STUDIA INFORMATICA 2004
Volume 25 Number 1 (57)

Katarzyna STĄPOR
Politechnika Śląska, Instytut Informatyki

PODZIAŁOWE I HIERARCHICZNE ALGORYTMY GRUPOWANIA DANYCH

Streszczenie. W artykule przedstawiono najważniejsze podziałowe oraz hierarchiczne algorytmy grupowania danych. Wśród algorytmów podziałowych omówiono algorytmy oparte na prototypach punktowych oraz liniowych. Przedstawiono algorytmy hierarchiczne dla różnych miar podobieństwa oraz omówiono skrótowo klasteryzację neuronową wykorzystującą sieć Kohonena.

Słowa kluczowe: klasteryzacja, macierz podziału, dendrogram podziału, miara podobieństwa

PARTITIONAL AND HIERARCHICAL DATA CLUSTERING METHODS

Summary. In this paper the most important partitional and hierarchical data clustering algorithms are described. Among partitional algorithms those based on point and linear prototypes are discussed. In hierarchical algorithms different similarity measures are described. Also, neural network clustering based on Kohonen network is described.

Keywords: clustering, partition matrix, dendrogram, similarity measure

1. Wprowadzenie

Grupowanie danych (inaczej klasteryzacja) jest jednym z rodzajów uczenia nienadzorowanego i zajmuje się zagadnieniami efektywnego podziału danego zbioru nieetykietowanych obiektów (tj. wejściowych wektorów cech $x_1, ..., x_N$) na grupy $G_1, ..., G_c$, stanowiące podzbiory podobnych elementów. Podziału dokonuje się tak, aby elementy danej grupy G_i były bardziej podobne do siebie niż do jakichkolwiek elementów innych grup.

Liczba klasterów c może być podana z góry przez użytkownika lub nieznana, tj. wyznaczona dopiero przez metodę. Podział dokonany przez proces klasteryzacji może stanowić podstawę do zdefiniowania klas obiektów. Każdemu uzyskanemu w wyniku podziału podzbiorowi przypisuje się nazwę klasy, zaś obiekty zaliczone do tego podzbioru uznaje się za przykładowych reprezentantów danej klasy (próbki uczące o znanej przynależności klasowej). Dalsze etapy postępowania (tzn. tworzenie obszarów decyzyjnych przypisanych klasom) są takie same jak w zadaniu klasyfikacji. Aby móc zrealizować grupowanie, musimy przyjąć pewne założenia wstępne.

1. Liczebność zbioru jest duża, aby przejawiła się w nim pełna różnorodność zjawisk występujących w danym zagadnieniu.
2. Wszystkie klasy są reprezentowane w zbiorze.
3. Podzbiory mogą być formowane w naturalne klastery, tak że każdy klaster odpowiada klasie, którą reprezentuje (ta właściwość silnie zależy od wyboru wektora cech).
4. Obiekty są opisywane przy użyciu cech tak dobranych, że zbliżone wartości cech odpowiadają obiektom o zbliżonych (w odczuciu użytkownika) własnościach.

Podstawowe etapy w procesie grupowania danych są następujące:

1. Selekcja i ekstrakcja cech, polegająca na identyfikacji zbioru cech istotnych.
2. Właściwy algorytm klasteryzacji.
3. Walidacja klasteryzacji.

W ogólnym przypadku optymalne (w sensie przyjętego kryterium) rozwiązanie podziału N wektorów zbioru na c niepustych podzbiorów można znaleźć poprzez testowanie wszystkich możliwych podziałów wejściowych wektorów. Jest to jednak nierealistyczne ze względu na ogromną liczbę możliwych podziałów, w przybliżeniu około $c^N / c!$.

Wynik klasteryzacji można przedstawić jako cxN macierz podziału U (tzw. twarda macierz podziału), której elementy są następujące:

$$u_{ik} = \begin{cases} 1 & x_k \in G_i \\ 0 & x_k \notin G_i \end{cases}$$

G_i - i-ty klaster, $i=1, ..., N$

V_{cN} - przestrzeń wektorowa (cxN) wymiarowych macierzy nad R.

Zbiór wszystkich możliwych macierzy podziału U definiujemy:

$$M_c = \{U \in V_{cN} \mid 1., 2., 3.\},$$

gdzie warunki 1-3 są określone następująco:

1. Każdy wektor x_k jest albo nie jest w i-tym klasterze:

$$\forall_{\substack{1 \le i \le c \\ 1 \le k \le N}} u_{ik} \in \{0,1\}$$

2. Każdy wektor x_k należy dokładnie do jednego z c klasterów:

$$\forall_{1 \le k \le N} \sum_{i=1}^{c} u_{ik} = 1$$

3. Żaden klaster nie jest pusty:

$$\forall_{1 \le i \le c} \; 0 < \sum_{k=1}^{N} u_{ik} < N$$

Istnieje wiele różnych podziałów algorytmów klasteryzacji, stosownie do różnych kryteriów. Najczęściej jednak dzieli się algorytmy klasteryzacji na 2 grupy:

1. Algorytmy hierarchiczne (wynikiem jest zbiór wzajemnie zawierających się podziałów zbioru).
2. Algorytmy podziałowe (wynikiem jest jeden podział zbioru), wśród których najpopularniejsze są algorytmy z *iteracyjną optymalizacją pewnej funkcji kryterialnej*.

Często spotyka się podział na:

1. Algorytmy twarde (ang. hard), w których każdy wektor należy w całości do klastera.
2. Algorytmy rozmyte (ang. fuzzy), w których przynależność wektora do danego klastera może być tylko częściowa).
3. Algorytmy posybilistyczne (ang. possibilistic), w których przynależność wektora do klastera zostaje zastąpiona pojęciem „typowości".

Należy nie zapomnieć również o zastosowaniu sieci neuronowych oraz algorytmów genetycznych i ewolucyjnych w rozwiązaniu zagadnienia grupowania zbioru.

2. Algorytmy iteracyjnej optymalizacji funkcji kryterialnej

2.1. Algorytm ISODATA

Aby przeprowadzić podział na podzbiory podobnych obiektów należy zdefiniować pojęcie podobieństwa. Zazwyczaj miarą podobieństwa jest odległość w przestrzeni cech między wektorami cech obiektów (im bliższe siebie wektory, tym bardziej podobne obiekty).

$$s(x,y) = \| x - y \|^2$$

Kolejnym zadaniem jest określenie pewnego wskaźnika jakości (tj. funkcji kryterialnej), który pozwoliłby spośród wszystkich możliwych wariantów podziału zbioru na podzbiory wybrać podział optymalny (w sensie przyjętego kryterium). Najczęściej stosuje się tzw. dyspersję zbioru będącą sumą kwadratów odchyleń wektorów od środków klastrów:

$$J_E(U,V) = \sum_{i=1}^{c} \sum_{x \in G_i} \| x - v_i \|^2 = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik} \| x_k - v_i \|^2$$

c - liczba grup (klastrów)
G_i - i-ty klaster (podzbiór)
v_i - średnia wartość wektorów w i-tej grupie
$U = [u_{ik}]$ - macierz podziału
$V = [v_1, ..., v_c] \in V_{pc}$ - macierz środków klasterów zdefiniowana następująco:

$$\underset{1 \le i \le c}{\forall} v_i = \frac{1}{N_i} \sum_{x \in G_i} x = \frac{\sum_{k=1}^{N} u_{ik} x_k}{\sum_{k=1}^{N} u_{ik}}$$

Każdy element v_i macierzy określa średnie położenie wektorów tworzących klaster. Podział uzyskany za pomocą takiego kryterium nazywa się podziałem minimalnej wariancji, gdyż minimalizuje sumę kwadratów odległości wektorów w klasterze od jego środka. Kryterium takie jest dobre dla klasterów zwartych, o podobnych kształtach w przybliżeniu hiperkulistych i dobrze oddzielonych od siebie.

Problem klasteryzacji polega więc na znalezieniu takiej macierzy podziału U_{opt} (z klasy M_c) i macierzy środków klasterów V_{opt} (z klasy V_{pc}), które minimalizują J_e:

$$\min_{M_c x V_{pc}} J_e(U,V) = J_e(U_{opt}, V_{opt})$$

Najbardziej popularny, uproszczony algorytm dla aproksymacji minimum funkcji J_e to iteracyjny algorytm ISODATA (Iterative Self-Organizing DATA clustering) [1,4].

Algorytm ISODATA

1. Ustalenie liczby klastrów c $(1<c<N)$, wartości błędu granicznego ε.
2. Inicjalizacja macierzy podziału $U^{(0)} \in M_c$.

   Obliczenie środków klasterów $V^{(j)} = [v_1^{(j)}, ..., v_c^{(j)}]$ przy użyciu $U^{(0)}$.:

$$\underset{1 \le i \le c}{\forall} v_i = \frac{1}{N_i} \sum_{x \in G_i} x = \frac{\sum_{k=1}^{N} u_{ik} x_k}{\sum_{k=1}^{N} u_{ik}}$$

3. Aktualizacja macierzy podziału $U^{(j+1)}$ dla $(j+1)$-szej iteracji:

$$\underset{\substack{1 \le i \le c \\ 1 \le k \le N}}{\forall} u_{ik}^{(j+1)} = \begin{cases} 1 & \min_{1 \le i \le c} \| x_k - v_i^{(j)} \| = \| x_k - v_i^{(j)} \| \\ 0 & otherwise \end{cases}$$

4. Jeśli $\| U^{(j+1)} - U^{(j)} \| > \varepsilon$, to $j := j+1$, idź do punktu 3, w przeciwnym przypadku zakończenie działania algorytmu.

Algorytm ten jest zbieżny do lokalnego minimum funkcji J_E, którego położenie zależy bardzo silnie od początkowego wyboru środków klastrów. Dlatego najczęściej powtarza się

każdą klasteryzację kilkakrotnie, a następnie wybiera się częściej występujący wynik. Nie ma gwarancji, że osiągnie się maksimum globalne (wynik optymalny).

2.2. Algorytmy rozmyte

2.2.1. Algorytmy oparte na prototypach punktowych

Rozmyty algorytm ISODATA (FCM)

Metoda ISODATA zakłada, że każdy wektor należy tylko do jednej klasy, co jest dobre tylko w przypadku klasterów zwartych i dobrze oddzielonych. W praktyce klastery najczęściej nakładają się i niektóre wektory mogą należeć częściowo do kilku klasterów. Najlepiej jest opisać taką sytuację poprzez zastosowanie teorii zbiorów rozmytych w ten sposób, że wektor x_k może należeć do i-tego klastra z wartością przynależności u_{ik} z przedziału $<0,1>$. Zbiór wszystkich możliwych tzw. rozmytych c-macierzy podziału U definiujemy jako:

$$M_{fc} = \{U \in V_{cN} \mid 1.*, 2., 3.\}$$

przy zmienionym warunku 1.*:

1.* Każdy wektor x_k może należeć do kilku klasterów:

$$\underset{\substack{1\le i\le c\\ 1\le k\le N}}{\forall} u_{ik} \in [0,1]$$

Jako funkcję kryterialną stosuje się:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m d_{ik}^2$$

gdzie:

$$d_{ik}^2 = \| x_k - v_i \|_A^2 = (x_k - v_i)^T A(x_k - v_i)$$

A macierz dodatnio określona

$m \in [1,+\infty)$ współczynnik wagowy, określający stopień rozmycia klastrów

Podział optymalny jest rozwiązaniem:

$$\min_{M_{fc} x V_{pc}} J_m(U,V) = J_m(U_{opt}, V_{opt})$$

Można udowodnić następujące twierdzenie, z którego korzysta się w celu znalezienia rozwiązania optymalnego.

Twierdzenie

Ustalmy m oraz c i zdefiniujmy następujące zbiory [2,4]:

$$\underset{1\le k\le N}{\forall}\begin{cases} I_k = \{i \mid 1 \le i \le c; d_{ik} = 0\} \\ \bar{I}_k = \{1,\dots,c\} \setminus I_k \end{cases}$$

$(U_{opt}, V_{opt}) \in (M_{fc} x V_{pc})$ może osiągnąć minimum globalne funkcji kryterialnej J_E tylko wtedy, gdy:

$$\underset{\substack{1\le i\le c \\ 1\le k\le N}}{\forall} u_{ik} = \begin{cases} \underset{i\in \bar{I}_k}{\forall} \; 0 \quad \sum_{i\in I_k} u_{ik} = 1 & I_k \ne \varnothing \\ \left(\frac{1}{d_{ik}}\right)^{\frac{2}{m-1}} \Big/ \left[\sum_{j=1}^{c}\left(\frac{1}{d_{jk}}\right)^{\frac{2}{m-1}}\right] & I_k = \varnothing \end{cases}$$

oraz

$$\underset{1\le i\le c}{\forall} v_i = \left[\sum_{k=1}^{N}(u_{ik})^m x_k\right] \Big/ \left[\sum_{k=1}^{N}(u_{ik})^m\right]$$

Następujący algorytm, zwany rozmytym algorytmem ISODATA lub FCM (od angielskiej nazwy fuzzy clustering method FCM), umożliwia osiągnięcie w sposób iteracyjny rozwiązania optymalnego [1]:

1. Ustalenie liczby klastrów c $(1<c<N)$, $m \in [1,+\infty)$, $\varepsilon>0$
2. Inicjalizacja $U^{(0)} \in M_{fc}$
3. Obliczenie rozmytych środków klasterów $V^{(j)} = [v_1^{(j)},\dots,v_c^{(j)}]$ przy użyciu $U^{(j)}$:

$$\underset{1\le i\le c}{\forall} v_i = \left[\sum_{k=1}^{N}(u_{ik})^m x_k\right] \Big/ \left[\sum_{k=1}^{N}(u_{ik})^m\right]$$

4. Aktualizacja macierzy podziału dla $(j+1)$-szej iteracji:

$$\underset{\substack{1\le i\le c \\ 1\le k\le N}}{\forall} u_{ik} = \begin{cases} \underset{i\in \bar{I}_k}{\forall} \; 0 \quad \sum_{i\in I_k} u_{ik} = 1 & I_k \ne \varnothing \\ \left(\frac{1}{d_{ik}}\right)^{\frac{2}{m-1}} \Big/ \left[\sum_{j=1}^{c}\left(\frac{1}{d_{jk}}\right)^{\frac{2}{m-1}}\right] & I_k = \varnothing \end{cases}$$

5. Jeśli $\|U^{(j+1)}-U^{(j)}\| > \varepsilon$, to $j:=j+1$, idź do punktu 3, w przeciwnym przypadku zakończenie działania algorytmu.

Parametr m określa stopień rozmycia klasterów, zazwyczaj przyjmuje się m=2. Dla m=1 algorytm przechodzi w omówiony algorytm ISODATA. Udowodniono, że algorytm FCM jest zbieżny do lokalnego minimum funkcji lub do punktu siodłowego [2].

Algorytm G-K (Gustafson-Kessel)

W algorytmie FCM zakłada się, że wszystkie klastery mają jednakowe kształty geometryczne, co nie zawsze ma miejsce w przypadku rzeczywistych danych. Użyta w algorytmie FCM norma ma wpływ na kształt wszystkich klasterów. Gustaffson i Kessel [6] zaproponowali modyfikację FCM poprzez wprowadzenie różnych norm dla każdego klastera, co umożliwiło stosowanie algorytmu FCM w przypadku klasterów różnego kształtu. Dla i-tego klastera wprowadzono normę:

$$\forall_{\substack{1\le k\le N\\ 1\le i\le c}} \quad d_{ik}^2 = \| x_k - v_i \|_{A_i}^2 = (x_k - v_i)^T A_i (x_k - v_i),$$

gdzie $A_i \in R^{p\times p}$. W ten sposób kształt i-tego klastera (w danej iteracji) będzie określony poprzez macierz A_i. Zmodyfikowana funkcja kryterialna ma postać:

$$J_m(U,V,A) = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m \| x_k - v_i \|_{A_i}^2,$$

gdzie $A=(A_1, ..., A_c)$ jest c-krotką macierzy.

Gustafson i Kessel pokazali, że przy dodatkowym założeniu stałości wyznacznika A_i-tej macierzy, tj.

$$det(A_i) = \rho_i > 0,$$

co gwarantuje jej dodatnią określoność, minimalizacja funkcji $J(U,V,A)$ ze względu na A_i prowadzi do następującego rozwiązania gwarantującego optymalność kształtu i-tego klastera:

$$\forall_{1\le i\le c} \quad A_i opt = \left[\rho_i \det(C_i)\right]^{1/p} C_i^{-1},$$

gdzie C_i jest tzw. macierzą kowariancji i-tego klastera:

$$C_i = \sum_{k=1}^{n} u_{ik}^m (x_k - v_i)(x_k - v_i)^T / \sum_{k=1}^{n} u_{ik}^m \qquad m \ge 1,$$

zaś v_i jest prototypem punktowym (centrum) i-tego klastra.

Przebieg algorytmu można zapisać w następujących punktach:

1. Ustalenie liczby klastrów c $(0<c<N)$, $m \in [1,+\infty)$, $\varepsilon>0$, $\rho_i \in R^+$
2. Inicjalizacja $U^0 \in M_{fc}$
3. Obliczenie rozmytych środków klastrów $V^{(j)} = [v_1^{(j)},...,v_c^{(j)}]$ przy użyciu $U^{(j)}$.

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

4. Obliczenie rozmytych macierzy kowariancji $[C_1, ..., C_c]$:

$$C_i = \sum_{k=1}^{n} u_{ik}^m (x_k - v_i)(x_k - v_i)^T / \sum_{k=1}^{n} u_{ik}^m$$

ich macierzy odwrotnych oraz wyznaczników macierzy kowariancji.

5. Obliczenie macierzy $[A_1 ,..., A_c]$ normujących i-ty klaster:

$$A_i = [\rho_i \det(C_i)]^{1/p} C_i^{-1}$$

6. Aktualizacja macierzy przynależności U^{j+1} dla $j=1$-szej iteracji

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

gdzie:

$$d_{ik}^2 = \| x_k - v_i \|_{A_i}^2 = (x_k - v_i)^T A_i (x_k - v_i)$$

7. Jeśli $\|U^{(j+1)} - U^{(j)}\| > \varepsilon$, to $j:=j+1$, idź do punktu *3*, w przeciwnym przypadku zakończenie działania algorytmu.

W celu inicjalizacji macierzy podziału często stosuje się algorytm FCM.

2.2.2. Algorytmy oparte na prototypach niepunktowych

Algorytm oparty na rozmytych liniowych rozmaitościach (ang. fuzzy c-varieties (FCV))

W algorytmie tym [1,4] rozmyte środki klasterów vi (prototypy punktowe) w przestrzeni p-wymiarowej zostały zastąpione poprzez *r*-wymiarowe liniowe rozmaitości (ang. varieties, manifolds) ($0 \le r < p$), stanowiące prototypy.

R-wymiarowa rozmaitość liniowa V_{ri} oparta na punkcie $v_i \in R^p$ i określona poprzez *r* liniowo niezależnych wektorów $(s_{i1}, ..., s_{ir})$, gdzie $s_{ir} \in R^p$ jest to następujący zbiór:

$$V_{ri}(v_i, s_{i1}, ..., s_{ir}) = \left\{ y \in R^p \mid y = v_i + \sum_{j=1}^{r} t_j s_{ij} \quad t_j \in R \right\}$$

Dla $r=0$ jest to punkt, dla $r=1$ prosta, dla $r=2$ płaszczyzna, zaś dla $r>3$ hiperpłaszczyzna. Prototypy tego rodzaju mogą być uważane jako „płaskie" zbiory w przestrzeni R^p. Wymiar r określa liczbę kierunków, w którym ta płaskość rozciąga się. Jeśli wektory $\{s_{ij}\}$ są ortogonalne, to odległość wektora x_k od i-tej liniowej rozmaitości R_i jest określona :

$$D_{ik}^2 = D(x_k, V_{R_i})^2 = \| x_k - v_i \|_A^2 - \sum_{j=1}^{r} \left[(x_k - v_i)^T A s_{ij}\right]^2,$$

gdzie *A* jest macierzą dodatnio określoną.

Funkcja kryterialna do optymalizacji ma następująca postać:

$$J(U, V, V_R) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m D_{ik}^2,$$

gdzie $V_R = (V_{R_1}, ..., V_{R_c})$ jest zbiorem c liniowych rozmaitości stanowiących *c* prototypów.

Dla $r=0$ algorytm FCV redukuje się do algorytmu FCM. Algorytm FCV jest taki sam jak omówiony już algorytm FCM z tą różnicą, że zamiast d_{ik} podstawiane jest D_{ik}, zaś warunkiem minimalizacji funkcji kryterialnej jest aktualizowanie w każdej iteracji algorytmu wektorów s_{ij}, definiujących liniowe rozmaitości w taki sposób, że s_{ij} jest j-tym jednostkowym wektorem własnym macierzy kowariancji C_i $(j=1,...,p)$, zaś $\{s_{ij}\}$ są uporządkowane zgodnie z malejącymi wartościami własnymi macierzy C_i.

Algorytm FCV posiada te wadę, że nie bierze się w nim pod uwagę rozciągłości klasterów (np. długości, powierzchni), co może powodować scalenie klasterów współliniowych. Istnieją ulepszone rozszerzenia tego algorytmu [4]. Podstawową wadą omówionych tu algorytmów, opartych na liniowych rozmaitościach jako prototypach, jest konieczność posiadania pewnej wiedzy a priori o kształcie klasterów w posiadanym zbiorze danych, co dla wymiary $p>3$ jest raczej trudne do przewidzenia.

Prototypy innych kształtów

Opracowano również wiele algorytmów, w których prototypy stanowią konkretne kształty, np. hipersfery, hiperelipsoidalne powierzchnie oraz inne uogólnione kształty [4,8,9].

3. Klasteryzacja posybilistyczna

Wskutek ograniczeń algorytmu FCM zwiększony stopień przynależności do jednego klastera powoduje automatycznie zmniejszenie stopnia przynależności do pozostałych klasterów, a intuicyjnie - stopień przynależności powinien być podyktowany odległością od środków klasterów. Zaproponowano więc podejście [10], w którym u_{ij} reprezentuje stopień typowości (kompatybilności) wektora x_k z i-tym klasterem reprezentowanym przez prototyp punktowy v_i.

Zbiór wszystkich możliwych tzw. posybilistycznych macierzy podziału U definiujemy [10]:

$$M_{pc} = \{U \in V_{cN} \mid 1.*, 2.*, 3.\}$$

przy zmienionym warunku 2.*, który usuwa konieczność probabilistycznego sumowania się do jedynki przynależności wektora do wszystkich klasterów:

2.*

$$\underset{1\le k\le N}{\forall} \sum_{i=1}^{c} u_{ik} \le 1$$

Inna jest również funkcja kryterialna:

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m d_{ik}^2 + \sum_{i=1}^{c} w_i \sum_{k=1}^{N}(1-u_{ik})^m ,$$

gdzie:

$$d_{ik}^2 = \| x_k - v_i \|_A^2 = (x_k - v_i)^T A (x_k - v_i)$$

A - macierz dodatnio określona,

$m \in [1,+\infty)$ - współczynnik wagowy, określający stopień rozmycia klasterów,

w_i - wagi.

Minimum globalne dla J_m uzyskuje się dla następującej macierzy U [4,10]:

$$\forall_{\substack{1 \le i \le c \\ 1 \le k \le N}} \quad u_{ik} = \frac{1}{1 + \left(\frac{d_{ik}^2}{w_i} \right)^{\frac{1}{m-1}}}$$

Widać stąd, że w każdej iteracji u_{ik} zależy tylko od odległości x_k od środka v_i, a nie zależy od odległości od pozostałych klasterów.

Jako wartości wag w_i często przyjmuje się odległości, przy których stopień przynależności jest równy *0.5*. Inny sposób, to określenie ich jako wielkości proporcjonalnych do średniej, rozmytej odległości wewnątrz klastera:

$$\forall_{\substack{1 \le i \le c \\ 1 \le k \le N}} \quad w_i = \xi \frac{\sum_{k=1}^{N} (u_{ik})^m d_{ik}^2}{\sum_{k=1}^{N} (u_{ik})^m},$$

gdzie ξ jest stałą zazwyczaj przyjmowaną jako *1*, zaś wartości u_{ik} pochodzą z algorytmu FCM, który używany jest jako sposób inicjalizacji macierzy U^0 w algorytmie posybilistycznym.

Algorytm posybilistyczny

1. Ustalenie liczby klasterów c $(1<c<N)$, $m \in [1,+\infty)$.
2. Inicjalizacja $U^{(0)} \in M_{pc}$, np. wykorzystując algorytm FCM.
3. Ustalenie wag w^i $i=1, ..., c$.
4. Obliczenie środków klasterów $V^{(j)} = [v_1^{(j)},...,v_c^{(j)}]$ przy użyciu $U^{(j)}$:

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

5. Aktualizacja macierzy podziału dla $(j+1)$-szej iteracji:

$$\underset{\substack{1\le i\le c\\ 1\le k\le N}}{\forall} \quad u_{ik} = \begin{cases} \underset{i\in I_k}{\forall} \; 0 \quad \sum\limits_{i\in I_k} u_{ik} = 1 & I_k \neq \varnothing \\ \dfrac{1}{1+\left(\dfrac{d_{ik}^2}{w_i}\right)^{\frac{1}{m-1}}} & I_k = \varnothing \end{cases}$$

6. Jeśli $||U^{(j+1)} - U^{(j)}|| > \varepsilon$, to $j:=j+1$, idź do punktu *4*, w przeciwnym przypadku zakończenie działania algorytmu.

Wprowadzenie takiej reprezentacji u_{ij} znacznie zwiększa odporność na szum i odstające wektory (dla których jest tutaj możliwość przydzielenia małych wartości u_{ij} we wszystkich klasterach, gdyż nie muszą one sumować się do jedynki).

4. Algorytmy hierarchiczne

Podstawową niedogodnością algorytmów iteracyjnej poprawy funkcji kryterialnej jest konieczność określenia przez użytkownika liczby podzbiorów podziału *c*. W przeciwnym przypadku należałoby przeprowadzić optymalizację dla wszystkich możliwych wartości c. Bardzo często liczba *c* nie jest jednak znana i jednym z rezultatów grupowania ma być również ustalenie najbardziej prawdopodobnej liczby grup. Znacznie skuteczniejsze jest w takim przypadku zastosowanie hierarchicznego algorytmu grupowania, którego ogólna idea jest następująca.

Aby pogrupować *N* obiektów w *c* klasterów najpierw te obiekty grupowane są w *N* klasterów (każdy obiekt stanowi oddzielny klaster). W następnym kroku znajduje się *2* klastery najbardziej podobne do siebie, które następnie ulegają połączeniu, co daje w wyniku *N-1* klasterów. W następnym kroku grupujemy znów *2* najbardziej podobne, uzyskując *N-2* klastery, później *N-3* i tak dalej aż do uzyskania c klasterów.

Metodę klasteryzacji hierarchicznej opisuje następująca procedura:

1. $C_i = \{x_i\}$ $i=1, ..., N$, $c^*=N$.
2. Znajdź *2* klastery najbardziej podobne C_k oraz C_j.
3. Połącz C_k oraz C_j, usuń Cj, $c^*=c^*-1$.
4. If $c^*>c$ idź do punktu *2*.

Początkowym podziałem jest podział, w którym każda próbka tworzy jednoelementowy zbiór. W każdym kroku łączy się ze sobą dwie najbardziej podobne grupy. Ostatecznym zaś wynikiem, jeśli algorytm nie zostanie wcześniej zatrzymany, jest podział, w którym istnieje tylko jedna grupa złożona z wszystkich próbek.

Należy zatem dodatkowo zdefiniować sposób określania podobieństwa między grupami elementów. Istnieje kilka podstawowych miar podobieństwa międzygrupowego. Podobieństwo pomiędzy klasterami może być mierzone poprzez odległość pomiędzy pojedynczymi obiektami w klasterach. Najpopularniejsze miary to:

metoda najbliższego sąsiada

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

metoda najdalszego sąsiada

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

poprzez odległość pomiędzy wszystkimi obiektami w klasterach

$$d_{avg}(X_i, X_j) = \frac{1}{n_i n_j} \sum_{x \in X_i} \sum_{y \in X_j} \| x - y \|^2$$

poprzez odległość pomiędzy środkami klasterów:

$$d_{mean}(X_i, X_j) = \| m_i - m_j \|$$

Użycie różnych definicji podobieństwa wpływa na różny kształt klasterów.

Klasteryzację hierarchiczną można przedstawić graficznie w postaci dendrogramu, czyli drzewa binarnego, w którym każdy węzeł reprezentuje klaster. Przecięcie poziome dendrogramu daje jedną z możliwych klasteryzacji.

W praktyce zadaniem użytkownika jest sformułowanie pewnej reguły, określającej w której iteracji należy przerwać działanie algorytmu. Regułą tą może być np. podanie żądanej liczby podzbiorów podziału lub podanie pewnej progowej wartości. Algorytm zatrzymuje się, gdy w danym kroku podobieństwo między dwiema najbardziej zbliżonymi grupami przekracza zadaną wartość progową. W takiej sytuacji użytkownik nie musi dokonywać oszacowania spodziewanej liczby grup.

5. Klasteryzacja neuronowa

Neuronowe sieci samoorganizujące się stosowane są również w zadaniu grupowania danych. Do najbardziej znanych przykładów należy zaliczyć sieć Kohonena [13] z regułą uczenia z rywalizacją (ang. competitive learning) oraz sieć rezonansową typu ART. [13]. Są to sieci jednowarstwowe. W sieci Kohonena na przykład uczenie polega na korekcji wag $w_m = (w_{m1}, w_{m2}, ..., w_{mp})^T$ tylko wygrywającego neuronu (tj. najbardziej podobnego do wektora wejściowego) wedle reguły:

$$\Delta w_m = \alpha(x - w_m),$$

gdzie $\alpha > 0$ jest dodatnim współczynnikiem zazwyczaj malejącym w miarę postępu nauki. Zwycięski neuron ustala się na podstawie kryterium:

$$w_m^T x = \max_{i=1,\dots,p}(w_i^t x),$$

które wykrywa największy spośród iloczynów skalarnych wektorów wag z wektorem wejściowym, czyli wskazuje wektor wag najbliższy wejściowemu. Po zakończeniu uczenia wektory wag (znormalizowane) wskazują środki ciężkości wykrytych grup obiektów ze zbioru.

Sieć ART ma większą zdolność do adaptacji oraz wykazuje większą stabilność reprezentacji grup danych przez poszczególne neurony od sieci Kohonena. Potrafi również wykrywać grupy bez informacji a priori o ich liczbie, jest jednak mało odporna na wszelkie zakłócenia (szum) w zbiorze danych do grupowania.

6. Klasteryzacja dużych zbiorów danych

W przypadku wielkich zbiorów danych, z jakimi mamy do czynienia np. w eksploracji danych (ang. data mining), czyli setek tysięcy wielowymiarowych wektorów, stosowanie omówionych algorytmów klasteryzacji nie jest sensowne. W tym celu opracowano szybkie, 'aproksymacyjne' algorytmy klasteryzacji, jak np. Clarans (clustering large applications based on random search) [11] czy Birch (balanced, iterative reducing and clustering) [12], w których wykorzystuje się efektywne, dynamiczne struktury danych. W przypadku jednak, gdy nie jest możliwe umieszczenie całego zbioru w pamięci wewnętrznej, stosuje się 2 podejścia:

1. Podejście 'dziel i rządź', w którym w pamięci wewnętrznej klasteryzacji podlega każdorazowo tylko pewna część zbioru, a części poklasteryzowane są później scalane.
2. Podejście inkrementacyjne, w którym w pamięci wewnętrznej znajduje się tylko reprezentacja klasterów, zaś do pamięci ładowane są kolejno pojedyncze wektory.

7. Walidacja klasteryzacji

Algorytmy klasteryzacji opierają się na wielu parametrach, np. liczbie klasterów c, stopniu rozmycia m. Zdefiniowano wiele miar jakości klasteryzacji (tzw. indeksów) w celu umożliwienia wyboru najlepszego podziału danego zbioru w sensie wybranego kryterium. Najczęściej stosuje się różne indeksy w celu określenia optymalnej liczby klasterów, choć można również użyć ich do celu określenia optymalnej wartości innych parametrów. Znajduje

się taki c-podział zbioru, który minimalizuje wartości indeksów. Dwa najprostsze indeksy zdefiniowano następująco [3]:

$$V_{PC}(U) = -\frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^2$$

$$V_{PE}(U) = -\frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N}u_{ik}\ln u_{ik}$$

Ich podstawową wadą jest brak bezpośredniego związku z geometrycznymi własnościami wektorów danych.

Przytoczymy dla przykładu jeszcze indeks Dunna [3]; omówienie niektórych innych indeksów można znaleźć np. w [3].

$$V_D(U) = \min_{1\le i\le c}\min_{\substack{1\le j\le c\\ j\ne i}}\left\{\frac{\delta(\Omega_i,\Omega_j)}{\max\limits_{1\le k\le c}\Delta(\Omega_k)}\right\},$$

gdzie:

$\delta(\Omega_i,\Omega_k) = \min\limits_{\substack{x\in\Omega_i\\ y\in\Omega_j}} d(x,y)$ - odległość i-tego klastera od j-tego

$\Delta(\Omega_k) = \max\limits_{x,y\in\Omega_k} d(x,y)$ - średnica k-tego klastera.

W indeksie Dunna oraz wielu innych wykorzystuje się twardą macierz podziału, którą można otrzymać z macierzy rozmytej poprzez np. następujące przekształcenie (tzw. defuzyfikacja):

$$\forall_{\substack{1\le i\le c\\ 1\le j\le N}}\quad u_{ik}^{(c)} = \begin{cases}1 & \max\limits_{1\le p\le c} u_{pk}^{(f)} = u_{ik}^{(f)}\\ 0 & \textit{przeciwnie}\end{cases}$$

(f) oznacza macierz rozmytą, (c) macierz twardą podziału.

LITERATURA

1. Bezdek J. C.: Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York 1982.
2. Bezdek J. C., et al: Convergence theory for fuzzy c-means: counterexamples and repairs. IEEE Trans. SMC, V.17, No5, 873-877, 1987.
3. Bezdek J. C: Some new indexes for cluster validity. IEEE Trans. SMC, V. 28, No3, 1998.
4. Czogała, J. Łęski: Fuzzy and neuro-fuzzy intelligent systems. Physica-Verlag, Heidelberg 2000.
5. Duda R., Hart P.: Pattern classification and scene analysis. John Wiley&Sons , New York 1973.

6. Gustafson D., Kessel W.: Fuzzy clustering with a fuzzy covariance matrix. Proc. IEEE CDC, San Diego, 761-766, 1979.
7. Jain A., Dubes R. C.: Algorithms for clustering data. Prentice-Hall, NJ 1988.
8. Krishnapuram R., et al.: Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. Part I. IEEE Trans. Fuzzy Systems, V. 3, No 1, 29-43, 1995.
9. Krishnapuram R., et al.: Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. Part II. IEEE Trans. Fuzzy Systems, V. 3, No 1, 44-60, 1995.
10. Krishnapuram R., Keller J.: A possibilistic approach to clustering. IEEE Trans. Fuzzy Systems, V 1, No 2, 98-110, 1993.
11. Ng R., Han J.: Very large databases. Proc. VLDB, Berkeley, 144-155, 1994.
12. Zhang T. et al.: An efficient data clustering method for very large databases. Sigmod, V. 25, No 2, 103-114, 1996.
13. Żurada J. et al.: Sztuczne sieci neuronowe. PWN, Warszawa 1997.

Recenzent: Dr inż. Krzysztof Fujarewicz

Wpłynęło do Redakcji 22 grudnia 2003 r.

Abstract

This paper presents an overview of pattern clustering methods with a goal of providing useful advice and references to fundamental concepts accessible to the broad community of clustering practitioners. We present a taxonomy of clustering techniques and describe the most important representatives of two main groups of clustering algorithms: partitional and hierarchical ones.

Among the partitional algorithms we described the algorithms based on point as well as non-point prototypes. Point-prototypes methods include: the popular ISODATA [1] clustering, fuzzy clustering method (FCM) of Bezdek [1] and the algorithm of Gustafson/Kessel [6] which uses different norms for each cluster. The fuzzy c-varieties algorithm of Bezdek [1] is the most simple representative of algorithms based on linear prototypes. The possibilistic approach to clustering of Krishapuram [10] in which the membership values may be interpreted as compatibilities of the points with class prototypes.

In hierarchical algorithms [5] we described the four different similarity measures: nearest and farthest neighbor, mean as well as prototype distances.

Finally, the problem of cluster validity together with simple validation indexes is shortly described.

Niniejsza praca była finansowana z grantu nr 5T12A00123.

Adresses

Katarzyna STĄPOR: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, delta@ivp.iinf.polsl.gliwice.pl .