УДК 681.322

Ю. Стех, Файсал М. Сардіх, М. Лобур, М. Домброва, В. Арцибасов
Національний університет “Львівська політехніка”,
кафедра систем автоматизованого проектування

ДОСЛІДЖЕННЯ І РОЗРОБЛЕННЯ МЕТОДІВ І АЛГОРИТМІВ НЕІЄРАРХІЧНОЇ КЛАСТЕРИЗАЦІЇ

Розроблено і досліджено методи й алгоритми неієрархічної кластеризації, які дають змогу визначити оптимальну початкову кількість кластерів без будь-якої початкової інформації про їхнє розміщення. Розроблені методи і алгоритми досліджено на відомому тестовому наборі Iris.

Ключові слова: метод, алгоритм, кластеризація, розміщення кластерів.

Developed and studied the methods and non-hierarchical clustering algorithms for determining the optimal initial number of clusters without any background information on the location of the clusters. The methods and algorithms are studied in the famous test set Iris.

Keywords: method, algorithm, clustering, location of the clusters.

Вступ

Кластеризація є однією з фундаментальних задач в області інтелектуального аналізу даних (Data Mining), інтелектуального аналізу даних в Internet (Web Mining), інтелектуального аналізу текстових даних і документів (Text Mining), машинного навчання [2, 5, 6]. Основним завданням процесу кластеризації є розподіл заданої множини образів на класи (кластери), котрі дають змогу досліджувати подібність і відмінність між образами в класах і робити обґрунтовані висновки про кластерні образи в кластерах. Під час кластеризації відсутня будь-яка інформація про наперед задані класи і тому процес кластеризації належить до задач некерованого машинного навчання (unsupervised learning). Результат процесу кластеризації залежить від ряду факторів, визначальними з яких є метод і алгоритм кластеризації, початкові параметри алгоритму кластеризації. Основні кроки процесу кластеризації зображено на рис. 1.

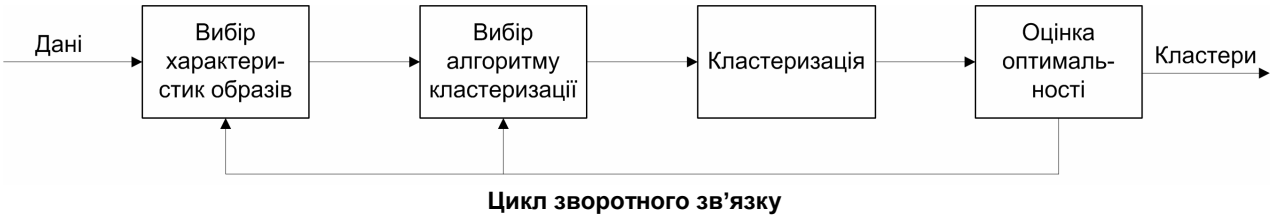


Рис. 1. Основні кроки процесу кластеризації

На етапі вибору характеристик образів формуються вектори ознак образів, які найповніше відображають властивості об’єктів, котрі кластеризуються.

На етапі вибору алгоритму кластеризації здійснюється вибір одного із алгоритмів, котрі містяться у бібліотеці алгоритмів кластеризації. Цей етап передбачає вибір міри подібності образів у кластері та критеріїв кластеризації. Міра подібності покладена в основу правила належності образу до певного кластера. Критерій кластеризації визначає процес зупинки алгоритму кластеризації.

На етапі кластеризації відбувається кластеризація заданої множини образів вибраним алгоритмом, якщо вибрані міри подібності і критерії кластеризації.

На етапі оцінки результатів кластеризації оцінюють оптимальність розбиття заданої множини образів на кластери. На цьому етапі застосовують декілька точних та/або наближених критеріїв оптимальності.

Тому для оптимального розбиття на кластери треба побудувати та дослідити ансамбль методів та алгоритмів кластеризації і застосувати декілька критеріїв оптимальності розбиття на кластери.

Формальна постановка задачі

Нехай $D = \{\overline{\mathbf{X}}_1, \overline{\mathbf{X}}_2, ..., \overline{\mathbf{X}}_n\}$ – множина n образів, кожен з яких володіє d ознаками. Нехай $L = \{A_1, A_2, ..., A_m\}$ – множина алгоритмів кластеризації. Кожен алгоритм кластеризації A_i генерує розбиття для множини D на кластери $P^{(i)} = \{S_1^{(i)}, S_2^{(i)}, ...S_m^{(i)}\}$. Нехай $S = \bigcup_{i=1}^{m} P^{(i)}$ – множина всіх кластерів, котрі отримані множиною алгоритмів L. Мета роботи ансамблю методів та алгоритмів – оптимальне розбиття на кластери $T = \{C_1, C_2, ...C_k\}$, яке відповідає оптимальним максимальним або мінімальним значенням критеріїв оптимальності. При цьому елементи множини T повинні задовольняти такі властивості:

- Кожен кластер повинен мати хоча б один образ $C_i \neq \emptyset \ \forall i \in \{1,2,...,k\}$.
- Кожен образ повинен належати хоча б одному кластеру $C_i \cap C_j = \emptyset \ \forall i \neq j, i, j = \{1,2,...,k\}$.
- Усі образи повинні бути рознесені по кластерах $\bigcup_{i=1}^{k} C_i = P$.

Отже, задача кластеризації зводиться до оптимізаційної задачі, яка вимагає розроблення і дослідження ансамблю методів і алгоритмів і аналізу певних критеріальних функцій.

Аналіз відомих розв'язків проблеми

Відомі алгоритми кластеризації можна поділити на ієрархічні та неієрархічні [2, 6].

В ієрархічних алгоритмах кластеризація не потребує визначення кількості кластерів. При цьому будують дерево вкладених кластерів (дендрограму). Кількість кластерів визначається із припущень, які не відносять до роботи алгоритму. Проблеми таких алгоритмів добре відомі: вибір міри близькості кластерів, проблема інверсної індексації в дендрограмах, негнучкість ієрархічної кластеризації.

В неієрархічних алгоритмах кластеризації характер їх роботи і умови зупинки необхідно задати заздалегідь за допомогою вхідних параметрів роботи алгоритму. Найважливішим з них є кількість бажаних кластерів.

Далі розглянемо неієрархічні методи й алгоритми кластеризації.

Розроблення методів і алгоритмів неієрархічної кластеризації

Проблему оптимального вибору початкової кількості кластерів пропонується розв'язати за допомогою розробки ансамблю методів і алгоритмів неієрархічної кластеризації.

Загальна структура і порядок використання алгоритмів наведені на рис. 2.

Основною особливістю розробленої бібліотеки алгоритмів неієрархічної кластеризації є те, що проблема вибору кількості кластерів вирішується за допомогою двох алгоритмів: евристичного алгоритму пошуку центрів кластерів і алгоритму пошуку центрів кластерів за допомогою нейронної мережі [7].

Евристичний алгоритм пошуку центрів кластерів (ЕАПЦК) призначений для знаходження центрів кластерів у заданій множині образів без жодної початкової інформації про розміщення центрів кластерів. Алгоритм реалізований у вигляді двох модифікацій. Перша модифікація знаходить найвіддаленіші між собою точки центрів кластерів. Після знаходження центрів кластерів решта образів розподіляються по кластерах за критерієм мінімуму евклідової відстані до центрів кластерів.

Рис. 2. Структура бібліотеки алгоритмів неієрархічної кластеризації

Нехай $\overline{\mathbf{X}} = \{\bar{x}_1, \bar{x}_2, ..., \bar{x}_n\}$ – множина точок образів, $\overline{\mathbf{Z}} = \{\bar{z}_1, \bar{z}_2, ..., \bar{z}_n\}$ – шукані центри кластерів.

Крок 1. Вибрати випадковим чином точку центра першого кластера k.

Крок 2. $\bar{z}_1 = \bar{x}_k, l = n-1$.

Крок 3. Обчислити евклідову відстань від решти точок множини образів до центра першого кластера $D_{i1} = \|\bar{x}_i - \bar{z}_1\|, i = 1(1)l$.

Крок 4. $K_i^{(1)} = \max_i \{D_{i1}\}, L_1 = K_i^{(1)}, p = i$.

Крок 5. Вибрати центр другого кластера $\bar{z}_2 = \bar{x}_p, l = l-1$.

Крок 6. Обчислити евклідову відстань від решти точок множини образів до центрів першого і другого кластерів. $D_{i1} = \|\bar{x}_i - \bar{z}_1\|$, $D_{i2} = \|\bar{x}_i - \bar{z}_2\|, i = 1(1)l$.

Крок 7. $A_i = \min_i \{D_{i1}, D_{i2}\}$.

Крок 8. $K_i^{(2)} = \max_i \{A_i\}, L_2 = K_i^{(2)}, p = i$.

Крок 9. Якщо $L_2 > S \cdot L_1$, тоді $\bar{z}_3 = \bar{x}_p, l = l-1$, в іншому випадку STOP.

Крок 10. Обчислити $L_{c.a.} = \dfrac{L_1 + L_2}{2}$.

Крок 11. Обчислити евклідову відстань від решти точок множини образів до центрів першого, другого та третього кластерів: $D_{i1} = \|\bar{x}_i - \bar{z}_1\|$, $D_{i2} = \|\bar{x}_i - \bar{z}_2\|$, $D_{i3} = \|\bar{x}_i - \bar{z}_3\|, i = 1(1)l$.

Крок 12. Обчислити $A_i = \min_i \{D_{i1}, D_{i2}, D_{i3}\}, i = 1(1)l$.

Крок 13. Обчислити $K_i^{(3)} = \max_i \{A_i\}, L_3 = K_i^{(3)}, p = i$.

Крок 14. Якщо $L_3 > S \cdot L_{c.a.}$, тоді $\bar{z}_4 = \bar{x}_p, l = l-1$, в іншому випадку STOP.

$\bar{x}_i \in A_k$, якщо $\|\bar{x}_i - \bar{z}_k\| < \|\bar{x}_i - \bar{z}_r\|, r = 1(1)l, m \neq k$.

…….....

.....

Параметр S вибирається в межах $S \in (0,1)$.

Однак така побудова алгоритму знаходить найвіддаленіші центри кластерів, що не завжди приводить до оптимального кінцевого результату.

Тому в другій модифікації спочатку знаходять геометричний центр множини точок образів $\overline{x_c}$.

$$\bar{x}_c = \frac{1}{|D|} \sum_{i=1}^{n} \bar{x}_i \qquad (1)$$

де $|D|$ – потужність множини точок образів.

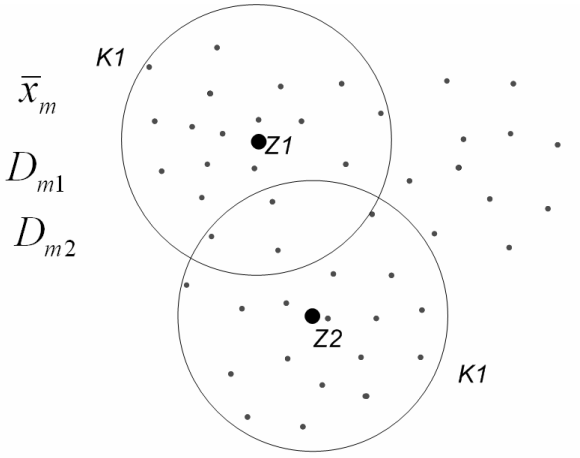
Надалі точка першого центра кластера вибирається як найбільш віддалена точка від $\bar{x}_c$. В багатьох випадках це дає змогу визначити оптимальніші центри кластерів.

Альтернативним методом пошуку початкових центрів кластерів у розробленому ансамблі методів і алгоритмів є запропонований у [7] алгоритм пошуку центрів кластерів за допомогою нейронної мережі. В цьому алгоритмі досліджувана множина точок образів подається за допомогою нейронної мережі, де кожному образу ставиться у відповідність нейрон, і досліджувана множина точок образів у d-вимірному просторі перетворюється на певний неорієнтований зважений граф нейронної мережі. Алгоритм працює так, що нейрони, які містяться на межі кластерних областей, передають свої активності нейронам, які містяться всередині областей кластерів. Процес навчання нейронної мережі сходиться до такого результату, коли в кожній області кластера залишається лише один активний нейрон – центр кластера.

Такий підхід до знаходження центрів кластерів дає змогу надалі використати відомі алгоритми k-means і k-medoids [4, 6] без задання початкових параметрів для них.

Бібліотека алгоритмів містить відомий алгоритм порогової величини [4]. Основним недоліком алгоритму є те, що він вимагає задання порогової величини T і результат його роботи залежить від вибору початкової точки – першого центра кластера. Окрім того, в результаті роботи алгоритму може утворитися ряд кластерів, котрі дають перетини областей кластеризації і вимагають додаткових евристичних процедур для визначення належності точок образів до певних кластерів.

Ми розробили і ввели в бібліотеку алгоритмів модифікований алгоритм порогової величини (рис. 3). Цей алгоритм являє собою комбінацію класичного алгоритму порогової величини й алгоритму k-means. Особливість розробленої комбінації полягає в тому, що, на відміну від алгоритму порогової величини, у кожній області кластеризації спочатку обчислюється геометричний центр множини образів і множину точок образів цієї області надалі вилучають з розгляду як черговий кластер. Процес кластеризації продовжується доти, доки ми не отримаємо пусту множину точок образів.



$$D_{i1} = \left\| \bar{x}_i - \bar{z}_{11} \right\|$$

$$D_{i1} < T \rightarrow \bar{x}_i \in K_1$$

$$\bar{z}_{12} = \frac{1}{n} \sum_{i=1}^{n} \bar{x}_i$$

$$\left\| \bar{z}_{1i+1} - z_{1i} \right\| \leq \varepsilon$$

Ai – множина образів (точок)в кластері Ki

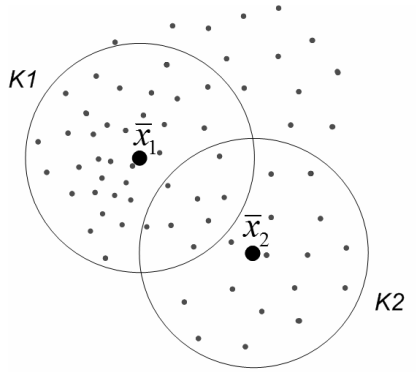
Bo – початкова множина образів

$$B_{i+1} = B_i \setminus A_i$$

$$B_{i+1} = \varnothing \rightarrow STOP$$

Рис. 3. Модифікований алгоритм порогової величини

Розроблений також алгоритм якісного порогу, що являє собою покращаний модифікований алгоритм порогової величини (рис. 4).



Bo – початкова множина образів

$$|B_0| = m$$

$$K_1 = n_1, K_2 = n_2, ..., K_m = n_m$$

$$g_i = \max_i (n_1, n_2, ..., n_m)$$

Ki – перший кластер

$$B_1 = B_0 \setminus K_i$$

.....

$$B_{p+1} = B_p \setminus K_i$$

$$B_{p+1} = \varnothing \rightarrow STOP$$

Рис. 4. Алгоритм якісного порогу

На відміну від модифікованого алгоритму порогової величини, в цьому алгоритмі спочатку обчислюються кластери для кожної точки образу заданої множини за допомогою кроків модофікованого алгоритму порогової величини.

Відтак обчислюють кількості точок образів у кожній області кластеризації і за перший кластер приймають область з максимальною кількістю точок образів. Ця множина точок вилучається з подальшого розгляду. Такий ітеративний процес продовжується доти, доки ми не отримаємо пусту множину точок образів. Очевидно, що цей алгоритм має найвищу складність серед усіх алгоритмів бібліотеки.

Розроблені алгоритми і комбінації алгоритмів протестовано на відомому тестовому наборі Iris [5]. Тестовий набір Iris містить дані про квіти ірису (Iris Setosa, Iris Versicolour, Iris Virginica). Кожна квітка характеризується чотирма атрибутами – довжина чашолистка, ширина чашолистка, довжина пелюстки, ширина пелюстки (рис. 5). Набір даних містить три кластери.

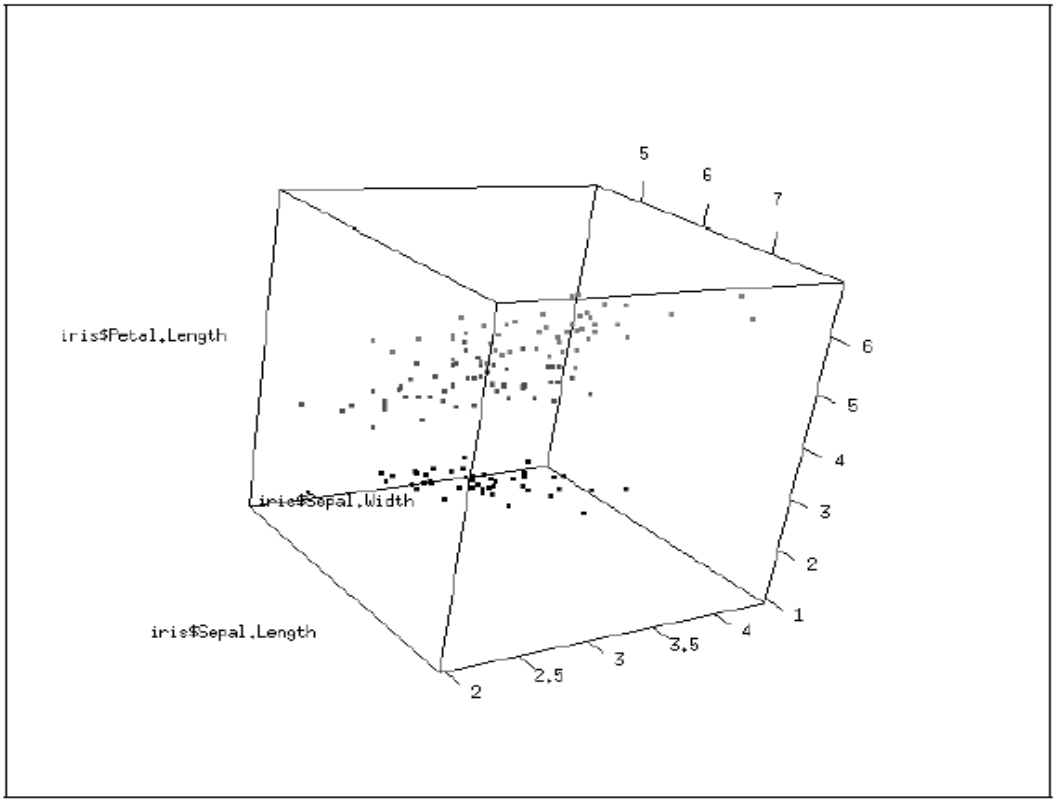


Рис. 5. Набір даних Iris

Результати тестування наведено в таблиці

Результати дослідження

Результати / Функція відстані	Алгоритм	Відхилення, %	Час роботи алгоритму, мс	Сер. розмір кластера	Сер. міжкластерна відстань	Сер. відстань між центрами кластерів
Евклідова відстань	K-means	14,67	6419	0,9212	3,473	3,366
	ЕАПЦК + K-maens	16	3510	0,9183	3,487	3,4
	ЕАПЦК + K-medoids	16	2787	0,9183	3,487	3,433
Квадрат евклідової відстані	K-means	12	2694	1,099	12,33	11,133
	ЕАПЦК + K-means	20	3026	1,068	14,6	13,766
	ЕАПЦК + K-medoids	20	2797	1,068	14,6	13,666
Степенева відстань (p=4, r=2)	K-means	10,67	3273	0,7582	9,159	8,433
	ЕАПЦК + K-means	20	4072	0,7431	10,77	10
	ЕАПЦК + K-medoids	17,33	2276	0,7425	10,6	10,466

Висновки

Розроблені методи і алгоритми неієрархічної кластеризації, а також бібліотека програм дають змогу знаходити оптимальне розбиття на кластери за допомогою обчислення кластерів різними алгоритмами при різних початкових параметрах і за допомогою визначення таких критеріальних функцій, як середня міжкластерна відстань, середній об’єм кластерів і середня відстань між їхніми центрами.

1. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Модели и методы анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 236 с. 2. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Технологии анализа данных:Data Mining, Visual Mining, Text Mining, OLAP. – СПб.: БХВ-Петербург, 2007. – 384 с. 3. Ким Дж.-О., Мьюллер Ч.У., Клекка У.Р. Факторный, дискриминантный и кластерный анализ: пер. с англ. – М.: Финансы и статистика 1989. – 215 с. 4. Ту Дж., Гонсалес Р. Принципы распознавания образов: пер. с англ. – М.: Мир, 1978. – 411 с. 5. Классификация и кластер / под ред. Дж. Вэн Райзина: пер. с англ. – М.: Мир. – 389 с. 6. Jain A.J., Murty M.N., Flynn P.J. Data clustering: a review // ACM Computing Surveys, 1999. – V. 31, № 3. – P. 264–323. 7. Стех Ю.В., Файсал М.Е. Сардіх, Лобур М.В., Керницький А.Б. Алгоритм пошуку оптимальної кількості кластерів // Вісник Національного університету «Львівська політехніка». – 2009. – № 651. – С. 129–132.