УДК 004.855.5
DOI: 10.18523/2617-3808.2020.3.83-87

Крещенко Т. О., Ющенко Ю. О.

МЕТОД КЛАСТЕРИЗАЦІЇ З ВИКОРИСТАННЯМ БАГАТОВИМІРНОГО АДРЕСНОГО СОРТУВАННЯ

У роботі розглянуто багатовимірне адресне сортування, зокрема декілька методів його реалізації. Описано декілька структур даних для збереження та використання результатів багатовимірного адресного сортування. На прикладі реалізованого програмного проекту продемонстровано корисність і доцільність використання багатовимірного адресного сортування для розв'язання задач класифікації сукупностей згрупованих даних. Визначено переваги використання багатовимірного адресного сортування при розв'язанні задач кластеризації порівняно з методами, які нині набули широкого використання.

Ключові слова: адресне сортування, списки, двозв'язні списки, дерева, індексація, класифікація, кластерування, кластеризація, кластерний аналіз, машинне навчання.

Багатовимірне адресне сортування – дуже цікава тема, яка недостатньо досліджена, на відміну від методів сортування даних, які вельми детально вивчені, описані та широко використовуються на практиці. Настав час, коли інформатизація всіх сфер діяльності та відпочинку людей потребує якісного стрибка до застосування засобів штучного інтелекту. Одним із напрямів інтелектуалізації програмного забезпечення, яке має широке застосування, є машинне навчання, яке безпосередньо пов'язано з методами кластеризації сукупностей згрупованих даних.

Запропоновано новий підхід до розв'язання задач кластеризації, в основу якого покладено застосування багатовимірного адресного сортування. Саме цим визначається актуальність і новизна цієї роботи. Доцільність такого підходу для розв'язання задач класифікації продемонстровано на прикладі розробленого програмного проекту.

У роботі описано декілька структур даних для збереження та використання результатів багатовимірного адресного сортування.

Від давніх часів люди шукали способи спростити взаємодію з різноманітними об'єктами та інформацією про них. Одним із найефективніших завжди було впорядкування. Прикладами впорядкування є розміщення книжок на полиці в бажаному порядку або інструментів на столі в тесляра. Людям завжди приємніше мати справу з тим, де панує порядок. Зокрема, порядок прискорює пошук потрібних елементів.

Із появою комп'ютерів при розв'язанні інформаційних задач програмісти розробили та реалізували різноманітні методи впорядкування даних про об'єкти. Жоден навчальний курс із програмування не оминає вивчення методів сортування даних, яке дає змогу за сталий час визначити перший/найменший та останній/найбільший елемент даних і прискорює час знаходження інформації про потрібний елемент.

Розвиток технічних і програмних засобів оброблення інформації дав людству змогу зберігати величезні обсяги інформації. Зокрема, обсяги інформації в інтернеті зростають дуже швидко, і цей процес не має обмежень. Тільки за 2016–2017 рр. було згенеровано 90 % наявних на той час інтернет-даних [3], а чим більше інформації, тим більшою є потреба її впорядкувати. Це досить легко довести, оскільки сортування дає можливість використовувати алгоритм бінарного пошуку. Він спрощує часову складність знаходження елемента в масиві з O(n) до O(log(n)), що на перший погляд може видатися невеликим покращенням. Проте коли мова йде про квадрильйони одиниць даних (Big Data), без логарифмічного часу не обійтися.

Загальний опис багатовимірного сортування

Відомо, що в ІТ задачу сортування досліджено майже досконало. Неодноразово було розглянуто велику кількість алгоритмів упорядкування та доведено складність цієї задачі. Тоді в чому ж полягає проблема такого сортування, до якого всі звикли? Невже настав час говорити про квантові комп'ютери та сортування за O(n)? На жаль, ця робота про дещо інше, але не менш важливе.

У реальному світі інформація про певний об'єкт, як правило, є у вигляді не простих (елементарних) даних, як-от число, рядок або булеан, а у вигляді запису (*англ*. record). У мові програмування Python ця структура більш відома як кортеж (*англ*. tuple), а саме, статичний масив елементів довільного типу. У базі даних – це рядок реляції, де міститься інформація про певний об'єкт. Коли мова йде про сортування кортежів, то, як правило, це відбувається за деяким елементом цих записів.

По-перше, у багатьох випадках доцільно мати можливість сортувати дані не лише за одним певним елементом, а за багатьма. По-друге, прийнято копіювати або переміщувати дані при сортуванні, що може бути дуже ресурсоємним, коли мова йде про кортежі великого обсягу.

Для розв'язання цих проблем пропонують багатовимірне адресне сортування. Ю. О. Ющенко дав таке визначення цьому терміна: *«Під багатовимірним впорядкуванням будемо розуміти сортування за багатьма ознаками чи критеріями та збереження цих результатів таким чином, що від будь-якого елемента сукупності можна швидко (миттєво, без використання пошуку та/або сортування) перейти до інформації щодо іншого елемента, який є наступним чи попереднім за однією з ознак/характеристик, за якою було здійснено сортування. Під виміром будемо розуміти ознаку, характеристику сукупності об'єктів, за якою було здійснено адресне сортування*» [1; 2]. При цьому створюється нова послідовність адрес даних, яка посилається на початковий масив, не змінюючи його. Таким способом, на виході є будь-яка кількість варіантів упорядкувань, досягнута без копіювання чи переміщення початкових даних. Різні структури даних можуть бути використані задля зберігання результатів багатовимірного сортування та подальшого використання.

Використання масивів індексації за багатовимірного сортування

Найпростіша структура, яку можна використати для зберігання варіантів сортування, – це масиви індексації, тобто масиви вказівників або посилань на кортежі в потрібному порядку. На рис. 1 наведено приклад початкової сукупності кортежів і масив індексації, в якому збережено результати тривимірного сортування.

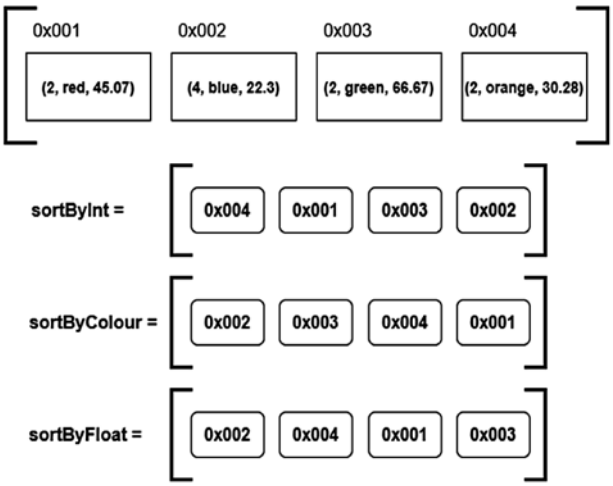

Рис. 1. Візуалізація результату тривимірного сортування з використанням масивів індексації

Масиви індексації використано для збереження результатів сортування за різними елементами кортежу. Часова складність отримання 1-го та *n*-го елемента за значенням елементів кортежів, за якими проведено сортування, – O(1). Також це найпростіша з погляду реалізації та використання структура. Недоліком є те, що перехід від елемента до попереднього (наступного) займає O(log(n)), якщо невідомий індекс поточного елемента, що є цілком можливим, насамперед коли змінюється варіант сортування.

Шляхом розв'язання цієї проблеми є використання зворотного масиву індексації, у якому елемент можна отримати не за індексом, а навпаки. Така структура дасть змогу за O(1) отримати індекс поточного елемента. Відповідно, кількість таких масивів має дорівнювати кількості вимірів, що дасть змогу в будь-який час перейти до наступного (попереднього) елемента в будь-якому вимірі за одиничний час. Часова складність алгоритму побудови такого масиву – O(n).

Використання двозв'язних списків за багатовимірного сортування

Така структура, як двобічно зв'язаний список, може бути корисною для переходу від певного об'єкта до наступного або попереднього за будь-яким із сортувань із часовою складністю O(1), тобто за одну одиницю часу. На рис. 2 зображено UML-діаграму класу вузла, що може бути використаний для реалізації такого двозв'язного списку.

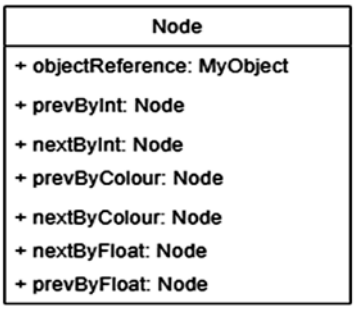

Рис. 2. Приклад реалізації вузла двобічно зв'язаного списку

У вузлі міститься відсилка на сам об'єкт, а також вказівники на наступні та попередні об'єкти для кожного з вимірів сортування. У разі якщо цей об'єкт є першим або останнім, то відповідний вказівник на наступний або попередній елемент матиме значення null. Таким способом досягається складність O(1) для переходу до наступних (попередніх) елементів. Головним недоліком такої реалізації є те, що часова складність отримання елемента за індексом – O(n).

Використання зовнішніх ключів за багатовимірного сортування

Ще одна можлива реалізація збереження результатів багатовимірного сортування – це використання зовнішніх ключів у реляційних базах даних. На рис. 3 зображено одну з можливих реалізацій сутності вузла.

Node	
PK	id
FK	objectReference
FK	prevByInt
FK	nextByInt
FK	prevByColour
FK	nextByColour
FK	prevByFloat
FK	nextByFloat

Рис. 3. Приклад сутності в реляційній базі даних

Така реалізація є дуже схожою на двобічно зв'язаний список, проте є декілька відмінностей. По-перше, наявність унікального номера як головного ключа (*англ*. Primary Key, PK) забезпечує можливість знаходження елемента за індексом за O(log(n)), що є краще, ніж O(n). Перехід до наступних (попередніх) елементів все ще займає O(1). По-друге, збереження сортувань у базі даних унеможливлює втрату даних у разі зупинки програми, що інакше змусило б знову виконати сортування елементів. Недоліком такої реалізації, на думку авторів, є загроза надмірного навантаження бази даних.

Отже, багатовимірне адресне сортування розв'язує проблему копіювання даних під час їх сортування, а також дає можливість зберігати декілька варіантів сортування, коли це необхідно. Зазначимо, що таке сортування є зручним і корисним не тільки для оброблення даних і насамперед розширення функціоналу програмного забезпечення, а й для покращення зручності інтерфейсу користувача [1; 2].

Класифікації даних із використанням багатовимірного сортування

Машинне навчання стає все більш широковживаним. Стрімкий розвиток інформаційних технологій у XXI ст. зробив можливим застосування методів машинного навчання майже до будь-якої галузі. У випадку застосування багатовимірного сортування має сенс поєднати його з використанням методів групування об'єктів.

Серед них було розглянуто задачу класифікації, в якій відомо назви класів та потрібно визначити, який об'єкт до якого класу належить; і задачу кластеризації, в якій не відомо назви класів і потрібно створити кластери об'єктів (згрупувати об'єкти) залежно насамперед від об'єктів.

Кластеризація, також відома як кластерний аналіз, має за мету знаходження груп схожих об'єктів у вибірці, що зазвичай використовується для подальшого аналізу кожного кластера. Одна з найбільших переваг багатовимірного сортування – це можливість зручно переглядати наступні та попередні за різними ознаками об'єкти щодо якогось об'єкта. Це може бути доволі корисно саме в аналізі вже створених кластерів, оскільки надається можливість аналізувати різниці в ознаках схожих об'єктів в одному кластері та наочно бачити їх близькість.

Приклад застосування методу класифікації даних із використанням багатовимірного сортування

Для демонстрації та опробування методу кластеризації з використання багатовимірного сортування обрано задачу класифікації країн світу за сукупністю ознак захворювання на COVID-19.

	country	casesPerMil	deathsPerMil	cases	deaths	recovered	mortality
0	Abkhazia	12	4	3	1	2	0.333333
1	Afghanistan	105	3	3392	104	458	0.185053
2	Albania	292	11	832	31	570	0.051581
3	Algeria	113	11	4997	476	2197	0.178077
4	Andorra	9685	593	751	46	514	0.082143
5	Angola	1	0	36	2	11	0.153846
6	Anguilla	202	0	3	0	3	0.000000
7	Antigua and Barbuda	249	31	24	3	11	0.214286
8	Argentina	111	6	5007	264	1459	0.153221
9	Armenia	941	16	2782	47	1111	0.040587
10	Artsakh	47	0	7	0	0	0.000000
11	Aruba	899	18	101	2	89	0.021978
12	Australia	264	4	6794	97	5980	0.015962
13	Austria	1751	68	15589	608	13639	0.042676
14	Azerbaijan	205	3	2127	28	1536	0.017903
15	Bahamas	231	29	89	11	26	0.297297
16	Bahrain	2410	5	3842	8	1860	0.004283
17	Bangladesh	70	1	11719	186	1403	0.117055
18	Barbados	286	24	82	7	47	0.129630
19	Belarus	2047	12	19255	112	4388	0.024889
20	Belgium	4403	723	50781	8339	12731	0.395776
21	Belize	44	5	18	2	16	0.111111
22	Benin	8	0	96	2	50	0.038462
23	Bermuda	1796	109	115	7	54	0.114754
24	Bhutan	9	0	7	0	5	0.000000
25	Bolivia	157	7	1802	86	187	0.315018
26	Bosnia and Herzegovina	590	24	1987	86	928	0.084813
27	Botswana	10	0	23	1	8	0.111111
28	Brazil	550	38	121600	8022	48221	0.142631
29	British Virgin Islands	200	33	6	1	3	0.250000

Рис. 4. Фрагмент датасету

Програму було написано на мові Python із використанням бібліотек numpy, pandas, seaborn, matplotlib, а також scikit-learn. Було взято інформацію про кількість випадків на мільйон осіб, кількість смертей на мільйон осіб, а також загальну кількість випадків, смертей та одужань у кожній країні, та збережено у CSV-файлі. Використано реалізацію алгоритму DBSCAN із бібліотеки scikit-learn.

Для розв'язку обраної задачі вибрано датасет (масив даних), фрагмент якого зображено на рис. 4.

Кластеризацію буде проведено за кількістю випадків на мільйон осіб (casesPerMil) і за смертністю (mortality). Наведений датасет було спрощено видаленням несуттєвих, «зайвих» ознак.

На рис. 5 показано, що розподіл даних за обраними ознаками є логарифмічним. Тому застосовано логарифмічну функцію і надалі порівнюються країни у логарифмічному співвідношенні.

На рис. 6 наведено розподіл після застосування логарифмічної функції.

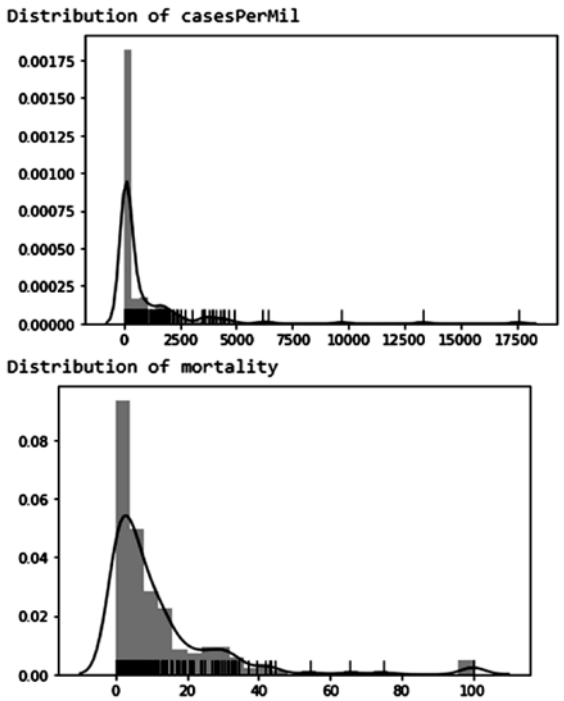

Рис. 5. Розподіл даних за обраними ознаками

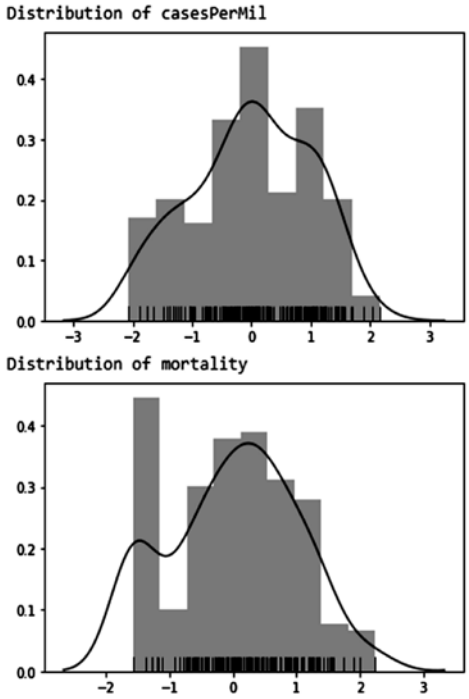

Рис. 6. Розподіл даних за обраними ознаками після застосування до них логарифмічної функції

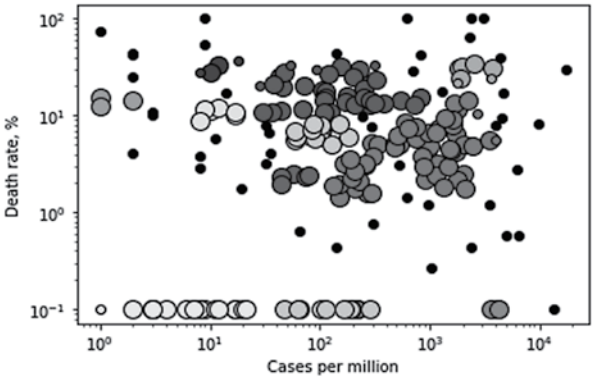

Рис. 7. Графік створених кластерів

Далі виконується алгоритм кластеризації DBSCAN і виводиться результат у вигляді графіка, де кожна точка має дві раніше обрані ознаки: смертність по вісі Oy та кількість випадків на мільйон осіб по вісі Ox (див. рис. 7). Різними відтінками сірого позначено різні кластери.

Далі було здійснено багатовимірне сортування. Створено шість масивів для сортування за кожною ознакою, кожен із яких містить масиви індексації для кожного окремого кластера.

Багатовимірне адресне сортування у поєднанні з кластеризацією – це чудовий ефективний спосіб наочно показати результат класифікації. Розроблена програма досить добре демонструє потенціал використання запропонованого методу сортування на практиці.

У подальшому планується вдосконалення використання багатовимірного адресного сортування для покращення наочності за сприймання результатів класифікації.

Список літератури

1. Ющенко Ю. О. Багатовимірне впорядкування та його використання для вдосконалення інтерфейсу користувачів інформаційних систем / Ю. О. Ющенко // Наукові записки НаУКМА. Комп'ютерні науки. – 2018. – Т. 1 . – С. 10–13.
2. Ющенко Ю. О. Використання багатовимірного впорядкування для наочного та зручного доступу до інформації / Ю. О. Ющенко // Матеріали XV Міжнародної науково-практичної конф. «Інформаційні технології в економіці, менеджменті і бізнесі. Проблеми науки, практики та освіти» (Київ, 25–26 листопада 2010 р.). – Київ : Вид-во Європ. ун-ту, 2010. – С. 114–115.
3. IBM Marketing Cloud. 10 key marketing trends for 2017 and ideas for exceeding customer expectations [Electronic resource]. – Mode of access: https://paulwriter.com/wp-content/uploads/2017/10/10-Key-Marketing-Trends-for-2017.pdf.

References

IBM Marketing Cloud. (2017). 10 key marketing trends for 2017 and ideas for exceeding customer expectations. Retrieved from https://paulwriter.com/wp-content/uploads/2017/10/10-Key-Marketing-Trends-for-2017.pdf.

Yushchenko, Yu. (2010). Vykorystannia bahatovymirnoho vporiadkuvannia dlia naochnoho ta zruchnoho dostupu do informatsii. In *Materialy XV Mizhnarodnoi naukovo-praktychnoi konferentsii "Informatsiini tekhnolohii v ekonomitsi, menedzhmenti i biznesi. Problemy nauky, praktyky ta osvity"* (pp. 114–115). Kyiv: Vydavnytstvo Yevropeiskoho universytetu [in Ukrainian].

Yushchenko, Yu. (2018). Bahatovymirne vporiadkuvannia ta yoho vykorystannia dlia vdoskonalennia interfeisu korystuvachiv informatsiinykh system. *Naukovi zapysky NaUKMA. Kompiuterni nauky*, *1*, 10–13 [in Ukrainian].

T. Kreshchenko, Yu. Yuschenko

CLUSTERING METHOD USING MULTIDIMENSIONAL ADDRESS SORTING

The paper examines multidimensional address sorting as a way to sort datasets by multiple columns (features of the objects) and to avoid copying data at the same time. Several data structures are proposed for storing and using the resulting data of multidimensional sorting. That includes the usage of indexing arrays, doubly linked lists, and foreign keys in a relational database. Each variant is analyzed in terms of time complexity of performing various tasks. The paper illuminates advantages and disadvantages of using each proposed data structure.

When using indexing arrays, by which arrays that store indices of the elements in the desired order are meant, it becomes impossible to access the next or previous element in any other sort in O(1) time. To resolve this problem the paper proposes to use inverted index arrays, which map data points to their indices in each sort.

The implementation that uses doubly linked lists shows promising time complexity results, but the one that uses foreign keys has proven to be better, because the presence of primary keys allows to get elements by indices in logarithmic time, or O(log(n)). It also reduces the risk of losing data if a critical error occurs in the main program.

One of the main goals of cluster analysis is to make it easier to analyze similar data within a dataset. Multidimentional sorting adds to this goal by simplifying the process of displaying the clustered data and making it possible to compare data points that are close by a chosen feature in any cluster.

The implemented software project is used to demonstrate expediency and convenience of using multidimensional address sorting to display and visualize the results of data clustering. The paper identifies advantages of using multidimensional address sorting in solving clustering problems over methods that are currently widely used.

Keywords: address sorting, lists, doubly linked lists, trees, index, classification, clustering, clusterization, cluster analysis, machine learning.

Матеріал надійшов 10.06.2020