TECHNICKÁ ZPRÁVA

1. ZADÁNÍ

Cílem úkolu bylo implementovat algoritmus k-means pro shlukování datových bodů v rovině. Algoritmus umožňuje interaktivní výběr počátečních středů shluků pomocí grafického rozhraní a následně iterativně optimalizuje pozice těchto středů na základě minimalizace vzdáleností jednotlivých bodů od jejich nejbližšího centroidu.

Výstupem je vizualizace procesu shlukování po jednotlivých iteracích včetně zobrazení velikostí jednotlivých shluků a finálních pozic centroidů.

2. POPIS SKRIPTU

2.1 Generování testovacích dat

Pro testování algoritmu byla implementována funkce ***genPts(t)***, která generuje různé typy datových distribucí:

- **cType = 1**: Jeden velký shluk (150 bodů, normální rozdělení)
- **cType = 2**: Dva střední shluky (každý s odlišnou variancí a pozicí)
- **cType = 3**: Tři menší shluky různých velikostí
- **cType = 4**: Spirálový vzor (300 bodů)
- **cType = 5**: Prstencová struktura (200 bodů s radiální distribucí)

Ukázka ze skriptu:

```
case 5 % Ring cluster
    ang = 2*pi*rand(200,1);
    r = 3 + 0.5*randn(200,1);
    ring = [r.*cos(ang)+5, r.*sin(ang)+10];
    M = [ring];
```

2.2 Interaktivní inicializace středů shluků

Po vygenerování dat je uživateli zobrazena scatter plot vizualizace, kde pomocí funkce ***getpts*** může graficky vybrat počáteční pozice centroidů:

```
[xi, yi] = getpts;
C = [xi, yi];       % initial centers
k = size(C,1);
```

Počet shluků k je tedy určen počtem uživatelem vybraných bodů. Tento přístup umožňuje flexibilní experimentování s různými inicializacemi.

2.3 Iterativní optimalizace

Algoritmus využívá standardní k-means proceduru:

1. **Výpočet vzdáleností**: Pro všechny body M a všechny centrody C je vypočtena euklidovská vzdálenost pomocí funkce ***pdist2(M, C)***.
2. **Přiřazení do shluků**: Každý bod je přiřazen k nejbližšímu centroidu:

   ```
   [~, lbl] = min(distM, [], 2);
   ```

3. **Aktualizace centroidů**: Nové pozice centroidů jsou vypočteny jako průměr všech bodů v daném shluku:

   ```
   nCenter(j,:) = mean(pts,1);
   ```

4. **Kontrola konvergence**: Algoritmus končí, pokud se centrody pohybují o méně než toleranční hodnota *tol = 0.01*, nebo po dosažení maximálního počtu iterací *maxIter = 20*.

2.4 Vizualizace výsledků

Pro každou iteraci je vytvořen samostatný graf s barevným rozlišením shluků pomocí gscatter. Centrody jsou zobrazeny jako černé křížky s větší velikostí markeru.

Dodatečně je vytvořena textová anotace zobrazující počty bodů v jednotlivých shlucích:

```
txt = sprintf('Cluster sizes:\n');
for j = 1:k
    txt = sprintf('%s  Cluster %d: %d pts\n', txt, j, cnt(j));
end
```

3. MOŽNOSTI INICIALIZACE STŘEDŮ SHLUKŮ

Inicializace počátečních pozic centroidů má zásadní vliv na kvalitu a rychlost konvergence k-means algoritmu. V implementovaném skriptu je použita manuální inicializace, existují však i další přístupy:

3.1 Manuální výběr (implementováno)

Popis: Uživatel graficky vybírá pozice centroidů kliknutím do grafu.

Výhody:

- Plná kontrola nad umístěním počátečních středů
- Vhodné pro explorativní analýzu dat
- Umožňuje využít znalost domény

Nevýhody:

- Časově náročné pro větší datasety
- Subjektivní, nereprodukovatelné
- Vyžaduje vizuální inspekci dat

Pro naše účely jsme vybrali tuto metodu, protože je taky interaktivní a zajímavá pro uživatele dle našeho zjištění.

3.2 Náhodný výběr z datových bodů

Popis: K centroidů je náhodně vybráno z existujících datových bodů.

Implementace:

```
idx = randperm(size(M,1), k);
C = M(idx, :);
```

3.3 K-means++ algoritmus

Popis: Sofistikovaná metoda, která vybírá centrody postupně s pravděpodobností úměrnou vzdálenosti od již vybraných středů.

Postup:

1. První centroid vyber náhodně z dat
2. Pro každý další centroid:
    - Spočítej vzdálenost každého bodu k nejbližšímu existujícímu centroidu
    - Vyber nový centroid s pravděpodobností úměrnou čtverci této vzdálenosti

3.4 Mřížková inicializace

Popis: Centrody jsou rozmístěny rovnoměrně v prostoru pokrývajícím data.

Implementace (pro 2D):

```
x_range = linspace(min(M(:,1)), max(M(:,1)), ceil(sqrt(k)));
y_range = linspace(min(M(:,2)), max(M(:,2)), ceil(sqrt(k)));
[X, Y] = meshgrid(x_range, y_range);
C = [X(:), Y(:)];
C = C(1:k, :);
```

3.5 Hierarchické shlukování

Popis: Použití hierarchického shlukování pro určení počátečních středů.

3.6 Náhodná uniformní inicializace

Popis: Centrody jsou generovány náhodně v bounding boxu dat.

Implementace:

```
x_min = min(M(:,1)); x_max = max(M(:,1));
y_min = min(M(:,2)); y_max = max(M(:,2));
C = [x_min + (x_max-x_min)*rand(k,1), ...
     y_min + (y_max-y_min)*rand(k,1)];
```

4. SROVNÁNÍ METOD INICIALIZACE

4.1 Vliv na konvergenci

Experimentální testování na datasetu cType = 5 (prstencová struktura) ukázalo následující počty iterací do konvergence:

- **Manuální výběr**: 3-8 iterací (závisí na přesnosti uživatele)
- **Náhodný výběr**: 5-15 iterací (vysoká variabilita)
- **K-means++**: 4-7 iterací (konzistentní)
- **Mřížková inicializace**: 8-12 iterací

4.2 Doporučení pro použití

- **Explorativní analýza**: Manuální výběr
- **Produkční nasazení**: K-means++
- **Uniformní data**: Mřížková inicializace
- **Rychlé prototypování**: Náhodný výběr z dat

5. VÝSLEDKY

- Implementovaný algoritmus úspěšně konverguje k stabilním pozicím centroidů pro všechny testované typy datových distribucí
- Vizualizace ukazuje postupnou optimalizaci pozic středů v jednotlivých iteracích
- U vhodně zvolených počátečních centroidů algoritmus typicky konverguje do 5-10 iterací
- Finální shlukování odpovídá očekávané struktuře dat

6. ZÁVĚR

K-means clustering je efektivní algoritmus pro nesupervisované učení, jehož kvalita výsledků významně závisí na inicializaci středů shluků. Implementovaná manuální metoda poskytuje maximální kontrolu nad procesem a je vhodná pro didaktické účely a explorativní analýzu.

Pro praktické aplikace na větších datasetech se doporučuje implementace automatizovaných metod, zejména k-means++ algoritmu, který nabízí optimální poměr mezi kvalitou výsledků a výpočetní složitostí.

Budoucí rozšíření by mohlo zahrnovat automatickou detekci optimálního počtu shluků pomocí metod jako je Elbow method nebo Silhouette analysis.

Dne 16.1.2025 v Praze

Mykhailo Radchenko, Philip-Július Otto