УДК 519.254

О.Б. Назаревич, канд. техн. наук. доц., Г.В. Шимчук
Тернопільський державний технічний університет імені Івана Пулюя

ОПИС ЧИСЕЛЬНОГО МЕТОДУ СИНГУЛЯРНО-СПЕКТРАЛЬНОГО АНАЛІЗУ ДЛЯ АНАЛІЗУ ЧАСОВИХ РЯДІВ

O. Nazarevych, G. Shymchuk

DESCRIPTION OF THE NUMERICAL METHOD OF SINGULAR-SPECTRAL ANALYSIS FOR ANALYSIS OF TIME SERIES

Сингулярний розклад матриці – один із найвідоміших матричних розкладів. Розклад або факторизація – мультиплікативне представлення матриці у вигляді кількох матриць (зазвичай двох або трьох), які мають ті чи інакші властивості. Процес факторизації здійснюється на основі різних лінійних перетворень у відповідних просторах над векторами, які ототожнюються з стовбцями або рядками вихідних матриць, а також матриць проміжних етапів в застосовуваних алгоритмах.

Однією із найбільших переваг є те, що SSA носить адаптивний характер реалізації процесу газоспоживання і не потребує апріорної інформації про його структуру, а регулюється лише одним параметром L –, який підбирається експериментально, опираючись на досвід дослідника.

У багатьох природничих науках склалось уявлення про можливість опису природних процесів за допомогою функції, що складається з декількох доданків (загальна адитивна модель):

$$f(x) = f_T(t) + f_n(t) + f_r(t) + \varepsilon(t), t \in [0, T] \tag{1.1}$$

Де повільна нерегулярна складова, здебільшого називають трендом, $f_n(t)$ – періодична або сума періодичних складових (сезонні, добові тощо), $f_r(t)$ – швидкі нерегулярні малі варіації, в котрі, як правило, включають все, що не вкладається у формальну модель, інколи включають і випадкові шуми, $\varepsilon(t)$ – випадкова складова, що описується випадковим процесом певного типу.

У результаті використання методу відбувається розкладання часового ряду на адитивні складові: річний тренд, гармонічні коливання з основним періодом 24 години (загалом 8, 12, 24, 168 год) та стохастичний залишок. До недоліків можна віднести інтерактивність методу його застосування, що потребує втручання фахівця з досвідом роботи на етапі групування окремих адитивних складових, а також неоднозначність вибору параметра L.

Базовий метод сингулярно-спектрального аналізу полягає в перетворені часового ряду в багатовимірний шляхом побудови траєкторної матриці за допомогою векторів вкладення. Його часто називають процедурою «нарізки» часового ряду довжиною L, групуванням членів сингулярного розкладу нової матриці, що утворилася за наступним відновленням. Інтерактивність методу, а відповідно і його складність, полягає в правильному виділені (групуванні) адитивних складових вихідного часового ряду газоспоживання, таких як тренд, періодичні та квазіперіодичні компоненти, а також стохастичний залишок (зміст якого часто розглядають як шум). Відповідно якісь і точність альтернативних прогнозів залежить від досвіду дослідника (вибору параметра L). Завдяки цьому методу можна отримувати прогнозні значення за рекурентним та векторним алгоритмом (як модифікація рекурентного).

Оскільки сингулярний розклад матриці відбувається за тою ж схемою, що й знаходження власних пар в симетричній матриці, то ми будемо розглядати відомі алгоритми для розв’язку останнього.

Для вирішення задачі знаходження власних значень можна застосувати наступні чотири алгоритми: алгоритм бісекції і зворотної ітерації (BI) [1], QR [2, 3], і алгоритму «розділяй і владарюй» (DC) [4, 5], алгоритм MRRR.

Всі ці алгоритми представлені в [6], але порівняння продуктивності [7] показали, що DC і MRRR найшвидші з існуючих. Також якісний аналіз дозволяє зазначити наступне. Як і алгоритм MRRR, метод зворотних ітерацій дозволяє розраховувати підмножини власних пар за зниженою обчислювальної вартістю, вимагаючи $O(nk^2)$ арифметичних операцій в найгіршому випадку. На противагу цьому, інші методи можуть тільки обчислити всі власні пари і може їх обчислювальна складність в найгіршому випадку складатиме $O(n^3)$ [8]. Для всіх цих алгоритмів існують реалізації як для послідовної так і для розподіленої пам'яті. Всебічне обговорення паралельних існуючих алгоритмів і їх продуктивності можна знайти в [9]. З погляду точності, QR і D C, як правило, кращі ніж BI і MRRR; DC вимагає $O(n^2)$ додаткової пам'яті а, отже, набагато більше, ніж решта алгоритмів, які вимагають тільки $O(n)$ додаткового місця для зберігання; DC і MRRR набагато швидші, ніж QR і BI; незважаючи на те, що MRRR використовує найменшу кількість операцій з плаваючою комою, DC може бути швидшим на деяких класах матриці. Тобто те, який алгоритм, DC чи MRRR, буде швидшим, залежатиме від спектрального розподілу вхідної матриці.

Література

1. Dhillon I. Relative Robust Representations of Symmetric Tridiagonal Linear Algebra / I. Dhillon, B. Parlett // Linear Algebra and its Applications, 2000. – Vol. 309, no. 1-3. – C. 121–151.
2. Dhillona Inderjit S. Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices / Inderjit S. Dhillona, Beresford N. Parlettb // Linear Algebra and its Applications, 2004. – Vol. 387. – C. 1–28.
3. Divide and Conquer Symmetric Tridiagonal Eigensolver for Multicore Architectures // Parallel and Distributed Processing Symposium / [G. Pichon, A. Haidar, M. Faverge та ін.]. – Hyderabad, 2015. – C. 51–60.
4. Dongarra J. A. Quark user' guide: Queuing and runtime for kernels / J. Dongarra, J. Kurzak, A. YarKhan // Innovative Computing Laboratory University of Tennessee, Technical Report, 2011.
5. Francis J. G. F. The QR Transformation A Unitary Analogue to the LR Transformation – Part 1 / J. G. F. Francis // Computer Journal, 1961. – vol. 4, no. 3 – C. 256–271.
6. Bashe, C. J. The Architecture of IBM's Early Computers [Electronic resource] / [C. J. Bashe, W. Buchholz, G. V. Hawkins та ін.] // IBM Journal of System Development, 1981 – Mode of access: URL: http://web.ece.ucdavis.edu/~vojin/CLASSES/EEC272/S2005/Papers/IBM-Architecture-Bashe_sep81.pdf.
7. Bentley Jon Writing Efficient Programs / Jon Bentley – Prentice Hall Ptr, 1982. – 170 с.– ISBN 978-0139702440.
8. Algorithm 880: A Testing Infrastructure for Symmetric Tridiagonals Eigensolver / J. Demmel, O. Marques, B. Parlett, C. Vömel. // ACM TOMS., 2008. – Vol. 30, no. 1.– C. 1508–1526.
9. Chapman B. Using OpenMP – Portable Shared Memory Parallel Programming / Barbara Chapman, Gabriele Jost, Ruud van der Pas – The MIT Press, 2007. – 384 с. – ISBN 978-0262533027.