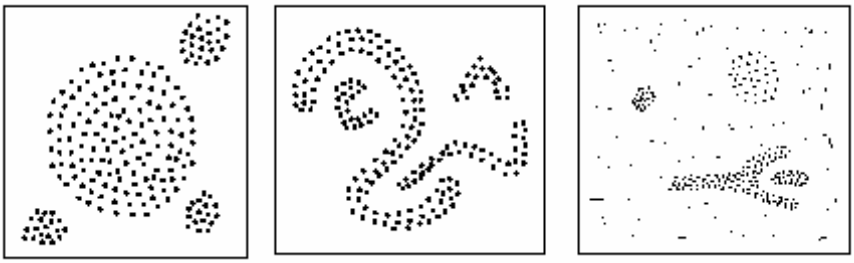
Eksploracja danych

grupowanie danych

Piotr Lipiński

Grupowanie danych

- Celem grupowania danych jest podział rekordów danych na grupy, tak aby elementy z tej samej grupy były do siebie podobne, a z różnych grup od siebie różne.
  - Zazwyczaj nie wiadomo czemu odpowiadają utworzone grupy (jak je interpretować merytorycznie).
  - Wiadomo jednak, jak je precyzyjnie zdefiniować.
  - Wiadomo też, że są statystycznie nieprzypadkowe.



Grupowanie danych

- Niech $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ będzie zbiorem danych złożonym z N obserwacji $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. Każda obserwacja $\mathbf{x}_i$ opisana jest przez d cech $x_{1i}, x_{2i}, \ldots, x_{di}$.
- Grupowanie danych polega na znalezieniu K elementowej partycji $C = \{C_1, C_2, \ldots, C_K\}$ zbioru D (tzn. K parami rozłącznych zbiorów $C_1, C_2, \ldots, C_K$ takich, że $C_1 \cup C_2 \cup \ldots \cup C_K = D$) maksymalizującej pewną miarę jakości grupowania danych F(C).
  - czasami dopuszcza się, że niektóre zbiory C_k są puste
  - liczba K jest zazwyczaj ustalona (parametr algorytmu grupowania)
  - w praktyce często wykonuje się kilka grupowań z różnymi liczbami K i wybiera najlepsze z nich

Kryterium oceny jakości grupowania danych

- Jak zdefiniować miarę jakości grupowania?
  - Chcemy, żeby każde dwa elementy należące do tej samej grupy były do siebie podobne, zaś każde dwa elementy należące do dwóch różnych grup były do siebie niepodobne.
  - Przyjmijmy, że potrafimy określić:
    - pewną miarę podobieństwa $\rho(x, y)$ mierząca podobieństwo między obserwacjami x i y
    - pewną miarę odległości $d(x, y)$ mierzącą odległość między obserwacjami x i y
    - zazwyczaj podobieństwo jest ujemnie skorelowane z odległością, na przykład $d(x, y) = 1 / \rho(x, y)$
  - Możliwe są różne podejścia do mierzenia jakości grupowania, które prowadzą do różnych algorytmów oraz różnych wyników grupowania tych samych danych. W konkretnej sytuacji wybór podejścia powinien zależeć od charakterystyki analizowanych danych oraz konkretnych potrzeb i konkretnych oczekiwań analityka danych.

Kryterium oceny jakości grupowania danych

- Podejście 1:
  - Dla każdej grupy C_k możemy zmierzyć średnie podobieństwo elementów w tej grupie

$$WCS(C_k) = \frac{1}{(|C_k|-1)|C_k|} \sum_{\mathbf{x}\in C_k} \sum_{\substack{\mathbf{y}\in C_k \\ \mathbf{x}\neq\mathbf{y}}} \rho(\mathbf{x},\mathbf{y})$$

  - Dla każdych dwóch grup C_k i C_l możemy zmierzyć średnie podobieństwo elementów tych grup

$$BCS(C_k, C_l) = \frac{1}{|C_k||C_l|} \sum_{\mathbf{x}\in C_k} \sum_{\mathbf{y}\in C_l} \rho(\mathbf{x},\mathbf{y})$$

  - Całkowita jakość grupowania C może być określona jako

$$F(C) = \frac{\sum_{k=1}^{K} WCS(C_k)}{\sum_{1\le k<l\le K} BCS(C_k, C_l)}$$

  - Podobne definicje można określić w oparciu o funkcję odległości.
  - Podejście to jest niepraktyczne ze względu na złożoność obliczeniową.

Kryterium oceny jakości grupowania danych

- Podejście 2:
  - Dla każdej grupy C_k możemy wyznaczyć jej centrum $\mathbf{r}_k$ określone jako środek ciężkości punktów tej grupy

$$\mathbf{r}_k = \frac{1}{|C_k|} \sum_{\mathbf{x}\in C_k} \mathbf{x}$$

  - Możemy określić odchylenie wewnątrzskupieniowe grupowania C jako

$$WCD(C) = \sum_{k=1}^{K} WCD(C_k) = \sum_{k=1}^{K} \sum_{\mathbf{x}\in C_k} d(\mathbf{x}, \mathbf{r}_k)$$

  - Możemy określić odchylenie międzyskupieniowe grupowania C jako

$$BCD(C) = \sum_{1\le k<l\le K} d(\mathbf{r}_k, \mathbf{r}_l)$$

  - Całkowita jakość grupowania C może być określona jako kombinacja WCD(C) i BCD(C), na przykład F(C) = BCD(C) / WCD(C).

Kryterium oceny jakości grupowania danych

- Podejście 2':
  - Jeśli wszystkie atrybuty są numeryczne, tzn. każda obserwacja $x \in R^d$, a miara odległości d to kwadrat odległości euklidesowej, to podejście 2 upraszcza się.
  - Dla każdej grupy C_k możemy określić macierz kowariancji (nieunormowaną) elementów grupy

$$\mathbf{W}_k = \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \mathbf{r}_k)(\mathbf{x} - \mathbf{r}_k)^T$$

  - wówczas odchylenie wewnątrzskupieniowe $WCD(C_k)$ to ślad tej macierzy (suma elementów przekątnej macierzy)

$$WCD(C_k) = tr(\mathbf{W}_k)$$

  - zatem

$$WCD(C) = \sum_{k=1}^{K} WCD(C_k) = \sum_{k=1}^{K} tr(\mathbf{W}_k) = tr(\mathbf{W})$$

  - gdzie

$$\mathbf{W} = \sum_{k=1}^{K} \mathbf{W}_k$$

  - Wniosek: Jeśli tr(W) jest małe, to WCD(C) jest małe, i odwrotnie. Powinno się więc dążyć do grupowania z małymi wariancjami elementów wewnątrz grup.

Kryterium oceny jakości grupowania danych

- Podejście 2':
  - Podobnie, można określić macierz **B**

$$\mathbf{B} = \sum_{k=1}^{K} | C_k | (\mathbf{r}_k - \hat{\mu})(\mathbf{r}_k - \hat{\mu})^T$$

gdzie μ to estymowana wartość średnia wszystkich punktów danych z D.

Kryterium oceny jakości grupowania danych

- Podejście 2':
  - Popularne funkcje oceny jakości grupowania danych opierają się na macierzach W i B, m.in.
    - tr(W)
    - det(W)
    - $tr(BW^{-1})$
  - Wadą miary tr(W) jest zależność od skali poszczególnych zmiennych. Zmieniając bowiem jednostkę jednej ze zmiennych (np. cm na m) możemy otrzymać inną strukturę grupowania. Miara tr(W) zazwyczaj prowadzi do kulistych kształtów grup, często dość zwartych i równolicznych.
  - Miara det(W) nie ma zależności skali, więc wykrywa też grupy eliptyczne. Preferuje również grupy równoliczne.
  - Miara $tr(BW^{-1})$ preferuje grupy równoliczne i o podobnych kształtach. Często tworzy grupy współliniowe.

Podstawowe algorytmy grupowania danych

- Różne miary jakości grupowania danych prowadzą do różnych algorytmów grupowania.
- Algorytmy wykrywające grupy definiowane w oparciu o centra grup:
  - algorytm k-means
  - algorytm oparty na algorytmie EM
- Algorytmy wykrywające grupy definiowane w oparciu o gęstość grup:
  - DBScan
- Algorytmy grupowania hierarchicznego

Algorytm k-means

- Niech D = {$\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_N$} będzie zbiorem danych złożonym z N obserwacji $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_N$. Niech K będzie liczbą grup, które należy utworzyć.
- Każda grupa C_k reprezentowana jest przez punkt $\mathbf{r}_k$ zwany centrum grupy. Każdy wektor danych jest przypisywany do grupy, której centrum jest mu najbliższe.
  - w przypadku równych odległości od kilku centrów, decyduje ustalona kolejność rozpatrywania grup lub przypisanie jest losowe
- Zadanie polega na znalezieniu K elementowej partycji C = {C_1, C_2, …, C_K} zbioru D (tzn. K parami rozłącznych zbiorów C_1, C_2, …, C_K takich, że $C_1 \cup C_2 \cup \ldots \cup C_K = D$) minimalizującej funkcję kryterium

$$F(C) = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} \| \mathbf{x} - \mathbf{r}_k \|^2$$

- Jednym z algorytmów rozwiązujących taki problem jest algorytm k-means.

Algorytm k-means

- Minimalizacja funkcji kryterium

$$F(C) = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} \| \mathbf{x} - \mathbf{r}_k \|^2$$

może przebiegać w dwóch krokach powtarzanych iteracyjnie:

- znając wektory $\mathbf{r}_k$, wyznaczyć optymalne przypisanie wektorów danych do grup – jest to oczywiste: każdy wektor danych powinien być przypisany do grupy reprezentowanej przez najbliższy mu wektor $\mathbf{r}_k$
- znając przypisanie wektorów danych do grup, wyznaczyć wektory $\mathbf{r}_k$
  - to jest mniej oczywiste – można użyć m.in. analizy matematycznej
  - rozwiązaniem jest ustawienie wektorów $\mathbf{r}_k$ w środkach geometrycznych zbioru punktów tworzących grupę

Algorytm k-means

- Algorytm k-means

FOR k = 1, 2, …, K
  r_k = losowo wybrany punkt z D
WHILE są zmiany w grupach C_k
  FOR k = 1, 2, …, K
    $C_k = \{x \in D : d(x, r_k) < d(x, r_l)$ dla każdego $l = 1, 2, \ldots, K, l \neq k\}$
  FOR k = 1, 2, …, K
    r_k = środek ciężkości C_k

Algorytm k-means

- Na wynik działania algorytmu k-means bardzo wpływa początkowe położenie centrów grup.
- Algorytm k-means tworzy podział przestrzeni danych na obszary Voronoya.
- Algorytm k-means nie będzie poprawnie grupował danych o nieregularnych kształtach grup, m.in..

Rozszerzenia algorytmu k-means

- Popularnych jest wiele modyfikacji algorytmu k-means:
  - algorytm k-means nazywa się czasem Hard C-Means (HCM)
  - algorytm Fuzzy C-Means (FCM)
  - algorytm Possibilistic C-Means (PCM)
  - algorytm Gustafsona-Kessela
  - algorytm Fuzzy Maximum Likelihood Estimation (FMLE)

Definicja odległości w grupowaniu danych

- Miara odległości w przestrzeni danych d(**x**, **y**) mierząca odległość między wektorami danych **x** i **y** ma kluczowe znaczenie dla grupowania.
- Odległość euklidesowa

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{j=1}^{d}(x_j - y_j)^2} = \sqrt{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}$$

  nie zawsze jest najlepszym wyborem.
- Odległość Minkowskiego to uogólnienie odległości euklidesowej

$$d(\mathbf{x},\mathbf{y}) = \sqrt[r]{\sum_{j=1}^{d}|x_j - y_j|^r}$$

  gdzie r jest pewną stałą.
  - dla r = 2 otrzymujemy odległość euklidesową
  - dla r = 1 otrzymujemy odległość Manhattan
  - odległość Manhattan dla binarnych wektorów danych to po prostu odległość Hamminga (liczba bitów na których różnią się dwa wektory binarne).

Definicja odległości w grupowaniu danych

- Częstym problemem jest nieodporność algorytmów grupowania na skalowanie poszczególnych wymiarów
  - na przykład zmiana jednostek jednego z atrybutów z mm na km może prowadzić do zupełnie innych wyników algorytmu grupowania
- Można tego uniknąć wprowadzając ważenie wymiarów w definicji odległości.
  - na przykład ważona odległość euklidesowa to

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{j=1}^{d} a_j (x_j - y_j)^2} = \sqrt{(\mathbf{x}-\mathbf{y})^T \mathbf{A} (\mathbf{x}-\mathbf{y})}$$

    gdzie $a_1, a_2, \ldots, a_d$ to wagi kolejnych wymiarów (pewne stałe), zaś **A** to macierz diagonalna z wartościami $a_1, a_2, \ldots, a_d$ na przekątnej.
- Ważenie wymiarów można rozszerzyć dopuszczając, aby macierz **A** nie była diagonalna.
- Jeśli $\mathbf{A} = \mathbf{R}^{-1}$, gdzie **R** to macierz kowariancji zbioru danych D, tzn.

$$\mathbf{R} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

  to otrzymana odległość jest zwana odległością Mahalanobisa.

Hard C-Means (HCM)

- Algorytm k-means można zapisać trochę inaczej.
- Macierzą przynależności wektora danych do grupy nazywamy macierz **M** rozmiaru N x K o elementach m_{ik} spełniającą
  - dla każdego i = 1, 2, .., N oraz k = 1, 2, …, K

$$m_{ik} \in \{0,1\}$$

  - dla każdego i = 1, 2, …, N

$$\sum_{k=1}^{K} m_{ik} = 1$$

  - dla każdego k = 1, 2, .., K

$$0 < \sum_{i=1}^{N} m_{ik} < N$$

- Macierzą środków grup nazywamy macierz **R** rozmiaru d x K, której kolejne kolumny to wektory $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K$.
- Algorytm k-means można więc zapisać przy użyciu macierzy **M** oraz **R**.

Hard C-Means (HCM)

- HCM – krok 0:
  - kolumny $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_K$ macierzy $\mathbf{R}$ inicjowane są losowo
- HCM – krok 1:
  - dla każdego i = 1, 2, …, N, dla każdego k = 1, 2, …, K
    - $m_{ik} = 1$, jeśli dla każdego $l \neq k$ zachodzi $d(\mathbf{x}_i, \mathbf{r}_k) < d(\mathbf{x}_i, \mathbf{r}_l)$
      - UWAGA: Jeśli dla pewnego wektora danych minimalna odległość jest realizowana przez więcej niż jeden środek grupy, to należy wybrać jeden z tych środków grup losowo bądź w inny ustalony sposób.
    - $m_{ik} = 0$, w przeciwnym przypadku
- HCM – krok 2:
  - dla każdego k = 1, 2, …, K

$$\mathbf{r}_k = \frac{\sum_{i=1}^{N} m_{ik}\mathbf{x}_i}{\sum_{i=1}^{N} m_{ik}}$$

- HCM – krok 3:
  - powtarzaj kroki 1 i 2 dopóki grupowanie nie ustabilizuje się (macierze **M** i **R** nie będą się zmieniać)

Fuzzy C-Means (FCM)

- Algorytm Fuzzy C-Means (FCM) używa rozmytej przynależności wektora danych do grupy (pozwala przypisać ten sam obiekt do kilku różnych grup z odpowiednimi stopniami przynależności).
- Macierzą rozmytej przynależności wektora danych do grupy nazywamy macierz **M** rozmiaru N x K o elementach m_{ik} spełniającą
  - dla każdego i = 1, 2, .., N oraz k = 1, 2, …, K

$$m_{ik} \in [0,1]$$

  - dla każdego i = 1, 2, …, N

$$\sum_{k=1}^{K} m_{ik} = 1$$

  - dla każdego k = 1, 2, .., K

$$0 < \sum_{i=1}^{N} m_{ik} < N$$

- Algorytm FCM minimalizuje kryterium (q to stała zwana stopniem rozmycia)

$$F(\mathbf{M}, \mathbf{R}) = \sum_{k=1}^{K}\sum_{i=1}^{N} m_{ik}^{q} d(\mathbf{x}_i, \mathbf{r}_k)$$

Fuzzy C-Means (FCM)

- FCM – krok 1:
  - dla każdego i = 1, 2, …, N, dla każdego k = 1, 2, …, K

$$m_{ik} = \left(\sum_{l=1}^{K} \frac{d(\mathbf{x}_i, \mathbf{r}_k)}{d(\mathbf{x}_i, \mathbf{r}_l)} \right)^{-\frac{2}{q-1}}$$

- FCM – krok 2:
  - dla każdego k = 1, 2, …, K

$$\mathbf{r}_k = \frac{\sum_{i=1}^{N} m_{ik}^q \mathbf{x}_i}{\sum_{i=1}^{N} m_{ik}^q}$$

- reszta algorytmu jak w HCM

Possibilistic C-Means (PCM)

- Algorytm Possibilistic C-Means (PCM) używa posybilistycznej przynależności wektora danych do grupy (pozwala przypisać ten sam obiekt do kilku różnych grup z odpowiednimi stopniami przynależności niekoniecznie sumującymi się do 1).
- Macierzą posybilistycznej przynależności wektora danych do grupy nazywamy macierz **M** rozmiaru N x K o elementach m_{ik} spełniającą
  - dla każdego i = 1, 2, .., N oraz k = 1, 2, …, K

$$m_{ik} \in [0,1]$$

  - dla każdego i = 1, 2, …, N, istnieje k = 1, 2, …, K, takie że

$$m_{ik} > 0$$

  - dla każdego k = 1, 2, .., K

$$0 < \sum_{i=1}^{N} m_{ik} < N$$

- Algorytm PCM minimalizuje kryterium (q to stała zwana stopniem rozmycia, a $\eta_1, \eta_2, \ldots, \eta_K$ to pewne współczynniki dodatnie)

$$F(\mathbf{M}, \mathbf{R}) = \sum_{k=1}^{K} \sum_{i=1}^{N} m_{ik}^q d(\mathbf{x}_i, \mathbf{r}_k) + \sum_{k=1}^{K} \eta_k \sum_{i=1}^{N} (1 - m_{ik})^q$$

Possibilistic C-Means (PCM)

- PCM – krok 1:
  - dla każdego i = 1, 2, …, N, dla każdego k = 1, 2, …, K

$$m_{ik} = \left(1 + \frac{d(\mathbf{x}_i, \mathbf{r}_k)}{\eta_k}\right)^{-\frac{2}{q-1}}$$

- PCM – krok 2:
  - dla każdego k = 1, 2, …, K

$$\mathbf{r}_k = \frac{\sum_{i=1}^{N} m_{ik}^q \mathbf{x}_i}{\sum_{i=1}^{N} m_{ik}^q}$$

- reszta algorytmu jak w HCM

Possibilistic C-Means (PCM)

- Współczynniki $\eta_1, \eta_2, \ldots, \eta_K$ określają tak zwaną szerokość rozkładu posybilistycznego.
- Współczynniki te:
  - mogą być stałe (parametry algorytmu)
  - mogą być zmienne (w czasie działania algorytmu)

$$\eta_k = \frac{\sum_{i=1}^{N} m_{ik}^q d(\mathbf{x}_i, \mathbf{r}_k)}{\sum_{i=1}^{N} m_{ik}^q}$$

Algorytm Gustafsona-Kessela (GK)

- We wszystkich omawianych dotąd algorytmach miara odległości w przestrzeni danych musi zostać z góry określona.
- Algorytm Gustafsona-Kessela (GK) to modyfikacja algorytmu FCM, w której wprowadza się różne miary odległości dla różnych grup:
  - dla k = 1, 2, …, K, odległość między wektorami danych **x** i **y** należącymi do C_k to

$$d_k(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T \mathbf{A}_k (\mathbf{x}-\mathbf{y})$$

    gdzie $\mathbf{A}_k$ to macierz rozmiaru d x d różna dla różnych grup.
- Algorytm GK minimalizuje kryterium (q to stała zwana stopniem rozmycia)

$$F(\mathbf{M},\mathbf{R}) = \sum_{k=1}^{K}\sum_{i=1}^{N} m_{ik}^q d_k(\mathbf{x}_i,\mathbf{r}_k)$$

- Macierze $\mathbf{A}_k$ muszą być w pewien sposób "ograniczone", na przykład przez wymuszenie $\det \mathbf{A}_k = \rho_k$, dla pewnych stałych ρ_k, bo inaczej minimalizacja będzie prowadzić do macierzy o bardzo małych elementach.

Algorytm Gustafsona-Kessela (GK)

- GK – krok 1:
  - dla każdego k = 1, 2, …, K, liczymy tzw. rozmytą macierz kowariancji

$$\mathbf{F}_k = \frac{\sum_{i=1}^{N} m_{ik}^q (\mathbf{x}_i-\mathbf{r}_k)(\mathbf{x}_i-\mathbf{r}_k)^T}{\sum_{i=1}^{N} m_{ik}^q}$$

  - dla każdego i = 1, 2, …, N, dla każdego k = 1, 2, …, K,

$$d(\mathbf{x}_i,\mathbf{r}_k) = (\mathbf{x}_i-\mathbf{r}_k)^T[(\rho_k \det(\mathbf{F}_k))^{1/d}\mathbf{F}_k^{-1}](\mathbf{x}_i-\mathbf{r}_k)$$

$$m_{ik} = \left(\sum_{j=1}^{K}\frac{d(\mathbf{x}_i,\mathbf{r}_k)}{d(\mathbf{x}_i,\mathbf{r}_j)}\right)^{-1}$$

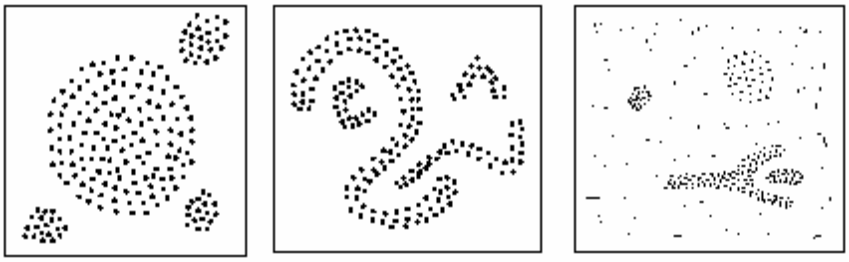
- GK – krok 2:
  - dla każdego k = 1, 2, …, K

$$\mathbf{r}_k = \frac{\sum_{i=1}^{N} m_{ik}^q \mathbf{x}_i}{\sum_{i=1}^{N} m_{ik}^q}$$

- reszta algorytmu jak w HCM

Algorytm DBScan

- Algorytm k-means tworzy podział przestrzeni danych na obszary Voronoya. Nie będzie więc poprawnie grupował danych o nieregularnych kształtach.
- Algorytm DBScan działa na innej zasadzie. Jest to przykład algorytmu grupowania opartego na gęstości.

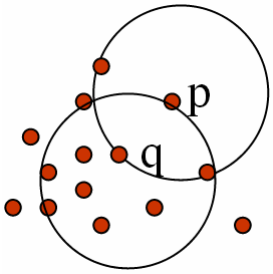


Algorytm DBScan

- Przez **sąsiedztwo** wektora danych **x** rozumiemy zbiór

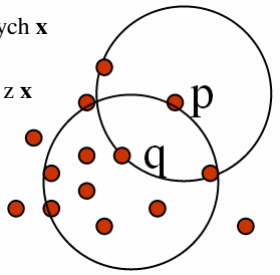
$$\{\mathbf{y} \in D : d(\mathbf{x}, \mathbf{y}) < \varepsilon\},$$

  gdzie wartość ε jest parametrem algorytmu DBScan.
- Sąsiedztwo wektora danych **x** jest **gęste**, jeśli zawiera co najmniej *m* wektorów danych, gdzie wartość *m* jest parametrem algorytmu DBScan.
- **Rdzeń** to wektor danych, którego sąsiedztwo jest gęste.
- **Punkt brzegowy** to wektor danych, którego sąsiedztwo nie jest gęste.



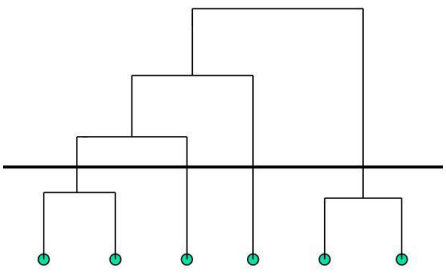
Algorytm DBScan

- Wektor danych **y** jest bezpośrednio osiągalny z wektora danych **x**, jeśli:
  - **y** należy do sąsiedztwa **x**,
  - sąsiedztwo **x** jest gęste.
- Wektor danych **y** jest osiągalny z wektora danych **x**, jeśli istnieje ciąg wektorów danych $\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$, taki że $\mathbf{x}_1 = \mathbf{x}$, $\mathbf{x}_n = \mathbf{y}$ oraz $\mathbf{x}_i$ jest bezpośrednio osiągalny z $\mathbf{x}_{i-1}$, dla każdego i = 2, 3, …, n.
- Wektory danych **x** i **y** są połączone, jeśli istnieje wektor danych **z**, taki że **x** i **y** są osiągalne z **z**.
- Grupa to maksymalny zbiór punktów połączonych.
- DBScan:
  - wybierz dowolny nierozpatrzony jeszcze wektor danych **x**
  - oznacz **x** jako już rozpatrzony
  - C := zbiór wszystkich wektorów danych osiągalnych z **x**
  - jeśli **x** jest rdzeniem, to uznaj C za grupę i oznacz wszystkie elementy C jako już rozpatrzone
  - powtarzaj powyższe kroki aż wszystkie wektory danych zostaną rozpatrzone



Grupowanie hierarchiczne

- Grupowanie hierarchiczne
  - Metody aglomeracyjne
  - Metody rozdzielające



Grupowanie hierarchiczne

- Grupowanie hierarchiczne
  - Metody aglomeracyjne
  - Metody rozdzielające
- Metody aglomeracyjne

FOR i = 1, 2, …, N

$C_i = \{\mathbf{x}_i\}$

WHILE pozostała więcej niż jedna grupa

wyznacz parę C_i i C_j dla której odległość $d(C_i, C_j)$ jest najmniejsza

połącz C_i i C_j zapisując w miejsce C_i

usuń C_j

- W zależności od definicji odległości między grupami, uzyskuje się inne dendrogramy.
  - Metoda Single Link (Nearest Neighbor) – odległość między grupami to odległość między najbliższymi punktami z tych grup.
    - Metoda podatna na zjawisko „łańcuchowania", w którym długie ciągi punktów są przypisywane do tej samej grupy.
    - Metoda wrażliwa na małe perturbacje danych i na punkty oddalone (outliers).
  - Metoda Complete Link (Furthest Neighbor) – odległość między grupami to odległość między najdalszymi punktami z tych grup.
  - Metoda oparta na odległości między środkami ciężkości grup.
  - Metoda oparta na średniej wszystkich odległości między punktami z obu grup

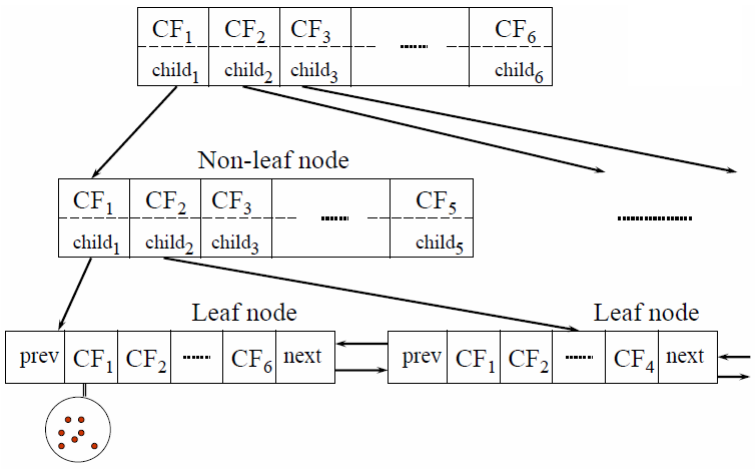
Grupowanie hierarchiczne

- Metody rozdzielające
  - monoteiczne – rozdzielają grupę używając tylko jednego atrybutu
  - politeiczne – rozdzielają grupę używając wszystkich atrybutów

BIRCH

- Clustering Feature Tree (CF Tree)
  - CF to (N, LS, SS)
    - gdzie N to liczba punktów danych, LS (linear sum) to suma punktów danych, SS (square sum) to suma kwadratów punktów danych
  - CF poszczególnych grup są przechowywane w drzewie CF
  - CF drzewo ma parametry: B (branching factor), T (threshold), L
  - Każdy węzeł wewnętrzny CF drzewa zawiera co najwyżej B par [CF, wskaźnik] opisujących poszczególne grupy (wskaźnik wskazuje na węzeł potomny odpowiadający danej grupie)
  - Każdy liść zawiera co najwyżej L pozycji [CF]

BIRCH



BIRCH

- Krok 1: tworzenie CF drzewa
  - Dla punktu danych $\mathbf{x}_i$ wyznacz liść L_j drzewa CF zawierający najbliższą mu grupę C_k.
  - Jeśli liść L_j zawiera mniej niż L punktów danych, to wstaw $\mathbf{x}_i$. W przeciwnym przypadku podziel liść na dwa liście (operacja split).
  - Popraw drzewo łącząc dwa najbliższe węzły (operacja merge) i ewentualnie powtórnie je dzieląc (operacja resplit)

BIRCH

- Krok 2 (opcjonalny): Popraw CF drzewo łącząc najbliższe liście.
- Krok 3: Pogrupuj punkty danych w liściach CF drzewa stosując wybrany algorytm grupowania
  - każda grupa jest reprezentowana przez swoje centrum, wyznaczone na przykład algorytmem k-means
- Krok 4 (opcjonalny): Popraw CF drzewo przez przeniesienie niektórych punktów danych między grupami.
  - wyznacz punkty danych, które są położone bliżej centrum innej grupy

Podejście probabilistyczne do grupowania danych

- Dotychczas nie zwracaliśmy uwagi na to, że wszystkie zarejestrowane dane są obarczone zaburzeniem losowym.
- Ignorowanie tego faktu często prowadzi do powstawania fałszywych grup danych.
- Nie powinniśmy więc skupiać się na samych wektorach danych, ale powinniśmy starać się znaleźć rozkład prawdopodobieństwa, z którego zostały wygenerowane.
  - Czy można powiedzieć, że nasze dane zostały wygenerowane losowo? Przecież są to konkretne wartości precyzyjnie wyliczone z danych zgromadzonych w bazach danych.
  - Można (patrz wykład o niepewności danych). Dla ilustracji można powiedzieć, że generatorem takich danych losowych mogło być urządzenie pomiarowe, które zamiast zwrócić nam dokładną wartość danej, zwróciło nam wartość zaburzoną losowym błędem pomiarowym.
- Skoro przypuszczamy, że dane dzielą się na K grup, a grupy danych mają różną charakterystykę, to przypuszczamy, że dane pochodzą z K różnych rozkładów prawdopodobieństwa (każda grupa danych została wygenerowana z innego rozkładu prawdopodobieństwa).
  - Gdyby wektory danych z dwóch różnych grup pochodziły z tego samego rozkładu prawdopodobieństwa, to nie można byłoby znaleźć między nimi różnic statystycznie istotnych.
  - Gdyby wektory danych z tej samej grupy pochodziły z dwóch różnych rozkładów prawdopodobieństwa, to powinny wystąpić między nimi różnice statystycznie istotne.

Podejście probabilistyczne do grupowania danych

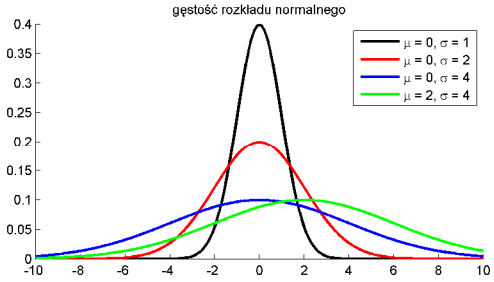
- Sytuacja odpowiada zatem następującemu modelowi generowania wektorów danych:
  - dane są rozkłady prawdopodobieństwa $P_1, P_2, ..., P_K$
  - wybieramy losowo jeden z nich (prawdopodobieństwa wyboru poszczególnych rozkładów $P_1, P_2, ..., P_K$ to odpowiednio $p_1, p_2, \ldots, p_K$)
  - losujemy wektor danych $\mathbf{x}$ używając wybranego rozkładu prawdopodobieństwa
- Model taki nazywa się mieszaniną rozkładów.
- Szczególnym przypadkiem jest mieszanina rozkładów gaussowskich (ang. Gaussian Mixture Model, GMM)
  - $P_1 = N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), P_2 = N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2), ..., P_K = N(\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)$
    gdzie $\boldsymbol{\mu}_k \in R^d$ to wartość oczekiwana, zaś $\boldsymbol{\Sigma}_k \in R^{d \times d}$ to macierz kowariancji

Parę słów o jednowymiarowym rozkładzie normalnym

- gęstość jednowymiarowego rozkładu normalnego $N(\mu, \sigma^2)$ o wartości oczekiwanej μ i wariancji σ^2 to

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

- wykresem gęstości jest krzywa Gaussa, wartość oczekiwana wpływa na przesunięcie krzywej, a wariancja na jej kształt

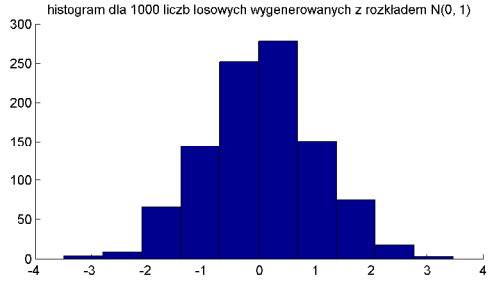


Parę słów o jednowymiarowym rozkładzie normalnym

- generując liczby losowe z rozkładem $N(\mu, \sigma^2)$ powinniśmy dostawać histogramy zgodne z odpowiednią krzywą Gaussa

  UWAGA: Jest kilka popularnych algorytmów generowania liczb pseudolosowych z rozkładem normalnym (zazwyczaj są one już zaimplementowane w popularnych narzędziach programistycznych), m.in. algorytm Boxa-Mullera czy algorytm Ziggurata.

- przykład: histogram dla 1000 liczb losowych wygenerowanych z rozkładem N(0, 1)

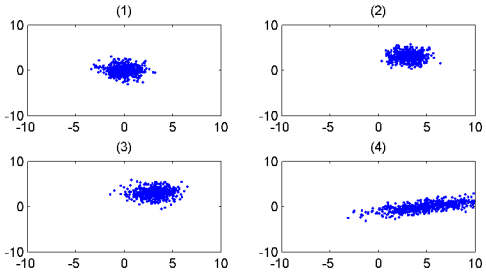


Parę słów o wielowymiarowym rozkładzie normalnym

- gęstość d-wymiarowego rozkładu normalnego $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ o wartości oczekiwanej $\boldsymbol{\mu} \in R^d$ i macierzy kowariancji $\boldsymbol{\Sigma} \in R^{d \times d}$ to

$$f(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d \,|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

- przykład: 500 losowych punktów wygenerowanych z dwuwymiarowym rozkładem normalnym (dla czterech różnych rozkładów)



Parę słów o wielowymiarowym rozkładzie normalnym

- Podstawowe własności wielowymiarowego rozkładu normalnego:
  - Jeśli $X \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, to $\mathbf{A}X + \mathbf{b} \sim N(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T)$, dla dowolnej macierzy $\mathbf{A}$ i dowolnego wektora $\mathbf{b}$ (odpowiednich rozmiarów).
  - estymator wartości oczekiwanej:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$$

  - estymator kowariancji (o największej wiarygodności):

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$$

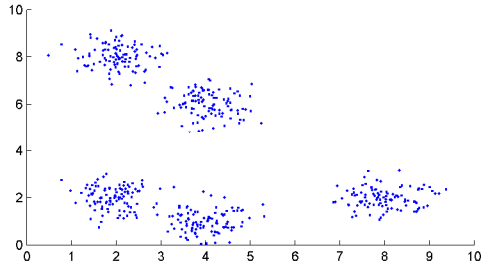
  - estymator kowariancji (nieobciążony):

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N} (\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$$

  - jak generować dane?
    - rozkład Choleskiego macierzy kowariancji $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^T$,
    - $X = \mathbf{A}Z + \boldsymbol{\mu}$,
      gdzie Z to zmienna losowa o standardowym rozkładzie normalnym

Mieszanina rozkładów gaussowskich

- Przykład:
  - mieszanina K = 5 dwuwymiarowych rozkładów gaussowskich
  - $p_1 = p_2 = p_3 = p_4 = p_5 = 1/5$
  - wszystkie macierze kowariancji są diagonalne z wariancjami 0.5
  - $\boldsymbol{\mu}_1 = [2, 2]$, $\boldsymbol{\mu}_2 = [4, 1]$, $\boldsymbol{\mu}_3 = [2, 8]$, $\boldsymbol{\mu}_4 = [4, 6]$, $\boldsymbol{\mu}_5 = [8, 2]$
  - rysunek poniżej przedstawia 500 wygenerowanych wektorów danych



Mieszanina rozkładów gaussowskich

- Łączny rozkład prawdopodobieństwa GMM ma postać

$$P(\mathbf{x}) = \sum_{k=1}^{K} P(R = k \mid \mathbf{p}) N(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  gdzie K to liczba rozkładów w rozważanej mieszaninie, $\mathbf{p} = (p_1, p_2, \ldots, p_K)$ to wektor prawdopodobieństw wyboru poszczególnych rozkładów mieszaniny, zaś $\boldsymbol{\mu}_k \in R^d$ i $\boldsymbol{\Sigma}_k \in R^{d \times d}$ to wartości oczekiwane i macierze kowariancji poszczególnych rozkładów mieszaniny.

- Przykład: Jeśli zmienne losowe są niezależne (macierze kowariancji są diagonalne), to GMM upraszcza się do

$$P(\mathbf{x}) = \sum_{k=1}^{K} P(R = k \mid \mathbf{p}) \prod_{i=1}^{N} N(x_i \mid \mu_{ki}, \sigma_{ki})$$

Podejście probabilistyczne do grupowania danych

- Grupowanie danych można więc potraktować jako zadanie znalezienie parametrów GMM opisującego rozważaną próbkę danych.
  - czyli należy wyznaczyć prawdopodobieństwa poszczególnych rozkładów $p_1, p_2, \ldots, p_K$ oraz ich parametry $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$ i $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots, \boldsymbol{\Sigma}_K$.
- Funkcja oceny modelu może być oparta na funkcji wiarygodności.
- Niech θ oznacza wektor parametrów modelu, zaś D próbkę danych. Funkcja wiarygodności to prawdopodobieństwo otrzymania próbki danych D pod warunkiem, że model ma parametry $\boldsymbol{\theta}$

$$L(\theta) = P(X \mid \theta) = \prod_{\mathbf{x} \in D} P(\mathbf{x} \mid \theta)$$

gdzie ostatnia równość zachodzi, jeśli wektory danych ze zbioru D były generowane niezależnie. Zamiast funkcji wiarygodności, często wygodniej jest rozpatrywać jej logarytm $l(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta})$.

Wiarygodność w modelach z wartościami ukrytymi

- W praktyce często występują modele z wartościami ukrytymi.
- W grupowaniu danych opartym na GMM za wartość ukrytą należy uznać zmienną losową R (tzn. numer rozkładu użytego do wygenerowania wektora danych), bo jej nie potrafimy zmierzyć (zmierzyć potrafimy jedynie ostateczny wynik, czyli wektor danych **x**).
- Jeśli w modelu występują pewne ukryte zmienne losowe H, to

$$l(\theta) = \log P(D \mid \theta) = \log \sum_H p(D, H \mid \theta)$$

- gdzie sumowanie jest po wszystkich możliwych wartościach zmiennych losowych H.
- Przekształcając dalej otrzymujemy (Q to rozkład prawdopodobieństwa dla H)

$$l(\theta) = \log \sum_H p(D, H \mid \theta) = \log \sum_H Q(H) \frac{p(D, H \mid \theta)}{Q(H)} \geq$$

$$\geq \sum_H Q(H) \log \frac{p(D, H \mid \theta)}{Q(H)} = \sum_H Q(H) \log p(D, H \mid \theta) + \sum_H Q(H) \log \frac{1}{Q(H)}$$

(nierówność wynika z nierówności Jensena dla funkcji wypukłych).

- Definiując

$$F(Q, \theta) = \sum_H Q(H) \log p(D, H \mid \theta) + \sum_H Q(H) \log \frac{1}{Q(H)}$$

otrzymujemy $l(\boldsymbol{\theta}) \geq F(Q, \boldsymbol{\theta})$. Zatem maksymalizując F maksymalizujemy też l.

Algorytm EM

- Rozważania te są podstawą ogólnego algorytmu EM (Estimation – Maximization).
- Algorytm EM składa się z dwóch kroków:
  - najpierw inicjalizujemy Q i $\boldsymbol{\theta}$ w losowy sposób
  - Krok E (estymacja):

    w którym szukamy Q maksymalizującego $F(Q, \boldsymbol{\theta})$ (odpowiada do estymacji rozkładu Q metodą największej wiarygodności)
    - $Q^{(t+1)} := \arg\max_Q F(Q^{(t)}, \boldsymbol{\theta}^{(t)})$
  - Krok M (maksymalizacja):

    w którym szukamy θ maksymalizującego $F(Q, \boldsymbol{\theta})$
    - $\boldsymbol{\theta}^{(t+1)} := \arg\max_\theta F(Q^{(t+1)}, \boldsymbol{\theta}^{(t)})$
  - powtarzaj kroki E i M dopóki rozkład nie ustabilizuje się

Podejście probabilistyczne do grupowania danych

- Dla uproszczenia załóżmy, że wszystkie rozkłady są rozkładami niezależnych zmiennych losowych, z taką samą wariancją.
- Krok 0:
  - wszystkie parametry, $p_1, p_2, \ldots, p_K$ oraz $\mu_1, \mu_2, \ldots, \mu_K$ i $\sigma_1, \sigma_2, \ldots, \sigma_K$, inicjujemy losowo
- Krok 1: (estymacja)
  - prawdopodobieństwo, że wektor danych $\mathbf{x}$ pochodzi z k-tej grupy można estymować przez

$$\hat{P}(R = k \mid \mathbf{x}) = \frac{p_k f(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{K} p_k f(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

- Krok 2: (maksymalizacja)

$$\hat{p}_k = \frac{1}{N} \sum_{i=1}^{N} \hat{P}(R = k \mid \mathbf{x}_i)$$

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{N\hat{p}_k} \sum_{i=1}^{N} \hat{P}(R = k \mid \mathbf{x}_i) \mathbf{x}_i$$

$$\hat{\sigma}_k = \frac{1}{N\hat{p}_k} \sum_{i=1}^{N} \hat{P}(R = k \mid \mathbf{x}_i)(\mathbf{x}_i - \boldsymbol{\mu}_k)^2$$

- Krok 3:
  - powtarzaj kroki 1 i 2 dopóki rozkład nie ustabilizuje się

Inne algorytmy grupowania danych

- Grupowanie danych jest jednym z kluczowych zagadnień eksploracji danych, więc podejść do niego jest wiele.
- Grupowanie danych nienumerycznych:
  - dotychczas rozważaliśmy wyłącznie grupowanie danych, w których wszystkie atrybuty były numeryczne
  - czy omówione algorytmy można zastosować do grupowania danych z atrybutami nienumerycznymi?
    - jeśli możliwe będzie określenie odległości na przestrzeni danych, to część algorytmów będzie działać
    - jeśli odległości dla pewnych atrybutów nie będzie można określić (np. atrybut wyrażający kolor: czerwony, zielony, niebieski), to trzeba użyć innego podejścia
  - algorytm PAM (Partitioning Around Medoids)
  - algorytm CLARA (Clustering Large Applications)
  - algorytm CLARANS

Inne algorytmy grupowania danych

- PAM (Partitioning Around Medoids)
  - Niech $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ będzie zbiorem danych złożonym z N obserwacji $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. Niech K będzie liczbą grup, które należy utworzyć.
  - Każda grupa C_k reprezentowana jest przez punkt $\mathbf{r}_k$ zwany centrum grupy. Każdy wektor danych jest przypisywany do grupy, której centrum jest mu najbliższe.
  - Centra grup ustawiamy w losowo wybranych punktach danych (tzn. wybieramy losowo bez zwracania K z N punktów danych).
  - Każdy punkt danych przypisujemy do najbliższego centrum.
  - Dopóki zmniejsza się łączna odległość:
    - Dla każdego centrum $\mathbf{r}_k$ i dla każdego punktu danych $\mathbf{x}_i$ niepokrywającego się z centrum próbujemy przenieść centrum $\mathbf{r}_k$ do punktu $\mathbf{x}_i$: obliczamy łączną odległość punktów danych od najbliższego centrum, jeśli łączna odległość wzrasta po takim przeniesieniu, to je cofamy, jeśli nie, to ją zostawiamy.
  - UWAGA: PAM można stosować dla danych nienumerycznych.
  - UWAGA: PAM można przyspieszyć obliczając wcześniej macierz odległości między punktami danych (centra grup zawsze są w punktach danych).

Metody oceny grupowania danych

- Ocena uzyskanego grupowania danych nie jest łatwa:
  - może zależeć od konkretnych zastosowań,
  - może zależeć od konkretnych danych.
- Naturalnym podejściem jest użycie miar jakości grupowania, dyskutowanych na początku wykładu.
- Używa się też kilku popularnych wskaźników poprawności grupowania (ang. clustering validity indices):
  - Dunn index,
  - Davies Bouldin index,
  - i inne.