УДК 519.688

К.т.н., доцент Соколова Н.А., студент Стратієвський О.М.

Національний технічний університет України
«Київський політехнічний інститут»

ВИКОРИСТАННЯ МОДИФІКАЦІЇ СИНГУЛЯРНОГО РОЗВИНЕННЯ МАТРИЦІ ДЛЯ ЗАДАЧ ЛАТЕНТНО-СЕМАНТИЧНОГО ІНДЕКСУВАННЯ

Abstract

Nadiya A. Sokolova, assoc. prof., PhD; Oleksandr Stratiyevs'kyy, student
Singular value decomposition for latent-semantic indexing

This paper concerns the task of singular value decomposition (SVD) for information filtering using latent-semantic indexing (LSI). The known SVD algorithms based on QR factorization, that are used for LSI are studied and discussed. The modifications are proposed with analysis of efficiency due to LSI. The ways for further research are proposed as well.

Вступ

Парадигми та методи векторної алгебри активно використовуються в процесі автоматичного аналізу текстів природною мовою (Natural Language processing). Так кожен термін (значуще слово) тексту чи декількох текстових документів аналізується, як векторна величина з використанням відповідного математичного апарату. Один із таких підходів, що отримав назву латентно-семантичного індексування (LSI) має за основу аналіз взаємозв'язків між колекцією документів та їх спільними термінами на предмет виявлення подібної тематики, зокрема в задачах інформаційного пошуку. Це статистичний метод, що автоматично пов'язує терміни тексту в семантичну структуру без синтаксичного чи семантичного аналізу [1]. В результаті документ представляється за концептами, які, на відміну від термінів, є деякою мірою статистично незалежними. Концепти використовуються в задачах аналізу тематичної орієнтації текстових документів, зокрема для удосконалення можливостей інформаційно-пошукових систем [1]. Ключовою в LSI являється процедура декомпозиції матриці з використанням методу сингулярного розвинення матриці (Singular Value Decomposition – SVD) [2]. Не менш важливою задачею являється процедура побудови вихідної матриці для SVD, яка визначає особливості задачі розвинення.

Постановка задачі

Задача полягає в комбінації модифікацій існуючого алгоритму SVD для LSI [2] з урахуванням особливостей побудови матриць, що використовуються в задачах LSI, з метою збільшення швидкості обчислень та зменшення розмірності задачі.

Сингулярне розвинення матриці в контексті LSI

Сингулярне розвинення — це розвинення прямокутної дійсної чи комплексної матриці, що використовується в багатьох областях прикладної математики[3]. Так в LSI за допомогою SVD матрицю W $m \times n$, елемент якої w_{ij} – це вага i-го ключового терміну в j-му документі колекції представляється у вигляді:

$$W = U * S * V$$

де U $m \times r$ – це матриця термів, S $r \times r$ – діагональна матриця, а V $r \times n$ – матриця документів. Це можна зробити так, що стовпчики матриці U, рядки матриці V та діагональні величини матриці S впорядковані вниз по діагоналі в порядку зменшення [3]. За допомогою методу LSI розмірність r цих матриць зменшується до s шляхом видалення r-s діагональних елементів S, що менші визначеного значення, та відповідних їм компонентів U та V [2].

Огляд існуючих алгоритмів сингулярного розвинення

На сьогодні можна виділити наступні класи алгоритмів сингулярного розвинення:

Алгоритм Якобі. Відзначається високою точністю, та в чистому вигляді його швидкість досить низька. Останнім часом набирають актуальність його модифікації, засновані на ідеях паралельного програмування.

Алгоритми сингулярного розвинення, що ґрунтуються на приведенні матриці за допомогою ортогональних перетворень до дводіагональної форми [3] та наступної діагоналізації ітеративним QR-алгоритмом. Найпоширенішими алгоритмами такого класу є модифікації алгоритму Голуба-Кахана-Рейнча (Golub Kahan Reinsch algorithm). До їх безперечних переваг слід віднести простоту та компактність [3]. Проте з ростом розмірності задачі значно підвищується кількість обчислень та відчутно падає точність розрахунків.

Наступне сімейство алгоритмів, що отримало назву divide-and-conquer(розділяй та владарюй), зводить задачу SVD-декомпозиції матриці великого розміру до низки задач з матрицями меншої розмірності.

Опис обраної модифікації алгоритму

Проста схема зведення матриці до дводіагонального вигляду з наступною діагоналізацією QR-алгоритмом часто на практиці себе виправдовує, та для використання SDV-декомпозиції в задачах LSI її можна покращити. У даній роботі за основу пропонується використати модифікацію вищеописаного алгоритму, розроблену для пакету LAPACK, що підвищує швидкість роботи алгоритму в залежності від вихідних умов в 2-6 разів [4].

По-перше, пропонується підвищити швидкість пошуку матриці U за рахунок адаптації алгоритму до особливостей проведення машинних розрахунків та принципів організації масивів в памяті – пропонується знаходити матрицю U^T. Таким чином операції, що здійснюються в процесі побудови матриці, проводяться над рядками, що є набагато ефективнішим. По-друге, пропонується замість зведення матриці до дводіагональної форми використовувати LQ або QR перетворення, в залежності від того, яка із сторін матриці більша. Тоді матриця представляється у вигляді добутку прямокутної ортогональної матриці Q та квадратної, відповідно нижньо або верхньотрикутної матриці [4].

Суттєвим недоліком запропонованих вище підходів є той факт, що для реалізації обох нововведень в загальному вигляді потрібно виділяти додаткову пам'ять. Та в LSI власне матриця U в чистому вигляді не використовується і ми можемо повноцінно працювати з транспонованою матрицею, не виділяючи додаткового місця.

Щодо другого нововведення, то проблема використання додаткової пам’яті є дещо складнішою. Пам’ять може стати критичним ресурсом у випадку, коли виникає необхідність одночасного виконання великої кількості подібних задач, зокрема за умови великої кількості запитів від користувачів інформаційної системи, що використовує LSI. Тому цією роботою пропонується використати додаткові механізми фільтрації [5] для перетворення початкової матриці на квадратну, аби одразу отримувати на вході більш оптимальну для QR-розвинення матрицю і не використовувати проблемний другий етап модифікації у цьому випадку. Та процедури фільтрації повинні бути відносно простими, і у випадку, коли у результаті матриця все-таки виявиться прямокутною, необхідно приймати вибір між подальшим використанням фільтрації, типовим пониження розмірності для LSI [2], чи використанням другого підходу модифікації пакету LAPACK.

Побудова та збереження вихідної матриці для SVD

На сьогоднішній день латентно-семантичне індексування є задачею аналізу текстових даних переважно в мережевих системах з клієнт-

серверною або розподіленою архітектурою. Тому важливим також є питання побудови вихідної матриці та ефективного її зберігання. Значення матриці змінюються відносно до наповнення інформаційної системи новими документами. Зважаючи на значну частоту пошукових запитів та складність динамічної побудови матриці з таблиць баз даних, пропонується будувати матрицю при першому запиті, а в подальшому лише оновлювати. Тому для звуження кола матеріалів, які впливатимуть на зміну матриці вводяться фільтри, що обмежують коло користувачів, які додають та переглядають документи за допомогою механізмів контентної та колаборативної фільтрації [5]. Так як в загальному випадку подібні матриці є розрідженими, то їх варто зберігати у запакованому вигляді та вживати додаткові процедури пониження розмірності [5].

Висновки

В роботі було запропоновано підхід до вдосконалення роботи сингулярного розвинення матриць в контексті задач латентно-семантичного індексування за допомогою комбінації алгоритму SVD з процедурами пониження розмірності, та фільтрації. Використавши особливості задач LSI певною мірою вирішені проблеми використання ресурсів інформаційної системи та скомпенсовано недоліки використаного алгоритму SVD.

В подальших дослідженнях найактуальнішим є питання підбору комбінації процедур фільтрації для вирішення задач пониження розмірності, побудови вихідної матриці та модифікацією SVD з метою подальшого покращення швидкодії системи.

Література

1. *Дерецький В.О.* Підхід до автоматичної побудови тематичної онтології документа для удосконалення інформаційного пошуку // Проблеми програмування. – 2005.– N3.– С.76-82.
2. *Berry M.W., Dumais S.T., O'Brein G.W.* Using linear algebra intelligent information retrieval // SIAM Review. – 1995. – 37(4). – P. 573-595.
3. *Голуб Дж., Ван Лоун Ч.* Матричные вычисления: Пер с англ. – М.: Мир, 1999. – 548 с.
4. ALGLIB – многоязыковая коллекция алгоритмов [Електронний ресурс]. – Режим доступу: http://alglib.sources.ru/
5. *Горностай М.П.* Алгоритми і методи надання рекомендацій та їх оптимізація // Проблеми програмування. – 2007.– N4.– С.69-76.