Автоматика и телемеханика. 2004. № 12. С. 3-14.

© 2004 г. В.С. Выхованец, канд. техн. наук
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

ПОЛИНОМИАЛЬНАЯ ФАКТОРИЗАЦИЯ СПЕКТРАЛЬНЫХ БАЗИСОВ

Рассматривается проблема полиномиальной факторизации спектральных базисов, где под полиномиальной факторизацией понимается представление системы спектральных функций, заданных целочисленной матрицей дискретного преобразования, в виде кронекеровского произведения матриц меньшей размерности. Такое представление позволяет выразить упорядоченную систему функций в виде единой формулы в базисе бинарных операций. Представлен алгоритм полиномиальной факторизации матриц, дается его теоретическое обоснование и приводятся результаты экспериментального исследования.

1. Введение

Традиционно под факторизацией спектрального базиса понимается представление системы спектральных функций, заданных матрицей прямого (обратного) преобразования, в виде произведения слабозаполненных матриц меньшего порядка [1]. Факторизация необходима для выполнения быстрых спектральных преобразований, заключающихся в умножении матрицы на вектор.

В результате факторизации получают процедуру быстрых преобразований, основанную на приведении подобных слагаемых на основе дистрибутивности умножения относительно сложения и исключении операций в тривиальных случаях, когда не выполняется умножение на ноль и единицу, сложение с нулем.

Уменьшение количества операций является первым и необходимым условием факторизации. Второе условие связано с представлением алгоритма преобразования, когда дополнительно требуется получение такого представления матрицы, при котором операции имеют регулярное расположение в структуре алгоритма. Последнее позволяет не только сократить время вычислений, но и упростить сам алгоритм, повысить его эффективность.

Таким образом, при традиционном подходе к факторизации ставится задача получения эффективных процедур спектральных преобразований. При этом эффективность вычисления самих функций (матриц) не принимается во внимание и предполагается, что функции (матрицы) уже заданы.

В настоящей статье ставится и решается задача полиномиальной факторизации, заключающаяся в поиске представления матрицы дискретного преобразования в виде кронекеровского произведения матриц меньшей размерности. Суть подхода заключается в вычислении матриц и операций, относительно которых выполняются кронекеровские произведения, таких что в результате факторизации может быть получена формула для системы спектральных функций и вычислена исходная матрица дискретного преобразования.

2. Постановка задачи факторизации

Пусть задана целочисленная матрица $\mathbf{D}$ размерности $m \times m$. Для факторизации $\mathbf{D}$ найдем такие матрицы $\mathbf{P}_j$ и $\mathbf{O}_t$ ($j = \overline{0, n-1}$, $t = \overline{0, n-2}$), для которых с точностью до расстановки скобок выполняется равенство

(1) $$\mathbf{D} = \mathbf{P}_0 \otimes_0 \mathbf{P}_1 \otimes_1 \ldots \otimes_{n-2} \mathbf{P}_{n-1} = \underset{i=0}{\overset{n-1}{\otimes}} \mathbf{P}_i ,$$

где элементы и размерность $\mathbf{P}_j$ не превосходят некоторого числа k_p, а кронекеровские произведения $\otimes_t$ (см. ниже определение 2) выполняются относительно бинарных операций, заданных матрицами $\mathbf{O}_t$, которые также имеют элементы и размерность, не превосходящие k_p.

Известно [2], что матрица $\mathbf{D}$ представляется в виде (1), если спектральные функции, составляющие ее столбцы, являются полиномиальными. Аналитическая конструкция полиномиальных функций, задаваемая с точностью до нумерации переменных и расстановки скобок, имеет вид:

(2) $$\theta_i(X) = x_0^{i_0} \circ_0 x_1^{i_1} \circ_1 \ldots \circ_{n-2} x_{n-1}^{i_{n-1}}$$

или после подстановки в (2) выражений $x_j^{i_j} = x_j \triangleright_j i_j$

(3) $$\theta_i(X) = (x_0 \triangleright_0 i_0) \circ_0 (x_1 \triangleright_1 i_1) \circ_1 \ldots \circ_{n-2} (x_{n-1} \triangleright_{n-1} i_{n-1})$$

где $X = \{x_0, x_1, \ldots, x_{n-1}\}$ – множество переменных со значностями k_0, k_1, …, k_{n-1} соответственно, $m = k_0 k_1 \ldots k_{n-1}$ – длина вектора полиномиальной функции, i_j – цифры индекса функции i в позиционной системе счисления с основаниями, задаваемыми значностями переменных (см. определение 1), $\triangleright_j$ – бинарные степенные операции (определяются матрицами $\mathbf{P}_j$), $\circ_j$ – бинарные соединительные операции (определяются матрицами $\mathbf{O}_t$).

Таким образом, представление (1) позволяет также получить и единую формулу для системы спектральных функций, которую легко отразить в структуру алгоритма вычисления матрицы $\mathbf{D}$ или отдельных ее элементов, которые получаются в результате традиционной факторизации.

Поставим задачу поиска эффективных формульных представлений для упорядоченной системы функций, заданной в виде целочисленной матрицы. Если найдены операции полиномиальной аналитической конструкции (3), которые имеют значность не более чем k_p, то факторизацию будем называть существующей или успешной, в противном случае – несуществующей. Число k_p назовем *значностью полиномиальной факторизации*.

3. Определения

Определение 1. Представлением числа i в n-разрядной позиционной системе счисления с основаниями k_0, k_1, ..., k_{n-1} называется упорядоченное множество из n чисел или цифр $(i_0\, i_1 \ldots i_{n-1})_{k_0 k_1 \ldots k_{n-1}}$ таких, что

$$i = \sum_{j=0}^{n-1} i_j w_j \,, \quad w_j = \prod_{t=0}^{j-1} k_t \,,$$

где $i_j \in \{0,1,\ldots,k_j - 1\}$ – j-я цифра числа i с весом w_j, причем цифра i_0 имеет наименьший вес w_0, равный единице.

Определение 2. Пусть заданы бинарная операция $\circ$ и две матрицы $\mathbf{A} = [a_{i_0 j_0}]$ размерности $n_0 \times m_0$ и $\mathbf{B} = [b_{i_1 j_1}]$ размерности $n_1 \times m_1$. Правым кронекеровским произведением матрицы $\mathbf{A}$ на матрицу $\mathbf{B}$ относительно операции $\circ$ называется матрица $\mathbf{C} = \mathbf{A} \vec{\otimes}_{\circ} \mathbf{B}$ размерности $n_0 n_1 \times m_0 m_1$ с элементами $c_{ij} = a_{i_0 j_0} \circ b_{i_1 j_1}$, где $i = (i_0\, i_1)_{n_0 n_1}$ и $j = (j_0\, j_1)_{m_0 m_1}$ – представления индексов i и j в системе счисления с основаниями n_0, n_1 и m_0, m_1 соответственно.

Покажем вычисление кронекеровского произведения матриц относительно операции сложения. Пусть размерность матрицы $\mathbf{A}$ равна 2×2, а матрицы $\mathbf{B}$ – 2×3. Положим $a_{i_0 j_0} = i_0 + 2 j_0$ и $b_{i_1 j_1} = 3 i_1 + j_1$. Тогда

$$\mathbf{C} = \begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \vec{\otimes}_{+} \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix} = \left[\begin{array}{cc:cc:cc} 0 & 2 & 1 & 3 & 2 & 4 \\ 1 & 3 & 2 & 4 & 3 & 5 \\ \hdashline 3 & 5 & 4 & 6 & 5 & 7 \\ 4 & 6 & 5 & 7 & 6 & 8 \end{array}\right].$$

В частности, $c_{23} = a_{01} + b_{11} = 2 + 4 = 6$, так как $2 = 01_{22}$ и $3 = 11_{23}$.

Аналогично может быть определено и левое кронекеровское произведение. В этом случае используется обозначение $\overleftarrow{\otimes}_{\circ}$, а индексы представляются так: $i = (i_1\, i_0)_{n_1 n_0}$, $j = (j_1\, j_0)_{m_1 m_0}$. Для матриц из примера выше имеем

$$\mathbf{C} = \begin{bmatrix} 0 & 2 \\ 1 & 3 \end{bmatrix} \overleftarrow{\otimes}_{+} \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix} = \left[\begin{array}{ccc:ccc} 0 & 1 & 2 & 2 & 3 & 4 \\ 3 & 4 & 5 & 5 & 6 & 7 \\ \hdashline 1 & 2 & 3 & 3 & 4 & 5 \\ 4 & 5 & 6 & 6 & 7 & 8 \end{array}\right]$$

и $c_{2\,3} = a_{11} + b_{0\,0} = 3 + 0 = 3$, так как $2 = 01_{2\,2}$ и $3 = 01_{3\,2}$.

Очевидно, что $\mathbf{A} \overrightarrow{\otimes} \mathbf{B} \neq \mathbf{A} \overleftarrow{\otimes} \mathbf{B}$. Однако из определения 2 могут быть получены следующие тождества:

(4) $$\mathbf{A} \overrightarrow{\otimes}_{\circ} \mathbf{B} = \mathbf{B} \overleftarrow{\otimes}_{\circ^T} \mathbf{A}, \quad \mathbf{A} \overleftarrow{\otimes}_{\circ} \mathbf{B} = \mathbf{B} \overrightarrow{\otimes}_{\circ^T} \mathbf{A},$$

где операция $\circ^T$ получается путем транспонирования матрицы операции $\circ$.

Далее, если это не оговорено особо, знаком $\otimes_{\circ}$ будем обозначать правое кронекеровское произведение матриц.

4. Полиномиальная факторизация

Из формулы (1) непосредственно следует рекуррентное правило для полиномиальной факторизации:

(5) $$\mathbf{T}_0^* = \mathbf{D}, \quad \mathbf{T}_{t+1}^* = \mathbf{T}_t^L \otimes_t \mathbf{T}_t^R \ (t = 0, 1, \dots), \quad \mathbf{P}_j = \mathbf{T}_k^*,$$

где $\mathbf{T}_{t+1}^*$ – левая $\mathbf{T}_{t+1}^L$ или правая $\mathbf{T}_{t+1}^R$ промежуточная матрица, полученная на предыдущем шаге. Если размерность $\mathbf{T}_k^*$ не превосходит k_p, то матрица отождествляется с некоторой матрицей степенной операции $\mathbf{P}_j$.

Вычисления завершаются, когда отождествляются все промежуточные матрицы. Если $\mathbf{T}_{t+1}^L$ или $\mathbf{T}_{t+1}^R$ не отождествлена, то результат последующей факторизации этих матриц заключается в скобки. В свою очередь, индексы степеней переменных i_j последовательно возрастают в порядке и в направлении вычислений.

Заметим, что в силу тождеств (4) без потери общности в рекуррентном правиле (5) под $\otimes_t$ достаточно подразумевать только правое кронекеровское произведение.

4.1. Полиномиальная декомпозиция матрицы

Базовой операцией в (5) является представление матрицы $\mathbf{T}$ размерности $m' \times m''$ в виде кронекеровского произведения матриц $\mathbf{L}$ и $\mathbf{R}$ относительно неизвестной операции $\circ$,

(6) $$\mathbf{T} = \mathbf{L} \otimes_{\circ} \mathbf{R}.$$

Найдем для заданной матрицы $\mathbf{T}$ такие матрицы $\mathbf{L}$ и $\mathbf{R}$, а также операцию $\circ$, определяемую матрицей $\mathbf{O}$, которые обращают выражение (6) в равенство. Для начала определим $\mathbf{L}$, $\mathbf{R}$ и $\mathbf{O}$ следующим образом:

(7) $$\begin{cases} \mathbf{L} = [l_{uv}], & l_{uv} = v + uk_l'' \quad (u = \overline{0, k_l' - 1}, \quad v = \overline{0, k_l'' - 1}), \\ \mathbf{R} = [r_{sw}], & r_{sw} = w + sk_r'' \quad (s = \overline{0, k_r' - 1}, \quad w = \overline{0, k_r'' - 1}), \\ \mathbf{O} = [o_{pq}], & o_{pq} = p \circ q \quad (p = \overline{0, k_l' k_l'' - 1}, \quad q = \overline{0, k_r' k_r'' - 1}), \end{cases}$$

где k_l' и k_r' – делители m', а k_l'' и k_r'' – делители m'' такие, что $m' = k_l' k_r'$ и $m'' = k_l'' k_r''$.

Так как матрицы $\mathbf{L}$ и $\mathbf{R}$ заданы, нам остается найти только матрицу $\mathbf{O}$. Из определения 2 и выражений (7) следует

(8) $$t_{ij} = l_{i_l j_l} \circ r_{i_r j_r} = (j_l + i_l k_l'') \circ (j_r + i_r k_r'') = o_{pq},$$

где t_{ij} – элемент $\mathbf{T}$, $i = (i_l, i_r)_{k_l' k_r'}$ и $j = (j_l, j_r)_{k_l'' k_r''}$ – представления индексов i и j в позиционной системе счисления с основаниями, задаваемыми делителями размерностей матрицы $\mathbf{T}$. Из (8) получаем выражение для вычисления соединительной операции $\mathbf{O}$,

(9) $$p \circ q = \mathbf{T}[(p'', q'')_{k_l'' k_r''} \ (p', q')_{k_l' k_r'}],$$

где $p = (p'', p')_{k_l'' k_l'}$, $q = (q'', q')_{k_r'' k_r'}$, $\mathbf{T}[j \ i]$ – элемент t_{ij} матрицы $\mathbf{T}$.

Формула (9) позволяет вычислить матрицу $\mathbf{O}$. Для этого операнды p и q искомой операции $\circ$ представляются в позиционной системе счисления с основаниями, являющимися делителями чисел m' и m''. Затем цифры p', p'', q' и q'' используются для вычисления индексов элемента матрицы $\mathbf{T}$ – результата операции $\circ$.

В итоге для произвольной матрицы $\mathbf{T}$ получаем полностью определенные матрицы $\mathbf{L}$, $\mathbf{R}$ и $\mathbf{O}$, связанные уравнением (6). Очевидно, сделать это можно различными способами. Число решений уравнения (6) равно количеству нетривиальных разбиений текущего множества переменных на два непересекающихся подмножества, произведение значностей k_l' и k_r' которых равно m', умноженное на количество представлений числа m'' в виде двух нетривиальных сомножителей k_l'' и k_r'' таких, что $k_l'' k_r'' = m''$. Выбираем то решение, у которого матрица соединительной операции $\mathbf{O}$

может быть тождественно преобразована (редуцирована) до размеров, не превосходящих значность факторизации k_p.

4.2. Полиномиальная редукция

Следующим шагом полиномиальной факторизации является редукция $\mathbf{O}$ и соответствующая модификация $\mathbf{L}$ и $\mathbf{R}$. Для этого строки и столбцы матрицы $\mathbf{O}$ разбиваются на классы эквивалентности. Сама матрица $\mathbf{O}$ преобразуется к виду, содержащему по одной строке (столбцу) их каждого класса.

Пусть $\{E_0, E_1, \ldots, E_{n_r-1}\}$ и $\{C_0, C_1, \ldots, C_{n_c-1}\}$ – классы эквивалентности строк и столбцов $\mathbf{O}$, состоящие из индексов одинаковых строк и столбцов (или сопоставимых строк и столбцов – в случае неполностью определенных матриц). Редуцированную матрицу строим следующим образом: в качестве i-й строки новой матрицы используем строку из класса E_i, а вместо j-го столбца – столбец из класса C_j.

Для сохранения равенства (6) все элементы $e \in E_i$ левой матрицы $\mathbf{L}$ должны быть заменены i – индексом класса эквивалентности, которому принадлежит элемент e. Аналогичным образом заменяются элементы правой матрицы. Все элементы $c \in C_j$ матрицы $\mathbf{R}$ заменяются j – индексом класса эквивалентности, в который входит элемент c.

В итоге получена редуцированная матрица $\mathbf{O}$ и модифицированные матрицы $\mathbf{L}$ и $\mathbf{R}$. Если размерность $\mathbf{O}$ не превосходит k_p по любому из измерений, то редукция считается успешной. В противном случае выбираем другое решение уравнения (6) и повторяем декомпозицию и редукцию.

4.3. Минимизация полиномиальных функций

Как показано выше, в результате полиномиальной факторизации для произвольной матрицы $\mathbf{D}$ может быть получено ее представление в виде формулы (3). Формула (3) позволяет по номеру столбца матрицы (задается индексом функции i) и номеру строки (задается значениями переменных X) вычислить соответствующий элемент матрицы $\mathbf{D}$.

При фиксированном значении i формула (3) описывает одну из функций. Так как индекс функции i единственным образом преобразуется в степени i_0, i_1, …, i_{n-1}, то формула (3) при рассмотрении фиксированной функции является избыточной.

Выполним тождественные структурные преобразования конструкции формулы. С учетом произвольной расстановки скобок возможны три типа вхождения переменных в формулу:

$$\theta_i(X) = \ldots(x_j^{i_j} \circ_j x_{j+1}^{i_{j+1}})\ldots(x_s^{i_s} \circ_s (\ldots))\ldots((\ldots)\circ_{t-1} x_t^{i_t})\ldots.$$

Рассмотрим первый тип как наиболее общий, включающий в виде частных случаев два оставшихся. Представим вычисление $x_j^{i_j} \circ_j x_{j+1}^{i_{j+1}}$ в виде следующей диаграммы:

$$\begin{array}{ccccccc} & i_j & & x_j^{i_j} \circ_j x_{j+1}^{i_{j+1}} & & i_{j+1} & \\ & \downarrow & & \uparrow & & \downarrow & \\ \overset{x_j}{\rightarrow} & \begin{bmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & \ddots & \\ * & * & \dots & * \end{bmatrix} & \overset{x_j^{i_j}}{\rightarrow} & \begin{bmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & \ddots & \\ * & * & \dots & * \end{bmatrix} & \overset{x_{j+1}^{i_{j+1}}}{\leftarrow} & \begin{bmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & \ddots & \\ * & * & \dots & * \end{bmatrix} & \overset{x_{j+1}}{\leftarrow} \\ & \rhd_j & & \circ_j & & \rhd_{j+1} & \end{array}.$$

На диаграмме показано, что значение переменной x_j определяет строку, а степень i_j – столбец результата операции $\rhd_j$, который, в свою очередь, задает строку для $\circ_j$, столбец которой получается аналогично и задается переменной x_{j+1}, ее степенью i_{j+1} и матрицей операции $\rhd_{j+1}$. Так как для рассматриваемой функции степени переменных постоянны, то конструкция формулы может быть представлена эквивалентной диаграммой:

$$\begin{array}{ccccccc} & & & \tilde{x}_j \circ_j \hat{x}_{j+1} & & & \\ & & & \uparrow & & & \\ \overset{x_j}{\rightarrow} & \begin{bmatrix} * \\ * \\ \vdots \\ * \end{bmatrix} & \overset{\tilde{x}_j}{\rightarrow} & \begin{bmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & \ddots & \\ * & * & \dots & * \end{bmatrix} & \overset{\hat{x}_{j+1}}{\leftarrow} & \begin{bmatrix} * \\ * \\ \vdots \\ * \end{bmatrix} & \overset{x_{j+1}}{\leftarrow} \\ & \sim & & \circ_j & & \wedge & \end{array},$$

где степенные операции представлены используемыми столбцами, что эквивалентно некоторым унарным операциям $\sim$ и $\wedge$.

Теперь каждая пара, состоящая из унарной и бинарной операций, преобразуется в эквивалентную им новую бинарную операцию $\bullet_j$, осуществляемую путем копирования и перестановки строк матрицы $\circ_j$:

$$\begin{array}{ccc} & x_j \bullet_j x_{j+1} & \\ & \uparrow & \\ \begin{array}{c} x_j \\ \rightarrow \end{array} & \begin{bmatrix} * & * & \dots & * \\ * & * & \dots & * \\ \vdots & \vdots & \ddots & \\ * & * & \dots & * \end{bmatrix} & \begin{array}{c} x_{j+1} \\ \leftarrow \end{array} \\ & \bullet_j & \end{array}.$$

Если окажется, что операция $\bullet_j$ содержит одинаковые строки (столбцы), то она несущественным образом зависит от левого (правого) операнда и этот операнд может быть исключен из формулы. В итоге получаем минимальную формулу функции

$$\theta_i(X) = x_{t_1} \bullet_1 x_{t_2} \bullet_2 \dots \bullet_{s-1} x_{t_s} ,$$

заданную с точностью до расстановки скобок и из которой исключены несущественные переменные.

4.4. Демонстрационный пример

Произведем полиномиальную факторизацию упорядоченной системы функций, заданных матрицей

$$\mathbf{T} = \begin{bmatrix} 0 & 1 & 1 & 2 & 0 & 1 \\ 2 & 1 & 2 & 2 & 0 & 1 \\ 2 & 0 & 2 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 1 & 2 \\ 2 & 2 & 2 & 1 & 2 & 2 \\ 2 & 1 & 2 & 0 & 2 & 1 \end{bmatrix}$$

размерности 6×6 при значности факторизации $k_p = 3$. Пусть $k'_l = 3$, $k'_r = 2$, $k''_l = 2$ и $k''_r = 3$. Тогда из (7) и (9) получим:

$$\mathbf{L} = \begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}, \ \mathbf{O} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & 2 & 1 & 2 \\ 2 & 2 & 0 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 & 1 & 2 \\ 2 & 2 & 0 & 2 & 2 & 2 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \ \mathbf{R} = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}.$$

В частности, для элемента o_{53} имеем

$$o_{53} = 5 \circ 3 = 1\,2_{23} \circ 0\,1_{32} = \mathbf{T}[1\,0_{23}\ 2\,1_{32}] = \mathbf{T}[1\ \ 5] = t_{51} = 1 .$$

Далее находим классы эквивалентности строк и столбцов:

$$E_0 = \{0, 5\},\ E_1 = \{1, 3\},\ E_2 = \{2, 4\};$$

$$C_0 = \{0, 4\},\ C_1 = \{1, 3, 5\},\ C_2 = \{2\}.$$

После построения редуцированной матрицы соединительной операции и замены элементов левой и правой матриц на индексы классов, в которые они входят, получим

$$\mathbf{L} = \begin{bmatrix} 0 & 1 \\ 2 & 1 \\ 2 & 0 \end{bmatrix}, \ \mathbf{O} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 2 & 2 & 0 \end{bmatrix}, \ \mathbf{R} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \end{bmatrix}.$$

В итоге имеем $\mathbf{T} = \mathbf{L} \otimes_{\circ} \mathbf{R}$, где размерности матриц $\mathbf{L}$, $\mathbf{R}$ и $\mathbf{O}$ не больше чем k_p. Следовательно, система функций, заданная матрицей $\mathbf{T}$, может быть вычислена по формуле

$$\theta_i(x_0, x_1) = x_0^{i_0} \circ x_1^{i_1} = (x_0 \triangleright_L i_0) \circ (x_1 \triangleright_R i_1),$$

где значность переменной x_0 равна 3, значность переменной x_1 – 2, индексы функций представляются в системе счисления с основаниями 2 и 3, а степенные и соединительная операции задаются редуцированными матрицами $\mathbf{L}$, $\mathbf{R}$ и $\mathbf{O}$.

Выполним проверку формулы путем вычисления θ_5 (последний столбец $\mathbf{T}$). Для этого представим индекс функции в системе счисления с основаниями 2 и 3, $5 = 12_{23}$. Откуда находим степени переменных $i_0 = 1$, $i_1 = 2$. Конструируем итоговую формулу $\theta_5(x_0, x_1) = x_0^1 \circ x_1^2$ и выполняем ее минимизацию

$$\theta_5(x_0, x_1) = (\triangleright_0 x_0) \circ (\triangleright_1 x_1) = x_0^{\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}} \circ x_1^{\begin{bmatrix} 2 \\ 1 \end{bmatrix}} = x_0 \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} x_1 = x_0 \bullet x_1, \tag{10}$$

где степенные операции преобразованы в унарные и заданы в виде векторов. В (10) также выполнено преобразование бинарной операции $\circ$ и унарных операций $\triangleright_0$ и $\triangleright_1$ в эквивалентную им бинарную операцию $\bullet$, благодаря чему имеем минимальную формулу функции. Вычисление этой функции по любой из полученных формул дает вектор $[1 1 0 2 2 1]$, совпадающий с пятым столбцом матрицы $\mathbf{T}$.

Из примера видно, что полиномиальная факторизация может быть использована для получения формульных представлений системы спектральных функций при различной их упорядоченности, для минимизации или модификации операций, изменения значностей переменных, их порядка в выражении и т.д. В этом случае по известной формуле для полиномиальных функций вычисляется, а затем факторизуется матрица дискретного преобразования $\mathbf{D}$. В результате может быть получено другое формульное представление для той же системы функций, если, конечно, оно существует.

5. Алгоритм полиномиальной факторизации

Ниже приведен алгоритм полиномиальной факторизации системы спектральных функций, зависящих от переменных X и заданных квадратной матрицей **T**.

```
scalar k_P
stack powers, connects
string function Formula(matrix T, set X)
        scalar u, v
        string F_l, F_r
        matrix L, O, R
        vector E, C
        if Count(X) = 1 then
                u := X(1)
                v := Push(powers, T)
                return "(" + "x_u" + "▷_v" + "i_v"+ ")"
        else
                First(X)
                do
                        O := Middle(T, X)
                        E := Rows(O)
                        C := Columns(O)
                        if Classes(E) <= k_P and Classes(C) <= k_P
                                L := Left(E)
                                R := Right(C)
                                O := Operation(O, E, C)
                                F_l := Formula(L, Implement(X))
                                F_r := Formula(R, Complement(X))
                                if F_l <> empty and F_r <> empty
                                        u := Push(connects, O)
                                        return "(" + F_l + "∘_u" + F_r + ")"
                                else
                                        return empty
                                end if
                        end if
                while Next(X)
                return empty
        end if
end function
```

Алгоритм реализует частный случай факторизации, когда степенные операции ищутся в виде квадратных матриц, т.е. когда значности переменных совпадают со значностью их степеней.

Функция Formula (строка 3) является рекурсивной и возвращает формулу для системы функций. Декомпозиция матрицы $\mathbf{T}$ осуществляется функцией Middle (строка 15), которая возвращает нередуцированную матрицу $\mathbf{O}$. Функция Rows (строка 16) и Columns (строка 17) формируют векторы эквивалентности строк и столбцов следующим образом: i-й элемент вектора описывает i-ю строку (столбец) и равен индексу класса эквивалентности, в который эта строка (столбец) входит. Такое представление позволяет по получаемым векторам $\mathbf{E}$ и $\mathbf{C}$ легко построить матрицы $\mathbf{L}$ и $\mathbf{R}$, для чего используется функция Left (строка 19) и функция Right (строка 20). Если количество классов эквивалентности, возвращаемое функцией Classes (строка 18), не превосходит значности факторизации k_p, то производится редукция матрицы $\mathbf{O}$, осуществляемая функцией Operation (строка 21).

Для работы с наборами переменных используются функции First (строка 11) и Next (строка 31), которые формируют начальное и следующее подмножества переменных, причем последняя функция возвращают логический ноль, если все подмножества просмотрены, а в противном случае – логическую единицу. Функции Implement (строка 22) и Complement (строка 23) формируют соответственно текущий набор переменных и набор переменных, не вошедших в текущий, а функция Count (строка 8) служит для получения количества переменных в наборе.

Результатом работы алгоритма является формула, представляющая собой строку, состоящую из имен переменных, знаков операций и скобок, задающих порядок вычисления (строки 11, 26). Сами операции получаются в виде матриц и сохраняются в стеке операций: степенные – в стеке $powers$ (строка 10), соединительные – в стеке $connects$ (строка 25). Если формулы не существует, то результатом работы алгоритма будет пустая строка и незаполненные стеки операций.

Представленный алгоритм основан на единственности полиномиальной декомпозиции матрицы. Оператор return (строки 26, 28) выполняет выход из цикла по подмножествам переменных X и из рекурсии по глубине формулы, если матрица соединительной операции $\mathbf{O}$ имеет успешную редукцию, причем независимо от того, факторизовались ли матрицы $\mathbf{L}$ и $\mathbf{R}$ или нет.

6. Единственность полиномиальной декомпозиции

Приведем обоснование единственности полиномиальной декомпозиции матрицы, лежащее в основе описанного выше алгоритма полиномиальной факторизации.

Автоматика и телемеханика. 2004. № 12. С. 3-14.

Теорема. Пусть задана нетривиальная матрица $\mathbf{T}$ *размерности* $m' \times m''$, *а также матрицы* $\mathbf{L}$ *размерности* $m'_L \times m''_L$ *и* $\mathbf{R}$ *размерности* $m'_R \times m''_R$, *такие, что* $\mathbf{T} = \mathbf{L} \otimes_\circ \mathbf{R}$, *где операция* $\circ$ *имеет размерность не более чем* $\frac{m'_L m'_R}{2} \times \frac{m''_L m''_R}{2}$. *Тогда не существует операции* $\bullet$ *и матриц* $\mathbf{A}$ *и* $\mathbf{B}$ *с размерностями* $m'_A \times m''_A$ *и* $m'_B \times m''_B$, *отличными от размерностей* $\mathbf{L}$ *и* $\mathbf{R}$, *таких, что* $\mathbf{T} = \mathbf{A} \otimes_\bullet \mathbf{B}$, *где размерность операции* $\bullet$ *не более чем* $\frac{m'_A m'_B}{2} \times \frac{m''_A m''_B}{2}$.

Доказательство. Благодаря тождествам (4) достаточно рассмотреть только правое кронекеровское произведение матриц.

Покажем структуру связей между элементами $\mathbf{T}$, налагаемых выражением $\mathbf{T} = \mathbf{L} \otimes_\circ \mathbf{R}$. Предположим, что для исходной матрицы $\mathbf{T}$ размерности 6×6 и матриц $\mathbf{L}$ и $\mathbf{R}$ с размерностями 2×3 и 3×2 построена матрица соединительной операции $\mathbf{O}_\circ$ и установлено равенство 2-ой и 4-ой строки и 0-го и 5-го столбца,

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}, \quad \mathbf{O}_\circ = \begin{bmatrix} a_\circ & * & * & * & * & a_\circ \\ b_\circ & * & * & * & * & b_\circ \\ c_\circ & \alpha_\circ & \beta_\circ & \gamma_\circ & \delta_\circ & c_\circ \\ d_\circ & * & * & * & * & d_\circ \\ c_\circ & \alpha_\circ & \beta_\circ & \gamma_\circ & \delta_\circ & c_\circ \\ f_\circ & * & * & * & * & f_\circ \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}.$$

Теперь построим матрицу $\mathbf{O}_\bullet$ для $\mathbf{T} = \mathbf{A} \otimes_\bullet \mathbf{B}$, где строки 0 и 3, и столбцы 1 и 4 равны, а $\mathbf{A}$ и $\mathbf{B}$ имеют размерности 3×2 и 2×3, отличные от размерностей $\mathbf{L}$ и $\mathbf{R}$:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}, \quad \mathbf{O}_\bullet = \begin{bmatrix} \alpha_\bullet & a_\bullet & \beta_\bullet & \gamma_\bullet & a_\bullet & \delta_\bullet \\ * & b_\bullet & * & * & b_\bullet & * \\ * & c_\bullet & * & * & c_\bullet & * \\ \alpha_\bullet & a_\bullet & \beta_\bullet & \gamma_\bullet & a_\bullet & \delta_\bullet \\ * & e_\bullet & * & * & e_\bullet & * \\ * & f_\bullet & * & * & f_\bullet & * \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \end{bmatrix}.$$

Из построения находим, что $\mathbf{T}$ в первом и втором случае разбивается на подматрицы с размерностью, равной размерности левой матрицы ($\mathbf{L}$ или $\mathbf{A}$), а количество таких подматриц равно числу элементов правой матрицы ($\mathbf{R}$ или $\mathbf{B}$),

$$\mathbf{T}=\left[\begin{array}{ccc:ccc} a_\circ & b_\circ & c_\circ & * & * & \alpha_\circ \\ d_\circ & c_\circ & f_\circ & * & \alpha_\circ & * \\ \hdashline * & * & \beta_\circ & * & * & \gamma_\circ \\ * & \beta_\circ & * & * & \gamma_\circ & * \\ \hdashline * & * & \delta_\circ & a_\circ & b_\circ & c_\circ \\ * & \delta_\circ & * & d_\circ & c_\circ & f_\circ \end{array}\right]=\left[\begin{array}{cc:cc:cc} \alpha_\bullet & * & a_\bullet & b_\bullet & \beta_\bullet & * \\ * & \alpha_\bullet & c_\bullet & a_\bullet & * & \beta_\bullet \\ * & * & e_\bullet & f_\bullet & * & * \\ \hdashline \gamma_\bullet & * & a_\bullet & b_\bullet & \delta_\bullet & * \\ * & \gamma_\bullet & c_\bullet & a_\bullet & * & \delta_\bullet \\ * & * & e_\bullet & f_\bullet & * & * \end{array}\right].$$

Если пронумеровать подматрицы и их элементы слева направо и сверху вниз, то видна связь подматриц и столбцов матрицы $\mathbf{O}_\circ$ ($\mathbf{O}_\bullet$): i-й столбец $\mathbf{O}_\circ$ ($\mathbf{O}_\bullet$) состоит из элементов i-й подматрицы, размещенных в порядке нумерации элементов. С другой стороны, j-я строка $\mathbf{O}_\circ$ ($\mathbf{O}_\bullet$) состоит из j-х элементов, размещенных в порядке нумерации подматриц.

Из примера находим, что $\mathbf{L}\otimes_\circ\mathbf{R}=\mathbf{A}\otimes_\bullet\mathbf{B}$, если $a_\circ=c_\circ=f_\circ=\gamma_\circ=\delta_\circ=a_\bullet=c_\bullet=\alpha_\bullet=\delta_\bullet$, $d_\circ=f_\bullet$ и $\beta_\circ=e_\bullet$. Очевидно, существуют матрицы, для которых указанные тождества удается выполнить. Однако если дополнительно отождествить оставшиеся строки и столбцы так, чтобы число классов эквивалентности равнялось 3, то получаемая при этом связь элементов порождает один класс эквивалентности элементов, т.е. матрица $\mathbf{T}$ становится константной.

Наличие элементов, помеченных знаком $*$, определяет некоторую степень свободы, позволяющую сохранять нетривиальность матрицы $\mathbf{T}$. Найдем количества классов эквивалентности строк и столбцов, при которых представление $\mathbf{T}=\mathbf{A}\otimes_\bullet\mathbf{B}$ невозможно. Для этого потребуем отсутствия элементов, помеченных знаком $*$, и найдем условия, при которых последнее имеет место.

Действительно, если число классов эквивалентности столбцов $\mathbf{O}_\circ$ меньше или равно $\frac{m_L'm_R'}{2}$, то отождествляется не менее половины элементов матрицы $\mathbf{T}$, оставшаяся половина отождествляется, если число классов эквивалентности строк меньше или равно $\frac{m_L''m_R''}{2}$. С другой стороны, число классов эквивалентности элементов, получаемых из выражений $\mathbf{T}=\mathbf{L}\otimes_\circ\mathbf{R}$ и $\mathbf{T}=\mathbf{A}\otimes_\bullet\mathbf{B}$, не превосходит количества элементов редуцированных матриц операций $\circ$ и $\bullet$ (равно соответственно $\frac{m_L'm_R'}{2}\times\frac{m_L''m_R''}{2}=\frac{m'm''}{4}$ и $\frac{m_A'm_B'}{2}\times\frac{m_A''m_B''}{2}=\frac{m'm''}{4}$). Так как размерности $\mathbf{L}$ и $\mathbf{A}$ различны, то классы эквивалентности элементов пересекаются. Последнее означает, что различные элементы попадают в один и тот же класс. Полученное противоречие доказывает теорему.

7. Бесповторная декомпозиция

Представленный выше алгоритм может быть использован для бесповторной декомпозиции функции. Бесповторной называется такая формула, в которую переменные входят не более одного раза.

Так как система спектральных функций в рассматриваемом случае состоит из одной функции, то матрица дискретного преобразования **D** вырождается в вектор-столбец и, как следствие этого, промежуточные матрицы **L** и **R** ищутся в виде векторов. Очевидно, что вычисляемые при этом степенные операции будут унарными.

Декомпозиция завершается, когда длина текущего вектора **T** совпадает со значностью переменной, оставшейся свободной. Если факторизация успешная, то может быть получена минимальная формула функции

$$f(X) = x_{t_1} \bullet_1 x_{t_2} \bullet_2 \ldots \bullet_{s-1} x_{t_s} ,$$

представленная с точностью до расстановки скобок и из которой исключены унарные степенные операции и несущественные переменные. Если факторизация не удалась, то декомпозируемая функция не имеет бесповторного представления при заданной значности факторизации.

8. Заключение

На рисунке представлены результаты экспериментального исследования алгоритма полиномиальной факторизации, где показана зависимость времени факторизации t от размерности квадратной матрицы m при неуспешной (A) и успешной (B) факторизации. Эксперимент проводился на множестве трехзначных функций при значности факторизации, равной 3. Из рисунка следует, что время факторизации оценивается как $t \sim m^{\alpha}$, где α изменяется от 2,2 до 3,6.

Приведем еще одно применение полиномиальной факторизации в области сжатия изображений. Известно, что при цифровой обработке изображений имеют дело с массивами (матрицами) данных, составные части которых сильно коррелированны между собой, а сами данные кодируются целыми числами. Традиционно ставится задача сжатия изображения с минимальными потерями качества при восстановлении.

Как показано ранее, при полиномиальной факторизации на каждом шаге производится разбиение исходной матрицы на части, причем число делений по горизонтали и по вертикали является изменяемым и определяется возможностью представления матрицы в виде совокупности одинаковых или близких друг к другу с заданной точностью областей. Если количество классов эквивалентности областей не превосходит некоторого числа, определяющего степень сжатия, то формируется соединительная операция (матрица небольшой размерности), а также правая и левые матрицы, которые подлежат дальнейшей факторизации. Если достигнута небольшая или приемлемая по другим соображениям размерность этих матриц, то процесс полиномиальной факторизации завершается.

В итоге исходное изображение $m \times m$ точек заменяется на $2n-1$ матриц размерности $k_p \times k_p$, где $k_p \approx \sqrt[n]{m}$ – степень сжатия изображения, а объем памяти, необходимый для хранения результата преобразования, сокращается примерно в $2n$ раз.

Понятно, что при жестком задании степени сжатия факторизация может оказаться невыполнимой. В этом случае возможна модификация алгоритма, при которой на каждом шаге выбирается своя, присущая только этому шагу степень сжатия. Тем самым обеспечивается безусловность получения положительного результата и адаптивность алгоритма сжатия изображения.

Особенностью описанного алгоритма является возможность изменения масштаба сопоставляемых областей, вплоть до нескольких точек. Другая особенность алгоритма – понижение значности элементов изображения с каждым шагом декомпозиции. На первом шаге соединительная матрица содержит все отличные друг от друга отсчеты сигнала. После редукции значность элементов левой и правой матрицы понижается до соответствующих размерностей редуцированной матрицы, т.е. с каждым шагом факторизации динамический диапазон сигнала уменьшается примерно в $\sqrt{d}$ раз, где d – начальное его значение.

Автоматика и телемеханика. 2004. № 12. С. 3-14.

Список литературы

1. *Ахмед Н., Рао К. Р.* Ортогональные преобразования при обработке цифровых сигналов. М.: Связь, 1980.
2. *Выхованец В.С.* Спектральные методы в логической обработке данных // АиТ. 2001. № 10. С. 28-53.

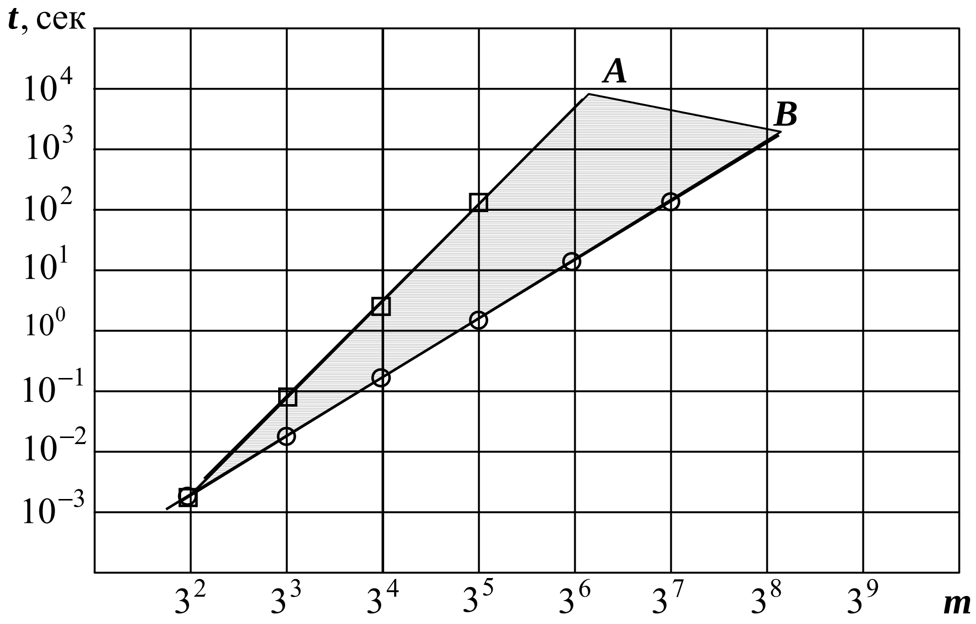

Рис. Зависимость времени факторизации t от размерности матрицы m при неуспешной (A) и успешной (B) факторизации