ОПТИМИЗАЦИЯ ВЫЧИСЛЕНИЯ ДВУМЕРНЫХ РЕКУРРЕНТНЫХ СВЕРТОК ПРИ ПОМОЩИ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ

Липницкий В. А., Сергей А. И.
Кафедра высшей математики, Военная академия Республики Беларусь
Факультет экономики и управления, Гродненский государственный университет имени Янки Купалы
Минск, Гродно, Республика Беларусь
E-mail: valipnitski@yandex.by, sergej.a.i@mail.ru

В статье приводится эффективный алгоритм решения одной из подзадач третьей проблемы Кэмерона. Рассматривается применение быстрого преобразования Фурье для вычисления двумерных сверток с рекуррентными зависимостями. Предлагаемый алгоритм снижает асимптотическую сложность расчетов с $O(n^4)$ (для базового алгоритма) до $O(n^{2.5}\log n)$ арифметических операций.

Введение

Рассматриваемая задача возникает при подсчете количества квадратных бинарных матриц n-го порядка, содержащих в точности n единиц, с точностью до перестановки строк/столбцов (третья проблема Кэмерона [1]).

Наибольшего прогресса в решении проблемы удалось достичь при помощи метода динамического программирования [2]. При таком подходе естественным образом возникает необходимость вычисления сложных сверток, которые в свою очередь являются узким местом алгоритма. Поэтому оптимизация расчетов приведенных ниже сверток напрямую влияет на итоговую сложность алгоритма решения проблемы Кэмерона.

I. Постановка задачи

Необходимо эффективно вычислить значение величины

$$r_{i,j} = \frac{1}{i}\sum_{k=0}^{i-1}\sum_{l=0}^{j} r_{k,l}t_{i-k,j-l}[2] \qquad (1)$$

для $1 \le i \le n, 0 \le j \le n$ по заданным $r_{0,j}, 0 \le j \le n$ и $t_{i,j}, 0 \le i,j \le n$.

Соотношение (1) напоминает формулу для вычисления дискретной двумерной свертки [3]. Основное отличие и сложность формулы (1) заключается в рекуррентной зависимости между значениями $r_{i,j}$.

Формула (1) отчетливо показывает двумерную структуру задачи, поэтому ее естественно рассматривать в матричном виде. Пусть

$$R = (r_{i,j})_{\substack{i=\overline{0,n}\\ j=\overline{0,n}}}, T = (t_{i,j})_{\substack{i=\overline{0,n}\\ j=\overline{0,n}}}.$$

Тогда по заданной матрице T и 0-й строке матрицы R нужно вычислить остальные элементы матрицы R по формуле (1).

Оценим асимптотическую сложность расчета значений $r_{i,j}$ пользуясь формулой (1) напрямую. Матрицу R можно вычислять построчно, поскольку элементы i-й строки зависят только от элементов из предыдущих строк. Сложность вычисления отдельно взятого элемента $r_{i,j}$ составляет $O(ij)$. Следовательно, вычисление всех недостающих элементов матрицы R «наивным» способом имеет сложность $\sum_{i=1}^{n}\sum_{j=0}^{n} O(ij) = O(n^4)$.

II. Оптимизация 1. Одномерные свертки

Будем оптимизировать вычисление $r_{i,j}$, сводя его к расчету одномерных сверток.

Пусть u и v – вектор-строки длины $n+1$: $u = (u_0u_1\dots u_n), v = (v_0v_1\dots v_n)$. Тогда их одномерной сверткой будем называть вектор-строку $conv1(u,v)$ длины $n+1$, j-й элемент которой вычисляется по формуле

$$[conv1(u,v)]_j = \sum_{l=0}^{j} u_l v_{j-l}, j = \overline{0,n}. \qquad (2)$$

Пусть $R_i = (r_{i0}\ r_{i1}\ \dots\ r_{in})$ - i-я строка матрицы R и, аналогично, $T_i = (t_{i0}t_{i1}\dots t_{in})$ - i-я строка матрицы T. Тогда Формулу (1) можно записать в виде:

$$R_i = \frac{1}{i}\sum_{k=0}^{i-1} conv1(R_k, T_{i-k}), i = \overline{1,n}. \qquad (3)$$

Вычисление свертки $conv1(R_k, T_{i-k})$ по формуле (2) потребует $O(n^2)$ операций, однако быстрое преобразование Фурье позволяет вычислять такие свертки со сложностью $O(n\log n)$ [3]. Таким образом, вычисление R_i по формуле (3) имеет сложность $O(in\log n)$, а общая сложность решения исходной задачи составляет $O(n^3\log n)$.

III. Оптимизация 2. Двумерные свертки

Выразим вычисление $r_{i,j}$ через расчет двумерных сверток. Пусть A и B – квадратные матрицы порядка $n+1$. Тогда их двумерной сверткой будем называть матрицу $conv2(A,B)$ порядка $n+1$, элемент $a_{i,j}$ которой вычисляется по формуле

$$[conv2(A,B)]_{i,j} = \sum_{k=0}^{i}\sum_{l=0}^{j} a_{i,j}b_{k-i,l-j}. \qquad (4)$$

Формулу (4) можно также выразить через применение одномерных сверток к строкам:

$$[conv2(A,B)]_i = \textstyle\sum_{k=0}^{i} conv1(A_k, B_{i-k}). \quad (5)$$

Рассмотрим способ, позволяющий вычислить очередную строку матрицы R целиком при помощи двумерных сверток. Обозначим через $R^{(i)}$ матрицу, полученную из R занулением строк с $(i+1)$-й по n-ю, т. е. матрицу $(R_0\ R_1\ \dots\ R_{i-1}\ R_i\ 0\ \dots\ 0)^T$.

Пользуясь формулами (5) и (3), получаем,

$$[conv2(R^{(i-1)},T)]_i = \sum_{k=0}^{i} conv1([R^{(i-1)}]_k, T_{i-k}) =$$

$$= \underbrace{conv1([R^{(i-1)}]_i, T_0)}_{=\,(0\,\dots\,0)} + \sum_{k=0}^{i-1} conv1(R_k, T_{i-k}) = iR_i.$$

Поэтому формула для вычисления i-й строки матрицы R на основе предыдущих имеет вид:

$$R_i = \frac{1}{i}[conv2(R^{(i-1)},T)]_i. \quad (6)$$

Вычисление свертки $conv2(R^{(i-1)},T)$ по формуле (4) потребует $O(n^4)$ операций, однако и в двумерном случае быстрое преобразование Фурье позволяет ускорить расчет сверток до $O(n^2\log n)$ [3]. Таким образом, вычисление R_i по формуле (6) имеет сложность $O(n^2\log n)$, а общая сложность решения исходной задачи таким способом составляет $O(n^3\log n)$.

IV. Оптимизация 3. Чередование одномерных и двумерных сверток

В результате оптимизаций 1 и 2 получились два независимых алгоритма решения исходной задачи, оба со сложностью $O(n^3\log n)$. В этом разделе эти два метода объединяются для получения более эффективного алгритма.

Обозначим $conv1(R_k, T_{i-k})$, промежуточную величину из оптимизации 1, через $C_{i,k}$, $i = \overline{1,n}, k = \overline{0, i-1}$. В этом разделе основное внимание уделяется именно значениям $C_{i,k}$ и их вычислению.

Также введем обозначение $P(i)$ для величины $conv2(R^{(i-1)},T)$, возникающей в оптимизации 2, $i = \overline{0,n}$. В соответствии с формулой (5), имеем

$$[P(i)]_i = \textstyle\sum_{k=0}^{i-1} C_{i,k}. \quad (7)$$

Заметим, что следующие за i-й строки матрицы $P(i)$ также несут в себе полезную информацию, например:

$$[P(i)]_{i+1} = \textstyle\sum_{k=0}^{i-1} C_{i+1,k}, [P(i)]_{i+2} = \sum_{k=0}^{i-1} C_{i+2,k}.$$

В общем случае

$$[P(i)]_{i+s} = \textstyle\sum_{k=0}^{i-1} C_{i+s,k}, s = \overline{0, n-i}.$$

Отсюда следует, что для $0 \le s \le n-i$

$$[P(i+s)]_{i+s} = [P(i)]_{i+s} + \textstyle\sum_{k=i}^{i+s-1} C_{i+s,k}. \quad (8)$$

Другими словами, имея вычисленное значение матрицы $P(i)$, можно получить значение $(i+s)$-й строки матрицы $P(i+s)$, посчитав дополнительно s недостающих значений $C_{i+s,k}$, для $k = \overline{i, i+s-1}$.

Из формулы (6) следует, что $R_i = \frac{1}{i}[P(i)]_i, i = \overline{1,n}$, поэтому для решения исходной задачи достаточно вычислить $[P(i)]_i$ для $i = \overline{1,n}$.

Сложность вычисления значения $C_{i,k}$ при помощи одномерной свертки составляет $O(n\log n)$, тогда как сложность вычисления матрицы $P(i)$ при помощи двумерной свертки равна $O(n^2\log n)$.

Будем вычислять строки матрицы R блоками. Выберем некоторое целое число b - размер блока, $1 \le b \le n+1$. Строки $1, 2, \dots, b-1$ матрицы R вычислим, пользуясь формулой (7). Остальные строки будем вычислять следующим образом. Для строк с номерами вида rb будем находить значение матрицы $P(rb)$ и использовать ее при вычислении строк с номерами $rb+s, s = \overline{1, b-1}$, используя формулу (8).

На рисунке 1 показан принцип вычисления строк матрицы R по блокам на примере $n = 11, b = 4$. Цвет ячейки (i,k) показывает вычислялось ли значение $C_{i,k}$ отдельно (белый) или было учтено в первом слагаемом формулы (8) (черный).

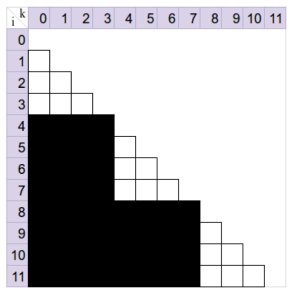

Рис. 1 – Вычисление $C_{i,k}$ блоками для $n = 11$, $b = 4$

Заметим, что при $b = n+1$ этот алгоритм вырождается в оптимизацию 1, а при $b = 1$ - в оптимизацию 2.

Выберем оптимальный размер блока b. Сложность вычисления матриц $P(rb)$ составляет $O(\frac{n}{b}b^2 n\log n)$. Сложность подсчета значений $C_{i,k}$ равна $O(\frac{n}{b}n^2\log n)$. Общая сложность алгоритма: $O(bn^2\log n + \frac{n^3}{b}\log n)$. Легко видеть, что оптимально выбирать b порядка $\sqrt{n}$, в этом случае сложность алгоритма получится $O(n^{2.5}\log n)$.

1. Cameron P. J. Problems on permutation groups // P. J. Cameron – [Электронный ресурс] – Режим доступа: http://www.maths.qmul.ac.uk/ pjc/pgprob.html. – Дата доступа: 07.02.2017.
2. Липницкий, В. А. Алгоритм развертки в подсчете количества орбит кэмероновских матриц / В. А. Липницкий, А. И. Сергей, Н. В. Спичекова // Веснiк Магiлёўскага дзяржаўнага унiверсiтэта iмя А.А. Куляшова, 2017, №2(50). – С. 30 – 43.
3. Быстрое преобразование Фурье и алгоритмы вычисления сверток / Г. Нуссбаумер; Изд-во: М.: Радио и связь, 1985 г. - 245 с.