Вісник Київського національного університету імені Тараса Шевченка
Серія фізико-математичні науки

2015, 2

Bulletin of Taras Shevchenko National University of Kyiv
Series Physics & Mathematics

УДК 681.3.062

Еміл М. Насіров, аспірант

Блочно-діагональний підхід до невід'ємної факторизації розріджених матриць і тензорів надвеликої розмірності

Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д, e-mail: enasirov@gmail.com

Emil M. Nasirov, Postgraduate Student

Blocks approach of non-negative huge sparse matrix and tensors factorization

Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova ave., 4d,
e-mail: enasirov@gmail.com

В роботі описано алгоритми невід'ємної факторизації розріджених матриць та тензорів, популярну технологію в штучному інтелекті взагалі, та в комп'ютерній лінгвістиці зокрема. Оброблювана матриця – матриця зв'язків Слова×Статті, яка містила частотну оцінку вживання слів в текстах статей Вікіпедії. Наведено існуючі підходи до факторизації матриць та тензорів. Запропоновано блочно-діагональний підхід до невід'ємної факторизації матриць та тензорів надвеликої розмірності. Використання пам'яті та мережевих операцій було проаналізовано та оптимізовано.

Ключові слова: штучний інтелект, комп'ютерна лінгвістика, паралельні обчислення

This paper proposes parallel method of non-negative large sparse matrix factorization - very popular technique in computational linguistics. The initial matrix contains frequency estimations of using terms in Wikipedia articles.

The main purpose of the paper is developing of a parallel method of factorization. Proposed blocks factorization approach, which can speedup non-negative sparse extra-large matrix and tensor factorization. Memory usage and data transmitting necessity of factorization algorithm was analysed and optimized. The described algorithms were compared by means of large sparse matrices and tensors processing. Presented method pretends to become a good basis for developing new effective tools for non-negative huge sparse matrix and tensors factorization. It can be used to perform the tasks of industrial scale to factorize sparse matrices of large dimension with an acceptable time using available computing resources. This perspective proves theoretical and practical value of the proposed algorithm.

Keywords: artificial intelligence, computational linguistics, parallel computations

Статтю представив чл.-кор. НАН України, д.ф.- м.н., проф. Анісімов А. В.

Сьогодні невід'ємна факторизація матриць та тензорів є дуже популярною технологією в штучному інтелекті взагалі, та в комп'ютерній лінгвістиці зокрема. Використовуючи невід'ємну факторизацію в рамках парадигми латентно-семантичного аналізу, комп'ютерні лінгвісти застосовують даний підхід для розв'язання таких прикладних задач, як класифікація, кластеризація текстів та термінів[1, 2], побудова мір семантичної близькості[3], автоматичне виділення лінгвістичних структур та відношень (*Selectional Preferences* [4] та *Verb Sub-Categorization Frames* [5]) і багато інших.

Дана робота описує побудову моделі паралелізації обчислення невід'ємної факторизації розріджених матриць надвеликої розмірності, що представляє особливий інтерес та актуальність з точки зору її застосування у великих NLP системах загального тематичного призначення, не обмежених використанням лише для вузьких предметних областей.

Задача невід'ємної факторизації розріджених матриць надвеликої розмірності постала в процесі розробки системи визначення міри семантичної близькості-зв'язності за технологією Латентного Семантичного Аналізу[6]. Для побудови системи був оброблений великий текстовий корпус статей англомовної Вікіпедії. Обробка корпусу полягала у лексичному аналізі та лематизації лексем речень статей та в обчисленні частот вживання множини слів та словосполучень англійської мови в складі різних статей англомовної Вікіпедії. В результаті була побудована велика матриця Слова×Статті яка містила частотну оцінку вживання слів в текстах статей Вікіпедії. Розмірність матриці дорівнює 2,437,234 слів-словосполучень на 4,475,180 статей англомовної Вікіпедії. Після цього для усунення з матриці випадкових даних був встановлений пороговий рівень T=3, частотні оцінки в матриці, які менші за пороговий рівень T, були обнулені). В результаті була побудована розріджена матриця надвеликого розміру - 156,236,043 ненульових елементів при розмірі 2,437,234×4,475,180. Для невід'ємної факторизації розрідженої матриці такого розміру знадобилася розробка спеціальної моделі паралелізації матричних обчислень, яка була реалізована із застосуванням паралельних розподілених обчислень та обчислень на GPU.

Алгоритм невід'ємної матричної факторизації

Алгоритм невід'ємної матричної факторизації виконує декомпозицію невід'ємної матриці V на невід'ємні матриці W та H таким чином, щоб

$$V \approx WH$$

В якості функції оцінки може бути використана функція виміру відстані між двома невід'ємними матрицями. Однією з таких мір є квадрат Евклідової метрики:

$$\mu = \|A - B\|^2 = \sum_{ij} \left(A_{ij} - B_{ij}\right)^2$$

Така цільова функція обмежена знизу. Нижня границя 0 досягається тоді і тільки тоді коли $A = B$.

Отже, при використанні Евклідової метрики, факторизація матриці полягає в мінімізації $\|V - WH\|^2$ при умові невід'ємності W та H.

Така цільова функція не зростаюча при наступних правилах:

$$H_{ij} \leftarrow H_{ij} \frac{(W^T V)_{ij}}{(W^T W H)_{ij}} \quad (1)$$

$$W_{ij} \leftarrow W_{ij} \frac{(V H^T)_{ij}}{(W H H^T)_{ij}} \quad (2)$$

Виконання ітерацій алгоритму продовжується дот тих пір, поки не буде досягнута стаціонарна точка, або не буде виконана максимальна кількість ітерацій[7].

Аналіз моделі

В Таблиці 1 представлено необхідні об'єми пам'яті для зберігання матриць W та H при різних k при використанні типу даних float (32bit).

K	*100*	*200*	*300*
W	0.98Gb	1.95Gb	2.92Gb
H	1.79Gb	3.58Gb	5.37Gb
Всього	2.76Gb	5.53Gb	8.29Gb

Таблиця 1. Необхідні об'єми пам'яті для зберігання матриць W та H при різних k

Описаний алгоритм на кожній ітерації потребує в 2 рази більше пам'яті для збережень матриць (не враховуючи потреби на збереження початкової матриці V). Враховуючи такі потреби в пам'яті, для виконання алгоритму на одному комп'ютері необхідно проводити збереження даних на жорсткий диск. Далі буде розглянуто та порівняно 2 підходи до паралелізації алгоритму: локальний та розподілений.

Застосування моделі до факторизації тензорів

Для того, щоб обчислити невід'ємні матриці-компоненти {A, B, C} зазвичай застосовується обмежений оптимізаційний підхід для мінімізації підходящої функції оцінки. Зазвичай мінімізують наступну функцію:

$$D_F(\underline{Y} \| [\![A, B, C]\!]) = \left\| \underline{Y} - [\![A, B, C]\!] \right\|_F^2 + \alpha_A \|A\|_F^2 + \alpha_B \|B\|_F^2 + \alpha_C \|C\|_F^2$$

, де $\alpha_A, \alpha_B, \alpha_C$ – невід'ємні регуляційні параметри.

Існує щонайменше три різні підходи до розв'язання такої оптимізаційної задачі. Перший підхід полягає у використанні векторної форми функції оцінки: $J(X) = vec(\underline{Y} - [A, B, C]) = 0$. Для розв'язання використовується метод найменших квадратів. Цей метод для факторизації тензорів вперше використав Паатеро [7]. Якобіан такої функції може мати розмір $ITQJ \times (I + T + Q)$, а отже такий підхід потребує значних розрахункових витрат.

У другомі підході Акар, Колда та Дунлаві запропонували штучно оптимізувати функцію оцінки використовуючи техніку нелінійної зв'язної градієнтної оптимізації.[1]

Третім, і найпоширенішим підходом є використання техніки ALS. В цьому підході підраховується градієнт функції оцінки для кожної матриці окремо.

$$\nabla_A D_F = -Y_1(C \odot B) + A[(C^T C) \circledast (B^T B) + \alpha_A I],$$
$$\nabla_B D_F = -Y_2(C \odot A) + B[(C^T C) \circledast (A^T A) + \alpha_B I],$$
$$\nabla_C D_F = -Y_3(B \odot A) + C[(B^T B) \circledast (A^T A) + \alpha_C I].$$

Прирівнюючи кожну компоненту градієнта до 0 та накладаючи умову невід'ємності можна отримати ефективні та прості правила оновлення матриць:

$$A \leftarrow \left[Y_{(1)}(C \odot B) + [(C^T C) \circledast (B^T B) + \alpha_A I]^{-1}\right]_+,$$
$$B \leftarrow \left[Y_{(2)}(C \odot A) + [(C^T C) \circledast (A^T A) + \alpha_B I]^{-1}\right]_+,$$
$$C \leftarrow \left[Y_{(3)}(B \odot A) + [(B^T B) \circledast (A^T A) + \alpha_C I]^{-1}\right]_+.$$

Основними перевагами такого підходу є висока швидкість збіжності та можливість розподілення для задач великої розмірності.

Блочно-діагональний підхід до факторизації матриць та тензорів

Так як розглядувана розріджена матриця є матрицею зв'язності Слова×Статті, отже можливі перестановки як рядків так і стовпців початкової матриці.

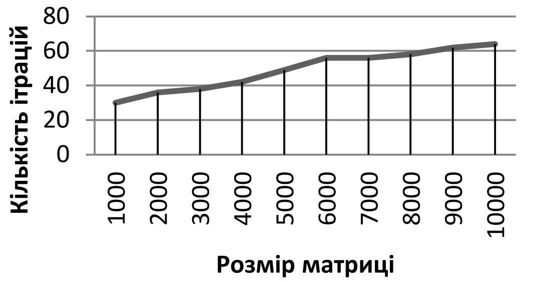
Розглядувана розріджена матриця має 156,236,043 ненульових елементів з 10,907,060,852,120 що складає лише 0.014%. Така розрідженість дозволяє привести матрицю до блочно-діагональної форми за допомогою перестановок рядків та стовпчиків. У випадку, коли слово може належати декільком блокам, рядок може бути розділений на два рядки, що відповідатимуть одному слову.

Аналогічно, до блочно-діагонального вигляду можуть бути приведені тензори за допомогою перестановок шарів.

Розміри блоків найкраще обирати такими, що можуть бути повністю завантаженні в пам'ять. В результаті буде отримано N блоків.

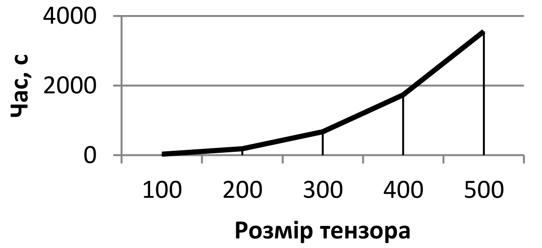
Переваги такого підходу:

1. Не потрібно мережевих операцій на кожній ітерації. Необхідне лише початкове розподілення блоків матриці або тензора між вузлами.



Графік 1.

2. Швидша збіжність. На графіку 1 та графіку 2 показано час необхідний для факторизації квадратних матриць та тензорів різних розмірів відповідно, в один потік без використання GPU. На графіку 3 показано необхідну кількість ітерацій для невід'ємної факторизації квадратних матриць різних розмірів.

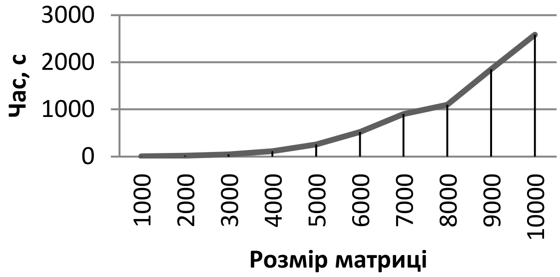


Графік 2.

3. Зменшення *k*, а отже і зменшення розмірів результатів та об'ємів необхідних даних для зберігання в пам'яті під час ітерацій.

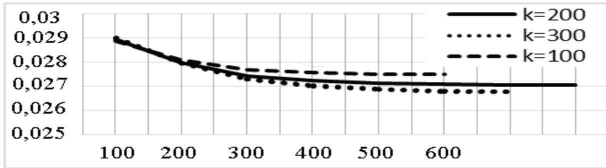
4. Швидший підрахунок функції оцінки μ. Швидший підрахунок добутку *W*H* так як менше k. Кількість елементів для підрахунку функції оцінки менша в N разів(для матриці), кількість необхідних перевірок менша так як зменшується загальна кількість ітерацій.

На графіку 4 показано залежність якості результатів факторизації від k та кількості ітерацій для звичайного алгоритму факторизації матриць.



Графік 3.

У випадку зведеної до блочно-діагональної матриці, в результаті факторизації окремих блоків для кожного слова буде отримано один або декілька рядків та ідентифікатор блоку в якому це слово знаходиться. При відновленні, якщо слово та стаття знаходяться в різних блоках – результатом буде 0, інакше результатом буде добуток відповідних рядка та стовпчика.



Графік 4

Висновки

В роботі описано алгоритм невід'ємної факторизації матриць та алгоритм невід'ємної факторизації тензорів. Запропоновано блочно-діагональний підхід до невід'ємної факторизації матриць та тензорів. Наведено переваги блочно-діагонального підходу щодо швидкодії.

Список використаних джерел

1. *Xu W.* Document-clustering based on n-negative matrix factorization / Xu W., Liu X., Gong Y. // In Proceedings of SIGIR'2003.–2003.– p. 267–273.
2. Document clustering using nonnegative matrix factorization / [F.Shahnaz and other] // Information Processing and Management.–2006. – vol. 42. –p. 373 – 386
3. Ukrainian WordNet: Creation and Filling / [A. Anisimov and other] // FQAS-2013. –2013. – p. 649-660
4. *T. Van de Cruys* A Non-negative Tensor Factorization Model for Selectional Preference Induction / Tim Van de Cruys // Journal of Natural Language Engineering. –2010.– vol.16(4).– p. 417–437.
5. *T. Van de Cruys.* Multi-way Tensor Factorization for Unsupervised Lexical Acquisition / T. Van de Cruys [and other] // Proceedings of COLING-2012. –2012. –p. 2703–2720.
6. Indexing by Latent Semantic Analysis / [Scott Deerwester and other] // Journal of the American Society for Information Science – 1990 – vol.41 (6) – p. 391–407.
7. *D. D. Lee.* Algorithms for non-negative matrix factorization / D.D. Lee and H. S. Seung // In NIPS. –2000. –p. 556-562.

References

1. XU W., LIU X., GONG Y. (2003) *Document-clustering based on n-negative matrix factorization*. In Proceedings of SIGIR'2003. pp. 267–273.
2. SHAHNAZ F. et al. (2006) *Document clustering using nonnegative matrix factorization*. Information Processing and Management. Vol 42, pp. 649-660.
3. ANISIMOV A. et al. (2013) *Ukrainian WordNet: Creation and Filling*. FQAS-2013. pp. 649-660.
4. VAN DE CRUYS T. (2010) *A Non-negative Tensor Factorization Model for Selectional Preference Induction*. Journal of Natural Language Engineering, Vol. 16(4), pp. 417–437.
5. VAN DE CRUYS T. et al. (2012) *Multi-way Tensor Factorization for Unsupervised Lexical Acquisition*. Proceedings of COLING-2012. pp. 2703–2720.
6. DEERWESTER S. et al. (1990) *Indexing by Latent Semantic Analysis*. Journal of the American Society for Information Science. Vol. 41 (6), pp. 391–407.
7. LEE D.D., SEUNG H. S. (2000) *Algorithms for non-negative matrix factorization*. In NIPS, MIT Press. pp. 556-562.

Надійшла до редколегії 30.06.15