků, které se v každé iteraci přepočítávají. Je citlivý na šum v datech a počáteční inicializaci centroidů.
- **Hierarchické shlukování**: Tento přístup postupně spojuje nebo dělí shluky na základě jejich podobnosti. Je vhodný pro analýzu hierarchických vztahů mezi daty. Výsledky lze vizualizovat jako dendrogram, což umožňuje snadno pochopit strukturu dat.
- **DBSCAN**: Algoritmus identifikuje husté oblasti jako shluky a řídké oblasti ignoruje šum. Vstupní parametry, jako hustota bodů a okolí ε, určují, které body patří do shluků, a to bez nutnosti předem definovat počet shluků.
- **Fuzzy shlukování**: Tento algoritmus umožňuje, aby každý bod patřil do více shluků s různou mírou příslušnosti, což je užitečné pro data, kde jsou hranice mezi shluky nejednoznačné nebo rozmazané.[2]

4 Popis metod

Pro výpočet shlukování byla vytvořena třída `Clustering`, která obsahuje metody pro různé algoritmy shlukování: k-means (`kmeans`), hierarchické shlukování (`hierar`) a DBSCAN (`dbscan`).

4.1 K-means

K-means přiřazuje body k nejbližším centroidům, probíhá v několika iteracích, kdy se na základě aktuálních poloh centroidů přiřazují body k jednotlivým shlukům a následně se centroidy přepočítávají. Tento proces pokračuje, dokud se pozice centroidů stabilizují nebo dokud není dosaženo maximálního počtu iterací. Popis výpočtu k-means je uveden v následujícím pseudokódu:

Metoda `kmeans`

Vstup: M – Matice bodů, k – Počet klastrů, max_iter – Počet iterací, PS – Tolerance konvergence
Výstup: S – Pozice centroidů, L – Přiřazení bodů ke klastrům
$Max, Min \leftarrow \max(M), \min(M)$ // Určení rozsahu dat
$S \leftarrow Min + (Max - Min) \cdot \text{rand}(k, \text{size}(M, 2))$ // Inicializace centroidů
$N \leftarrow 0$, $N\_max \leftarrow max\_iter$
while $N < N\_max$ **do**
  $D \leftarrow \text{pdist2}(M, S)$ // Výpočet vzdáleností
  $L \leftarrow \text{argmin}(D, \text{axis} = 2)$ // Přiřazení bodů
  $S\_new \leftarrow \text{zeros}(k, \text{size}(M, 2))$
  for $j \leftarrow 1$ **to** k **do**
    $cluster\_points \leftarrow M[L == j, :]$
    if isempty($cluster\_points$) **then**
      $S\_new[j, :] \leftarrow Min + (Max - Min) \cdot \text{rand}(1, \text{size}(M, 2))$ // Inicializace nového centroidu
    else
      $S\_new[j, :] \leftarrow \text{mean}(cluster\_points, \text{axis} = 1)$ // Výpočet nového centroidu
    end if
  end for
  $diff \leftarrow \text{norm}(S\_new - S)$ // Kontrola konvergence
  if $diff < PS$ **then**
    break
  end if
  $S \leftarrow S\_new$, $N \leftarrow N + 1$
end while
return S, L

4.2 Hierarchické shlukování

Hierarchické shlukování je metoda, která vytváří hierarchii shluků tím, že postupně spojuje (aglomerační přístup) data na základě vzdálenosti mezi jednotlivými body. Popis výpočtu hierarchického shlukování je uveden v následujícím pseudokódu:

Metoda `hierar`

Vstup: M – Matice bodů, k – Počet klastrů
Výstup: *clusters* – Seznam shluků, kde každý shluk obsahuje indexy bodů v daném shluku
$distance\_matrix \leftarrow \text{pdist2}(M, M)$ // Výpočet matice vzdáleností
$n \leftarrow \text{size}(M, 1)$ // Počet bodů
$clusters \leftarrow \text{num2cell}(1 : n)$ // Inicializace klastrů, každý bod je ve svém vlastním shluku
for $i \leftarrow 1$ **to** $n + 1 - k$ **do**
  $min\_val \leftarrow \min(distance\_matrix(:))$ // Minimální hodnota v matici
  $[row\_indices, col\_indices] \leftarrow \text{find}(distance\_matrix == min\_val)$ // Indexy min hodnot
  $valid\_pairs \leftarrow (row\_indices \neq col\_indices)$ // Filtrace párů, kde indexy jsou různé
  $row\_indices \leftarrow row\_indices(valid\_pairs)$, $col\_indices \leftarrow col\_indices(valid\_pairs)$
  for $j \leftarrow 1$ **to** length(row_indices) **do**
    $row \leftarrow row\_indices(j)$, $col \leftarrow col\_indices(j)$
    $clusters[row] \leftarrow clusters[row] \cup clusters[col]$ // Spojení shluků
    $clusters[col] \leftarrow []$ // Vyprázdnění sloučeného shluku
  end for
  $distance\_matrix \leftarrow \text{inf}(\text{length}(clusters))$ // Resetování matice
  for $m \leftarrow 1$ **to** length(clusters) **do**
    for $n \leftarrow 1$ **to** length(clusters) **do**
      if $m \neq n$ **and** $clusters[m] \neq []$ **and** $clusters[n] \neq []$ **then**
        $dist \leftarrow \min(\text{pdist2}(M(clusters[m], :), M(clusters[n], :)), [], \text{all})$
        $distance\_matrix[m, n] \leftarrow dist$
        $distance\_matrix[n, m] \leftarrow dist$
      end if
    end for
  end for
end for
$empty\_indices \leftarrow cellfun(\texttt{'isempty'}, clusters)$ // Hledání prázdných shluků
$clusters \leftarrow clusters(\neg empty\_indices)$ // Výběr neprázdných shluků
return *clusters*

4.3 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) se zaměřuje na identifikaci hustých oblastí v prostoru a ignoruje šum (tj. body, které nepatří k žádnému shluku). Tento algoritmus nevyžaduje stanovení počtu shluků předem. Popis výpočtu DBSCAN je uveden v následujícím pseudokódu:

Metoda `dbscan`

Vstup: M – Matice bodů, *epsilon* – Maximální vzdálenost mezi dvěma body pro označení sousedů, *minPts* – Minimální počet bodů pro označení husté oblasti (jádro)
Výstup: *clusters* – Seznam shluků, kde každý shluk obsahuje indexy bodů v daném shluku
$n \leftarrow \text{size}(M, 1)$ // Počet bodů
$labels \leftarrow \text{zeros}(n, 1)$ // Inicializace popisků klastrů
$cluster\_id \leftarrow 0$ // Počáteční ID pro shluky
$clusters \leftarrow \text{cell}(n, 1)$ // Inicializace seznamu pro shluky
for $i \leftarrow 1$ **to** n **do**
  if $labels[i] \neq 0$ **then**
    continue // Pokud je bod již navštívený, přeskoč ho
  end if
  $neighbors \leftarrow \text{find}(\text{pdist2}(M(i,:), M) \leq \text{epsilon})$ // Hledání sousedů bodu
  if $\text{numel}(neighbors) < minPts$ **then**
    $labels[i] \leftarrow -1$ // Označení bodu jako šumu
    continue
  end if
  $cluster\_id \leftarrow cluster\_id + 1$ // Vytvoření nového shluku
  $labels[i] \leftarrow cluster\_id$ // Přiřazení bodu k novému shluku
  $current\_cluster \leftarrow neighbors$ // Aktuální seznam sousedních bodů
  $k \leftarrow 1$
  while $k \leq \text{numel}(current\_cluster)$ **do**
    $j \leftarrow current\_cluster(k)$
    if $labels[j] = -1$ **then**
      $labels[j] \leftarrow cluster\_id$ // Změna šumu na okrajový bod
    end if
    if $labels[j] = 0$ **then**
      $labels[j] \leftarrow cluster\_id$ // Přiřazení bodu k aktuálnímu shluku
      $new\_neighbors \leftarrow \text{find}(\text{pdist2}(M(j,:), M) \leq \text{epsilon})$
      if $\text{numel}(new\_neighbors) \geq minPts$ **then**
        $current\_cluster \leftarrow \text{union}(current\_cluster, new\_neighbors)$ // Nový sousedé do seznamu
      end if
    end if
    $k \leftarrow k + 1$
  end while
  $clusters[cluster\_id] \leftarrow current\_cluster$ // Uložení bodů do aktuálního shluku
end for
$empty\_indices \leftarrow cellfun(\texttt{'isempty'}, clusters)$ // Hledání prázdných shluků
$clusters \leftarrow clusters(\neg empty\_indices)$ // Výběr neprázdných shluků
return *clusters*

5 Postup

Úloha byla zpracována v softwaru MATLAB.

5.1 Vstupy a nastavení

Na začátku skriptu je požadováno, aby uživatel zadal dimenzi prostoru v němž bude shlukování probíhat. Tento vstup je validován, aby bylo zajištěno, že uživatel zadá platné číslo dimenze. Pokud je zvolen prostor o dvou dimenzích, skript umožňuje dále dvě možnosti: buď uživatel zadá body manuálně kliknutím do grafu, nebo jsou body automaticky generovány. Pro jiné dimenze jsou body vždy generovány automaticky. Parametry pro různé, shlukovací algoritmy jsou vloženy přímo do skriptu v němž je lze měnit.

5.2 Metody shlukování

Pro shlukování dat jsou použity následující metody:

- **K-means**: K-means je použit jak jako vlastní implementace, tak jako vestavěná funkce v MATLABu.
- **Hierarchické shlukování**: Použita byla vlastní implementace.
- **DBSCAN**: Použita byla vlastní implementace.

5.3 Vizualizace

Pro vizualizaci výsledků shlukování jsou body zobrazeny do grafů v závislosti na dimenzi dat. V případě 1D dat jsou zobrazeny na jedné ose, pro 2D jsou body zobrazeny v rovinném grafu, pro 3D jsou body vykresleny ve 3D prostoru. Pro 4D jsou zobrazeny čtyři řezy 4D prostoru jako čtyři 3D grafy. Pro vyšší dimenze nejsou grafy vkreslovány. V grafech jsou jednotlivé shluky barevně odlišeny. Pro K-means jsou také zobrazeny centroidy každého shluku.

6 Vstupní data

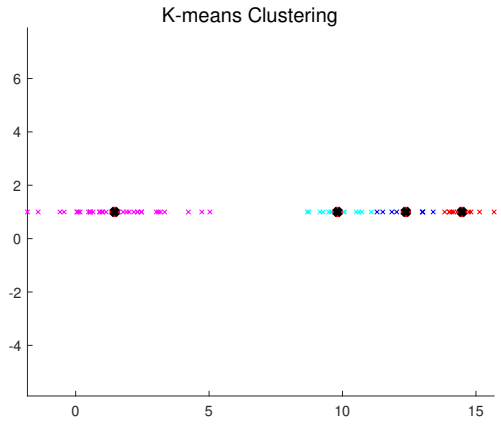
Vstupní data se generují nebo zadávají uživatelem následujícím způsobem:

- Uživatel nejprve zadá počet rozměrů dat (`n_dim`). Pokud uživatel zadá neplatný vstup, skript požaduje opětovné zadání, dokud nebude vstup správný.
- Pokud `n_dim` $= 2$, uživatel má možnost zadat data interaktivně kliknutím na graf, nebo je nechat náhodně vygenerovat.
- Pokud `n_dim` $\neq 2$, data jsou vždy generována náhodně.

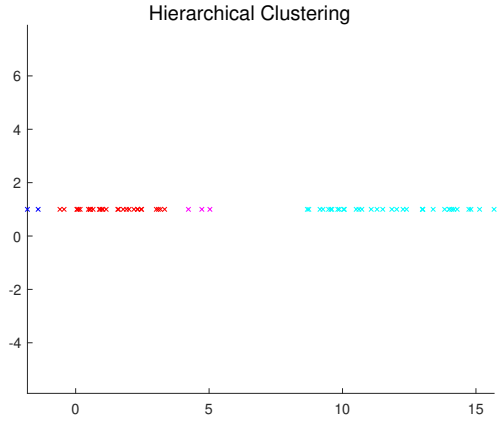
7 Výstupní data

Výstupní data obsahují body obarvené podle clusterů, ke kterým patří. U algoritmu K-means byly navíc černě vykresleny centroidy vypočítané vlastní implementací algoritmu a červeně centroidy určené integrovanou funkcí `kmeans` v MATLABu. Tato vizualizace umožňuje porovnat přesnost a odlišnosti obou přístupů.

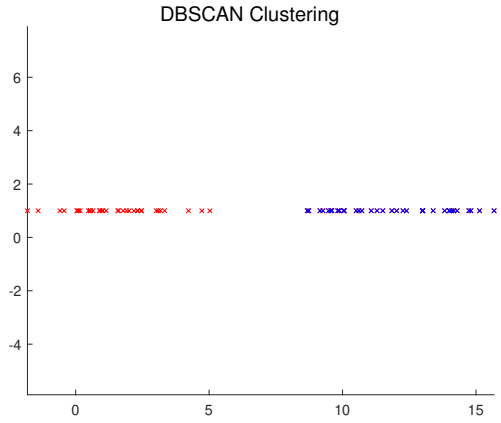
7.1 Jednorozměrná data



Obrázek 1: K-means pro jednorozměrná data.

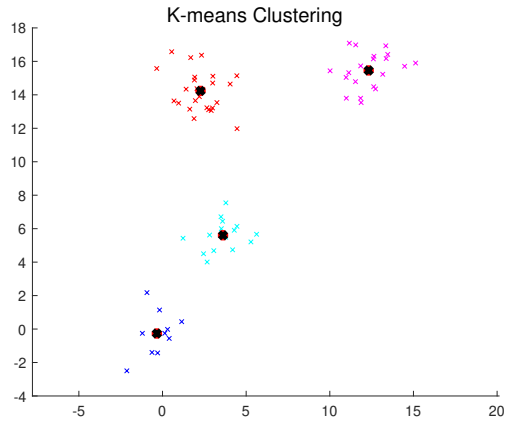


Obrázek 2: Hierarchické shlukování pro jednorozměrná data.

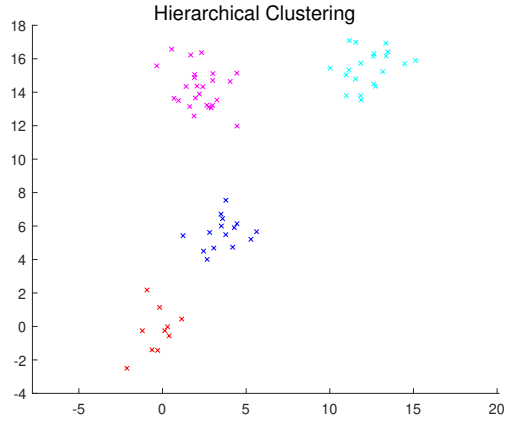


Obrázek 3: DBSCAN pro jednorozměrná data.

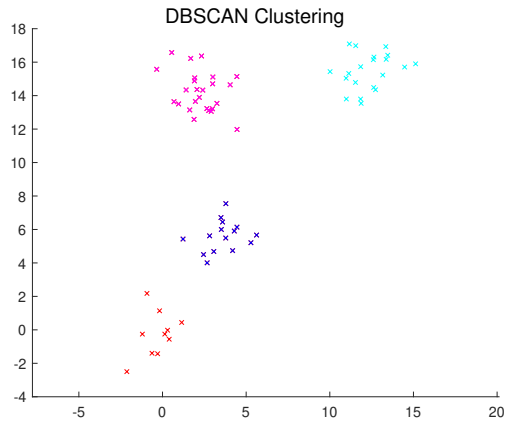
7.2 Dvourozměrná data



Obrázek 4: K-means pro dvojrozměrná data.

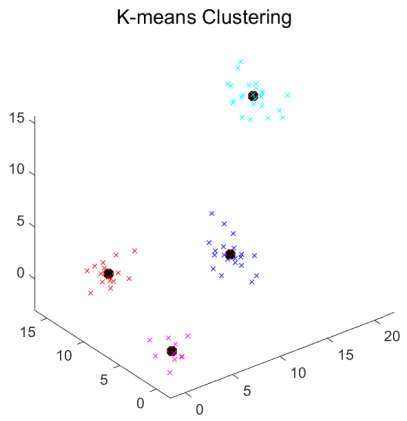


Obrázek 5: Hierarchické shlukování pro dvojrozměrná data.

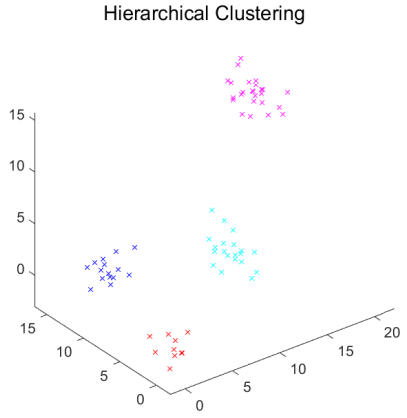


Obrázek 6: DBSCAN pro dvojrozměrná data.

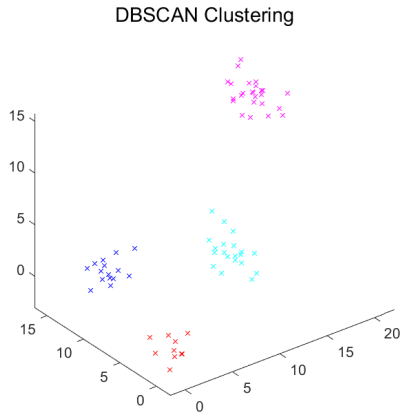
7.3 Trojrozměrná data



Obrázek 7: K-means pro trojrozměrná data.

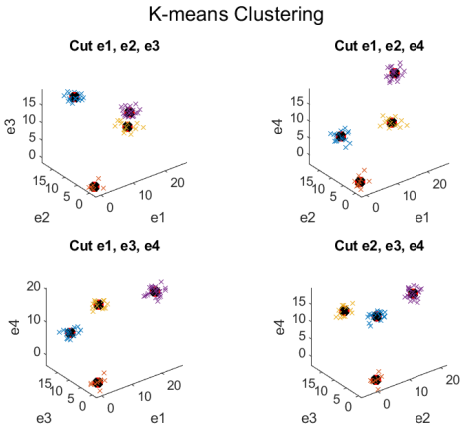


Obrázek 8: Hierarchické shlukování pro trojrozměrná data.

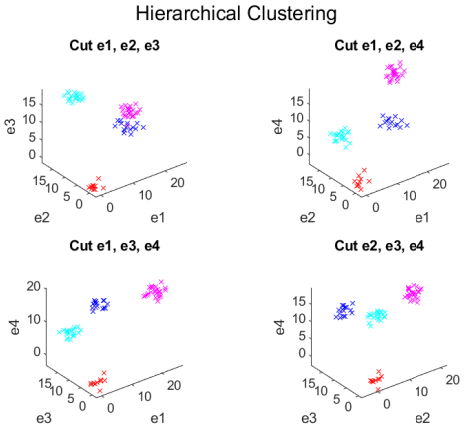


Obrázek 9: DBSCAN pro trojrozměrná data.

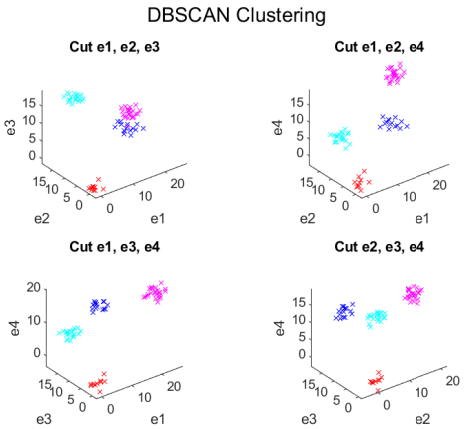
7.4 Čtyřrozměrná data



Obrázek 10: K-means pro čtyřrozměrná data.



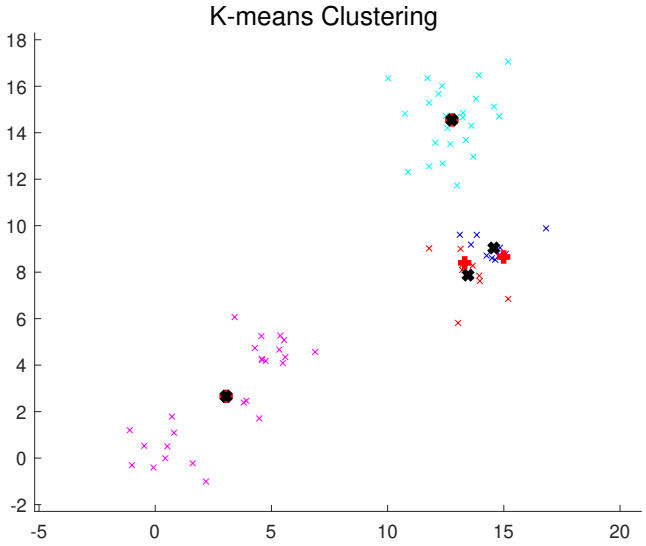
Obrázek 11: Hierarchické shlukování pro čtyřrozměrná data.



Obrázek 12: DBSCAN pro čtyřrozměrná data.

7.5 Rozdílný výsledek K-means

V některých případech se výsledky vlastní implementace K-means a implementace v MATLABu liší. Tento rozdíl je pravděpodobně způsoben náhodnou inicializací centroidů při startu algoritmu.



Obrázek 13: Rozdíly mezi vlastní implementací a MATLAB implementací K-means.

8 Závěr

Byl vyhotoven skript v MATLABu, který provádí shlukování dat pomocí různých algoritmů, včetně K-means, hierarchického shlukování a DBSCAN. Algoritmus K-means byl porovnán s vestavěnou funkcí v MATLABu. Bylo zjištěno, že v některých případech se výsledky shlukování liší, v jiných se naopak shodují. To může být způsobeno náhodnou inicializací počátečních centroidů, která ovlivňuje výsledky K-means algoritmu.

8.1 Možné oblasti pro vylepšení

- **Další metody shlukování**: Lze přidat další metody clusterizace, jako je například ISODATA nebo fuzzy shlukování.
- **Automatizace výběru parametrů**: Parametry pro shlukování, jako je počet shluků pro K-means nebo hodnoty epsilon a minPts pro DBSCAN, by mohli být nastaveny automaticky na základě analýzy dat před samotným zpracováním.
- **Vizualizace pro vyšší dimenze**: Pro dimenze vyšší než 3 by bylo vhodné implementovat pokročilé metody vizualizace více dimenzionálních dat, jako je například metoda hlavních komponent (PCA) nebo t-SNE, které umožňují redukci dimenzí pro lepší zobrazení dat.
- **Zrychlení algoritmů**: Optimalizace algoritmů, by mohla urychlit zpracování velkých souborů dat.

Odkazy

[1] Markéta Potůčková. *GEOINFORMATIKA: Shluková analýza, Algoritmy neřízené klasifikace*. Prezentace k přednášce. Katedra aplikované geoinformatiky a kartografie, Přf UK, 2024. URL: `marketa.potuckova@natur.cuni.cz`.

[2] K. Koutroumbas a S. Theodoridis. *Pattern Recognition*. 2008. URL: `https://github.com/free-educa/books/issues/27` (cit. 05. 12. 2024).