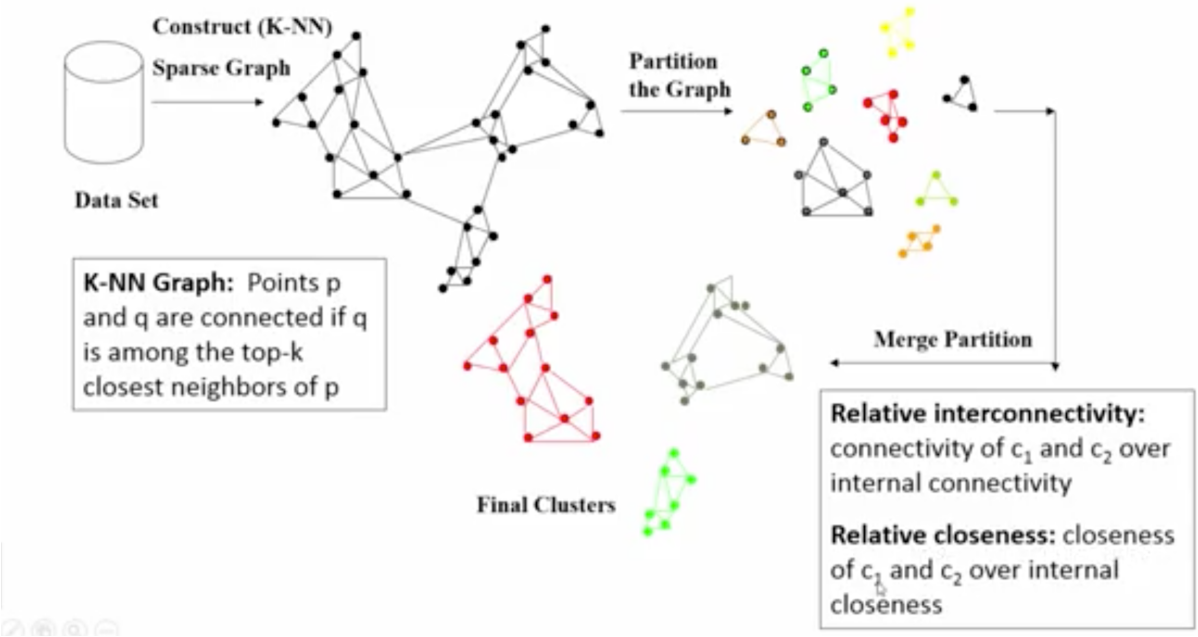
SPDB – dokumentacja wstępna
Projekt kluczowych elementów

Autor: Jacek Janczura
Temat: Implementacja wybranego algorytmu eksploracji danych.
Wybrany algorytm na podstawie artykułu z listy:
CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling [1]
Źródło: https://www-users.cs.umn.edu/~hanxx023/dmclass/chameleon.pdf

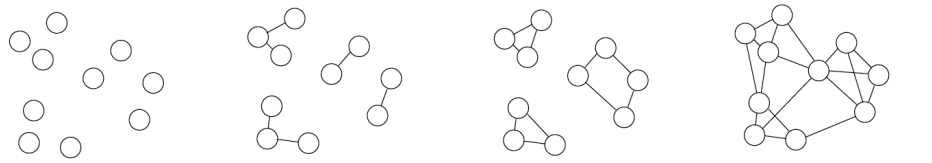
1. Opis algorytmu

Poniższy rysunek przedstawia trzy kroki działania algorytmu, które dokładniej zostały opisane pod Rysunkiem 1. Działanie algorytmu składa się z trzech faz.



Rysunek 1. Prezentacja zasady działania algorytmu CHAMELEON [2]

1. Faza pierwsza
   a. Tworzony jest graf zawierający wszystkie punkty ze zbioru testowego
   b. Zostawienie krawędzi zgodnie z algorytmem KNN
   c. Podział na początkowe klastry poprzez znalezienie składowych spójnych w grafach – przeszukiwanie w głąb.



(a) Original Data in 2D (b) 1-nearest neighbor graph (c) 2-nearest neighbor graph (d) 3-nearest neighbor graph

Rysunek 2. Grafy k najbliższych sąsiadów [1]

2. Faza druga
   a. Dopóki nie osiągnięto początkowej liczby grup następuje podział wierzchołków największego klastra na dwie równoliczne części
      - Wybór największego klastra
      - Podział klastra na dwa równoliczne podgrafy w zależności od rozkładu punktów
3. Dopóki nie osiągnięto określonej na wejściu końcowej liczby k klastrów
   a. Wyliczenie metryk współczynnika powiązania (RI oraz RC) dla wszystkich par klastrów

      EC(Ci) – średnia wartość wagi (1/odległość) między wierzchołkami w grafie

      RI(Ci,Cj) – współczynnik powiązań wewnętrznych w klastrze (Relative Inter-Connectivity)

```
RI = bothClustersEC * (firstClusterPointsNr + secondClusterPointsNr) /
           (secondClusterPointsNr * firstClusterEC + firstClusterPointsNr * secondClusterEC)
```

      RC(Ci,Cj) – współczynnik powiązania dwóch klastrów (Relative Closeness)

```
RC = bothClustersEC * (firstClusterPointsNr + secondClusterPointsNr) /
           (secondClusterPointsNr * firstClusterEC + firstClusterPointsNr * secondClusterEC)
```

   b. Dla każdej pary klastrów wyliczany jest iloczyn RI oraz RC.
   c. Para z najmniejszą wartością iloczynu RI i RC jest złączana
   d. Dodanie nowego klastra do listy klastrów oraz usunięcie starych klastrów

2. Dane

Wybrany został zbiór miast z podziałem administracyjnym na stany - https://simplemaps.com/data/us-cities

Plik z danymi zawiera 4 dane o każdym mieście.

Są to nazwą miasta(**city**), stan(**state_name),** długość (**lng**) i szerokość (**lat**) geograficzna.

Dane w zbiorze us-cities przechowywane są w formacie tekstowym .csv

city	state_name	lat	lng
Navajo Dam	New Mexico	36.7981	-107.7045
Gascon	New Mexico	35.8867	-105.4461
Taos Ski Valley	New Mexico	36.5908	-105.4374
Crownpoint	New Mexico	35.6880	-108.1494
Playas	New Mexico	31.9126	-108.5364
Tse Bonito	New Mexico	35.6525	-109.0361
Pleasant Hill	New Mexico	34.5204	-103.0738

Rysunek 3. Przykład danych zawartych w zbiorze

3. Implementacja

- Dane przechowywane są w tekstowym formacie .CSV
- Algorytm zostanie zaimplementowany w języku Java 8
- Opis najważniejszych metod dostępny jest w JavaDoc dołączonym do sprawozdania oraz w komentarzach do kodu

4. Dane wejściowe

- Plik z danymi
- **k** - Liczba sąsiadów do algorytmu knn – krok 1 algorytmu
- **m** – oczekiwana liczba grup
- **n** – liczba podgrup – krok 2 algorytmu

5. Dane wyjściowe

Metryki oceniające poprawność klasteryzacji.
Grafika prezentująca klasteryzację.

6. Testowanie algorytmu

Testowanie algorytmu będzie polegało na wyliczeniu dokładności (accuracy) oraz czystości (purity) dla pojedynczej grupy jak i dla całości.

Dokładność wyliczana jest jako stosunek liczby punktów poprawnie zgrupowanych(zgodnie z pierwotną przynależnością do stanów) do liczby wszystkich punktów.

Czystość grupowania wyliczana będzie jako liczba różnych stanów w grupie. Założono, że stan zaliczany jest jako obecny w grupie gdy są co najmniej trzy punkty należące do tego stanu.

Badany będzie wpływ liczby najbliższych sąsiadów – pierwsza część algorytmu, liczby podgrup – 2 krok algorytmu na dokładność grupowania i czystość grup.

- **Test 1 – Stany blisko siebie –** Sprawdzenie działania algorytmu dla grup znajdujących się obok siebie

  a. **Zmiana początkowej liczby grup 5 – 15 – 30 – 60 – 120** przy zachowaniu reszty parametrów stałych.

Parametr k: 3
Początkowa liczba grup: 5 - 15
Końcowa liczba grup: 3
liczba punktów: 2152

Rezultaty
Accuracy: 0.78, Purity: 2.0
Cluster: Kansas, Accuracy: 1.0, Purity: 1.0
Cluster: Colorado, Accuracy: 0.95, Purity: 2.0
Cluster: Nebraska, Accuracy: 0.58, Purity: 3.0

Rezultaty : Początkowa liczba grup: 30
Accuracy: 0.82, Purity: 2.0
Cluster: Colorado, Accuracy: 1.0, Purity: 1.0
Cluster: Nebraska, Accuracy: 0.69, Purity: 3.0
Cluster: Kansas, Accuracy: 0.82, Purity: 2.0

Rezultaty : Początkowa liczba grup: 60
Accuracy: 0.85, Purity: 2.0
Cluster: Colorado, Accuracy: 1.0, Purity: 1.0
Cluster: Kansas, Accuracy: 0.89, Purity: 2.0
Cluster: Nebraska, Accuracy: 0.7, Purity: 3.0

Rezultaty : Początkowa liczba grup: 120
Accuracy: 0.83, Purity: 2.0
Cluster: Colorado, Accuracy: 1.0, Purity: 1.0
Cluster: Kansas, Accuracy: 0.99, Purity: 2.0
Cluster: Nebraska, Accuracy: 0.66, Purity: 3.0

Zwiększanie liczby klastrów powoduje że każdy z nich jest coraz mniejszy dlatego gdy liczba klastrów będzie zbyt duża – liczba punktów w każdym z nich będzie niewielka – prowadzić to może do błędnych decyzji łączenia klastrów na podstawie wyliczonych metryk RI i RC.

Z drugiej strony dla zbyt małej liczby klastrów wyliczane metryki mogą być zbyt zgrubne i również może to prowadzić do błędnych decyzji łączenia klastrów.

b. **Zmiana liczby najbliższych sąsiadów** – parametr k - 2-3-4- przy zachowaniu reszty parametrów stałych

Parametr k: 2
Początkowa liczba grup: 5 - 15
Końcowa liczba grup: 3
liczba punktów: 2152

Rezultaty : k=2
Accuracy: 0.63, Purity: 2.33
Cluster: Kansas, Accuracy: 0.54, Purity: 3.0
Cluster: Kansas, Accuracy: 0.49, Purity: 3.0
Cluster: Colorado, Accuracy: 1.0, Purity: 1.0

Rezultaty : k = 3
Accuracy: 0.78, Purity: 2.0
Cluster: Kansas, Accuracy: 1.0, Purity: 1.0
Cluster: Colorado, Accuracy: 0.95, Purity: 2.0
Cluster: Nebraska, Accuracy: 0.58, Purity: 3.0

Rezultaty : k = 4
Accuracy: 0.72, Purity: 2.33
Cluster: Nebraska, Accuracy: 0.54, Purity: 3.0
Cluster: Colorado, Accuracy: 0.96, Purity: 2.0
Cluster: Kansas, Accuracy: 0.72, Purity: 2.0

Rezultaty : k = 6
Accuracy: 0.72, Purity: 2.33
Cluster: Nebraska, Accuracy: 0.54, Purity: 3.0
Cluster: Colorado, Accuracy: 0.96, Purity: 2.0
Cluster: Kansas, Accuracy: 0.72, Purity: 2.0

Przy liczbie najbliższych sąsiadów zmniejszonej do 2, dokładność grupowania oraz ogólna czystość grup została zmniejszona. Brak dominującej grupy ze stanem Nebraska.

Przy wzroście parametru k, a więc zwiększonej liczbie najbliższych sąsiadów spowodowała również nieznaczne obniżenie dokładności grupowania – powstają duże klastry z dużą liczbą punktów

- **Test 2 – Stany blisko siebie, liczba punktów mniej liczna – liczba stanów 4**
  Test ma na celu sprawdzenie jak radzi sobie algorytm z większą liczbą klastrów – 4 które położone są blisko siebie oraz liczność punktów jest ponad dwukrotnie niższa niż w poprzednim teście

  Parametr k: 4
  Początkowa liczba grup: 15
  Końcowa liczba grup: 4
  liczba punktów: 1901

  Resultaty:
  Accuracy: 0.80, Purity: 1.75
  Cluster: Colorado, Accuracy: 1.0, Purity: 1.0
  Cluster: Arizona, Accuracy: 0.99, Purity: 2.0
  Cluster: New Mexico, Accuracy: 0.97, Purity: 2.0
  Cluster: Colorado, Accuracy: 0.57, Purity: 2.0

  Parametr k: 4
  Początkowa liczba grup: 30
  Końcowa liczba grup: 4
  liczba punktów: 1901

  Rezultaty: Początkowa liczba grup: 30
  Accuracy: 0.71, Purity: 2.0
  Cluster: Colorado, Accuracy: 1.0, Purity: 1.0
  Cluster: Arizona, Accuracy: 0.88, Purity: 3.0
  Cluster: Utah, Accuracy: 0.74, Purity: 2.0
  Cluster: New Mexico, Accuracy: 0.55, Purity: 2.0

  Parametr k: 3
  Początkowa liczba grup: 15
  Końcowa liczba grup: 4
  liczba punktów: 1901

  Rezultaty: Parametr k = 3
  Accuracy: 0.66, Purity: 2.75
  Cluster: Arizona, Accuracy: 0.47, Purity: 4.0
  Cluster: Colorado, Accuracy: 0.95, Purity: 2.0
  Cluster: Utah, Accuracy: 1.0, Purity: 1.0
  Cluster: New Mexico, Accuracy: 0.36, Purity: 4.0

  Parametr k: 6
  Początkowa liczba grup: 15
  Końcowa liczba grup: 4
  liczba punktów: 1901

Rezultaty: Parametr k = 6
Accuracy: 0.80, Purity: 1.75
Cluster: Colorado, Accuracy: 1.0, Purity: 1.0
Cluster: Arizona, Accuracy: 0.94, Purity: 2.0
Cluster: Utah, Accuracy: 0.76, Purity: 2.0
Cluster: New Mexico, Accuracy: 0.66, Purity: 2.0

Parametr k: 4
Początkowa liczba grup: 8
Końcowa liczba grup: 4
liczba punktów: 1901

Rezultaty: Początkowa liczba grup = 8
Accuracy: 0.92, Purity: 1.75
Cluster: Colorado, Accuracy: 1.0, Purity: 1.0
Cluster: Arizona, Accuracy: 0.99, Purity: 2.0
Cluster: Utah, Accuracy: 0.74, Purity: 2.0
Cluster: New Mexico, Accuracy: 0.97, Purity: 2.0

Dla większej liczby stanów dokładność grupowania wyniosła 80% co jest akceptowalnym wynikiem i porównywalnym z wynikami testu dla trzech stanów.

Zmniejszenie liczby najbliższych sąsiadów zgodnie z wcześniejszymi wynikami i biorąc pod uwagę zmniejszoną liczbę punktów, mocno popsuło grupowanie znacznie zmniejszając dokładność. Spowodowane to zostało utworzeniem małolicznych niewielkich grafów po pierwszej fazie algorytmu.

Zgodnie z artykułem najlepsze wyniki osiągnięto dla liczby najbliższych sąsiadów w algorytmie knn równej liczbie wyjściowych przewidywanych grup.

Zwiększenie początkowej liczby grup również zgodnie z przewidywaniami spowodowało problem w trzeciej fazie algorytmu i z uwagi na małą liczbę punktów w klastrze błędne łączenie klastrów.

Zmniejszenie początkowej liczby grup do 8 spowodowało zwiększenie liczby punktów w każdym klastrze oraz lepsze wyliczenie metryk współczynników powiązania. Spowodowało to wzrost dokładności grupowania do 98%

- **Test 3 – Stany daleko od siebie –** Sprawdzenie działania grupowania dla odległych grup.

  Dla oddalonych od siebie grup z widocznym wyraźnym podziałem algorytm radzi sobie bardzo dobrze:
  Przykładowe parametry początkowe

  Parametr k: 3
  Początkowa liczba grup: 15
  Końcowa liczba grup: 3
  liczba punktów: 4543

  Rezultaty:
  Ogólne: Accuracy: 1, Purity: 1
  Cluster: Colorado, Accuracy: 1, Purity: 1
  Cluster: Utah, Accuracy: 1, Purity: 1
  Cluster: New Mexico, Accuracy: 1, Purity: 1

  Rezultaty : Początkowa liczba grup =5
  Accuracy: 0.985, Purity: 1.67
  Cluster: Texas, Accuracy: 1.0, Purity: 1.0
  Cluster: North Dakota, Accuracy: 0.88, Purity: 3.0
  Cluster: California, Accuracy: 1.0, Purity: 1.0

  Dla oddalonych od siebie grup wynik grupowania jest niezależny od parametru k i początkowej liczby grup.
  Dopiero od wartości początkowej liczby grup mniejszej niż 5 rezultaty zaczęły odbiegać od idealnych, jednakże nadal pozostając bardzo dobre dokładność całkowita na poziomie 98,5%.

7. Wnioski

Zgodnie z wnioskami przedstawionymi w artykule algorytm dobrze radzi sobie z grupowaniem odległych od siebie grup. Doskonale przeprowadza klasyfikacje. Dla zbiorów punktów blisko siebie algorytm również radzi sobie bardzo dobrze.

Największym problemem jest dobór parametrów grupowania do zbioru na jakim wykonywany jest algorytm.

Dla zbiorów niewielkich trzeba wziąć pod uwagę, że zbyt duże zmniejszenie parametru k odpowiedzialnego za pierwszą fazę działania algorytmu spowoduje utworzenie wielu niezorganizowanych małolicznych klastrów, z kolei zbyt duża wartość tego parametru sprawi, że powstaną klastry ze zbyt dużą liczbą punktów.

Z przeprowadzonych testów wynika, że najlepszą dokładność działania algorytmu uzyskano dla parametru k równego wyjściowej przewidywanej liczbie grup. Dobranie zbyt małej wartości parametru k zdecydowanie bardziej znacząco wpływa na późniejsze propagowanie się błędów, niż zbyt duża wartość tego parametru.

Początkowa liczba grup, a więc parametr odpowiedzialny za drugą fazę działania algorytmu wpływa znacząco na jakość wyliczania metryk RI i RC dla każdego klastra. Im mniejsza wartość tego parametru tym grupy są bardziej liczne z kolei zbyt duża wartość początkowej liczby grup powoduje, że przez niewielkie rozmiary klastrów w ostatniej fazie łączenia. Za mała jak i za duża liczba punktów w klastrze może spowodować niepoprawne wyliczanie metryk, a więc błędne łączenie klastrów w grupy.

Liczba punktów, na których będzie działał algorytm również wpływa na sposób jego działania – do poprawnego wyliczenia metryk RC i RI jak i EC potrzebne są wystarczająco duże klastry aby można było wyznaczyć zależności między nimi, jak i wystarczająco małe aby grupa nie zawierała zbyt wielu punktów należących oryginalnie do innych klastrów.

Biorąc powyższe pod uwagę, algorytm działa bardzo dobrze. Wpływ parametrów wejściowych jest zgodny z przewidywaniami i z artykułem opisującym działanie algorytmu.

Bibliografia

[1] G. Karypis, E.-H. (Sam) i H. V. Kumar, „CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling”.

[2] „CHAMELEON graph partitioning,” [Online]. Available: https://www.coursera.org/learn/cluster-analysis/lecture/0MQCo/4-7-chameleon-graph-partitioning-on-the-knn-graph-of-the-data.

[3] „METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering,” [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/metis/overview.

[4] „United States Cities Database,” [Online]. Available: https://simplemaps.com/data/us-cities.

[5] „Ocena jakości kwalifikacji,” [Online]. Available: http://mathspace.pl/tag/ocena-jakosci-klasyfikacji/page/2/.

[6] [Online]. Available: http://mlwiki.org/index.php/Chameleon_Clustering.