

UCZENIE KLASYFIKACJI NIENADZOROWANEJ

- Zadanie klasteryzacji polega na dokonaniu efektywnego podziału zadanego zbioru danych Ω na pewną ilość podzbiorów $\Omega_1, \Omega_2, \dots, \Omega_c$, elementów do siebie podobnych. Podzbiory te nazywać będziemy **grupami** (klastrami).

$$\Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_c = \Omega$$

$$\Omega_i \cap \Omega_j = \Phi$$

- Elementy zbioru uczącego są wektorami $x_1, x_2, \dots, x_N \in R^p$
- Dokonany podział może stanowić podstawę do zdefiniowania klas obiektów.

PRZETWARZANIE WSTĘPNE

- Zapewnia bezpośrednią porównywalność atrybutów, które mogą być wyrażone w różnych jednostkach.
- Normalizacja danych zbioru uczącego odbywa się zatem w następujący sposób:

$$x_{ij} := \frac{x_{ij} - \bar{x}_j}{s_j} \text{ dla } i = 1, \dots, N, \quad j = 1, \dots, p,$$

gdzie \bar{x}_j oznacza średnią j -tego atrybutu w zbiorze uczącym,

a s_j odchylenie standardowe ze względu na j -ty atrybut, czyli:

$$\bar{x}_j = \frac{1}{N} \sum_{i=1}^N x_{ij}, \text{ natomiast } s_j = \frac{1}{N} \sum_{i=1}^N (x_{ij} - \bar{x}_j)^2.$$

PRZETWARZANIE WSTĘPNE

- Skalowanie - operację tą stosujemy dla zapewnienia jednakowej istotności i porównywalności każdego z atrybutów w zbiorze uczącym.
- Polega ona na liniowym, bądź nieliniowym skalowaniu danych do odpowiedniego zakresu..
- Operację skalowania liniowego zbioru uczącego na przedział $[0,1]$ definiujemy następująco:

$$x_{ij} := \frac{(x_{ij} - x_{\min j})}{\max_{i=1, \dots, N} \{x_{ij} - x_{\min j}\}} \text{ dla } i = 1, \dots, N, \quad j = 1, \dots, p,$$

gdzie $x_{\min j} = \min_{i=1, \dots, N} \{x_{ij}\}$ jest najmniejszą wartością j -tego atrybutu przyjmowaną w zbiorze uczącym dla $j = 1, \dots, p$.

MIARY PODOBIEŃSTWA

- Podobieństwo obiektów jest zatem wprost proporcjonalne do wartości miary podobieństwa.
- Wśród najczęściej stosowanych miar podobieństwa są jednak różne miary odległości - miary niepodobieństwa (ang. dissimilarity measure).
- Przyjmując, że każdy z obiektów zbioru uczącego jest reprezentowany w postaci p wymiarowego wektora cech, założmy zatem że określamy miarę odległości dla wektorów $x_i, x_j \in \Omega \subset R^p$. Miarę tą oznaczamy symbolem $d(x_i, x_j)$.

•

MIARY PODOBIEŃSTWA

- Popularnym sposobem reprezentacji całego zbioru danych poprzez miarę odległości, podobieństwa czy niepodobieństwa pomiędzy każdą parą obiektów, jest tak zwana *macierz sąsiedztwa* lub *bliskości* (ang. proximity matrix).
- Symetryczna macierz kwadratowa $D = [d_{ij}]$ rozmiaru $N \times N$, taka że:

$$d_{ij} = d(x_i, x_j) \text{ dla } i, j = 1, \dots, N$$

MIARY PODOBIEŃSTWA

- Podział zbioru na grupy, uzyskany w wyniku działania algorytmu grupowania, zwracać możemy również w postaci macierzy sąsiedztwa.
- W takim przypadku określamy miarę niepodobieństwa w ten sposób, że dla obiektów tej samej grupy wynosi 0, a dla obiektów należących do różnych grup jest równe 1, czyli:

$$D(i, j) = \begin{cases} 0 & \text{gdy } x_i, x_j \text{ należą do tej samej grupy} \\ 1 & \text{w przeciwnym wypadku} \end{cases},$$

GRUPOWANIE SEKWENCYJNE

- $d(x, \Omega)$ - miara podobieństwa (niepodobieństwa) wektora cech x do grupy Ω .
- Może być ona zdefiniowana jako odległość od pewnego reprezentanta, czy od środka grupy Ω .
- W algorytm nie wymaga na wejściu liczby grup na jakie ma zostać dokonany podział, a nowe grupy tworzone są w trakcie wykonywania algorytmu.
- Parametry algorytmu - próg niepodobieństwa T , oraz maksymalna liczba grup q

GRUPOWANIE SEKWENCYJNE

Algorytm grupowania sekwencyjnego

- $m := 1, \Omega_1 := \{x_1\}$
- Dla $i := 2$ do N
Znajdź grupę Ω_k , taki że $d(x_i, \Omega_k) = \min_{1 \leq j \leq m} d(x_i, \Omega_j)$
Jeżeli $d(x_i, \Omega_k) > T$ oraz $m < q$ to
 $m := m + 1, \Omega_m := \{x_i\}$
w przeciwnym razie
 $\Omega_k := \Omega_k \cup \{x_i\}$
Koniec jeżeli
Koniec dla.

GRUPOWANIE SEKWENCYJNE

- Za miarę podobieństwa wektora x_i do grupy Ω_k przyjmujemy odległość tego wektora od środka rozpatrywanej grupy v_k :

$$d(x_i, \Omega_k) = \|x_i - v_k\|, \text{ gdzie } v_k = \frac{1}{|\Omega_k|} \sum_{x \in \Omega_k} x.$$

Uwagi

- Każdy z wektorów podawany jest tylko raz
- Wynik działania algorytmu uzależniony może być od kolejności podawanych wektorów

GRUPOWANIE SEKWENCYJNE

Przykład

Rozważmy zbiór $X = \{x_1, \dots, x_7\}$, gdzie $x_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, x_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$
 $x_3 = \begin{bmatrix} 1,5 \\ 1,5 \end{bmatrix}, x_4 = \begin{bmatrix} 2 \\ 1,5 \end{bmatrix}, x_5 = \begin{bmatrix} 2,5 \\ 1,5 \end{bmatrix}, x_6 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, x_7 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}.$

Prześledźmy wykonanie algorytmu podając wektory po kolei od x_1 do x_7 ustalając próg $T = 1$ oraz maksymalną liczbę tworzonych grup $q = 3$.

GRUPOWANIE SEKWENCYJNE

$\Omega_1 = \{x_1\}, d(x_2, \Omega_1) = 1$, otrzymujemy $\Omega_1 = \{x_1, x_2\}$.
Podajemy próbkę x_3 . Środkiem klastra Ω_1 jest punkt $[1 \ 1,5]^T$,
zatem $d(x_3, \Omega_1) = 0,5$, więc próg T nie został przekroczony co
oznacza dołożenie wektora x_3 do grupy Ω_1 i otrzymujemy
 $\Omega_1 = \{x_1, x_2, x_3\}$ ze środkiem $[1,67 \ 1,5]^T$. Dalej dla wektora x_4
mamy: $d(x_4, \Omega_1) = \sqrt{(2-1,67)^2 + (1,5-1,5)^2} \approx 0,83$, więc
 $\Omega_1 = \{x_1, x_2, x_3, x_4\}$, a środek tej grupy wynosi $[1,375 \ 1,5]^T$.

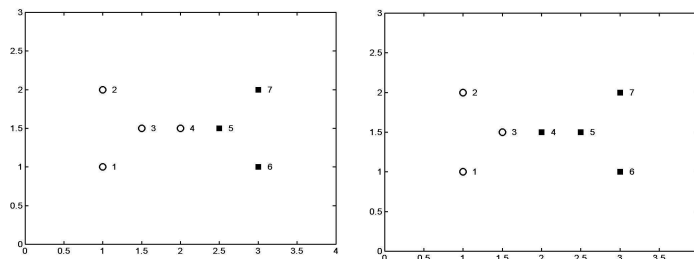
GRUPOWANIE SEKWENCYJNE

Dla wektora $x_5: d(x_5, \Omega_1) = 1,125$, próg T zostaje przekroczony,
tworzona nowa grupa $\Omega_2 = \{x_5\}$. Dla $x_6: d(x_6, \Omega_1) \approx 1,70$ oraz
 $d(x_6, \Omega_2) \approx 0,71$, więc $\Omega_2 = \{x_5, x_6\}$. Dla ostatniego wektora
 $d(x_7, \Omega_1) \approx 1,70, d(x_7, \Omega_2) \approx 0,79$ więc zostaje on dołożony do
grupy Ω_2 . Ostatecznie otrzymujemy zatem podział:
 $\Omega_1 = \{x_1, x_2, x_3, x_4\}, \Omega_2 = \{x_5, x_6, x_7\}.$

GRUPOWANIE SEKWENCYJNE

Weźmy teraz x_1 , czyli: $\Omega_1 = \{x_1\}$, następnie dla x_6 $d(x_6, \Omega_1) = 2$, więc $\Omega_2 = \{x_6\}$. Dla x_5 mamy $d(x_5, \Omega_1) \approx 1,58$, $d(x_5, \Omega_2) \approx 0,71$, otrzymując $\Omega_2 = \{x_5, x_6\}$. dla próbki x_4 : $d(x_4, \Omega_1) \approx 1,12$ oraz $d(x_4, \Omega_2) \approx 0,79$, więc $\Omega_2 = \{x_4, x_5, x_6\}$. dla x_3 : $d(x_3, \Omega_1) \approx 0,71$, $d(x_3, \Omega_2) \approx 1,01$, otrzymujemy $\Omega_1 = \{x_1, x_3\}$. Dalej dla x_2 : $d(x_2, \Omega_1) \approx 0,79$, $d(x_2, \Omega_2) \approx 1,64$, więc $\Omega_1 = \{x_1, x_2, x_3\}$. Wreszcie dla x_7 : $d(x_7, \Omega_1) \approx 1,90$, $d(x_7, \Omega_2) \approx 0,83$, ostatecznie otrzymujemy $\Omega_1 = \{x_1, x_2, x_3\}$, $\Omega_2 = \{x_4, x_5, x_6, x_7\}$.

GRUPOWANIE SEKWENCYJNE



Podział dokonany przez algorytm sekwencyjny w zależności od kolejności podawania wektorów.

ALGORYTMY GRUPOWANIA

- Podział algorytmów grupowania:
 - algorytmy hierarchiczne**, które w wyniku zwracają całą hierarchię podziałów.
 - algorytmy podziałowe**, które na ogół jako wynik zwracają jeden konkretny podział (reprezentowany np. macierzą podziału).
 - najpopularniejsze tutaj algorytmy optymalizacji funkcji kryterialnej.

GRUPOWANIE HIERARCHICZNE

- Wynikiem działania algorytmów hierarchicznych jest cała hierarchia podziałów zaczynając od podziału, w którym każdy obiekt stanowi oddzielną grupę, kończąc na podziale, w którym istnieje tylko jedna złożona ze wszystkich próbek
- Każdy krok algorytmu polega na połączeniu dwóch grup do siebie najbardziej podobnych (najmniej niepodobnych) - algorytmy aglomeracyjne
- Wariant odwrotny - podział początkowy będący jedną grupą i podział jednej grupy na dwie w każdym kroku – algorytmy podziału

GRUPOWANIE HIERARCHICZNE

- Aglomeracyjny algorytm grupowania hierarchicznego**
 - $\Omega_k := \{x_k\}$, dla $k = 1, \dots, N$, $c := N$
 - Znajdź 2 najbardziej do siebie podobne grupy Ω_i , Ω_j , za pomocą jednej z miar podobieństwa międzygrupowego.
 - Złącz grupy Ω_i , Ω_j to znaczy:

$$\Omega_i := \Omega_i \cup \Omega_j, i < j$$
 Usuń Ω_j (to znaczy: $k := k - 1$, dla $k > j$)
 $c := c - 1$
 - Jeżeli $c > 1$ idź do 2.

GRUPOWANIE HIERARCHICZNE

- Określanie podobieństwa międzygrupowego w decydujący sposób wpływa na rezultaty grupowania hierarchicznego i prowadzi w efekcie do innych algorytmów.
- Najczęściej spotykane metody określania miary podobieństwa międzygrupowego:

- Metoda najbliższego sąsiada (single link algorithm):

$$d_{\min}(\Omega_i, \Omega_j) = \min_{x \in \Omega_i, y \in \Omega_j} \|x - y\|^2$$

- Metoda najdalszego sąsiada (complete link algorithm):

$$d_{\max}(\Omega_i, \Omega_j) = \max_{x \in \Omega_i, y \in \Omega_j} \|x - y\|^2$$

GRUPOWANIE HIERARCHICZNE

3. Poprzez średnią odległość (weighted pair group method average (WPGMA) algorithm):

d_avg(Omega_i, Omega_j) = 1/(n_i * n_j) * sum_{x in Omega_i} sum_{y in Omega_j} ||x - y||^2

4. Poprzez odległość pomiędzy środkami grup (unweighted pair group method centroid (UPGMC) algorithm):

d_mean(Omega_i, Omega_j) = ||m_i - m_j||^2, gdzie: m_i = 1/n_i * sum_{x in Omega_i} x

GRUPOWANIE HIERARCHICZNE

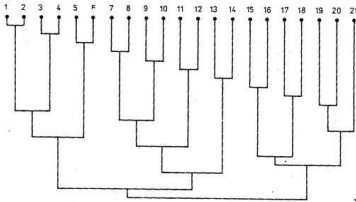
- W przypadku wszystkich metod 1-4 w każdym kroku łączymy dwie grupy Omega_i, Omega_j dla których:

d(Omega_i, Omega_j) = min_{k, l: k != l} d(Omega_k, Omega_l)

- Własności
- Metoda d_min generuje grupy o wydłużonych kształtach
 - Metoda d_max generuje grupy o kształtach maksymalnie zwartych
 - Metoda d_mean oraz d_avg ma pośrednie własności, tzn o kształcie grup decydują kształty obszarów, w których występuje maksymalne zagęszczenie próbek.

GRUPOWANIE HIERARCHICZNE

- Wynik działania algorytmu hierarchicznego przedstawiamy za pomocą tak zwanego dendrogramu
- **Dendrogram** - drzewo binarne którego liśćmi są klasyfikowane wektory, a do grupy utworzonej na poziomie węzła należą wszystkie wektory do których prowadzi droga z tego węzła



GRUPOWANIE HIERARCHICZNE

- Długość gałęzi dendrogramu może oznaczać wielkość podobieństwa między grupami
- Ucinając dendrogram na i-tym poziomie, licząc od korzenia włącznie, otrzymujemy podział na i grup
- O poziomie odcięcia decydować może pewna wartość progowa podobieństwa T - dendrogram odcina się w miejscu gdzie wartość podobieństwa łączonych grup jest mniejsza od T (niepodobieństwa większa)
- Aglomeracyjny algorytm hierarchiczny, można zmodyfikować by nie tworzył całej hierarchii podziałów, a zatrzymywał się w określonym miejscu

GRUPOWANIE HIERARCHICZNE

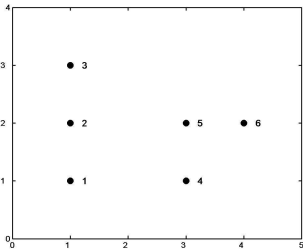
Przykład

Niech X = {x_1, ..., x_N}, gdzie x_1 = [1; 1], x_2 = [1; 2], x_3 = [1; 3],

x_4 = [3; 1], x_5 = [3; 2], x_6 = [4; 2].

Prześledzimy algorytm hierarchiczny z miarą podobieństwa d_min z normą euklidesową.

Mamy: Omega_1 = {x_1}, ..., Omega_6 = {x_6}.



GRUPOWANIE HIERARCHICZNE

Niech P(X) oznacza macierz podobieństwa dla k-tej iteracji, gdzie element i,j jest równy d_min(Omega_i, Omega_j):

P^0(X) = [[0, 1, 2, 2, 2,24, 3,16], [1, 0, 1, 2,24, 2, 3], [2, 1, 0, 2,83, 2,24, 3,16], [2, 2,24, 2,83, 0, 1, 1,41], [2,24, 2, 2,24, 1, 0, 1], [3,16, 3, 3,16, 1,41, 1, 0]]

GRUPOWANIE HIERARCHICZNE

W macierzy $P(X)$ poszukujemy wartości minimalnej poza przekątną, minimum dla grup Ω_1 i Ω_2 , które łączymy, mając:

$\Omega_1 = \{x_1, x_2\}$, $\Omega_2 = \{x_3\}$, $\Omega_3 = \{x_4\}$, $\Omega_4 = \{x_5\}$, $\Omega_5 = \{x_6\}$. Nowa macierz podobieństwa jest postaci:

$$P^1(X) = \begin{bmatrix} 0 & 1 & 2 & 2 & 3 \\ 1 & 0 & 2,83 & 2,24 & 3,16 \\ 2 & 2,83 & 0 & 1 & 1,41 \\ 2 & 2,24 & 1 & 0 & 1 \\ 3 & 3,16 & 1,41 & 1 & 0 \end{bmatrix}$$

GRUPOWANIE HIERARCHICZNE

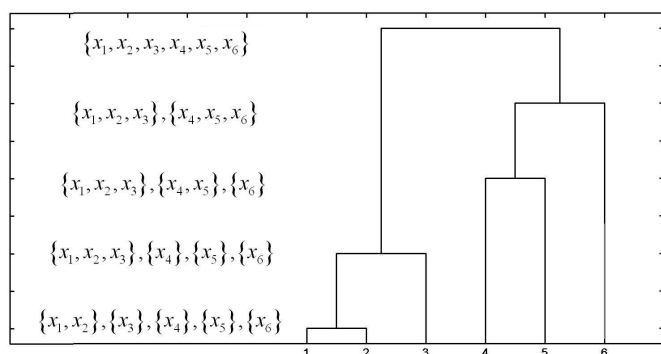
Łączymy grupy Ω_1 i Ω_2 , otrzymując: $\Omega_1 = \{x_1, x_2, x_3\}$, $\Omega_2 = \{x_4\}$, $\Omega_3 = \{x_5\}$, $\Omega_4 = \{x_6\}$. Macierz podobieństwa tego podziału:

$$P^2(X) = \begin{bmatrix} 0 & 2 & 2 & 3 \\ 2 & 0 & 1 & 1,41 \\ 2 & 1 & 0 & 1 \\ 3 & 1,41 & 1 & 0 \end{bmatrix}, \text{ dalej łączymy grupy } \Omega_2 \text{ i } \Omega_3$$

mając $\Omega_1 = \{x_1, x_2, x_3\}$, $\Omega_2 = \{x_4, x_5\}$, $\Omega_3 = \{x_6\}$. Dalej

$$P^3(X) = \begin{bmatrix} 0 & 2 & 3 \\ 2 & 0 & 1 \\ 3 & 1 & 0 \end{bmatrix}, \text{ więc } \Omega_1 = \{x_1, x_2, x_3\}, \Omega_2 = \{x_4, x_5, x_6\}.$$

GRUPOWANIE HIERARCHICZNE



GRUPOWANIE PODZIAŁOWE

- Wspólną cechą algorytmów tej grupy jest poszukiwanie możliwie najlepszego podziału, minimalizując iteracyjnie pewną funkcję kryterialną.
- Funkcja kryterialna jest miernikiem jakości podziału zbioru, który pozwala spośród wszystkich możliwych wariantów podziału wybrać ten, który uznajemy za optymalny.
 - Zwykle jest miarą dyspersji zbioru, na ogół ma postać sumy kwadratów odległości obiektów odpowiednich grup od prototypów je opisujących.

GRUPOWANIE PODZIAŁOWE

- Wynik podziału zbioru N elementów na C grup $\Omega_1, \Omega_2, \dots, \Omega_C$ przedstawiamy w postaci tak zwanej **macierzy podziału**
- Macierz podziału - macierz o wymiarze $C \times N$, o następujących własnościach:

$$1. \quad \forall_{\substack{1 \leq i \leq C \\ 1 \leq k \leq N}} u_{ik} = \begin{cases} 0 & \text{gdy } x_k \notin \Omega_i \\ 1 & \text{gdy } x_k \in \Omega_i \end{cases}$$

$$2. \quad \forall_{1 \leq k \leq N} \sum_{i=1}^C u_{ik} = 1$$

$$3. \quad \forall_{1 \leq i \leq C} 0 < \sum_{k=1}^N u_{ik} < N$$

ALGORYTM ISODATA (HCM)

- CEL - znalezienie macierzy podziału oraz macierzy środków grup, które minimalizują funkcję kryterialną:

$$J_e(U, V) = \sum_{i=1}^C \sum_{k=1}^N u_{ik} \|x_k - v_i\|^2,$$

gdzie: $U = [u_{ik}]$, gdzie $u_{ik} = \begin{cases} 0 & \text{gdy } x_k \notin \Omega_i \\ 1 & \text{gdy } x_k \in \Omega_i \end{cases}$ - macierz podziału

$$V = [v_1, \dots, v_C] \in R^{p \times C}, \text{ gdzie } \forall_{1 \leq i \leq C} v_i = \frac{1}{N_i} \sum_{x \in \Omega_i} x = \frac{\sum_{k=1}^N u_{ik} x_k}{\sum_{k=1}^N u_{ik}} -$$

macierz środków grup

ALGORYTM ISODATA (Iterative Self-Organizing DATA)

1. Inicjalizacja $U^{(0)}$.
2. Oblicz środki grup $V^{(j)} = [v_1^{(j)}, v_2^{(j)}, \dots, v_C^{(j)}]$ według

$$\text{wzoru: } \forall_{1 \leq i \leq C} v_i = \frac{1}{N_i} \sum_{x \in \Omega_i} x = \frac{\sum_{k=1}^N u_{ik} x_k}{\sum_{k=1}^N u_{ik}}$$

3. Aktualizuj macierz podziału dla $(j+1)$ iteracji według

$$\text{wzoru: } \forall_{\substack{1 \leq i \leq C \\ i \leq k \leq N}} u_{ik} = \begin{cases} 1, & \min_{1 \leq l \leq C} \|x_k - v_l^{(j)}\| = \|x_k - v_i^{(j)}\| \\ 0, & \text{w pozostałych przypadkach} \end{cases}$$

4. Jeżeli $U^{(j+1)} \neq U^{(j)}$, wtedy $j := j+1$ i idź do 2.

ALGORYTM ISODATA (HCM)

- Algorytm jest wykonywany tak długo dopóki macierz podziału ulega zmianom, jest zbieżny do lokalnego minimum funkcji kryterialnej, zalecane jest więc powtórzenie jego wykonania z różnymi macierzami początkowego podziału $U^{(0)}$.
- Algorytm może być również inicjowany środkami grup zamiast losowej macierzy podziału
- Problem – oszacowanie liczby grup na którą ma być dokonywany podział

ALGORYTM ISODATA (HCM)

Przykład

Niech $X = \{x_1, \dots, x_{10}\}$, gdzie $x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$, $x_2 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$, $x_3 = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$,

$x_4 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$, $x_5 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$, $x_6 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$, $x_7 = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$, $x_8 = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$, $x_9 = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$,

$x_{10} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$ a początkowa macierz podziału:

$$U^{(0)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

ALGORYTM ISODATA (HCM)

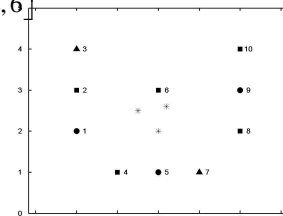
Obliczamy środki grup:

$$v_1 = \frac{1}{3} \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 4 \end{bmatrix} \right) = \begin{bmatrix} 3 \\ 2 \end{bmatrix},$$

$$v_2 = \frac{1}{5} \left(\begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 5 \\ 2 \end{bmatrix} + \begin{bmatrix} 5 \\ 3 \end{bmatrix} \right) = \begin{bmatrix} 3,2 \\ 2,6 \end{bmatrix},$$

$$v_3 = \frac{1}{2} \left(\begin{bmatrix} 1 \\ 4 \end{bmatrix} + \begin{bmatrix} 4 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 2,5 \\ 2,5 \end{bmatrix},$$

$$\text{czyli } V^{(1)} = \begin{bmatrix} 3 & 3,2 & 2,5 \\ 2 & 2,6 & 2,5 \end{bmatrix}.$$



ALGORYTM ISODATA (HCM)

Wyliczamy nową macierz podziału, przydzielając każdy z wektorów do grupy, której środek v_i jest mu najbliższy

Oznaczmy przez D macierz odległości, której elementy są równe $d_{ik} = \|x_k - v_i\|$ dla $i = 1, \dots, C$, $k = 1, \dots, N$. Mamy:

$$D^{(1)} = \begin{bmatrix} 2 & 2,24 & 2,83 & 1,41 & 1 & 1 & 1,41 & 2 & 2,24 & 2,83 \\ 2,28 & 2,24 & 2,61 & 2 & 1,61 & 0,45 & 1,79 & 1,9 & 1,84 & 2,28 \\ 1,58 & 1,58 & 2,12 & 1,58 & 1,58 & 0,71 & 2,12 & 2,55 & 2,55 & 2,91 \end{bmatrix}.$$

$$\text{Czyli: } U^{(1)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

ALGORYTM ISODATA (HCM)

Macierz $U^{(1)} \neq U^{(0)}$, więc wyliczamy macierz środków:

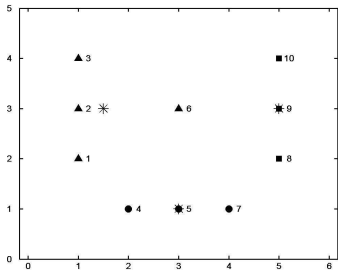
$$V^{(2)} = \begin{bmatrix} 3 & 5 & 1,5 \\ 1 & 3 & 3 \end{bmatrix}, \text{ macierz odległości:}$$

$$D^{(2)} = \begin{bmatrix} 2,24 & 2,83 & 3,61 & 1 & 0 & 2 & 1 & 2,24 & 2,83 & 3,61 \\ 4,12 & 4 & 4,12 & 3,61 & 2,83 & 2 & 2,23 & 1 & 0 & 1 \\ 1,12 & 0,5 & 1,12 & 2,06 & 2,5 & 1,5 & 3,2 & 3,64 & 3,5 & 3,64 \end{bmatrix},$$

$$\text{a stąd macierz podziału: } U^{(2)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

ALGORYTM ISODATA (HCM)

Ponieważ $U^{(2)}=U^{(1)}$, koniec algorytmu. Otrzymany podział::
 $\Omega_1=\{x_4,x_5,x_7\}$, $\Omega_2=\{x_8,x_9,x_{10}\}$, $\Omega_3=\{x_1,x_2,x_3,x_6\}$.



ALGORYTM FCM

Rozmyta macierz podziału - macierz spełniająca następujące warunki:

- i) $\forall_{\substack{1\leq i\leq C\\1\leq k\leq N}} u_{ik}\in [0,1]$,
ii) $\forall_{1\leq k\leq N} \sum_{i=1}^C u_{ik}=1$,
iii) $\forall_{1\leq i\leq C} 0<\sum_{k=1}^N u_{ik}<N$.
- Element u_{ik} rozmytej macierzy podziału możemy odczytywać jako prawdopodobieństwo z jakim element x_k należy do grupy Ω_i , lub stopień przynależności wektora x_k do Ω_i .
 - Algorytm, który w wyniku zwraca rozmytą macierz podziału -algorytm FCM (Fuzzy c-means)

ALGORYTM FCM

1. Inicjalizacja $U^{(0)}$, Ustal $\varepsilon>0$, $m>1$
2. Oblicz środki grup $V^{(j)}=[v_1^{(j)},v_2^{(j)},...,v_c^{(j)}]$ na podstawie $U^{(j)}$, według wzoru:

$$\forall_{1\leq i\leq c} \quad v_i=\left[\sum_{k=1}^N(u_{ik})^m x_k\right] / \left[\sum_{k=1}^N(u_{ik})^m\right]$$

3. Dla j+1 iteracji aktualizuj macierz podziału według wzoru:

ALGORYTM FCM

$$\forall_{\substack{1\leq i\leq c\\1\leq k\leq N}} u_{ik}=\begin{cases} \forall_{i\in I_k} \quad 0 & \sum_{i\in I_k} u_{ik}=1 & I_k\neq \emptyset \\ \left(\frac{1}{d_{ik}}\right)^{\frac{2}{m-1}} / \left[\sum_{j=1}^C \left(\frac{1}{d_{jk}}\right)^{\frac{2}{m-1}}\right] & I_k=\emptyset \end{cases},$$

gdzie: $d_{ik}^2=\|x_k-v_i\|^2=(x_k-v_i)^T(x_k-v_i)$,

$$\forall_{1\leq k\leq N} \begin{cases} I_k=\{i\mid 1\leq i\leq c; d_{ik}=0\}, \\ \tilde{I}_k=\{1,2,...,c\}\setminus I_k \end{cases}.$$

4. Jeżeli $\|U^{(j+1)}-U^{(j)}\|\geq \varepsilon$ wtedy j:=j+1 i idź do 2

ALGORYTM FCM

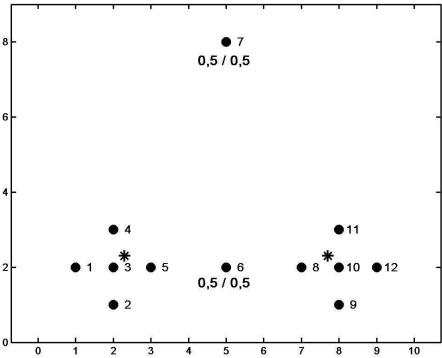
- Parametr m wpływa wprost proporcjonalnie na stopień rozmycia grup.
- Najczęściej przyjmuje się, że $m=2$.

- Algorytm FCM minimalizuje tę samą funkcję kryterialną co algorytm HCM tzn:

$$J_m(U,V)=\sum_{i=1}^C\sum_{k=1}^N(u_{ik})^m d_{ik}^2 \text{ przy warunku } \forall_{1\leq k\leq N} \sum_{i=1}^C u_{ik}=1.$$

ALGORYTM FCM

Przykład



INDEKS XIE-BENI

Indeks Xie-Beni pozwala ocenić jakość podziału dokonanego przez algorytm grupowania w zależności od ilości klasterów na jakie zostaje dokonywany, na podstawie wynikowej rozmytej macierzy podziału. Jest on zdefiniowany według wzoru:

V_{XB}^{(m)}(U) = \frac{\sum_{k=1}^N \sum_{i=1}^C (u_{ik})^m d_{ik}^2}{N \left(\min_i \|v_i - v_j\|^2 \right)}.

Najlepszym jest ten podział, który minimalizuje wspomniany współczynnik.

Algorytm posybilistyczny (Keller)

- Algorytm FCM minimalizuje funkcję kryterialną postaci $J_m(U,V) = \sum_{i=1}^C \sum_{k=1}^N (u_{ik})^m d_{ik}^2$ przy warunku $\forall_{1 \leq k \leq N} \sum_{i=1}^C u_{ik} = 1$.
- u_{ik} określa stopień przynależności k -tej próbki do i -tej grupy
- Rezygnując z tego założenia, przy podstawieniu $u_{ik} = 0$ dla wszystkich i, k funkcja kryterialna osiąga minimum - taki wynik z oczywistych względów nas nie zadowala.
- Założenie osłabiamy do postaci $\forall_i \max_k u_{ik} > 0$

Algorytm posybilistyczny (Keller)

- Chcemy by wartość przynależności próbek o dużym stopniu podobieństwa do grupy była możliwie duża a o stopniu podobieństwa jak najmniejsza. Funkcja kryterialna: $J_m(U,V) = \sum_{i=1}^C \sum_{k=1}^N (u_{ik})^m d_{ik}^2 + \sum_{i=1}^C \eta_i \sum_{k=1}^N (1-u_{ik})^m$,
gdzie $\eta_i > 0$
- Pierwsza część powyższego wyrażenia wymaga by odległość próbek od środków była jak najmniejsza natomiast druga by wartość przynależności u_{ik} była możliwie wysoka.

Algorytm posybilistyczny (Keller)

- Inicjalizuj $U^{(0)}$, ustal $\varepsilon > 0, m > 1$
- Inicjalizacja parametrów $\eta_i, i=1,...,C$ np. według wzoru: $\forall_{1 \leq i \leq C} \eta_i = \frac{\sum_{k=1}^N u_{ik}^m d_{ik}^2}{\sum_{k=1}^N u_{ik}^m}$
- Oblicz środki grup podstawie $U^{(0)}$, według wzoru: $\forall_{1 \leq i \leq c} v_i = \left[\sum_{k=1}^N (u_{ik})^m x_k \right] / \left[\sum_{k=1}^N (u_{ik})^m \right]$

Algorytm posybilistyczny (Keller)

- Dla $j+1$ iteracji aktualizuj macierz podziału według wzoru: $\forall_{\substack{1 \leq i \leq c \\ 1 \leq k \leq N}} u_{ik} = \frac{1}{1 + \left(\frac{d_{ik}}{\eta_i} \right)^{\frac{2}{m-1}}}$
- Jeżeli $\|U^{(j+1)} - U^{(j)}\| \geq \varepsilon$ wtedy $j := j + 1$ i idź do 3

Algorytm Gustafsona-Kessela

- Algorytm FCM nie sprawuje się najlepiej w przypadku gdy grupy nie mają w przybliżeniu eliptycznego kształtu, gdy różnice ich wielkości są bardzo duże.
- Miarę odległości definiujemy ważąc ją dodatkowo określoną macierzą A , a mianowicie: $d_{ik}^2 = |A_i|^{1/N} (x_k - v_i)^T A_i^{-1} (x_k - v_i)$.
- A jest kolejną zmienną otrzymujemy więc funkcję kryterialną: $\min_{\substack{(U,V,A) \\ \det(A_i) = \rho_i}} J_m(U,V,A) = \sum_{i=1}^C \sum_{k=1}^N (u_{ik})^m d_{ik}^2 = \sum_{i=1}^C \sum_{k=1}^N (u_{ik})^m \|x_k - v_i\|_{A_i}^2$,
gdzie U macierz podziału, V –macierz środków grup, $A = (A_1, ..., A_C)$, gdzie $A_i, i=1,...,C$ macierze dodatnio określone stopnia p . W miejsce A bierzemy estymator macierzy kowariancji.

Algorytm Gustafsona-Kessela

1. Inicjalizacja $U^{(0)}$, Ustal $\varepsilon > 0$, $m > 1$, $\eta_i > 0$

Oblicz środki grup $V^{(j)} = [v_1^{(j)}, v_2^{(j)}, \dots, v_c^{(j)}]$ podstawie $U^{(j)}$, według wzoru:

$$\forall_{1 \leq i \leq c} \quad v_i = \left[\sum_{k=1}^N (u_{ik})^m x_k \right] / \left[\sum_{k=1}^N (u_{ik})^m \right]$$

2. Oblicz macierze wag według wzoru:

$$\forall_{1 \leq i \leq c} \quad A_{i+1} = [\rho_i \det(C_i)]^{1/p} C_i^{-1},$$

$$\text{gdzie } C_i = \frac{\sum_{k=1}^N u_{ik}^m (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^N u_{ik}^m}.$$

Algorytm Gustafsona-Kessela

3. Dla j+1 iteracji aktualizuj macierz podziału według wzoru:

$$\forall_{\substack{1 \leq i \leq c \\ 1 \leq k \leq N}} \quad u_{ik} = \begin{cases} \forall_{i \in I_k} \quad 0 & \sum_{i \in I_k} u_{ik} = 1 & I_k \neq \emptyset \\ \left(\frac{1}{d_{ik}} \right)^{\frac{2}{m-1}} / \left[\sum_{j=1}^c \left(\frac{1}{d_{jk}} \right)^{\frac{2}{m-1}} \right] & I_k = \emptyset \end{cases}$$

4. Jeżeli $\|U^{(j+1)} - U^{(j)}\| \geq \varepsilon$ wtedy j:=j+1 i idź do 2

METODY JĄDROWE

Często w zbiorze danych nie ma regionów o eliptycznym kształcie, co jest warunkiem poprawnego działania takich algorytmów jak HCM czy FCM, w tym przypadku przestrzeni danych można odwzorować w nową przestrzeń, podobnie jak w SVM.

Funkcja jądra - iloczyn kartezjański w nowej przestrzeni.

$$H(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j).$$

Przykłady funkcji jądra:

- Wielomianowa: $H(x_i, x_j) = (x_i \cdot x_j + 1)^d$
- Gaussowska: $H(x_i, x_j) = \exp\left(-r \|x_i - x_j\|^2\right)$
- Neuronowa: $H(x_i, x_j) = \tanh(ax_i \cdot x_j + b)$

METODY JĄDROWE

Algorytm ISODATA zbiór N danych $x_1, \dots, x_N \in R^d$ dzieli na C grup $\Omega_1, \dots, \Omega_C$ i zwraca środek każdego z nich $m_1, \dots, m_C \in R^d$:

1. Ustal m_1, \dots, m_C .
2. Każdą próbkę x_i $i=1, \dots, N$ przydziel do klastra odpowiadającego najbliższemu z środków m_1, \dots, m_C czyli oblicz:

$$\delta(x_i, \Omega_k) = \begin{cases} 1 & D(x_i, m_k) < D(x_i, m_j) \text{ dla } j \neq k \\ 0 & \text{w pozostałych przypadkach} \end{cases}$$

3. Oblicz nowe środki klastrów m_1, \dots, m_C według wzoru:

$$m_k = \frac{1}{|\Omega_k|} \sum_{i=1}^N \delta(x_i, \Omega_k) x_i, \text{ gdzie } |\Omega_k| = \sum_{i=1}^N \delta(x_i, \Omega_k).$$

METODY JĄDROWE

4. Powtarzaj 2 i 3 do momentu gdy podział nie będzie uległ zmianom.
5. Zwróć m_1, \dots, m_C .

- $D(x_i, m_k)$ jest odległością euklidesową próbki x_i od środka m_k , to znaczy $D^2(x_i, m_k) = \|x_i - m_k\|^2$.

- Kluczem do zdefiniowania algorytmu ISODATA w nowej przestrzeni jest obliczenie w niej odległości.

Niech $u_i = \Phi(x_i)$. Odległość u_i od u_j wyraża się wzorem:

$$\begin{aligned} D^2(u_i, u_j) &= \|u_i - u_j\|^2 = \|\Phi(x_i) - \Phi(x_j)\|^2 = \Phi^2(x_i) - 2\Phi(x_i)\Phi(x_j) + \Phi^2(x_j) = \\ &= H(x_i, x_i) - 2H(x_i, x_j) + H(x_j, x_j) \end{aligned}$$

METODY JĄDROWE

Niech z_k będzie środkiem k -tej grupy w przestrzeni Q .

$$m_k = \frac{1}{|\Omega_k|} \sum_{i=1}^N \delta(u_i, \Omega_k) u_i.$$

Mamy:

$$\begin{aligned} D^2(u_i, z_k) &= \left\| u_i - \frac{1}{|\Omega_k|} \sum_{j=1}^N \delta(u_j, \Omega_k) u_j \right\|^2 = \\ &= \Phi^2(x_i) - \frac{2}{|\Omega_k|} \sum_{j=1}^N \delta(u_j, \Omega_k) \Phi(x_i) \Phi(x_j) + \frac{1}{|\Omega_k|^2} \left(\sum_{j=1}^N \delta(u_j, \Omega_k) \Phi(x_j) \right) \left(\sum_{l=1}^N \delta(u_l, \Omega_k) \Phi(x_l) \right) = \\ &= H(x_i, x_i) - \frac{2}{|\Omega_k|} \sum_{j=1}^N \delta(u_j, \Omega_k) H(x_i, x_j) + \sum_{j=1}^N \sum_{l=1}^N \delta(u_j, \Omega_k) \delta(u_l, \Omega_k) H(x_j, x_l) \end{aligned}$$

METODY JĄDROWE

czyli $D^2(u_i, z_k) = H(x_i, x_i) + f(x_i, \Omega_k) + g(\Omega_k),$

gdzie:

$$f(x_i, \Omega_k) = -\frac{2}{|\Omega_k|} \sum_{j=1}^N \delta(u_j, \Omega_k) H(x_i, x_j),$$
$$g(\Omega_k) = \sum_{j=1}^N \sum_{l=1}^N \delta(u_j, \Omega_k) \delta(u_l, \Omega_k) H(x_j, x_l).$$

Wstawiając powyższe obliczenia w miejsce tradycyjnego algorytmu ISODATA otrzymujemy nowy kernelowy (jądrowy) algorytm HCM.

Kernel c-means

1. Inicjalizacja $\delta(x_i, \Omega_k)$ dla $i=1,...,N, k=1,...,C$.
2. Dla każdej grupy Ω_k oblicz $|\Omega_k|$ oraz $g(\Omega_k)$.
3. Dla każdej próbki x_i oraz grupy Ω_k oblicz $f(x_i, \Omega_k)$.
Następnie każdą próbkę przypisz do najbliższej grupy:
$$\delta(x_i, \Omega_k) = \begin{cases} 1 & f(x_i, \Omega_k) + g(\Omega_k) < f(x_i, \Omega_j) + g(\Omega_j) \text{ dla } j \neq k \\ 0 & \text{w pozostałych przypadkach} \end{cases}$$
6. Powtarzaj 2 i 3 do momentu gdy podział nie będzie ulegał zmianom.
7. Dla każdego klastrera Ω_k wybierz próbkę, która jest najbliższa jego środkowi jako reprezentanta tej grupy.

$$m_k = \arg \min_{x_i: \delta(x_i, \Omega_k)=1} D(\Phi(x_i), z_k).$$